软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(8):1853−1870 [doi: 10.13328/j.cnki.jos.004838]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向随机模型检验的模型抽象技术*

刘 阳[1,2], 李宣东[1], 马 艳[3]

[1](计算机软件新技术国家重点实验室(南京大学),江苏 南京 210093)

[2](Department of computer Science, School of Computing, National University of Singapore, Singapore 117417, Singapore)

[3](南京航空航天大学 计算机科学与技术学院,江苏 南京 210016)

通讯作者: 刘阳, E-mail: yangliu@seg.nju.edu.cn

摘 要: 随机模型检验是经典模型检验理论的延伸和推广,由于其结合了经典模型检验算法和线性方程组求解或线性规划算法等,并且运算处理的是关于状态的概率向量而非经典模型检验中的位向量,所以状态爆炸问题在随机模型检验中更为严重.抽象作为缓解状态空间爆炸问题的重要技术之一,已经开始被应用到随机模型检验领域并取得了一定的进展.以面向随机模型检验的模型抽象技术为研究对象,首先给出了模型抽象技术的问题描述,然后按抽象模型构造技术分类归纳了其研究方向及目前的研究进展,最后对比了目前的模型抽象技术及其关系,总结出其还未能给出模型抽象问题的满意答案,并指出了有效解决模型抽象问题未来的研究方向.

关键词: 随机模型检验;状态空间爆炸;模型抽象;定量抽象精化

中图法分类号: TP311

中文引用格式: 刘阳,李宣东,马艳.面向随机模型检验的模型抽象技术.软件学报,2015,26(8):1853−1870. http://www.jos.org.cn/1000-9825/4838.htm

英文引用格式: Liu Y, Li XD, Ma Y. Model abstraction for stochastic model checking. Ruan Jian Xue Bao/Journal of Software, 2015,26(8):1853−1870 (in Chinese). http://www.jos.org.cn/1000-9825/4838.htm

Model Abstraction for Stochastic Model Checking

LIU Yang[1,2], LI Xuan-Dong[1], MA Yan[3]

[1](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210093, China)

[2](Department of computer Science, School of Computing, National University of Singapore, Singapore 117417, Singapore)

[3](College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: Stochastic model checking is a recent extension and generalization of the classical model checking. Stochastic model checking combines the classical model checking algorithms and linear equation solving or linear programming algorithms, moreover, it processes the probability vector instead of the bit vector. Consequently, the state explosion problem is more severe in stochastic model checking than classical model checking. Abstraction is an important means to tackle the state explosion problem, and it has made some progress in applying to the field of stochastic models testing. This study focus on model abstraction for stochastic model checking. First, the problem of model abstraction is formally presented. Then, the advances in the research area are classified and summarized according to the construction technology of abstraction model. At last, the various abstraction technologies are compared in regard to the effectiveness of solving the model abstraction problem, and the future research topics for improvement in solving the model abstraction problem are pointed out.

Key words: stochastic model checking; state space explosion; model abstraction; quantitative abstraction and refinement

* 基金项目: 国家自然科学基金(61021062, 61472179); 中国博士后科学基金(2013M531328); 山东省自然科学基金(ZR2012FQ013); 山东省高等学校科技计划(J13LN10); 泰安市科技发展计划(201330629)

收稿时间: 2014-07-08; 修改时间: 2015-03-10; 定稿时间: 2015-03-27

计算机软件是信息基础设施的灵魂,它已经渗透到国民经济、国防建设和个人生活的各个方面[1].软件系统正变得日趋庞大和复杂,很多实际的软件系统被赋予随机行为特征[2],其原因可分为 3 类:

(1) 系统本身包含随机性,如概率算法或随机化算法(randomized algorithm)的使用;

(2) 系统运行环境的复杂,造成系统构件间调用过程的失败或传递消息的丢失等随机故障的发生;

(3) 对系统进行性能评价和分析,需要人为地增加随机变量来刻画其相应的性能指标.

用模型检验(model checking)[3−5]的方法对具有随机行为的复杂系统进行定量验证,即称为随机模型检验(stochastic model checking)或概率模型检验(probabilistic model checking)[6,7].

随机模型检验是经典模型检验理论的延伸和推广,可对随机系统进行基于模型的自动的形式验证.

近年来,随机模型检验引起了形式验证等领域的广泛关注,取得了较大的进展,其代表性成果有:英国牛津大学 Kwiatkowska 课题组的 PRISM、德国亚琛工业大学(RWTH Aachen University)Katoen 课题组的 MRMC 和德国萨尔兰大学(Saarland University)Hermanns 课题组的 PARAM 等.目前,随机模型检验已开始应用到概率程序的正确性验证[8]、系统性能分析[9]、通信协议可靠性分析[10]、服务流程的服务质量优化[11],甚至系统生物学[12]等领域.

由于随机模型检验算法结合了经典模型检验算法和线性方程组求解或线性规划算法等,并且其运算处理的是关于状态的概率向量而非经典模型检验中的位(bit)向量,所以状态爆炸问题(state explosion problem)在随机模型检验中更为严重[6,9].因此对于规模较大的随机系统,如何缓解状态爆炸问题,将随机模型检验应用于此类软件系统的定量验证和分析,是随机模型检验研究面临的一个重要挑战.在 Turing Lecture 中,图灵奖得主 Clarke 把如何缓解随机模型检验中的状态爆炸问题列为模型检验未来研究领域的一个重要方向[6].

抽象技术[13]作为一种有效的缓解状态爆炸问题方法,在经典模型检验中取得了较好的效果.其实,抽象方法在软件工程的其他很多领域也扮演着重要的角色[14],如对于无限状态系统进行模型检验验证、建模开放系统运行环境的非确定性因素、构件组合式软件的规约和软件模型检验等.本文将以缓解随机模型检验中状态爆炸问题的模型抽象技术为研究对象,对其进行分类、总结和论述.

本文第 1 节给出本文中所用到的随机模型检验的基本概念.第 2 节给出面向随机模型检验的模型抽象问题描述.第 3 节对随机模型检验的模型抽象技术分类和归纳.第 4 节总结对比目前的模型抽象技术,并指出应对模型抽象挑战的几种未来的研究方向.

1 随机模型检验的基本概念

随机模型检验将经典模型检验的系统模型扩展到随机系统模型,将描述功能性质的时序逻辑扩展到可同时描述功能和性能的定量时序逻辑.目前,可验证的随机系统模型有 DTMC(discrete-time Markov chain,离散时间马尔可夫链)、CTMC(continuous-time Markov chain,连续时间马尔可夫链)、MDP(Markov decision process,马尔可夫决策过程)、PA(probabilistic automata,概率自动机)、NPPN(Nondeterministic Probabilistic Petri net,非确定概率 Petri 网)[15]、PTA(probabilistic timed automata,概率时间自动机)[16,17]、CTMDP(continuous-time MDP,连续时间马尔可夫决策过程)[18,19]和 IMC(interactive Markov chain,交互式马尔可夫链)[20]等,其分类见表 1.

Table 1 Stochastic system model

表 1 随机系统模型分类

	全概率	非确定+概率
离散时间	DTMC	MDP, PA, NPPN
连续时间	CTMC	PTA, CTMDP, IMC

在随机系统模型中,DTMC,CTMC 和 MDP 是基本的随机系统模型,它们与其他随机系统模型的关系如图 1 所示,其定义如下:

定义 1(DTMC). DTMC 可定义为 $M=(S,P,s_0,AP,L)$,其中,

- S 是可数的非空状态集合,表示被建模系统的所有可能配置状态;

- $P:S\times S\to[0,1]$是迁移概率函数,对于$\forall s\in S, \sum_{s'\in S}P(s,s')=1$,表示被建模系统状态间迁移的概率值;
- s_0是初始状态;
- AP是原子命题的集合;
- $L:S\to 2^{AP}$是标号函数,可用于表示用户的需求,即系统应该满足的性质.

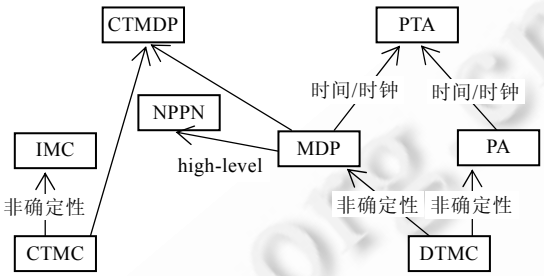

Fig.1 Relationship among stochastic system models

图 1 随机系统模型间的关系

定义 2(CTMC). CTMC 可定义为 $M=(S,R,s_0,AP,L)$,其中,

- S,s_0,AP,L 表示的含义与 DTMC 相似;
- $R:S\times S\to\mathfrak{R}_{\geqslant 0}$是退出速率(exit rate)函数,$\mathfrak{R}_{\geqslant 0}$是非负实数.

定义 3(MDP). MDP 可定义为 $M=(S,Act,s_0,AP,L)$,其中,

- S,s_0,AP,L 与 DTMC 定义的含义相似;
- Act 表示动作的集合,$P:S\times Act\times S\to[0,1]$是迁移概率函数,对于$\forall s\in S$,动作 $a\in Act, \sum_{s'\in S}P(s,a,s')\in\{0,1\}$.

目前,可验证的定量时序逻辑有 PCTL(probabilistic computation tree logic,概率计算树逻辑)、LTL+probability bounds(linear temporal logic with probability bounds,带有概率界的线性时序逻辑)、CSL(continuous stochastic logic,连续随机逻辑)、PCTL*、PTCTL(probabilistic timed computation tree logic,概率时间计算树逻辑)[21]和某些定量时序逻辑的动作扩展(如 aCSL[22],asCSL[23−25],aPCTL[15],asPCTL[26])等.其分类见表 2.

Table 2 Quantitative temporal logic

表 2 定量时序逻辑

名称	公式符号	可刻画的随机系统模型
PCTL	Φ	DTMC, MDP
LTL+probability bounds	$Prob(s,\Psi)$	DTMC, MDP
CSL	Φ	CTMC, CTMDP, IMC
PCTL*	Φ	DTMC, MDP
PTCTL	Φ	PTA
aCSL	Φ	CTMC+action
asCSL	Φ	CTMC+action
aPCTL	Φ	NPPN
asPCTL	Φ	NPPN

在定量时序逻辑中,PCTL,LTL,PCTL*和 CSL 是基本的定量时序逻辑,其他定量时序逻辑均可看作是它们的扩展或综合,其定义如下:

定义 4(PCTL). 关于原子命题集合 AP 的 PCTL 状态公式Φ可定义为

$$\Phi::=\text{true}|a|\Phi\wedge\Phi|\neg\Phi|P_{\sim p}(\Psi),$$

其中,$a\in AP$,Ψ是路径公式,$\sim\in\{\leqslant,<,\geqslant,>\}$,$p\in[0,1]$是概率界限值;

PCTL 路径公式Ψ可定义为

$$\Psi::=\mathrm{X}\Phi|\Phi\cup\Phi|\Phi\cup^{\leqslant n}\Phi,$$

其中,X(next)和∪(until)与 CTL 路径运算符语义相同,$\cup^{\leqslant n}$(bounded until)是∪的变体,要求 n 次迁移或小于 n 次迁

移内满足∪语义,n 是非负整数,Φ是状态公式.

定义 5(LTL). LTL 公式 Ψ可定义为

$$\Psi ::= \text{true}|a|\Psi \wedge \Psi|\neg \Psi|\text{X}\Psi|\Psi \cup \Psi,$$

其中,$a \in AP$,AP 是原子命题集合.LTL 公式最终是一个路径公式,其他的路径公式可以由定义推出.

定义 6(PCTL*). 关于原子命题集合 AP 的 PCTL*状态公式Φ可定义为

$$\Phi ::= \text{true}|a|\Phi \wedge \Phi|\neg \Phi|P_{\sim p}(\Psi),$$

其中,$a \in AP$, Ψ是路径公式,$\sim \in \{\leqslant, <, \geqslant, >\}$,$p \in [0,1]$是概率界限值;

PCTL*路径公式 Ψ可定义为

$$\Psi ::= \Phi|\Psi \wedge \Psi|\neg \Psi|\text{X}\Psi|\ \Psi \cup \Psi|\Psi \cup^{\leqslant n} \Psi,$$

其中,Φ是状态公式,n 是非负整数.

定义 7(CSL). 关于原子命题集合 AP 的 CSL 状态公式Φ可定义为

$$\Phi ::= \text{true}|a|\Phi \wedge \Phi|\neg \Phi|P_{\sim p}(\Psi)|S_{\sim p}(\Phi),$$

其中,$a \in AP$, Ψ是路径公式,$\sim \in \{\leqslant, <, \geqslant, >\}$,$p \in [0,1]$是概率界限值;

路径公式 Ψ可定义为

$$\Psi ::= \text{X}\Phi|\Phi \cup^{I} \Phi,$$

其中,Φ是状态公式,I 是一个实数区间.

2 面向随机检验的模型抽象问题描述

随机模型检验是通过穷尽遍历随机系统模型判定其是否满足给定的定量性质规约,包含 3 个元素,可表示为 $M \vDash \Phi$,其中,M是随机系统模型,如 DTMC,MDP,CTMC 等;Φ是用定量时序逻辑表示的定量性质规约,如 PCTL, PCTL*,LTL+概率等;$\vDash$是基于算法的满足关系判断.而基于抽象的随机模型检验(关于统一的基于抽象的随机模型检验理论可参见文献[14])是指在对任一随机模型检验元素抽象的基础上进行的随机模型检验,可表示为 $M^{\alpha} \vDash^{\alpha} \Phi^{\alpha}$,其中,$M^{\alpha}$是抽象的随机系统模型,$\vDash^{\alpha}$是基于算法抽象的满足关系,$\Phi^{\alpha}$是抽象的定量性质规约.其实,抽象模型往往会导致算法抽象和定量性质规约抽象;抽象算法也会导致定量性质规约的抽象,文献[27−31]均可看作是对算法进行的抽象的研究;抽象定量性质规约也可能会导致模型抽象和算法抽象,但关于定量性质规约本身的抽象研究较少[14].

面向随机检验的模型抽象,是通过验证比原具体模型小的抽象模型来实现验证原具体模型,即,通过 $M^{\alpha} \vDash \Phi$ 相关的信息得到 $M \vDash \Phi$相关的信息,其过程如图 2 所示.

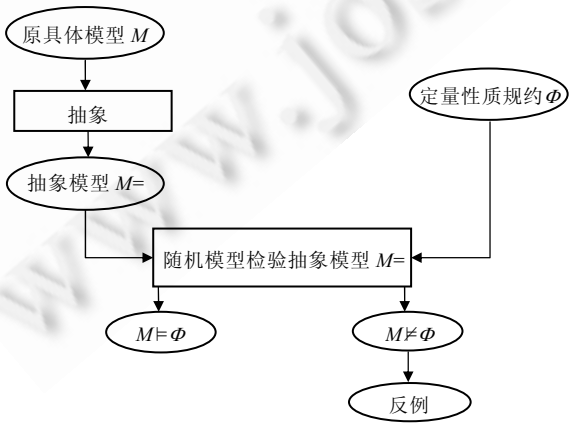

Fig.2 Model abstraction for stochastic model checking

图 2 面向随机检验的模型抽象

抽象技术的目标是使抽象模型拥有足够小的状态空间却包含丰富的具体系统模型信息,但是模型抽象的过程往往会引起模型相关信息丢失,理想的模型抽象技术要满足以下条件:

(1) 抽象模型 M^{α}要明显小于原具体模型 M;

(2) $M^{\alpha}\models\Phi\rightarrow M\models\Phi, M^{\alpha}\not\models\Phi\rightarrow M\not\models\Phi$.

因此,面向随机检验的模型抽象主要面临以下问题,我们称其为 MA4SMC(model abstraction for stochastic model checking)问题:

① 如何构造和表示抽象模型?

② 如何随机模型检验验证抽象模型?

③ 能够验证哪些定量性质?

④ 如何表示抽象模型验证中的反例?

其中,如何回答 MA4SMC 问题①是实现基于抽象的随机模型检验方法的关键,它将影响后续问题②~问题④的解决方法和结果.

3 模型抽象研究方向及进展

本节我们按照 MA4SMC 问题①的解决方法对模型抽象技术进行分类总结,分析其满足理想抽象技术条件的程度,并论述其目前主要的研究进展.

3.1 基于模拟关系的抽象

3.1.1 强概率互模拟

强概率互模拟[32]也称为概率互模拟,是对 LTS(labelled transition system,标号迁移系统)上的强互模拟[33,34]的定量扩展,即,强概率互模拟状态到其他每个在强概率互模拟下的等价类状态,要拥有相同的迁移概率.

定义 8(强概率互模拟). 令 $M=(S,P,s_0,AP,L)$是一个 DTMC,在 M 上的强概率互模拟是关于 S 的等价关系 R,且对于任意$(s_1,s_2)\in R$,满足以下条件:

(1) $L(s_1)=L(s_2)$;

(2) 对于每个等价类 $C\in S/R, P(s_1,C)=P(s_2,C)$,其中, $P(s_i,C)=\sum_{c\in C}P(s_i,c)$.

如果$(s_1,s_2)\in R$,R 是 M 上的强概率互模拟,则状态 s_1 和 s_2 称为强概率互模拟等价,可标记为 $s_1\sim_M s_2$.

在 CTMC 上的强概率互模拟定义[35],需要将条件(2)变成:$R(s_1,C)=R(s_2,C)$,其中, $R(s_i,C)=\sum_{c\in C}R(s_i,c)$. 在 MDP 上的强概率互模拟定义[36],需要对每个非确定性选择的动作 Act,存在一个动作选择使其满足条件(2).

强概率互模拟可以减少具体模型的状态空间,达到抽象目的,其用强概率互模拟商来表示抽象模型.

定义 9(强概率互模拟商). 令 $M=(S,P,s_0,AP,L)$是一个 DTMC,M 的强概率互模拟商可定义为

$$M/\sim=(S',P',AP,s_0,L'),$$

其中,$S'=S/\sim=\{[s]_\sim|s\in S\}$,$[s]_\sim$表示状态 s 的强概率互模拟状态;$P'([s]_\sim,C)=P'(s,C)$;$L'([s]_\sim)=L(s)$.

在文献[37]的基础上,文献[38]最早给出了自动构造 DTMC.M 的强概率互模拟商 $M/\sim$的算法,即划分-精化算法,其算法步骤如下:

(1) 根据标号函数,对具体模型状态进行初始划分为$\{S_1,S_2,\ldots,S_n\}$,集合 S_i 中的状态满足相同的标号函数;

(2) 找出需要状态分离的集合 S_i,即,若 $s,s'\in S_i$,对于某集合 S_k,$P(s,S_k)\neq P(s',S_k)$;

(3) 把集合 S_i 划分为子集合,使每个子集合中的状态到达 S_k 的概率一致;

(4) 重复步骤(2)、步骤(3),直到没有可划分的状态集合为止.

通过划分-精化算法可得到最粗糙的(coarsest)强概率互模拟商,但其依赖于初始划分,即,在初始划分时没有在一个状态集合的某些状态将不会得到其等价关系.

如表 3 所示,基于强概率互模拟等价的抽象,用 $M/\sim$表示抽象模型,用划分-精化算法来构造抽象模型.其优点是可保持 PCTL*/CSL 性质[39],即,对于此类性质满足理想抽象技术条件(2),模型检验 $M/\sim$得到的结果与模型检

验 M 得到的结果是一致的;其缺点是抽象模型状态空间过大,且构造抽象模型的时间有可能超过模型检验具体模型的时间.为此,文献[40]采用组合的方法,将具体模型的强概率互模拟分解为子构件的强概率互模拟,并将其用于 POTS(plain-old telephone system)的定量分析;文献[41]首次用基于 MTBDD(multi-terminal binary decision diagrams)的符号算法实现构造 DTMC 的商模型,其在时间上和处理大规模状态空间上均表现出了较好的优势;文献[42]利用互模拟求解器(satisfiability modulo theories,简称 SMT)从 PRISM 建模语言直接构造最粗糙的互模拟商,避免了在求商模型之前要构造整个模型的状态空间;文献[43]以概率自动机为随机系统模型,在定义了强 1 步互模拟及强分支互模拟的基础上给出了一种新的强概率互模拟概念,与目前的强概率互模拟相比,其可在保持 PCTL*性质前提下得到最小的强概率互模拟商模型.

Table 3 Abstraction based on strong probabilistic bisimulation equivalence

表 3 基于强概率互模拟等价的抽象

	DTMC	CTMC	MDP
抽象模型	DTMC $M/\sim$	CTMC $M/\sim$	MDP $M/\sim$
构造方法	划分-精化算法	划分-精化算法	划分-精化算法
时间复杂度	$O(\|M\|(\log(\|N\|))+\|AP\|\log(\|N\|))$	$O(\|M\|(\log(\|N\|))+\|AP\|\log(\|N\|))$	$O(\|N\|\|M\|(\log(\|N\|))+\log(\|M\|))$
可验证性质	PCTL*	CSL	PCTL*

$|M|$:模型状态数;$|N|$:模型迁移数;$|AP|$:模型的原子命题数.

3.1.2 弱概率互模拟

与强概率互模拟相比,弱概率互模拟[44,45]是一种较为粗糙的模拟关系,可以得到较小的抽象模型.

定义 10(弱概率互模拟). 令 $M=(S,P,s_0,AP,L)$是一个 DTMC,在 M 上的弱概率互模拟是关于 S 的等价关系 R,且对于任意$(s_1,s_2)\in R$,满足以下条件:

(1) $L(s_1)=L(s_2)$;

(2) 对于每个等价类 $C\in S/R, C\neq[s_1]_R$,$[s_1]_R$表示状态 s_1 的弱概率互模拟状态,s_1 可到达$[s_1]_R$ 以外的状态当且仅当 s_2 可到达$[s_2]_R$ 以外的状态,并且 $P(s_1,C)/(1-P(s_1,[s_1]_R))=P(s_2,C)/(1-P(s_2,[s_2]_R))$,如果 $P(s_i,[s_i]_R)<1$, $i=1,2$.

如果$(s_1,s_2)\in R$,R 是 M 上的弱概率互模拟,则状态 s_1 和 s_2 称为弱概率互模拟等价,可标记为 $s_1\approx_M s_2$.

与强概率互模拟类似,MDP 上的弱概率互模拟[43]及弱概率互模拟商的定义均可做相应的修改而得到.

弱概率互模拟可保持不包含 X(next)路径运算符的 PCTL*性质,即,此类性质满足理想抽象技术条件(2),其虽然可以得到较小的抽象模型,但构造抽象模型需要计算可达概率,其算法复杂度较高.对于 DTMC,文献[44]给出了构造抽象模型的 3 次方复杂度算法;对于 MDP,文献[46]给出了构造抽象模型的指数级算法;直到 2012 年,文献[47]给出了构造抽象模型的多项式算法.

3.1.3 强概率模拟

强概率模拟[32,48]可以看作是强模拟[49,50]的定量扩展,即,强概率模拟状态的分布(distribution)需要匹配(match)其模拟状态的分布.分布的概念可以用权函数(weight function)[48]来刻画.

定义 11(权函数). 令 $Dist(S)$表示关于状态集合 S 的分布集合,$\mu,\mu'\in Dist(S)$,$R\subseteq S\times S$,$F\in Dist(S)$是一个(μ,μ')和 R 的权函数,需要满足以下条件:

(1) $F(s,s')\rightarrow(s,s')\in R$;

(2) $\mu(s)=\sum_{s'\in S}F(s,s')$;

(3) $\mu(s')=\sum_{s\in S}F(s,s')$.

如果存在(μ,μ')和 R 的权函数,可标记为$\mu\subseteq_R\mu'$.

定义 12(强概率模拟). 令 $M=(S,P,s_0,AP,L)$是一个 DTMC,$R\subseteq S\times S$ 是强概率模拟,则对于任意$(s_1,s_2)\in R$,满足以下条件:

(1) $L(s_1)=L(s_2)$;

(2) $P(s_1,*)\subseteq_R P(s_2,*)$, $P(s_i,*)=\sum_{s\in S}P(s_i,s)$.

如果$(s_1,s_2)\in R$,R 是 M 上的强概率模拟,则状态 s_1 和 s_2 称为强概率模拟等价,可标记为 $s_1 \precsim_M s_2$.

在 CTMC 上的强概率模拟定义[51]需要满足的条件是:

(1) 其嵌入式 DTMC 满足强概率模拟;

(2) s_2 要具有比 s_1 更快的退出速率.

在 MDP 上的强概率模拟定义[36]需要对每个非确定性选择的动作 Act,存在一个动作选择使其满足条件(2).

强概率模拟也是用其商 $M/\precsim$[51,52]表示抽象模型,可保持 PCTL*中的定量安全真值性质,即,满足理想抽象技术条件(2)的 $M^{\alpha}\models\Phi\rightarrow M\models\Phi$;但与强概率互模拟相比,其得到的抽象模型的状态空间较小.其构造抽象模型的算法也是基于划分-精化思想.对于 DTMC 和 CTMC,文献[51]已经证明,强概率模拟等价与强概率互模拟是一致的,因此构造抽象模型的时间复杂度类似.对于 MDP,其强概率模拟商严格粗糙于强概率互模拟商,其构造算法涉及到参数最大流算法(parametric maximum flow algorithm)[53],文献[52,54]对此做了一些改进工作.

3.1.4 弱概率模拟

弱概率模拟[51,55]是模拟关系中最为粗糙的模拟关系,其定义和构造商模型的过程与上述模拟关系类似.弱概率模拟同样用商表示抽象模型,得到的抽象模型也较小,可保持不包含 X(next)路径运算符的 PCTL*的定量安全真值性质,即,此类性质满足理想抽象技术条件(2)的 $M^{\alpha}\models\Phi\rightarrow M\models\Phi$,但其时间复杂度也较高[52,54].目前,关于基于弱概率模拟的抽象研究文献较少.

3.2 基于偏序化简的抽象

偏序化简[56−59]是基于并发操作的执行顺序一般不会改变系统性质,固定一种交替执行的顺序可以减少系统模型的状态和迁移数目.因为依赖和独立关系体现了迁移发生之间的偏序关系,所以相应的方法称为偏序化简.直到近七八年,足够集(ample set)[57,60]的偏序化简方法才开始用于随机模型检验中的模型抽象.

足够集偏序化简方法的大体思想是:对于 MDP M 的任一状态 s,令其足够集为 $ample(s)\subseteq Act(s)$,然后用 $ample(s)$来替代 $Act(s)$构造偏序化简的 MDP $\hat{M}$.$\hat{M}$ 即为偏序化简的抽象模型,可定义为 $\hat{M}=(\hat{S},Act,\hat{P},s_0,AP,\hat{L})$,其中,

- $\hat{S}\subseteq S$, 包括初始状态 s_0 和状态$\{t|P(s,a,t)>0, s\in\hat{S}, a\in ample(s)\}$;
- $\hat{L}:\hat{S}\rightarrow 2^{AP}$;
- $\hat{P}(s,a,t)=\begin{cases}P(s,a,t), & \text{if } a\in ample(s)\\ 0, & \text{if } a\notin ample(s)\end{cases}$.

足够集定义所能满足的条件,决定了化简后的抽象模型可保持定量性质的范围.文献[61]率先将足够集的偏序化简方法引入到随机模型检验,其针对 MDP 特性,对经典的足够集条件[57,60]进行了扩充,要求足够集必须满足以下 5 个条件:

(1) 非空条件,即,对于任一 $s\in S$,$\Phi\neq ample(s)\subseteq Act(s)$;

(2) 哑(stutter)条件,即,如果 $s\in\hat{S}$ 并且 $ample(s)\neq Act(s)$,那么所有的动作 $a\in ample(s)$均为哑动作;

(3) 依赖条件,即,对于每条路径,$\sigma=s\xrightarrow{\alpha_1}s_1\xrightarrow{\alpha_2}\ldots\xrightarrow{\alpha_n}s_n\xrightarrow{\gamma}\ldots, s\in\hat{S}$ 并且γ依赖于 $ample(s)$,存在一个 $i\in\{1,\ldots,n\}$,使得 $a_i\in ample(s)$;

(4) 终端分量条件,即:在模型 $\hat{M}$ 的每个终端分量(T,A)中存在一个状态 $s\in T$,满足 $ample(s)=Act(s)$;

(5) 分支条件,此为新增条件,即,如果在存在 M 中的一条路径 $\sigma=s\xrightarrow{\alpha_1}s_1\xrightarrow{\alpha_2}\ldots\xrightarrow{\alpha_n}s_n\xrightarrow{\gamma}\ldots$,其中,$s\in\hat{S}$,$a_1,\ldots a_n,\gamma\notin ample(s)$并且$\gamma$是概率的,那么$|ample(s)|=1$.

其实,文献[61]中的条件(4)是对经典足够集条件(4)的弱化.在经典足够集条件中,条件(4)为循环条件,即,对于 $\hat{M}$ 中的每条路径 $s\xrightarrow{\alpha_1}s_1\xrightarrow{\alpha_2}\ldots\xrightarrow{\alpha_n}s_n=s$,存在一个状态 s_i 使得 $ample(s_i)=Act(s_i)$.文献[61]定义的足够集条件使偏序化简后的抽象模型与原具体模型的关系是概率哑等价(probabilistic stutter equivalence),可保持不带 X 算子的概率界的 LTL 性质.文献[62]加强了文献[61]新增的足够集条件(5),将其变为:对于任一状态 $s\in\hat{S}$,

ample(*s*)=*Act*(*s*)或者 *ample*(*s*)是单一元素的集合.文献[62]定义的足够集条件使偏序化简后的抽象模型与原具体模型的关系是完全向前概率模拟(complete forward simulation),可保持最大概率可达的定量性质和不带 X 算子的概率界的 LTL 性质,该方法已经在工具 LiQuor[63]中使用基于动态深度优先搜索算法 on-the-fly 产生化简模型实现.文献[64]进一步加强了文献[61]新增的足够集条件(5),将其变为:对于任一状态 $s \in \hat{S}$, 满足 *ample*(*s*)=*Act*(*s*)或者 *ample*(*s*)={*a*}且 *a* 是非概率动作.其实,该条件是文献[56]为实现两个非概率系统模型间的可视互模拟定义的.文献[64]定义的足够集条件使偏序化简后的抽象模型与原具体模型的关系是弱概率互模拟关系,可保持不带有 X 算子的 PCTL*性质.

虽然文献[61,62,64]加强了足够集条件并将其用于随机模型抽象,但实验结果显示,状态和迁移的化简过程和顺序与应用于非随机模型抽象一样[65,66],其过程的局限性在于需要用线性规划求解显式表示的化简模型.针对上述问题,文献[65]以静态偏序化简[67−69]为基础,首次将偏序化简与静态分析技术结合应用于 MDP 模型化简,给出了一个基于静态偏序化简的随机模型抽象框架:将化简标准插入到系统的控制流图,并用通过分析和修改概率控制流图算法来实现对应的标准,其可保持咂不变量的定量线性和分支时间性质.

与用足够集用于随机模型偏序化简不同,文献[70]首次将固执集(stubborn set)[71]方法用于随机模型化简,给出了一种基于弱固执集的 MDP 模型的偏序化简方法,并用 on-the-fly 化简和最优的 Tarjan 算法[72]在 PRISM[73]中实现了原型工具,其可以保持无条件公平性定量性质和公平调度下的最大概率可达性质.

3.3 基于对称化简的抽象

对称化简[74−77]是利用系统模型本身具有的状态对称重复特点对其进行化简,其本质是计算系统模型在不同局部状态对称构件的个数.对称在很多随机系统模型中均有所体现,如随机分布式算法、通信协议和生物系统等.

定义 13(状态对称). 令 M 是一个 DTMC,$\Delta: S \to S$ 是变换函数,若Δ满足 $P(\Delta(s),\Delta(s'))=P(s,s')$,则 s 和 s'是对称状态.

在 CTMC 上的对称状态的定义,只需将 P 变成退出速率函数 R;在 MDP 上的对称状态定义,需要对非确定性求解后,要求其满足定义中的条件.

对称化简通过合并对称的状态达到抽象目的,基于对称化简的抽象用商表示抽象模型.

定义 14(对称商). 对于一组关于状态 S 的变换函数 G 和相应的 orbit 关系θ,化简后的 DTMC 商模型可定义为 $(\overline{S},\overline{P})$,其中,$\overline{S}$ 是包含每个 orbit 等价类一个代表状态,$\overline{P}(\overline{s},\overline{s}') = \sum_{\{s' \in S \mid rep(s')=\overline{s}'\}} P(\overline{s},s')$, 函数 $rep: S \to \overline{S}$ 用来为每一个状态 s 选择一个对应的代表状态 $rep(s) \in \overline{S}, \overline{s}, \overline{s}' \in \overline{S}$.

关于 CTMC 和 MDP 的对称商,可根据相应的状态对称定义得到.

基于对称化简的抽象用其商表示抽象模型,商模型与原具体模型是强概率互模拟关系[28],因此其可保持 PCTL*或 CSL 性质,但构造对称商要比构造概率互模拟商所花费代价小.文献[78]扩展了文献[79−81]的方法,给出一种了面向建模语言层面的随机模型的对称化简方法,把对称化简和 MTBDD 的符号表示结合在一起,定义了 PRISM 建模语言关于 DTMC 和 MDP 语义的对称子集 SP(symmetric PRISM),并开发了工具 GRIP[82],自动地将 SP 模型转化为化简的模型.文献[83]采用动态对称化简[84,85]的方法给出了一种面向模型层面的随机模型对称化简方法,将其用于 DTMC,CTMC 和 MDP 模型抽象,提出了基于 MTBDD 的一种高效构造对称商模型的算法,并将其在 PRSIM 工具中实现.文献[86]认为:由于建模语言的限制约束的存在,以文献[78]为代表的面向语言层面的随机模型的对称化简方法很难用于复杂模型的化简.因此,文献[86]扩展了文献[78]的工作,提出了一种表达能力更强的 SPSL(symmetric probabilistic specification language)及其对应的求解抽象模型的转换算法,并在工具 GRIP 中实现,转换后的模型可以直接用 PRISM 工具验证.文献[87]把对称化简的抽象方法引入到显式状态的随机模型检验,提出一种较为完整的基于对称化简的随机模型抽象方法:定义了功能较为强大的 PSS 语言(probabilistic symmetric system language),通过抽取图,给出了自动发现 PSS 对称结构的方法,借助爬山算法和局部求解算法给出了有效的计算 PSS 对称等价类方法,并用改造的图遍历算法,可以直接从 PSS 构造对称商模型.

3.4 基于概率反例引导精化的抽象

上述抽象技术得到的抽象模型过于细化,即,抽象模型的状态空间并无明显减少,且对无限状态随机系统模型的抽象效果不理想.为此,基于概率反例引导精化的抽象(概率反例引导的抽象精化)把缓解经典模型检验中状态爆炸问题的 CEGAR(counterexample-guided abstraction-refinement)[88]技术应用于随机系统模型的抽象,其关键是如何定义反例和表示抽象模型,抽象模型本质上属于存在抽象(existential abstraction)[89].如图 3 所示,基于概率反例引导精化的抽象是一个自动化过程,其大体思想是:先构造一个较为粗糙的随机系统模型的抽象模型,用随机模型加以检验,用检验结果的无效反例增量精化初始的抽象模型,直到验证的性质在抽象模型上正确或者是产出了经过确认的有效反例.

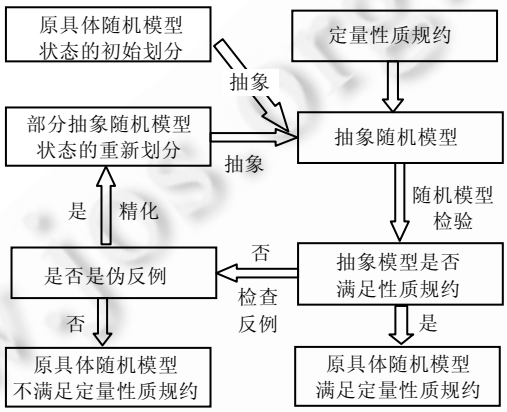

Fig.3 Probabilistic counterexample guided abstraction-refinement

图 3 概率反例引导精化的抽象框架

文献[89]可看作其雏形,文献[90]以 PRISM 语言规约的 MDP 为随机系统模型,用 MDP 作为抽象模型,使用类似路径结构的 DTMC 作为反例,通过基于插值的谓词推理来分析反例,给出了第一种概率反例引导精化的抽象方法,并实现了相应的工具 PASS[91],但其只能保持最大可达定量性质.文献[92]针对文献[90]保持性质范围的限制,并指出其原因是反例定义包含的信息不充分.为此,文献[92]使用图状结构 MDP 作为反例,给出了多项式时间的反例生成算法,用 MDP 作为抽象模型,提出了一种可验证 PCTL 定量安全和活性性质的概率反例引导精化的抽象方法.

3.5 基于误差引导精化的抽象

如图 4 所示,基于误差引导精化的抽象(误差引导的抽象精化)也是一种自动的抽象精化过程,过程本质上与基于概率反例引导精化的抽象有相似之处,其主要区别在于:该方法以抽象获得的误差(上界与下界的差别)作为抽象模型的精确度,如果误差过大,则需要精化抽象模型.

文献[93]可以看作是基于误差引导精化的抽象方法的雏形,其用 MDP 模型作具体 MDP 模型的抽象模型,只能保持最大概率可达类定量性质,其精化终止的条件是从抽象模型得到的可达概率满足给定的阈值,并实现了相应工具 RAPTURE.但由于该方法只能产生可达概率最小值的下界或最大值的上界这种方法,或许其更适合于 DTMC 模型的抽象.

针对 MDP 作为 MDP 抽象模型的不足——只能产生可达概率最小值的下界或最大值的上界,文献[94]首次提出用随机博弈[95]作为 MDP 的抽象模型,以区别原具体模型 MDP 的不确定性和因抽象形成的新的不确定性,其可以同时得到可达概率最小值或最大值的上下界.但是文献[94]方法的自动化程度不高,需要手动地划分模型状态空间.为此,文献[96]在文献[90,92–94]的基础上,给出了一种完整的自动的基于误差引导精化的抽象方法,其可看作是在文献[90,92]的抽象精化框架下,用文献[94]中的抽象模型表示方法、用文献[91]的精化方法精

化抽象模型.文献[96]的方法已被用来扩展工具 PASS[97],可以使其快速得到较小的抽象模型.但是文献[96]的方法也有局限性,即,只能验证定量可达性质,目前甚至还不能用于 PCTL 定量安全性质验证.

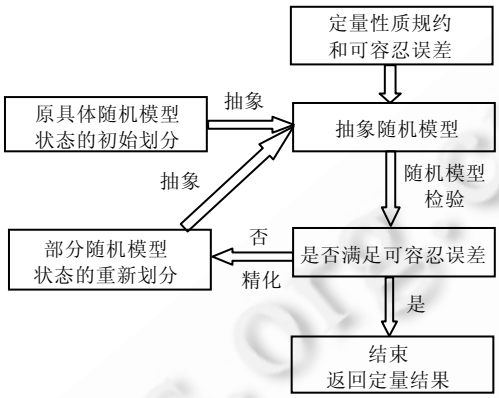

Fig.4 Error guided abstraction-refinement

图 4 误差引导精化的抽象框架

3.6 基于不确定结果引导精化的抽象

在文献[90,92,94,96]的抽象精化方法中,文献[92]是可保持性质范围最大的方法,但其也只能保持 PCTL 定量安全/活性的真值性质,即,抽象模型满足此性质,则原具体模型也满足此性质;若抽象模型不满足此性质,原具体模型也有可能满足此性质.基于不确定结果引导精化的抽象(3 值抽象精化)技术可以弥补上述不足,其用 3 值抽象方法[98,99]扩展抽象-精化框架,大体思想如图 5 所示.

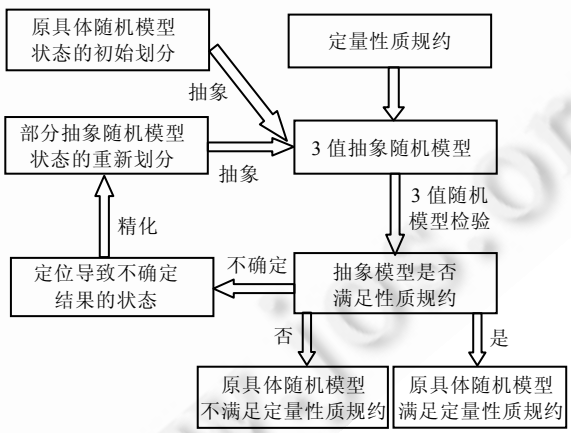

Fig.5 Indefinition guided abstraction-refinement

图 5 不确定结果引导精化的抽象框架

文献[100]最先给出随机模型的 3 值抽象精化方法,以 DTMC 作为随机系统模型,用概率区间[101]表示迁移概率:给出一种抽象马尔可夫链模型(abstract Markov chain,简称 AMC)并用其表示抽象模型,区间下界可用于表示下界逼近(under approximation),区间上界可用于表示上界逼近(over approximation)信息,原具体模型与抽象模型间依然是概率模拟关系;给出了 PCTL 的 3 值语义及其随机模型检验 AMC 算法,其可保持 PCTL 性质,即,此类性质满足理想抽象技术条件(2),但验证结果有可能是不确定结果.

文献[102]将文献[100]的方法应用于 CTMC 抽象.文献[103,104]对文献[100,102]的方法进行了改造和扩展,针对其抽象过程中的不精确性,研究了规范化 DTMC 和统一化 CTMC 的 3 值抽象方法,为随机模型的 3 值抽象

提供了严格的理念基础.

4 总结与展望

本文以随机模型检验中的模型抽象技术为研究对象,对其进行了问题描述,归纳了其主要的研究方向和进展.在模型抽象技术中,每种技术均有其优点和缺点,很难同时满足理想的模型抽象技术的两个条件.它们之间的对比和关系如下:

(1) 基于模拟关系的抽象是最基本的抽象技术,其他抽象技术往往可以归结为某种概率模拟关系.

(2) 概率互模拟、对称化简、偏序化简属于准确抽象,其构造的是一个等价抽象模型,可保持的性质范围较大,但抽象模型过于细化造成其状态空间没有明显比原具体模型小;其余抽象技术属于近似抽象,其构造的抽象模型较为粗糙,但可保持的性质范围较小.

(3) 对于准确抽象,抽象模型往往与原具体模型属于同一类模型,其随机模型检验过程和求解反例方法可以使用一般的随机模型检验理论验证和反例求解.对于近似抽象,抽象模型是 MDP 的可直接使用一般的随机模型检验理论验证和反例求解;抽象模型是随机博弈模型的可用线性规划或值迭代等方法求解,但尚未涉及反倒求解;抽象模型是 AMC 的要用 3 值随机模型检验方法验证,但也没有反例相关研究.

(4) 在准确抽象技术中,基于对称化简和偏序化简的抽象技术只能对具备相应特点的随机系统模型进行抽象.

(5) 在近似抽象技术中,以 MDP 和 AMC 作为抽象模型,其拥有相同规模的抽象模型状态空间.但 AMC 作为抽象模型会使其迁移变少,且其抽象程度更高,即概率界限值精确程度低.

(6) 在近似抽象技术中,除 3 值抽象外,其余抽象技术均可看作是 2 值语义抽象范畴,其本质上属于存在抽象,是对具体模型的上界逼近,即,抽象模型比具体模型拥有更多的行为;而 3 值抽象则可以同时对具体模型进行上界逼近和下界逼近,但需要定义相关的 3 值抽象模型及其 3 值时序逻辑语义和与之相应的随机模型检验算法.

虽然目前面向随机模型检验的模型抽象技术已取得了一定的进展,但其要么是可保持性质范围太小,要么构造的抽象模型过于细化,即,还没能给出解决 MA4SMC 问题的满意答案.我们认为,可以从以下几个方面来应对上述挑战:

(1) 抽象技术的完备性理论

面向随机模型检验的抽象技术的最基础理论是,该抽象技术是否具有完备性和有效性.完备性是指:若具体模型 M 满足性质 Φ,则存在一个小的有限状态模型 M^{α} 作为模型 M 的抽象,且 M^{α} 满足性质 Φ;有效性是指:若抽象模型 M^{α} 满足性质 Φ,则具体模型 M 满足性质 Φ.目前的抽象技术研究仅涉及到其有效性,除文献[105,106]之外,还几乎没有关于完备抽象技术的研究工作.

(2) 抽象技术本身的优化

1) 寻找更好的概率模拟关系定义及其有效的构造商模型的算法,使得抽象模型状态空间少且包含更丰富的信息.以下文献对该问题做了一些尝试工作:文献[107]用抽象解释[108,109]刻画概率互模拟和概率模拟,为算法求解概率互模拟和概率模拟提供了一个通用框架,并基于此给出了一种新的高效的概率模拟算法;文献[43]给出了一种新的关于 PLTS(probabilistic labeled transition system)的概率互模拟和概率模拟定义,使其可以准确地刻画 PCTL*逻辑等价.

2) 加强/消弱足够集满足的条件,寻求偏序化简可满足性质范围与抽象模型状态空间大小间的平衡.

3) 在抽象-精化技术中,研究更好的方法来实现精化抽象模型,如局部精化[110]或者启发式引导精化的抽象技术.

4) 组合抽象技术,将组合验证方法,如假设-保证推理与抽象精化技术相结合,形成组合式抽象精化方法,如文献[111,112]的工作属于这一方向的探索.

(3) 抽象技术的融合或集成

抽象技术各有其优缺点,可以考虑将两种或两种以上抽象技术集成使用,以扬长避短,得到更好的抽象结果:

1) 概率互模拟与抽象精化方法的集成,如文献[106]用增量计算 may/must 商来自动实现抽象精化方法.
2) 偏序与对称方法的融合,偏序化简是去掉随机系统模型的迁移冗余,而对称化简是去掉随机系统模型的冗余状态,这为将两种方法融合提供了理论上的可能性.

(4) 新的随机模型抽象技术

研究针对随机系统模型的新的抽象技术,是应对挑战的重要方法.如,文献[66,113]定义了一种新的关于概率自动机的 confluence 概念,并证明基于 confluence 化简的抽象可保持分支互模拟,并可以 on-the-fly 使用;学习算法[114]在经典模型检验的抽象中取得了较好的效果,并且文献[115]的研究已说明,学习算法用在构造随机模型的可行性.我们认为,基于学习算法的抽象在随机模型的抽象中应该会有所作为.

(5) 抽象技术应用于其他缓解状态爆炸问题的方法

1) 假设保证推理式组合随机模型检验.假设保证推理式组合随机模型检验[112,116]的关键是产生假设,理论上,抽象技术可用于产生假设[112].
2) 限界随机模型检验.限界随机模型检验[29,30]是通过搜索有限步长随机模型的反例,得到验证结果的方法.目前的抽象-精化方法仅应用于一般的随机模型检验,若将其用于限界随机模型检验,应该会使限界随机模型检验更快,并且也可以用限界随机模型检验产生精化需要的相关信息.

(6) 抽象技术应用范围的扩展

目前,面向随机模型检验的抽象技术还只是用于基本的随机系统模型,如 DTMC,CTMC 和 MDP,虽开始应用于概率时间自动机验证[117],但对于一些复杂的或 high-level 的随机系统模型,还有待进一步地研究如何将抽象技术应用于此类随机模型的验证,如连续时间马尔可夫决策过程[18,19]、交互式马尔可夫链[20]、随机混成系统模型[118]、非确定概率 Petri 网[15]和随机多人博弈模型[119]等.

致谢 本文的工作得到江苏省软件新技术与产业化协同创新中心的资助,在此表示感谢.

References:

[1] Liu K, Shan ZG, Wang J, He JF, Zhang ZT, Qin YW. Overview on major research plan of trustworthy software. Bulletin of National Natural Science Foundation of China, 2008,22(3):145−151 (in Chinese with English abstract).

[2] Baier C, Katoen JP. Principles of Model Checking. London: The MIT Press, 2008. 745−790.

[3] Clarke EM, Emerson EA. Design and synthesis of synchronization skeletons using branching time temporal logic. In: Proc. of the Workshop on Logic of Programs. Berlin, Heidelberg: Springer-Verlag, 1981. 52−71. [doi: 10.1007/BFb0025774]

[4] Queille JP, Sifakis J. Specification and verification of concurrent systems in CESAR. In: Proc. of the 5th Colloquium on Int'l Symp. on Programming. Berlin, Heidelberg: Springer-Verlag, 1982. 337−351. [doi: 10.1007/3-540-11494-7_22]

[5] Lin HM, Zhang WH. Model checking: Theories, techniques and applications. Acta Electronica Sinica, 2002,30(12A):1907−1912 (in Chinese with English abstract).

[6] Clarke EM, Emerson EA, Sifakis J. Model checking: Algorithmic verification and debugging. Communications of the ACM, 2009, 52(11):74−84. [doi: 10.1145/1592761.1592781]

[7] Kwiatkowska M, Norman G, Parker D. Stochastic model checking. In: Proc. of the 7th Int'l Conf. on Formal Methods for Performance Evaluation. Berlin, Heidelberg: Springer-Verlag, 2007. 220−270. [doi: 10.1007/978-3-540-72522-0_6]

[8] Ndukwu U, Mclver A. An expectation transformer approach to predicate abstraction and data independence for probabilistic programs. Electronic Proceedings in Theoretical Computer Science, 2010,28:129−143. [doi: 10.4204/EPTCS.28.9]

[9] Baier C, Haverkort BR, Hermanns H, Katoen JP. Performance evaluation and model checking join forces. Communication of the ACM, 2010,53(9):74−85. [doi: 10.1145/1810891.1810912]

[10] Duflot M, Kwiatkowska M, Norman G, Parker D, Peyronnet S, Picaronny C, Sproston J. Practical applications of probabilistic model checking to communication protocols. In: Gnesi S, Margaria T, eds. Proc. of the Formal Methods for Industrial Critical Systems: A Survey of Applications. Washington: IEEE Computer Society Press, 2012. 133−150. [doi: 10.1002/9781118459898.ch7]

[11] Calinescu R, Grunske L, Kwiatkowska M, Mirandola R, Tamburrelli G. Dynamic QoS management and optimisation in service-based systems. IEEE Trans. on Software Engineering, 2011,37(3):387−409. [doi: 10.1109/TSE.2010.92]

[12] Kwiatkowska M, Norman G, Parker D. Probabilistic model checking for systems biology. In: Iyengar MS, ed. Proc. of the Symbolic Systems Biology. Sudbury: Jones and Bartlett Publishers, 2010. 31−59.

[13] Clarke EM, Grumberg O, Long DE. Model checking and abstraction. In: Proc. of the 19th ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages. New York: ACM Press, 1992. 343−354.

[14] Huth M. An abstraction framework for mixed non-deterministic and probabilistic systems. In: Baier C, Haverkort B, Hermanns H, Katoen J P, Siegle M, eds. Proc. of the Validation of Stochastic Systems. Berlin, Heidelberg: Springer-Verlag, 2004. 419−444. [doi: 10.1007/978-3-540-24611-4_12]

[15] Liu Y, Miao HK, Zeng HW, Ma Y, Liu P. Nondeterministic probabilistic Petri net: A new method to study qualitative and quantitative behaviors of system. Journal of Computer Science and Technology, 2013,28(1):203−216. [doi: 10.1007/s11390-013-1323-7]

[16] Beauquier D. On probabilistic timed automata. Theoretical Computer Science, 2003,292(1):65−84. [doi: 10.1016/S0304-3975(01)00215-8]

[17] Kwiatkowska M, Norman G, Segala R, Sproston J. Automatic verification of real-time systems with discrete probability distributions. Theoretical Computer Science, 2002,282(1):101−150. [doi: 10.1016/S0304-3975(01)00046-9]

[18] Howard RA. Dynamic Programming and Markov Processes. Hoboken: John Wiley and Sons, 1960.

[19] Bertsekas DP. Dynamic Programming and Optimal Control. 3rd ed., Cambridge: MIT Press, 2011.

[20] Hermanns H. Interactive Markov Chains and the Quest for Quantified Quality. Berlin, Heidelberg: Springer-Verlag, 2002.

[21] Kwiatkowska M, Norman G, Sproston J, Wang F. Symbolic model checking for probabilistic timed automata. Information and Computation, 2007,205(7):1027−1077. [doi: 10.1016/j.ic.2007.01.004]

[22] de Alfaro L. Formal verification of probabilistic systems [Ph.D. Thesis]. Stanford: Stanford University, 1997.

[23] Cloth L, Haverkort B, Hermanns H, Katoen JP, Baier C. Model checking pathCSL. In: Proc. of the 6th Int'l Workshop Performability Modeling of Computer and Communication Systems. 2003. 19−22.

[24] Kuntz M, Siegle M. A stochastic extension of the logic PDL. In: Proc. of the 6th Int'l Workshop Performability Modeling of Computer and Communication Systems. 2003. 58−61.

[25] Baier C, Cloth L, Haverkort BR, Kuntz M, Siegle M. Model checking Markov chains with actions and state labels. IEEE Trans. on Software Engineering, 2007,33(4):209−224. [doi: 10.1109/TSE.2007.36]

[26] Liu Y. Quantitative verification of trustworthy service flow by stochastic model checking. Journal of Software Engineering, 2014, 8(3):152−168. [doi: 10.3923/jse.2014.152.168]

[27] de Alfaro L, Roy P. Magnifying-Lens abstraction for Markov decision processes. In: Proc. of the 19th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2007. 325−338. [doi: 10.1007/978-3-540-73368-3_38]

[28] Didier F, Henzinger T, Mateescu M, Wolf V. Sabre: A tool for stochastic analysis of biochemical reaction networks. In: Proc. of the 7th Int'l Conf. on Quantitative Evaluation of Systems. Washington: IEEE Computer Society Press, 2010. 193−194. [doi: 10.1109/QEST.2010.33]

[29] Della PG, Intrigila B, Melatti I, Tronci E, Venturini ZM. Bounded probabilistic model checking with the murphi verifier. In: Proc. of the 5th Int'l Conf. on Formal Methods in Computer-Aided Design. Berlin, Heidelberg: Springer-Verlag, 2004. 214−229. [doi: 10.1007/978-3-540-30494-4_16]

[30] Zhou CH, Liu ZF, Wang CD. Bounded model checking for probabilistic computation tree logic. Ruan Jian Xue Bao/Journal of Software, 2012,23(7):1656−1668 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4089.htm [doi: 10.3724/SP.J.1001.2012.04089]

[31] Filieri A, Ghezzi C, Tamburrelli G. Run-Time efficient probabilistic model checking. In: Proc. of the 33rd ACM/IEEE Int'l Conf. on Software Engineering. Washington: IEEE Computer Society Press, 2011. 341−350. [doi: 10.1145/1985793.1985840]

[32] Larsen K, Skou A. Bisimulation through probabilistic testing. Information and Computation, 1991,94(1):1−28. [doi: 10.1016/0890-5401(91)90030-6]

[33] Milner R. A Calculus of Communicating Systems. Berlin, Heidelberg: Springer-Verlag, 1980.

[34] Park D. Concurrency and automata on infinite sequences. In: Proc. of the 5th GI- Conf. on Theoretical Computer Science. Berlin, Heidelberg: Springer-Verlag, 1981. 167−183. [doi: 10.1007/BFb0017309]

[35] Buchholz P. Exact and ordinary lumpability infinite Markov chains. Journal of Applied Probability, 1994,31(1):59−75. [doi: 10.2307/3215235]

[36] Segala R, Lynch N. Probabilistic simulations for probabilistic processes. Nordic Journal of Computing, 1995,2(2):250−273.

[37] Paige R, Tarjan R. Three partition refinement algorithms. SIAM Journal of Computation, 1987,16(6):973−989. [doi: 10.1137/0216062]

[38] Huynh T, Tian L. On some equivalence relations for probabilistic processes. Fundamenta Informaticae, 1992,17(3):211−234.

[39] Aziz A, Singhal V, Balarin F, Brayton RK, Sangiovanni-Vincentelli AL. It usually works: The temporal logic of stochastic systems. In: Proc. of the 7th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 1995. 155−165. [doi: 10.1007/3-540-60045-0_48]

[40] Hermanns H, Katoen J. Automated compositional Markov chain generation for a plain-old telephone system. Science of Computer Programming, 2000,36(1):97−127. [doi: 10.1016/S0167-6423(99)00019-2]

[41] Derisavi S. A symbolic algorithm for optimal Markov chain lumping. In: Proc. of the 13th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 2007. 139−154. [doi: 10.1007/978-3-540-71209-1_13]

[42] Christian D, Katoen JP, Parker D. SMT-Based bisimulation minimization of markov models. In: Proc. of the 14th Int'l Conf. on Verification, Model Checking and Abstract Interpretation. Berlin, Heidelberg: Springer-Verlag, 2013. 28−47. [doi: 10.1007/978-3-642-35873-9_5]

[43] Song L, Zhang LJ, Godskesen JC, Nielson F. Bisimulations meet PCTL equivalences for probabilistic automata. Logical Methods in Computer Science, 2013,9(2):1−34.

[44] Baier C, Hermanns H. Weak bisimulation for fully probabilistic processes. In: Proc. of the 9th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 1997. 119−130. [doi: 10.1007/3-540-63166-6_14]

[45] Philippou A, Lee I, Sokolsky O. Weak bisimulation for probabilistic systems. In: Proc. of the 11th Int'l Conf. on Concurrency Theory. Berlin, Heidelberg: Springer-Verlag, 2000. 334−349. [doi: 10.1007/3-540-44618-4_25]

[46] Cattani S, Segala R. Decision algorithms for probabilistic bisimulation. In: Proc. of the 13th Int'l Conf. on Concurrency Theory. Berlin, Heidelberg: Springer-Verlag, 2002. 371−385. [doi: 10.1007/3-540-45694-5_25]

[47] Hermanns H, Turrini A. Deciding probabilistic automata weak bisimulation in polynomial time. In: Proc. of the 32nd Int'l Conf. on Foundations of Software Technology and Theoretical Computer Science. Saarbrücken/Wadern: Dagstuhl Publishing, 2012. 435−447.

[48] Jonsson B, Larsen KG. Specification and refinement of probabilistic processes. In: Proc. of the 6th Annual IEEE Symp. on Logic in Computer Science. Washington: IEEE Computer Society Press, 1991. 266−277. [doi: 10.1109/LICS.1991.151651]

[49] Milner R. An algebraic definition of simulation between programs. In: Proc. of the 2nd Int'l Joint Conf. on Artificial Intelligence. London: William Kaufmann Inc., 1971. 481−489.

[50] Clarke EM, Grumberg O, Long DE. Model checking and abstraction. ACM Trans. on Programming Languages and Systems, 1994, 16(5):1512−1542. [doi: 10.1145/186025.186051]

[51] Baier C, Katoen JP, Hermanns H, Wolf V. Comparative branching-time semantics for Markov chains. Information and Computation, 2005,200(2):149−214. [doi: 10.1016/j.ic.2005.03.001]

[52] Zhang LJ. Decision algorithms for probabilistic simulations [Ph.D. Thesis]. Saarbrücken: Saarland University, 2008.

[53] Gallo G, Grigoriadis MD, Tarjan RE. A fast parametric maximum flow algorithm and applications. SIAM Journal on Computing, 1989,18(1):30−55. [doi: 10.1137/0218003]

[54] Zhang LJ, Hermanns H, Eisenbrand F, Jansen DN. Flow faster: Efficient decision algorithms for probabilistic simulations. Logical Methods in Computer Science, 2008,4(4:6):1−23. [doi: 10.2168/LMCS-4(4:6)2008]

[55] Baier C, Katoen JP, Hermanns H, Haverkort B. Simulation for continuous time Markov chains. In: Proc. of the 13th Int'l Conf. on Concurrency Theory. Berlin, Heidelberg: Springer-Verlag, 2002. 338−354. [doi: 10.1007/3-540-45694-5_23]

[56] Gerth R, Kuiper R, Peled D, Penczek W. A partial order approach to branching time logic model checking. In: Proc. of the 3rd Israel Symp. on the Theory of Computing Systems. Washington: IEEE Computer Society Press, 1995. 130−139. [doi: 10.1109/ISTCS.1995.377038]

[57] Peled D. All from one, one for all: On model checking using representatives. In: Proc. of the 5th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 1993. 409−423. [doi: 10.1007/3-540-56922-7_34]

[58] Peled D, Pratt V, Holzmann G. Partial order methods in verification. In: DIMACS Series in Discrete Mathematics and Theoretical Computer Science. 1997.

[59] Valmari A. A stubborn attack on state explosion. Formal Methods in System Design, 1992,1(4):297−322. [doi: 10.1007/BF00709154]

[60] Peled D. Partial order reduction: Linear and branching temporal logics and process algebras. In: Proc. of the DIMACS Workshop on Partial Order Methods in Verification. New York: AMS Press, 1996. 79−88.

[61] Baier C, Groser M, Ciesinski F. Partial order reduction for probabilistic systems. In: Proc. of the 1st Int'l Conf. on Quantitative Evaluation of Systems. Washington: IEEE Computer Society Press, 2004. 230−239. [doi: 10.1109/QEST.2004.1348037]

[62] D'Argenio PR, Niebert P. Partial order reduction on concurrent probabilistic programs. In: Proc. of the 1st Int'l Conf. on Quantitative Evaluation of Systems. Washington: IEEE Computer Society Press, 2004. 240−249. [doi: 10.1109/QEST.2004.1348038]

[63] Ciesinski F. High-Level modelling and efficient analysis of randomized protocols [Ph.D. Thesis]. Dresden: Dresden University of Technology, 2011.

[64] Baier C, D'Argenio P, Groesser M. Partial order reduction for probabilistic branching time. Electronic Notes in Theoretical Computer Science, 2005,153(2):97−116. [doi: 10.1016/j.entcs.2005.10.034]

[65] Groesser M, Baier C. Partial order reduction for markov decision processes: A survey. In: Proc. of the 4th Int'l Conf. on Formal Methods for Components and Objects. Berlin, Heidelberg: Springer-Verlag, 2005. 408−427. [doi: 10.1007/11804192_19]

[66] Hansen H, Timmer M. A comparison of confluence and ample sets in probabilistic and non-probabilistic branching time. Theoretical Computer Science, 2014,538:103−123. [doi: 10.1016/j.tcs.2013.07.014]

[67] Fernandez-Diaz A, Baier C, Benac-Earle C, Fredlund LA. Static partial order reduction for probabilistic concurrent systems. In: Proc. of the 9th Int'l Conf. on Quantitative Evaluation of Systems. Washington: IEEE Computer Science Press, 2012. 104−113. [doi: 10.1109/QEST.2012.22]

[68] Kurshan R, Levin V, Minea M, Peled D, Yenigün H. Static partial order reduction. In: Proc. of the 4th Int'l Conf. on Tools and Algorithms for Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 1998. 345−357. [doi: 10.1007/BFb0054182]

[69] Kurshan RP, Levin V, Minea M, Peled D, Yenigün H. Combining software and hardware verification techniques. Formal Methods in System Design, 2002,21(3):251−280. [doi: 10.1023/A:1020383505582]

[70] Hansen H, Kwiatkowska M, Qu H. Partial order reduction for model checking Markov decision processes under unconditional fairness. In: Proc. of the 8th Int'l Conf. on Quantitative Evaluation of SysTems. Washington: IEEE Computer Science Press, 2011. 203−212. [doi: 10.1109/QEST.2011.35]

[71] Hansen H, Wang X. Compositional analysis for weak stubborn sets. In: Proc. of the Int'l Conf. on Application of Concurrency to System Design. Washington: IEEE Computer Science Press, 2011. 36−43. [doi: 10.1109/ACSD.2011.25]

[72] Nuutila E, Soisalon-soininen E. On finding the strongly connected components in a directed graph. Information Processing Letters, 1994,49(1):9−14. [doi: 10.1016/0020-0190(94)90047-7]

[73] Kwiatkowska M, Norman G, Parker D. PRISM 4.0: Verification of probabilistic real-time systems. In: Proc. of the 23rd Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2011. 585−591. [doi: 10.1007/978-3-642-22110-1_47]

[74] Clarke EM, Jha S, Enders R, Filkorn T. Exploiting symmetry in temporal logic model checking. Formal Methods in System Design, 1996,9(1-2):77−104. [doi: 10.1007/BF00625969]

[75] Emerson EA, Sistla A. Symmetry and model checking. Formal Methods in System Design, 1996,9(1-2):105−131. [doi: 10.1007/BF00625970]

[76] Ip CN, Dill D. Better verification through symmetry. Formal Methods in System Design, 1996,9(1-2):41−75. [doi: 10.1007/BF00625968]

[77] Miller A, Donaldson A, Calder M. Symmetry in temporal logic model checking. ACM Computing Surveys, 2006,38(3:8):1−40. [doi: 10.1145/1132960.1132962]

[78] Donaldson A, Miller A. Symmetry reduction for probabilistic model checking using generic representatives. In: Proc. of the 4th Int'l Conf. on Automated Technology for Verification and Analysis. Berlin, Heidelberg: Springer-Verlag, 2006. 9−23. [doi: 10.1007/11901914_4]

[79] Emerson EA, Trefler R. From asymmetry to full symmetry: New techniques for symmetry reduction in model checking. In: Proc. of the 8th IFIP WG Advanced Research Working Conf. on Correct Hardware Design and Verification Methods. Berlin, Heidelberg: Springer-Verlag, 1999. 142−156. [doi: 10.1007/3-540-48153-2_12]

[80] Emerson EA, Wahl T. On combining symmetry reduction and symbolic representation for efficient model checking. In: Proc. of the 12th IFIP WG Advanced Research Working Conf. on Correct Hardware Design and Verification Methods. Berlin, Heidelberg: Springer-Verlag, 2003. 216−230. [doi: 10.1007/978-3-540-39724-3_20]

[81] Emerson EA, Wahl T. Efficient reduction techniques for systems with many components. Electronic Notes in Theoretical Computer Science, 2005,130:379−399. [doi: 10.1016/j.entcs.2005.03.019]

[82] Donaldson A, Miller A, Parker D. GRIP: Generic representatives in PRISM. In: Proc. of the 4th Int'l Conf. on the Quantitative Evaluation of Systems. Washington: IEEE Computer Science Press, 2007. 115−116. [doi: 10.1109/QEST.2007.30]

[83] Kwiatkowska M, Norman G, Parker D. Symmetry reduction for probabilistic model checking. In: Proc. of the 18th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2006. 234−248. [doi: 10.1007/11817963_23]

[84] Emerson EA, Wahl T. Dynamic symmetry reduction. In: Proc. of the 11th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 2005. 382−396. [doi: 10.1007/978-3-540-31980-1_25]

[85] Wahl T, Blanc N, Emerson EA. SVISS: Symbolic verification of symmetric systems. In: Proc. of the 14th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 2008. 459−462. [doi: 10.1007/978-3-540-78800-3_34]

[86] Donaldson A, Miller A, Parker D. Language-Level symmetry reduction for probabilistic model checking. In: Proc. of the 6th Int'l Conf. on Quantitative Evaluation of Systems. Washington: IEEE Computer Science Press, 2009. 289−298. [doi: 10.1109/QEST.2009.21]

[87] Christopher P. Probabilistic symmetry reduction [Ph.D. Thesis]. University of Glasgow, 2012.

[88] Clarke EM, Grumberg O, Jha S, Lu Y, Veith H. Counterexample-Guided abstraction refinement. In: Proc. of the 12th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2000. 154−169. [doi: 10.1007/10722167_15]

[89] Wachter B, Zhang LJ, Hermanns H. Probabilistic model checking modulo theories. In: Proc. of the 4th Int'l Conf. on Quantitative Evaluation of Systems. Washington: IEEE Computer Science Press, 2007. 129−138. [doi: 10.1109/QEST.2007.10]

[90] Hermanns, H, Wachter, B, Zhang L. Probabilistic CEGAR. In: Proc. of 2008 the 20th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2007. 162−175. [doi: 10.1007/978-3-540-70545-1_16]

[91] Hahn EM, Hermanns H, Wachter B, Zhang L. PASS: Abstraction refinement for infinite probabilistic models. In: Proc. of the 16th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 2010. 353−357. [doi: 10.1007/978-3-642-12002-2_30]

[92] Chadha R, Viswanathan M. A counterexample guided abstraction-refinement framework for Markov decision processes. ACM Trans. on Computational Logic, 2010,12(1):1−49. [doi: 10.1145/1838552.1838553]

[93] D'Argenio P, Jeannet B, Jensen H, Larsen K. Reachability analysis of probabilistic systems by successive refinements. In: Proc. of the Joint Int'l Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification. Berlin, Heidelberg: Springer-Verlag, 2001. 39−56. [doi: 10.1007/3-540-44804-7_3]

[94] Kwiatkowska M, Norman G, Parker D. Game-Based abstraction for Markov decision processes. In: Proc. of the 3rd Int'l Conf. on Quantitative Evaluation of Systems. Washington: IEEE Computer Science Press, 2006. 157−166. [doi: 10.1109/QEST.2006.19]

[95] Condon A. The complexity of stochastic games. Information and Computation, 1992,96(2):203−224. [doi: 10.1016/0890-5401(92)90048-K]

[96] Kattenbelt M, Kwiatkowska M, Norman G, Parker D. A game-based abstraction refinement framework for Markov decision processes. Formal Methods in System Design, 2010,36(3):246−280. [doi: 10.1007/s10703-010-0097-6]

[97] Wachter B, Zhang LJ. Best probabilistic transformers. In: Proc. of the 11th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation. Berlin, Heidelberg: Springer-Verlag, 2010. 362−379. [doi: 10.1007/978-3-642-11319-2_26]

[98] Huth M, Jagadeesan R, Schmidt D. Modal transition systems: A foundation for three-valued program analysis. In: Proc. of the 10th European Symp. on Programming Languages and Systems. Berlin, Heidelberg: Springer-Verlag, 2001. 155−169. [doi: 10.1007/3-540-45309-1_11]

[99] Godefroid P, Jagadeesan R. On the expressiveness of 3-valued models. In: Proc. of the 4th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation. Berlin, Heidelberg: Springer-Verlag, 2003. 206−222. [doi: 10.1007/3-540-36384-X_18]

[100] Fecher H, Leucker, M, Wolf, V. Don't know in probabilistic systems. In: Proc. of the 13th Int'l Conf. on Model Checking Software. Berlin, Heidelberg: Springer-Verlag, 2006. 71−88. [doi: 10.1007/11691617_5]

[101] Wang Y. Reasoning about uncertain information compositionally. In: Proc. of the 3rd Int'l School and Symp. on Real-Time and Fault-Tolerant Systems. Berlin, Heidelberg: Springer-Verlag, 1994. 680−693.

[102] Katoen JP, Klink D, Leucker M, Wolf V. Three-Valued abstraction for continuous-time Markov chains. In: Proc. 19th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2007. 316−329. [doi: 10.1007/978-3-540-73368-3_37]

[103] Katoen JP, Klink D, Leucker M, Wolf V. Three-Valued abstraction for probabilistic systems. Journal on Logic and Algebraic Programming, 2012,81(4):356−389. [doi: 10.1016/j.jlap.2012.03.007]

[104] Klink D. Three-Valued abstraction for stochastic systems [Ph.D. Thesis]. Aachen: RWTH Aachen University, 2010.

[105] Huth M, Piterman N, Wagner D. p-Automata: New foundations for discrete-time probabilistic verification. Performance Evaluation, 2012,69(7-8):356−378. [doi: 10.1016/j.peva.2012.05.005]

[106] Song L, Zhang LJ, Hermanns H, Godskesen JC. Incremental bisimulation abstraction refinement. In: Proc. of the 13th Int'l Conf. on Application of Concurrency to System Design. Washington: IEEE Computer Science Press, 2013. 98−107. [doi: 10.1109/ACSD.2013.5]

[107] Crafa S, Ranzato F. Bisimulation and simulation algorithms on probabilistic transition systems by abstract interpretation. Formal Methods in System Design, 2012,40(3):356−376. [doi: 10.1007/s10703-012-0147-3]

[108] Cousot P, Cousot R. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Proc. of the 4th ACM SIGACT-SIGPLAN Symp. on Principles of programming languages. New York: ACM Press, 1977. 238−252. [doi: 10.1145/512950.512973]

[109] Cousot P, Cousot R. Systematic design of program analysis frameworks. In: Proc. of the 6th ACM SIGACT-SIGPLAN Symp. on Principles of programming languages. New York: ACM Press, 1979. 269−282. [doi: 10.1145/567752.567778]

[110] Draeger K, Kwiatkowska M, Parker D, Qu HY. Local abstraction refinement for probabilistic timed programs. Theoretical Computer Science, 2014,538:37−53. [doi: 10.1016/j.tcs.2013.07.013]

[111] Sher F, Katoen JP. Compositional abstraction techniques for probabilistic automata. In: Proc. of the 7th IFIP Int'l Conf. on Theoretical Computer Science. Berlin, Heidelberg: Springer-Verlag, 2012. 325−341. [doi: 10.1007/978-3-642-33475-7_23]

[112] Komuravelli A, Pasareanu CS, Clarke EM. Assume-Guarantee abstraction refinement for probabilistic systems. In: Proc. of the 24th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2012. 310−326. [doi: 10.1007/978-3-642-31424-7_25]

[113] Timmer M, Stoelinga M, van de Pol J. Confluence reduction for probabilistic systems. In: Proc. of the 17th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 2011. 311−325. [doi: 10.1007/978-3-642-19835-9_29]

[114] Mitchell TM. Machine Learning. New York: McGraw-Hill Science/Engineering/Math, 1997.

[115] Mao H, Chen Y, Jaeger M, Nielsen TD, Larsen KG, Nielsen B. Learning Markov decision processes for model checking. Electronic Proc. in Theoretical Computer Science, 2012,103:49−63. [doi: 10.4204/EPTCS.103.6]

[116] Kwiatkowska M, Norman G, Parker D, Qu H. Assume-Guarantee verification for probabilistic systems. In: Proc. of the 16th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 2010. 23−37. [doi: 10.1007/978-3-642-12002-2_3]

[117] Fioriti LMF, Hermanns H. Heuristics for probabilistic timed automata with abstraction refinement. In: Proc. of the 16th Int'l GI/ITG Conf. on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance. Berlin, Heidelberg: Springer-Verlag, 2012. 151−165. [doi: 10.1007/978-3-642-28540-0_11]

[118] Hahn EM. Model checking stochastic hybrid systems [Ph.D. Thesis]. Saarbrücken: Saarland University, 2013.

[119] Simaitis A. Automatic verification of competitive stochastic systems [Ph.D. Thesis]. Oxford: University of Oxford, 2014.

附中文参考文献:

[1] 刘克,单志广,王戟,何积丰,张兆田,秦玉文."可信软件基础研究"重大研究计划综述.中国科学基金,2008,22(3):145−151.

[5] 林惠民,张文辉.模型检测:理论、方法与应用.电子学报,2002,30(12A):1907−1912.

[30] 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. http://www.jos.org.cn/1000-9825/4089.htm [doi: 10.3724/SP.J.1001.2012.04089]

刘阳(1981－),男,山东滕州人,博士,主要研究领域为软件工程,形式化验证.

马艳(1981－),女,博士生,主要研究领域为软件工程,软件验证.

李宣东(1963－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为软件建模与分析,软件测试与验证.