随机模型检验研究

刘 阳[1),2)] 李宣东[1)] 马 艳[3)] 王林章[1)]

[1)](南京大学计算机软件新技术国家重点实验室 南京 210093)

[2)](新加坡国立大学计算学院计算机科学系 新加坡 117417)

[3)](南京航空航天大学计算机科学与技术学院 南京 210016)

摘 要 随机模型检验作为模型检验理论的延伸和推广,可用于验证分析系统模型的定性或定量性质,其已经应用到随机分布式算法验证、通信协议性能分析甚至是系统生物学等跨学科领域.从 20 世纪 90 年代末至今,随机模型检验引起了形式验证等领域的广泛关注,并取得了很大的进展.该文追溯了随机模型检验的渊源,系统地概括了其最基本的原理及几类典型的 PCTL、概率的 LTL、PCTL* 和 CSL 模型检验随机系统的算法框架.然后归纳总结了随机模型检验的主要研究方向及其进展,分析了基于随机模型检验的验证过程及其优势与劣势,并分类列出了目前出现的随机模型检验工具.最后介绍了随机模型检验的应用领域并指出了其未来的应用挑战.

关键词 形式验证;马尔可夫随机过程;随机模型检验;定量分析

中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2015.02145

Survey for Stochastic Model Checking

LIU Yang[1),2)] LI Xuan-Dong[1)] MA Yan[3)] WANG Lin-Zhang[1)]

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

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

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

Abstract Stochastic model checking is extension and generalization of the theory of model checking, which can verify and analyze system model quantitatively and qualitatively, and has been applied in the areas of verification of randomized distributed algorithms, performance analysis of communication protocols, and even the cross-disciplinary fields such as systems biology. Since the late 1990s, stochastic model checking has received widespread concern in the formal verification filed, and has made great progress. In this paper, we retrospect the origin of stochastic model checking, and discuss the basic principle of stochastic model checking systematically including the PCTL, LTL with probability bounds, PCTL* and CSL model checking algorithm. Then we summarize the main research direction and progress of stochastic model checking in recent years, analyze the verification process and advantages/disadvantages of stochastic model checking deeply, classify and list tools for stochastic model checking. Finally, we introduce the application areas of stochastic model checking and point out its future challenge.

Keywords formal verification; Markov stochastic process; stochastic model checking; quantitative analysis

收稿日期:2013-06-05;最终修改稿收到日期:2015-04-10.本课题得到国家自然科学基金项目(61303022,91318301)、中国博士后科学基金项目(2013M531328)、山东省自然科学基金项目(ZR2012FQ013)、山东省高等学校科技计划项目(J13LN10)和山东省泰安市科技发展计划项目(201330629)资助.**刘 阳**,男,1981 年生,博士,中国计算机学会(CCF)会员,主要研究方向为软件工程、形式化验证. E-mail: yangliu@seg.nju.edu.cn. **李宣东**,男,1963 年生,博士,教授,中国计算机学会(CCF)会员,主要研究领域为软件建模与分析、软件测试与验证. **马 艳**,女,1981 年生,硕士,讲师,主要研究方向为软件工程、混成系统验证. **王林章**,男,1973 年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为软件工程、软件测试.

1 引 言

系统验证是检验系统是否满足已经识别的需求,其作为保证计算机系统正确性的有效方法一直是业界和学术界研究的焦点问题.目前常用的系统验证方法有:同行审查(peer review)、测试、模拟、结构化分析、形式验证(formal verification)及某些方法的组合等[1].其中,形式验证借助严格的数学方法来完备地检查系统是否满足性质规约,同其他验证方法相比,其具有集成于系统的设计过程、较高的覆盖率和完备性及节省验证成本等优势[2].形式验证的发展大体上经历了以下有代表性的阶段:(1)1936 年,Turing 关于 Turing 机停机问题的思考[3],及其在 1949 年提出需人工证明程序的终止和正确性[4]等,可以看作是形式验证的雏形;(2)20 世纪 60 年代末和 70 年代,Floyd 和 Hoare 等开展的程序正确性证明研究[5-6]推动了形式验证的发展,他们试图用数学的方法来证明程序的正确性并发展成了各种程序验证方法,但受到程序规模的限制,这些方法并未完全达到预期的效果;(3)1977 年,Pnueli 提出反应式系统规格说明和时序逻辑(Temporal Logic,TL)的验证方法,将时序逻辑引入形式验证领域[7]及 20 世纪 80 年代初,Clarke 和 Emerson[8]、Queille 和 Sifakis[9]分别提出模型检验(model checking)的形式验证方法.

模型检验通过将待验证系统约束为有限状态模型、待验证需求性质约束为命题时序逻辑[10],把系统验证这一不可判定问题变换为可判定问题.模型检验是一种自动地完备判定有限状态系统模型是否满足给定时序逻辑描述性质的方法,其强调的是系统功能行为的绝对正确,回答的是系统是否满足用户的功能需求性质.目前,模型检验已成功应用于诸多工业领域.

然而计算机系统正变得日趋庞大和复杂,很多实际的系统被赋予随机行为特征[1],其原因可分为 3 类:(1)系统本身包含随机性,如随机化算法(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.具有以上各种随机行为特征的系统被称为随机系统.用模型检验的方法对随机系统进行自动的形式验证,定量分析其满足用户的功能或非功能需求性质的程度,即随机模型检验(stochastic model checking)或概率模型检验(probabilistic model checking)[11-12].Clarke、Emerson 和 Queille 在"Turing Lecture"中把随机模型检验列为模型检验研究的一个重要方向[11].近十年随机模型检验取得较大进展,出现了一些著名的研究组,如英国牛津大学 Kwiatkowska 教授课题组、德国亚琛工业大学(RWTH Aachen University) Katoen 教授课题组和德国萨尔大学(Saarland University) Hermanns 教授课题组等.

本文将从随机模型检验的渊源、原理、进展、验证过程及应用与挑战等方面,对其作阐述分析.本文第 2 节追溯随机模型检验的渊源,介绍其相关的最基本理论,给出典型的随机模型检验算法框架;第 3 节总结归纳出随机模型检验的 7 个重要研究方向,并分别对其进展作对比分析;第 4 节分析基于随机模型检验的验证过程及其优缺点,并对其支撑工具作分类介绍;第 5 节指出随机模型检验应用的几个方面的挑战;最后对全文进行总结.

2 随机模型检验原理

关于随机模型检验的研究可以追溯到:20 世纪 80 年代初,Harts、Pnueli 和 Sharir 用离散时间马尔可夫过程建模概率程序,研究概率并发程序的终止性质[13-14]和概率程序性质的证明方法[15].在后续的随机模型检验研究中,有以下里程碑式的成果:(1)Vardi 和 Wolper 提出一种基于自动机理论的定性线性时间性质的验证方法[16-17],Pnueli 和 Zuck 给出这一类问题的模型检验算法[18],Courcoubetis 和 Yannakakis 研究了线性时间框架下的定性、定量验证离散时间马尔可夫链,并分析了其时间复杂性[19-20];(2)Hansson 和 Jonsson 提出一种新的描述系统时间和可靠性的时序逻辑 PCTL,并给出了其模型检验算法验证随机系统[21],Bianco 和 de Alfaro 对该时序逻辑进行扩展并给出其模型检验算法验证概率和非确定性系统[22];(3)Aziz、Sanwal、Singhal 和 Brayton 提出一种新的时序逻辑 CSL,并给出其模型检验连续时间马尔可夫链算法[23-24],Baier、Haverkort、Hermanns 和 Katoen 对这一理论做了进一步的完善[25-26].

近年来,随机模型检验取得了较大进展,本节将简要介绍其最基本理论.如图 1 所示,随机模型检验实际上是经典模型检验理论和应用方面的延伸和推广.

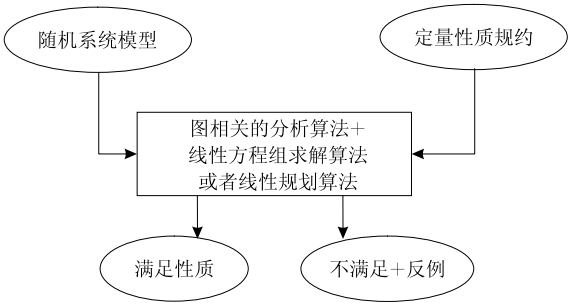

图 1　随机模型检验原理示意图

2.1　基本的随机系统模型及其度量

随机模型检验所验证的系统模型是一类具有马尔可夫特性①的随机过程式系统模型. 此类随机过程由俄罗斯数学家马尔可夫于 1907 年提出，可根据其状态空间和索引参数性质类型对其进行分类，其中：离散时间马尔可夫链(Discrete-Time Markov Chain，DTMC)是离散状态空间离散时间的马尔可夫随机过程，连续时间马尔可夫链(Continuous-Time Markov Chain，CTMC)是连续状态空间连续时间的马尔可夫随机过程，而马尔可夫决策过程(Markov Decision Process，MDP)是离散状态空间离散时间且带有非确定性选择的马尔可夫随机过程，是基于马尔可夫过程理论的动态随机系统的最优决策过程.

与随机过程中把马尔可夫过程作为随机变量序列的定义有所不同，在随机模型检验中其定义是基于状态的观点，可看作是标号迁移系统(Labelled Transition System，LTS)的变体，其关系如图 2 所示.

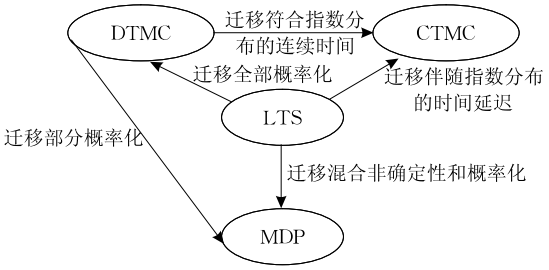

图 2　随机系统模型与 LTS 的关系

(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. DTMC 把 LTS 中某一状态后继状态间的非确定性选择变换成了概率选择，时间在 DTMC 模型中以离散的时间步向前推进，可以将其理解为精确的离散时间单元，或者是对时间的抽象，即迁移不伴随任何具体的时间信息. 在随机模型检验中，假定：(1) DTMC 是时间齐次(time-homogenous)②的，即迁移概率独立于时间；(2) DTMC M 是有限的；(3) 出于算法运算目的，迁移概率是有理数.

DTMC的路径表示被建模系统的执行，即系统可能的事件行为，可描述为状态的无限序列 $s_0s_1s_2\cdots\in S^\omega$，其中 s_0 是初始状态，对于 $\forall i\geq 0,P(s_i,s_{i+1})>0$. 令 $Paths(s)$ 表示从某状态 s 开始的所有无限路径的集合，$Paths_{fin}(s)$ 表示从某状态 s 开始的所有有限路径的集合. 为了度量 DTMC M 模型中某一事件的概率，需要定义关于模型 M 的概率空间 $(Outc,\zeta,Prob)$[27]，其计算方法[28]为：(1) 样本空间 $Outc=Paths(s_0)$；(2) ζ 是关于包含 $Cyl(\hat{\pi})$ 的路径 $Paths(s_0)$ 的最小 σ-代数，其中 $Cyl(\hat{\pi})=\{\pi\in Paths(s_0)\mid\hat{\pi}\in pref(\pi)\}$，$\hat{\pi}$ 是路径 π 的前缀，即 $\hat{\pi}=s_0\cdots s_n\in Paths_{fin}(s_0)$；(3) 存在唯一关于与 M 关联最小 σ-代数 ζ 的概率度量 $Prob$，即

$$Prob(Cyl(\hat{\pi}))=\begin{cases}1, & \text{若 } \hat{\pi}=s_0\\ \prod\limits_{0\leq i\leq n}P(s_i,s_{i+1}), & \text{其他}\end{cases}.$$

(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，$R:S\times S\to\Re_{\geq 0}$ 是迁移速率函数，$\Re_{\geq 0}$ 是非负实数. CTMC 把 LTS 中某一状态后继状态间的非确定性选择变换成在该状态停留指数分布时间后的概率选择；CTMC 也可以看作是把连续时间引入到 DTMC，把 DTMC 中的离散时间步迁移变成一个符合指数分布的连续时间延迟迁移. 在随机模型检验中，对 CTMC 也在存在诸如对 DTMC 的前两点假设.

度量 CTMC 涉及到 uniformised DTMC 和 embedded DTMC. CTMC M 的 uniformised DTMC 可定义为 DTMC $unif(M)=(S,P^{unif(M)},s_0,AP,L)$，其中 S,s_0,AP,L 与 CTMC 定义中的一致，而对于 $s,s'\in S,P^{unif(M)}=\boldsymbol{I}+\boldsymbol{Q}/q$，其中 $q=E(s)=\sum_{s'\in S}R(s,s')$，

① 马尔可夫特性是指随机过程的概率分布函数满足以下等式：$F\{X(t)\leq x\mid X(t_n)=x_n,X(t_{n-1})=x_{n-1},\cdots,X(t_0)=x_0\}=F\{X(t)\leq x\mid X(t_n)=x_n\},t>t_n>t_{n-1}>\cdots>t_0$.

② 时间齐次的是指系统行为不依赖于观察时间，即 $F\{X(t)\leq x\mid X(t_n)=x_n\}=F\{X(t-t_n)\leq x\mid X(0)=x_n\}$.

$$Q(s,s')=\begin{cases}R(s,s'), & \text{若 } s\neq s'\\ -\sum_{s''\neq s}R(s,s''), & \text{其他}\end{cases}$$，$\boldsymbol{I}$ 是单位矩阵. CTMC M 的 embedded DTMC 也可定义为一个 DTMC $emb(M)=(S,P^{emb(M)},s_0,AP,L)$，其中 S，s_0，AP，L 与 CTMC 定义中的一致，$P^{emb(M)}$ 可定义为

$$P^{emb(M)}(s,s')=\begin{cases}R(s,s')/E(s), & E(s)\neq 0\\ 1, & E(s)=0 \text{ 且 } s=s'\\ 0, & \text{其他}\end{cases}.$$

使用 embedded DTMC 可将 CTMC 的行为分析为在某一状态延迟一段关于 $E(s)$ 指数分布的时间后，按照概率值 $P^{emb(M)}(s,s')$ 触发迁移.

CTMC 的路径含义与 DTCM 的相同，CTMC 的一个无限路径 π 可表示为 $s_0t_0s_1t_1s_2t_2,\cdots$，其中 t_i 是非负实数，对于 $\forall i, R(s_i,s_{i+1})>0$. 关于 CTMC 模型 M 的概率空间 $(Outc,\zeta,Prob)$ 的计算方法[26]为：① 样本空间 $Outc=Paths(s_0)$，$Paths(s_0)$ 代表从初始状态 s_0 开始的所有路径集合；② ζ 是关于包含 $Cyl(s_0,I_0,\cdots,I_{n-1},s_n)$ 的路径 $Paths(s_0)$ 的最小 σ-代数，其中，$s_0\cdots s_n$ 是模型 M 中的 $R(s,s')>0$ 状态序列，$I_0,\cdots,I_{n-1}$ 是实数集上非空的区间序列；③ 存在唯一关于与 M 关联的最小 σ-代数 ζ 的概率度量 $Prob$，即：如果 $n=0$，$Prob(Cyl(s_0,I_0,\cdots,I_{n-1},s_n))=1$，否则 $Prob(Cyl(s_0,I_0,\cdots,I_{n-1},s_n))=Prob(Cyl(s_0,I_0,\cdots,I_{n-2},s_{n-1}))\cdot P^{emb(M)}(s_{n-1},s_n)\cdot(e^{-E(s_{n-1})\cdot \inf I_{n-1}}-e^{-E(s_{n-1})\cdot \sup I_{n-1}})$.

(3) MDP

MDP 可定义为 $M=(S,Act,P,s_0,AP,L)$，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 $\alpha\in Act$，$\sum_{s'\in S}P(s,\alpha,s')\in\{0,1\}$. MDP 把 LTS 中某一状态后继状态间的非确定性选择变换成了非确定性选择后的概率选择；MDP 也可以看作是把非确定性选择引入到 DTMC 模型，把原来的全概率选择变成了非确定性和概率选择混合存在. 若对于任何状态 s，$Act(s)$ 是只有一个元素的集合，那么该 MDP 就是一个 DTMC；反之亦然，任何一个 DTMC 可以看作是一个 MDP，在该 MDP 中，对于任何状态 s，$Act(s)$ 是只有一个元素的集合. 在随机模型检验中，关于 MDP 也存在诸如对 DTMC 的 3 点假设.

MDP 的一个路径是状态和动作的无限序列，无限路径可表示为 $s_0\alpha_1s_1\alpha_2s_2\alpha_3\cdots\in(S\times Act)^{\omega}$，其中对于 $\forall i\geq 0,\alpha_{i+1}\in Act(s_i)$ 且 $P(s_i,\alpha_{i+1},s_{i+1})>0$. MDP 的概率空间与其非确定性有关，在非确定性选择求解后，其概率空间就变换成了 DTMC 的概率空间.

MDP 的非确定性选择求解称为策略(strategy)，可形式化描述为 MDP M 的一个策略可表示为函数 $A:\hat{\pi}\to Act(s_n)$，其中 $\hat{\pi}$ 是有限路径 $s_0\alpha_1s_1\alpha_2s_2\alpha_3\cdots s_n$. 常用的策略有无记忆策略(memoryless adversary)、有限记忆策略(finite-memory adversary)、公平性策略(fair adversary)和随机化策略(randomized adversary)等.

2.2 常用的定量性质规约

定量性质规约是对系统行为的定量描述，如系统模型实现某一期望行为的概率要满足一定的界限(大于或小于用户给定的阈值). 定性性质规约是定量性质规约的特例，它的概率界限值是 0 或 1，但其含义也与经典模型检验中的性质规约不同，如系统模型到达一个“坏”状态的概率是 0 的含义是：存在到达“坏”状态的路径且其概率是 0，或者不存在到达“坏”状态的路径；系统模型完成期望的系统行为的概率是 1 的含义是：系统模型只能完成期望的行为，或者还可以有其他的行为但其概率是 0.

常用的定量性质规约有概率计算树逻辑(Probabilistic Computation Tree Logic，PCTL)[21]、带有概率的线性时序逻辑(Linear Temporal Logic，LTL)[7]、PCTL*[21]和连续随机逻辑(Continuous Stochastic Logic，CSL)[24,26]，其中 PCTL、概率的 LTL 和 PCTL* 的关系如图 3 所示.

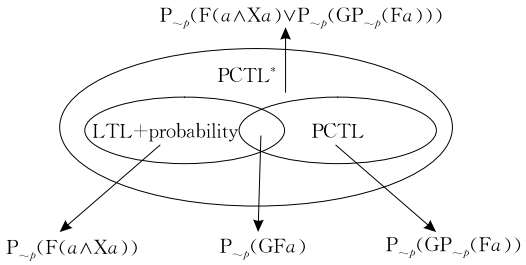

图 3 PCTL、概率的 LTL 和 PCTL* 间的关系

(1) PCTL

PCTL 是对计算树逻辑(Computation Tree Logic，CTL)[8]的概率扩展，用概率运算符 $P_{\sim p}$ 定量地扩展了 CTL 的路径量词 A(all)和 E(exists). PCTL 可描述关于 DTMC 和 MDP 这两类随机系统模型的定量分支(branching)时间性质，其与文献[29]的时序逻辑 pCTL 基本上一致. 关于原子命题集合 AP 的 PCTL 状态公式 Φ 可定义为 $\Phi::=\text{true}\mid a\mid\Phi\wedge\Phi\mid\neg\Phi\mid P_{\sim p}(\Psi)$，其中 $a\in AP$，Ψ 是路径公式，$\sim\in\{\leq,<,\geq,>\}$，$p\in[0,1]$ 是概率界限值；PCTL 路径公式 Ψ 可定义为 $\Psi::=X\Phi\mid\Phi\cup\Phi\mid\Phi\cup^{\leq n}\Phi$，其中

X(next)和$\cup$(until)与 CTL 路径运算符语义相同，$\cup^{\leq n}$(bounded until)是$\cup$的变体，要求 n 次迁移或小于 n 次迁移内满足$\cup$语义，n 是非负整数，Φ 是状态公式. PCTL 性质规约最终是一个状态公式，路径公式是概率运算符 $P_{\sim p}$ 的参数. 其他的 PCTL 公式，如 F(future)和 G(globally)可以由定义推出：$F^{\leq n}\Phi=\text{true}\cup^{\leq n}\Phi$，$P_{\leq p}(G\Phi)=P_{\geq 1-p}(F\neg\Phi)$，$F\Phi=\text{true}\cup\Phi$. PCTL 路径公式描述的事件是可度量的[21].

(2) 概率的 LTL

概率的 LTL 可描述关于 DTMC 和 MDP 这两类随机系统模型的定量线性时间性质规约，尤其是用于表示随机系统模型的 long-run 性质. 概率的 LTL 与 PCTL 在表达能力上相互补充，不存在包含关系. 与 PCTL 定义中概率可以语法地通过运算符 P 表示不同，LTL 概率度量只能在语义层面上体现. LTL 语法即经典模型检验中的 LTL 语法定义，其公式 Ψ 可定义为 $\Psi::=\text{true}\mid a\mid\Psi\wedge\Psi\mid\neg\Psi\mid X\Psi\mid\Psi\cup\Psi$，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. LTL 是隐式的关于所有路径的量词，LTL 路径公式描述的事件是可度量的[16].

(3) PCTL*

PCTL* 是 PCTL 的扩展(或 CTL*[8] 的扩展)，其不要求时序逻辑运算符后必须是状态公式，并允许路径公式的布尔运算. PCTL* 可描述关于 DTMC 和 MDP 这两类随机系统模型的定量分支时间性质规约. 关于原子命题集合 AP 的 PCTL* 状态公式 Φ 可定义为 $\Phi::=\text{true}\mid a\mid\Phi\wedge\Phi\mid\neg\Phi\mid P_{\sim p}(\Psi)$，其中 $a\in AP$，Ψ 是路径公式，$\sim\in\{\leq,<,\geq,>\}$，$p\in[0,1]$ 是概率界限值；PCTL* 路径公式 Ψ 可定义为 $\Psi::=\Phi\mid\Psi\wedge\Psi\mid\neg\Psi\mid X\Psi\mid\Psi\cup\Psi\mid\Psi\cup^{\leq n}\Psi$，其中 Φ 是状态公式，n 是非负整数. 其他的 PCTL* 公式可由定义推出，PCTL* 路径公式描述的事件是可度量的[21].

(4) CSL

CSL 可看作是对 CTL 的扩展：① 如同 PCTL 扩展 CTL 类似，增加了概率运算符 P；② 增加了稳状运算符 S. CSL 可描述关于 CTMC 这类随机系统模型的定量线性性质规约. 关于原子命题集合 AP 的 CSL 状态公式 Φ 可定义为 $\Phi::=\text{true}\mid a\mid\Phi\wedge\Phi\mid\neg\Phi\mid P_{\sim p}(\Psi)\mid S_{\sim p}(\Phi)$，其中 $a\in AP$，Ψ 是路径公式，$\sim\in\{\leq,<,\geq,>\}$，$p\in[0,1]$ 是概率界限值；路径公式 Ψ 可定义为 $\Psi::=X\Phi\mid\Phi\cup^{I}\Phi$，其中 Φ 是状态公式，I 是一个实数区间. 其他的 CSL 公式可由以上定义推出. CSL 路径公式描述的事件是可度量的[25].

2.3　随机模型检验算法

随机模型检验问题可描述为：给定一个随机系统模型 M 和一个性质规约公式 Φ，(1)判断模型 M 是否满足公式 Φ，即 $(M,s_0)\models\Phi$ 是否成立；或者(2)求模型 M 中满足公式 Φ 的所有状态. 其中，(1)称作局部随机模型检验问题，(2)称作全局随机模型检验问题，但本质是一致的，即判断状态 s 是否满足公式 Φ. 对于(1)可采用全局随机模型检验算法来完成判定，即：先求出模型 M 中满足公式 Φ 的所有状态集合 $Sat(\Phi)$，然后判断 s_0 是否属于该集合，若属于则 $(M,s_0)\models\Phi$ 成立，否则 $(M,s_0)\models\Phi$ 不成立. 本小节讨论的随机模型检验算法属于全局随机模型检验.

2.3.1　PCTL 模型检验 DTMC

PCTL 模型检验 DTMC 算法[19-21] 的基本过程是递归地由底向上遍历公式 Φ 的语法解析树，并在此过程中计算满足子公式 φ 的状态集合 $Sat(\varphi)$.

对于 PCTL 公式中的非概率状态运算符，其满足关系计算与经典模型检验相似，即

$$Sat(\text{true})=\{s\in S\mid s\models\text{true}\}=S,$$
$$Sat(a)=\{s\in S\mid a\in L(s)\},$$
$$Sat(\Phi_1\wedge\Phi_2)=Sat(\Phi_1)\cap Sat(\Phi_2),$$
$$Sat(\neg\Phi)=S-Sat(\Phi),$$

其中 S 表示系统模型的所有状态集合，$Sat(\varphi)$ 表示满足公式 φ 的状态集合.

对于 $P_{\sim p}[\Psi]$，需要计算状态 $s\in S$ 满足路径公式 Ψ 的概率值，即 $Prob(s,\Psi)$. 可分为 3 种情况求解：

(1) $P_{\sim p}[X\Phi]$.

满足公式 $P_{\sim p}[X\Phi]$ 的状态集合 $Sat(P_{\sim p}[X\Phi])$，是指满足路径公式 $X\Phi$ 的概率度量符合 $\sim p$ 的所有状态 s，即 $Sat(P_{\sim p}[X\Phi])=\{s\in S\mid Prob(s,X\Phi)\sim p\}$，其中，$Prob(s,X\Phi)=\sum_{s'\in Sat(\Phi)}P(s,s')$.

(2) $P_{\sim p}[\Phi_1\cup^{\leq k}\Phi_2]$.

满足公式 $P_{\sim p}[\Phi_1\cup^{\leq k}\Phi_2]$ 的状态集合 $Sat(P_{\sim p}[\Phi_1\cup^{\leq k}\Phi_2])$，是指满足路径公式 $\Phi_1\cup^{\leq k}\Phi_2$ 的概率度量符合 $\sim p$ 的所有状态 s，即 $Sat(P_{\sim p}[\Phi_1\cup^{\leq k}\Phi_2])=\{s\in S\mid Prob(s,\Phi_1\cup^{\leq k}\Phi_2)\sim p\}$，令 $S^{no}=S\backslash(Sat(\Phi_2)\cup Sat(\Phi_1))$，$S^{yes}=Sat(\Phi_2)$，$S^{?}=S\backslash(S^{yes}\cup S^{no})$，则

$$Prob(s,\Phi_1\cup^{\leq k}\Phi_2)=\begin{cases}1, & \text{若}s\in S^{yes}\\ 0, & \text{若}s\in S^{no}\\ 0, & \text{若}s\in S^{?}\text{且}k=0\\ \sum_{s'\in S}P(s,s')\cdot Prob(s',\Phi_1\cup^{\leq k-1}\Phi_2), & \text{若}s\in S^{?}\text{且}k>0\end{cases}$$

(3) $P_{\sim p}[\Phi_1 \cup \Phi_2]$.

满足公式 $P_{\sim p}[\Phi_1 \cup \Phi_2]$ 的状态集合 $Sat(P_{\sim p}[\Phi_1 \cup \Phi_2])$，是指满足路径公式 $\Phi_1 \cup \Phi_2$ 的概率度量符合 $\sim p$ 的所有状态 s，即 $Sat(P_{\sim p}[\Phi_1 \cup \Phi_2]) = \{s \in S | Prob(s, \Phi_1 \cup \Phi_2) \sim p\}$，计算 $Prob(s, \Phi_1 \cup \Phi_2)$ 的过程为：① 用图论算法计算 $Sat(E[\Phi_1 \cup \Phi_2])$，然后计算 $S^{no} = Sat(P_{\leq 0}[\Phi_1 \cup \Phi_2]) = S \backslash Sat(P_{>0}[\Phi_1 \cup \Phi_2])$，即 $S^{no} = S \backslash Sat(E[\Phi_1 \cup \Phi_2])$；② 用图论算法计算 $Sat(E[\Phi_1 \cup S^{no}])$，然后计算 $S^{yes} = Sat(P_{\geq 1}[\Phi_1 \cup \Phi_2]) = S \backslash Sat(P_{>0}[\Phi_1 \cup \Phi_2])$，即 $S^{yes} = S \backslash Sat(E[\Phi_1 \cup S^{no}])$；③ 用值迭代或高斯消元法等方法求解线性方程组：

$$Prob(s, \Phi_1 \cup \Phi_2) = \begin{cases} 1, & \text{若 } s \in S^{yes} \\ 0, & \text{若 } s \in S^{no} \\ \sum_{s' \in S} P(s,s') \cdot Prob(s', \Phi_1 \cup \Phi_2), & \text{其他} \end{cases}.$$

求解 $Sat(P_{\sim p}[\Phi_1 \cup^{\leq k} \Phi_2])$ 和 $Sat(P_{\sim p}[\Phi_1 \cup \Phi_2])$ 也可以通过将某些状态变换为吸收(absorbing)状态，然后将其化简为可达性概率问题的求解，而可达概率的计算可以通过图分析算法和线性方程组求解获得[1]. PCTL 模型检验 DTMC 的时间复杂性与公式的大小 $|\Phi|$ 成线性关系，与模型的大小 $|M|$ 成多项式关系.

2.3.2 PCTL 模型检验 MDP

PCTL 模型检验 MDP 算法[21-22]与 PCTL 模型检验 DTMC 的算法框架相似，其区别在于：需要根据 $\sim$ 的取值用线性规划等方法计算路径概率的最大值或最小值，即 $P_{max}(s, \Phi)$ 或 $P_{min}(s, \Phi)$. 与 PCTL 模型检验 DTMC 类似，也可以通过将某些状态变换为吸收(absorbing)状态，然后将其化简为可达性概率问题的求解. PCTL 模型检验 MDP 的时间复杂性与公式的大小 $|\Phi|$ 成线性关系，与模型的大小 $|M|$ 成多项式关系.

2.3.3 LTL 模型检验 DTMC

LTL 模型检验 DTMC[16]涉及到自动机、叉乘等相关理论，其过程步骤如下：(1) 先将公式 Ψ 转换成非确定 Büchi 自动机[30]，然后再将其转换成确定 Rabin 自动机 DRA[31] $\mathcal{A}$；(2) 求 DTMC M 与 DRA 的积 $M \otimes \mathcal{A}$；(3) 对于DTMC $M \otimes \mathcal{A}$，找出其可接受的强连通分量(Bottom Strongly Connected Component，BSCC)；(4) 用线性方程组计算到达可接受的 BSCC 的概率 p_{BSCC}；(5) 对比 p_{BSCC} 和 $\sim p$ 的值，求出满足的状态.

在 LTL 模型检验 DTMC 算法过程中，步 1 得到的 DRA 的大小是 LTL 公式大小的 2^{2^n} 倍，步 3 求强连通分量的复杂性与模型大小成线性关系，步 4 是求解线性方程组，其时间复杂性是关于积模型大小的三次方. 因此 LTL 模型检验 DTMC 的时间复杂性与公式的大小 $|\Psi|$ 成双指数关系，与模型的大小 $|M|$ 成多项式关系. 文献[19-20]将其复杂性化简为关于 $|\Psi|$ 的单指数关系，定性 LTL 模型检验 DTMC 是 PSPACE 完全问题[16].

2.3.4 LTL 模型检验 MDP

LTL 模型检验 MDP 与 2.3.3 节 LTL 模型检验 DTMC 的算法框架相似，其过程步骤如下：(1) 通过等价变换 $P_{>p}[\Psi] = P_{<1-p}[\neg \Psi]$，把概率的 LTL 规约变换成类型 $P_{<p}[\Psi]$，即求解最大概率；(2) 与 2.3.3 节的步 1 一致，将公式 Ψ 转换成 NBA[30]，再将 NBA 转换成 DRA[31] $\mathcal{A}$；(3) 求 MDP M 与 DRA 的积 $M \otimes \mathcal{A}$；(4) 对于 MDP $M \otimes \mathcal{A}$，找出可接受的终端分量(End Components，EC)；(5) 用线性规划计算到达可接受 EC 的概率 p_{EC}；(6) 对比 p_{EC} 和 p 的值，求出满足 $P_{<p}[\Psi]$ 的状态.

在 LTL 模型检验 MDP 算法过程中，步 2 得到的 DRA 的大小是 LTL 公式大小的 2^{2^n} 倍，步 4 求终端分量的复杂性与模型大小成线性关系. 因此 LTL 模型检验 MDP 的时间复杂性与公式的大小 $|\Psi|$ 成双指数关系，与模型的大小 $|M|$ 成多项式关系. 与 LTL 模型检验 DTMC 算法不同，该算法的时间复杂性目前没能改进，定性 LTL 模型检验 MDP 是 2EXPTIME 问题[20].

2.3.5 PCTL* 模型检验 DTMC

PCTL* 模型检验 DTMC 可以看作是 PCTL 和概率的 LTL 模型检验 DTMC 的联合，算法复杂度为 PSPACE 完全问题，其过程步骤为：(1) 把 PCTL* 状态公式 $P_{\sim p}[\Psi]$ 的 Ψ 转换成 LTL 公式 Ψ'；(2) 将 LTL 公式 Ψ' 换成 DRA $\mathcal{A}$；(3) 求 DTMC M 与 DRA 的积 $M \otimes \mathcal{A}$；(4) 对于 DTMC $M \otimes \mathcal{A}$，找出其可接受的强连通分量(Bottom Strongly Connected Component，BSCC)；(5) 计算到达可接受的 BSCC 的概率 p_{BSCC}；(6) 对比 p_{BSCC} 和 $\sim p$ 的值，求出满足的状态.

2.3.6 PCTL* 模型检验 MDP

类似 2.3.5 节，PCTL* 模型检验 MDP 可以看作是 PCTL 和概率的 LTL 模型检验 MDP 的联合，其算法复杂度为 2EXPTIME 完全问题.

2.3.7 CSL 模型检验 CTMC

CSL 模型检验 CTMC 算法已经被证明是可

判定的[23]，文献[25]首次给出其算法过程，文献[26,32]对该算法做了改进，其算法框架与 2.3.1 节算法过程类似，其时间复杂度与模型大小成多项式关系、与 CSL 公式大小成线性关系且与 $q.t_{\max}$ 成线性关系，$t_{\max}$ 是时间区间内的最大值. 其计算方法为：

对于 CSL 公式中非概率状态运算符，其满足关系计算与经典模型检验相似.

对于公式 $\mathrm{P}_{\sim p}(\mathrm{X}\Phi)$，$Sat(\mathrm{P}_{\sim p}[\mathrm{X}\Phi])=\{s\in S\mid Prob(s,\mathrm{X}\Phi)\sim p\}$，$Prob(s,\mathrm{X}\Phi)$ 的计算与 CTMC 的实时时间没有任何关系，只需计算到达下一状态的概率，可以直接用 2.3.1 节对于 $\mathrm{P}_{\sim p}[\Psi]$ 中的(1)方法对 CTMC 的 embedded DTMC 模型 $emb(M)$ 进行计算.

对于公式 $\mathrm{P}_{\sim p}(\Phi_1\bigcup^I\Phi_2)$，$I$ 是非负实数区间，$Sat(\mathrm{P}_{\sim p}[\Phi_1\bigcup^I\Phi_2])=\{s\in S\mid Prob(s,\Phi_1\bigcup^I\Phi_2)\sim p\}$，$Prob(s,\Phi_1\bigcup^I\Phi_2)=Prob(s,\Phi_1\bigcup^{cl(I)}\Phi_2)$，其中 $cl(I)$ 是区间 I 的闭包，可将 I 分成以下 4 种情况进行计算：

(1) $I=[0,\infty)$.

$Prob(s,\Phi_1\bigcup^{[0,\infty)}\Phi_2)$ 求解可用 2.3.1 节对于 $\mathrm{P}_{\sim p}[\Psi]$ 中的(3)方法对 CTMC 的 embedded DTMC 模型 $emb(M)$ 进行计算，即 $Prob(s,\Phi_1\bigcup^{[0,\infty)}\Phi_2)=Prob^{emb(M)}(s,\Phi_1\bigcup\Phi_2)$.

(2) $I=[0,t]$，$t\in\Re_{\geqslant 0}$.

令 $S^{\mathrm{yes}}=Sat(\Phi_2)$，$S^{\mathrm{no}}=S\backslash(Sat(\Phi_2)\bigcup Sat(\Phi_1))$，$S^{?}=S\backslash(S^{\mathrm{yes}}\bigcup S^{\mathrm{no}})$，则

$$Prob(s,\Phi_1\bigcup^{[0,t]}\Phi_2)=\begin{cases}1, & \text{若 } s\in S^{\mathrm{yes}}\\ 0, & \text{若 } s\in S^{\mathrm{no}}\\ \int_0^t\sum_{s'\in S}P^{emb(M)}(s,s')\cdot E(s)\cdot e^{-E(s)\cdot x}\cdot Prob(s',\Phi_1\bigcup^{[0,t-x]}\Phi_2)\mathrm{d}x, & \text{若 } s\in S^{?}\end{cases}.$$

关于该积分的解，文献[25]提出通过 Volterra 积分方程组的近似解来求解，而文献[33]的实验表明一般来说该方法比较慢，文献[34]中给出了一种比较简洁的方法来求解，即将其化简为瞬时状态概率分析.

(3) $I=[t,t']$，$t,t'\in\Re_{\geqslant 0}$.

分为两部分计算：① 直到时间 t 停留在满足 Φ_1 的状态，可以把满足 $\neg\Phi_1$ 的状态变成吸收状态后，用瞬时概率计算可得；② 在时间区间 $[0,t'-t]$ 内到达满足 Φ_2 的状态，在这之前停留在满足 Φ_1 的状态，通过(2)的计算可得.

(4) $I=[t,\infty)$，$t\in\Re_{\geqslant 0}$.

同(3)相似，也可把其分为两部分计算，且第一部分相同，第二部分变成了无界的区间，可以直接用 embedded DTMC 求解.

对于公式 $S_{\sim p}[\Phi]$，状态 s 满足 $S_{\sim p}[\Phi]$，即 $\sum_{s'\models\Phi}\pi_s(s')\sim p$，分两种情况计算稳态概率 $\pi_s(s')$：

(1) 若 CTMC M 是不可约简的，则通过求解以下线性方程组可得 $\pi\cdot Q=0$，且 $\sum_{s\in S}\pi(s)=1$.

(2) 若 CTMC M 是可约简的，则① 根据图的深度优先搜索算法，求出 M 的所有 BSCC；② 每个 BSCC 可算作是一个不可约简的 CTMC，根据上述(1)计算其稳态概率分布 $\pi^{BSCC}(s')$；③ 对于任一个 BSCC，计算 M 的每个状态到达其概率 $Prob^{emb(M)}(s,BSCC)$；④ 根据以下等式求解 M 的稳态概率分布：

$$\pi_s(s')=\begin{cases}Prob^{emb(M)}(s,BSCC)\cdot\pi^{BSCC}(s'), & \text{若 } s'\in BSCC\\ 0, & \text{其他}\end{cases}.$$

3　随机模型检验研究方向及其进展

总的来说，根据随机模型检验原理，关于随机模型检验的研究可分为：(1) 增强随机系统模型刻画能力，使随机模型检验的可验证随机系统模型范围扩大；(2) 增强定量性质规约描述能力，使随机模型检验可验证定量需求类型扩大；(3) 有效的反例产生和表示，使随机模型检验的实用性扩充；(4) 化简随机系统模型的状态空间，使随机模型检验的复杂度降低；(5) 更有效的检验算法或模型存储表示方法，使随机模型检验的效率提高. 其中，(1)、(2)和(3)属于随机模型检验功能横向上的扩展；(4)和(5)是随机模型检验功能纵向上的优化，是缓解随机模型检验状态空间爆炸[35]的重要手段，其二者往往交织或组合使用以达到更好地应对随机模型检验中的状态空间爆炸问题.

3.1　新的随机系统模型

随机系统模型的分类如表 1 所示，如何更具体或更广泛地建模和验证随机系统，即新的随机系统模型的分析与验证是目前随机模型检验的一个重要研究方向. 我们在文献[36]给出一种建模非确定性和概率系统的 high-level 方法非确定概率 Petri 网(Nondeterministic Probabilistic Petri Net，NPPN)，并给出其相应的代数语义和逻辑语义及定量验证方法，其是以通用网论[37]为指导，在保证非确定性前提下，将概率度量理论引入到经典的 Petri 网模型.

概率时间自动机(Probabilistic Timed Automata，PTA)[38-39]可以看作是 MDP 和概率自动机(Probabilistic Automata，PA)的实数时钟扩展，或者是时

间自动机[40]的概率扩展,其语义是无限状态空间MDP.目前PTA的分析方法有基于数字时钟(digital clock)[41]和基于区域带(zone)的方法:数字时钟方法假设时钟值是整数并将PTA转换成有限状态MDP;区域带(zone)方法是用有界差值矩阵(difference-bound matrices)符号化的获取时钟值的集合,包括向前可达性[39]、向后可达性[42]等分析方法.连续时间马尔可夫决策过程(Continuous-Time MDP,CTMDP)[43-44]结合CTMC的时间延迟方式和MDP的非确定性选择策略,文献[45-47]分别从离散化和均匀一致化等方面探讨了其可达性和模型检验算法.交互式马尔可夫链(Interactive Markov Chain,IMC)[48]正交的组合时间延迟和非确定性动作的选择,可以对随机系统进行组合式建模与分析,文献[49]给出一种CSL模型检验IMC算法.文献[50]将随机博弈作为建模两类非确定性及概率行为系统的形式模型,并将其用于控制器合成问题的验证.

表1 随机系统模型

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

3.2 新的定量性质描述方法

定量性质的分类如表2所示,如何更全面和更细致地描述定量需求性质,即新的定量性质规约方法与验证是目前随机模型检验的一个重要研究方向.概率时间计算树逻辑(Probabilistic Timed Computation Tree Logic,PTCTL)[51]是结合PCTL概率算子和TCTL(Timed Computation Tree Logic)[52-53]时间约束算子的一种关于时间和概率的定量性质规约描述方法.通过改进传统的区域图(region-graph)构造方法[40,52],PTCTL模型检验PTA过程是可判定的[39],遗憾的是这种方法很容易导致状态空间爆炸,文献[51]给出了一种基于符号化的模型检验算法.多目标定量性质(quantitative multi-objective properties)[54]是结合概率的ω正则性质和基于回报(reward)的度量性质[55],文献[54]给出了基于多目标定量性质的模型检验算法,并将其用于系统的合成.PCTL*、CSL、PTCTL和多目标定量性质规约等均是面向状态的定量性质描述方法,不能显式刻画随机系统的动作执行序列,因此有一定的局限性.aCSL[56]是基于动作(action)的CSL变体,文献[56]将其用于随机进程代数的模型检验,文献[57]将aCSL用于系统的性能建模.与aCSL类似,我们给出了一种基于动作的PCTL变体(aPCTL)[36]及其语义和模型检验算法,并将其用于NPPN的定量验证.借鉴了基于路径的回报变量[58]和命题动态逻辑[59],文献[60-62]扩展LTL,把基于动作和基于状态的时序逻辑联合起来,给出一种asCSL(CSL with action and state labels)逻辑,其可用于分析验证带有动作标号的CTMC.基于动作和状态的时序逻辑可以更直观和更有效地规约随机系统模型的性质.

表2 定量性质规约

名称	公式符号	随机系统模型
PCTL	Φ	DTMC MDP
LTL+probability	$Prob(s,\Psi)$	DTMC MDP
CSL	Φ	CTMC CTMDP IMC
PCTL*	Φ	DTMC MDP
PTCTL	Φ	PTA
quantitative multi-objective property	Φ	MDP+reward
aCSL	Φ	CTMC+action
asCSL	Φ	CTMC+action
aPCTL	Φ	NPPN

3.3 随机模型检验的反例

对于不满足的性质给出反例(counterexample)使模型检验不仅可以验证系统而且可以作为调试工具使用,如何有效地产生和表示反例是随机模型检验的一个重要研究方向.与经典模型检验相似[63],随机模型检验中的反例要满足3个条件:(1)可以作为随机系统模型为什么不满足性质的解释(explanation);(2)含有丰富的信息能代表不满足一类定量性质;(3)可以简单、具体地识别出随机系统模型的错误,并且可以有效地分析这些错误.文献[64]为随机模型检验的反例产生提供了理论和算法基础,是最早关注随机模型检验中反例产生的研究,其给出了关于模型检验DTMC的反例概念,将反例求解转换为图论中最短路径问题的变体.文献[65]对文献[64]的研究工作做了进一步完善,给出了表示反例的简洁正则表达式方法.文献[64]的反例产生方法可以扩展到模型检验MDP的反例产生[66]和模型检验CTMC的反例产生[67].与上述方法不同,文献[68-69]将模型检验MDP的反例定义为一般的DTMC.文献[66,68-69]的方法不能完全有效地表示模型检验MDP的反例,文献[70]进而将反例定义为一般的MDP,并给出求解最小反例的算法.文献[71]用符号状态交集等方法研究了PTCTL模型检验PTA的反例产生方法,但其不能保证可得到证据最少的反例.

3.4 随机系统模型的抽象

抽象(abstraction)是应对模型检验中状态爆炸问题的主要技术之一[72],在经典的模型检验中得到

很好的应用.抽象通过隐藏与验证性质无关的具体系统模型状态,用尽可能少的抽象状态来表示系统模型.对随机系统模型的抽象是一个更为复杂的过程,它是随机模型检验的一个重要研究方向.在随机模型检验中关于抽象的研究涉及到两个问题:如何表示抽象模型和如何构造抽象模型.

(1)抽象模型的表示

因随机模型检验中有不同的随机系统模型,所以也存在不同的抽象模型表示方法.对于 DTMC,可将抽象模型定义为抽象马尔可夫链[73],即抽象状态间的迁移概率是一个概率区间,通过类似的方法可将 CTMC 的抽象模型定义为 CTMDP.DTMC 和 CTMC 的抽象模型也可以通过概率模拟(probabilistic simulation)[74]或其变体的熵表示.文献[75]给出了一种 DTMC 和 CTMC 三值抽象模型的表示方法.对于 MDP,可将其抽象模型定义为另一个 MDP[76],或者将其抽象模型定义为二人随机博弈[77].

(2)抽象模型的构造

构造抽象模型是对原随机系统模型具体状态的划分,一个好的抽象模型是在保证可以正确分析用户关注的定量性质的同时使得随机系统模型的状态空间足够小.一种常用的构造抽象模型的方法是抽象精化(abstraction refinement),其过程是把一个简单的粗糙的抽象模型反复精化成一个足够精确的抽象模型.文献[76]第 1 次提出用抽象精化方法对 MDP 进行抽象;文献[69]运用谓词抽象、反例等技术给出一种关于随机系统的反例引导的抽象精化(Counterexample-Guided Abstraction Refinement,CEGAR)方法;文献[70]基于概率反例的定义给出一种关于 MDP 的反例引导的抽象精化算法框架,其抽象精化过程可表示为图 4;文献[78-80]提出一种基于随机博弈的用容忍误差引导的定量抽象精化框架来抽象 MDP 和 PTA 等随机系统模型,其抽象精化过程可表示为图 5.还有一些构造等价抽象模型方法,如:互模拟最小化(bisimulation minimization)[81]、对称归约(symmetry reduction)[82]或偏序归约(partial-order reduction)[83]等.

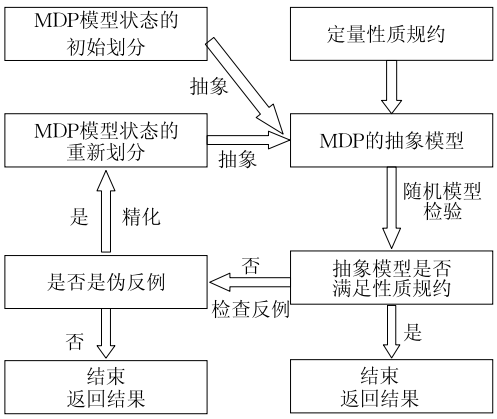

图 4　反例引导的抽象精化

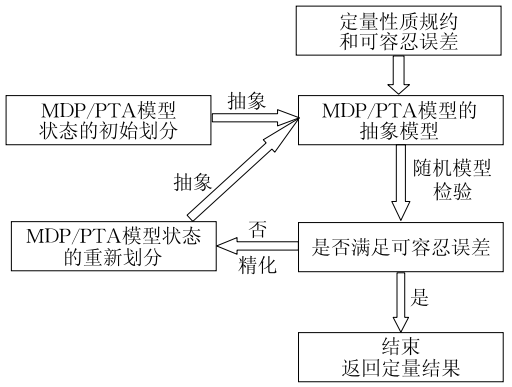

图 5　基于随机博弈的抽象精化

3.5　组合随机模型检验

组合验证利用分而治之的思想,分别对系统模块进行模型检验,是应对模型检验中状态空间爆炸的主要方法之一.实际上系统模块往往只能在特定的上下文环境中才能满足其性质,这就导致出现了假设-保证(assume-guarantee)式的组合验证技术,其验证过程是将验证两个模块的系统 $M_1 \parallel M_2$ 是否满足性质 P 变换成:(1)在假设 A 成立的前提下,模型检验 M_2 是否满足性质 P,即 $\langle A\rangle M_2\langle P\rangle$;(2)模型检验 M_1 是否在任何情况下总是满足假设 A,即 $\langle \text{true}\rangle M_1\langle A\rangle$.如何把假设-保证式组合验证技术应用于随机模型检验,即组合随机模型检验,是目前随机模型检验的一个重要研究方向.基于已有的关于组合分析 MDP 的理论[84],文献[85]提出了一种假设-保证的方法组合验证随机系统,给出了其假设-保证推理规则,其中 $\langle A\rangle_{\geq pA}$ 和 $\langle G\rangle_{\geq pG}$ 是用有限自动机表示的安全性质,即当系统模块 M 以最小概率 pA 满足 A 时,系统模型将以最小概率 pG 满足性质 G,其基本思想如图 6 所示.为了有效且自动化实现组合随机模型检验,假设-保证式组合随机模型检验可以转换成定量多目标随机模型检验问题[54].关于假设 A 的构造是假设-保证式组合随机模型检验的关键,其可以通过反例引导的人工交互式的产生[86],也可以通过 L^* 学习算法自动产生[86-87].文献

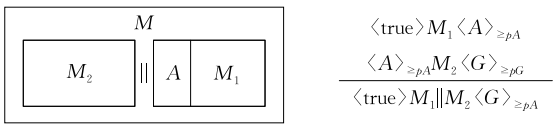

图 6　假设-保证式组合随机模型检验

[88]用模拟的方法产生假设-保证推理的假设,并将其用于概率标号迁移系统模型的组合验证.

3.6 统计随机模型检验

第2节论述的随机模型检验算法属于数值(numerical)方法,这种方法可以得到精确的度量值,但是会因随机系统模型与性质规约方式的不同而导致计算方法不同,并且其计算过程需要较多的时间和存储空间,易出现状态空间爆炸问题.另一种定量分析验证随机系统的方法是统计(statistical)方法[89-91].如何用统计方法实现随机模型检验,即统计随机模型检验,是目前随机模型检验研究的一个重要方向,其基本过程思想如图7所示.文献[92]最先给出一种统计随机模型检验方法,即通过有限多次的模拟随机系统的运行,并用假设检验(hypothesis testing)来推断样本是否提供了一个随机系统模型满足或违反性质规约的统计证据(evidence),其理论依据是既然随机系统的样本运行是根据系统定义的分布得到的,那么它就可以用来评估系统执行的概率度量.针对文献[92]只能验证时间限界的PCTL性质,文献[93]给出了一种基于Monte Carlo模拟和假设检验的验证无界Until PCTL性质的方法,文献[94]也给出一种验证无界Until PCTL性质的方法,但是这两种方法需要预先知道随机系统的一些信息,而文献[95]给出一种不需要这种先知条件的验证无界Until PCTL性质的方法.文献[96-97]将统计随机模型检验由原来的验证白盒随机系统推广到黑盒随机系统,文献[98]在对比数值和统计随机模型检验的基础上,提出将二者结合的思想.文献[99-100]给出了一类基于Monte Carlo技术的统计随机模型检验方法.

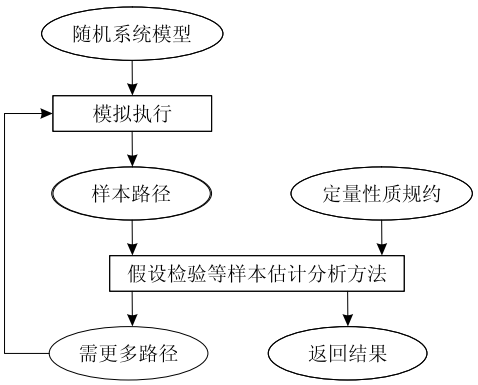

图7 统计随机模型检验基本过程

统计随机模型检验提供了一种验证随机系统模型DTMC和CTMC的统一方法,该随机系统模型可以具有无限状态空间或甚至不具备马尔可夫性质,但是现有的统计随机模型检验方法不能用于验证带有非确定性的随机系统模型.

3.7 参数随机模型检验

在建模随机系统的早期阶段或者为了建模系统的鲁棒性,往往把模型状态间的一些迁移概率设置为参数而不是具体值[101],如何对这类随机系统模型进行模型检验验证,即参数随机模型检验,是目前随机模型检验的一个重要研究方向.关于参数系统模型的研究,最初是分析与验证实时系统,文献[102]给出一种参数分析实时系统的时间性质的方法,文献[103]提出一种新的时间自动机模型参数时间区间自动机(Parametric Time-Interval Automata,PTIA),并给出其参数模型检验算法.文献[104-105]最先开始研究参数随机系统的分析与验证.文献[104]把满足概率的路径公式表示成一个正则表达式,然后递归地计算正则表达式,当概率值是有理数时,则正则表达的值也是有理数,当概率是参数时,则正则表达的值是一个有理数函数,给出一种基于正则语言的符号参数模型检验DTMC算法.文献[105]分析参数随机系统模型的可达性问题,给出确定满足性质的参数取值范围的方法.文献[106]把文献[105]中参数的取值域扩展到实数范围,并给出一种求解参数最大值和最小值的方法.文献[107]在文献[104]的研究基础上,给出一种更有效的参数随机系统模型的可达性计算方法.文献[108]通过把参数MDP可能的状态空间划分成超矩形(hyper-rectangles),然后使用已有的判定过程检验性质是否在该超矩形上成立,给出了一种关于参数MDP模型的PCTL合成方法.

除了以上主要的随机模型检验研究方向,还有其他一些主题也值得作深入的探讨,如有效的随机系统模型的表示与存储[109]、限界的随机模型检验[110-111]、运行时随机模型检验[112]等.

4 基于随机模型检验的验证

4.1 验证过程

与基于经典模型检验的验证过程类似[1],基于随机模型检验的验证过程如图8所示,可分为3个阶段:建模、运行和分析.

(1)建模待验证的系统和性质

随机模型检验器的输入是被验证系统的模型和需求性质规约,所以为了得到正确的验证结果,要对系统行为和需求性质进行严格、准确和精确的刻画,

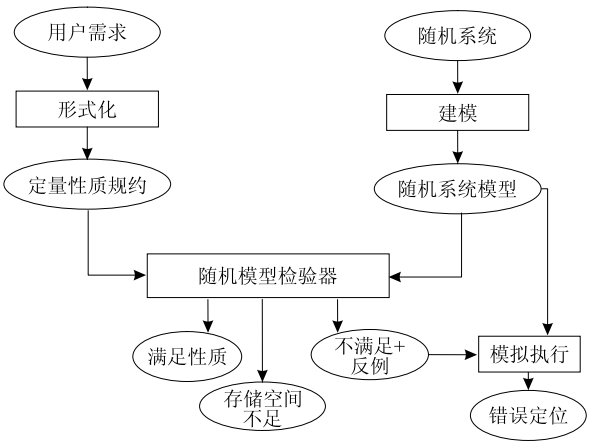

图 8　基于随机模型检验的验证过程

即验证过程的建模阶段. 建立系统模型的过程是用随机模型检验器的模型描述语言表示待验证的系统，这其实是将待验证系统或其马尔可夫链模型转换为模型检验器可识别的系统模型的过程. 为了提高模型质量，通常在模型检验前对系统模型进行模拟(simulation)执行，模拟可以发现一些简单错误，排除这些错误有利于减少模型检验过程的时间耗费. 性质规约是用随机模型检验器的性质描述语言表示需求性质，不同的随机模型检验器的性质描述语言略有不同.

(2) 运行随机模型检验器

根据随机系统模型类别和验证性质，适当地设置随机模型检验器的某些参数和选项，对其初始化并运行随机模型检验器.

(3) 检验结果分析

随机模型检验的返回结果有 3 种：① 系统模型满足性质；② 系统模型不满足性质，并给出反例；③ 系统模型状态空间太大，运行随机模型检验器的系统内存不足. 对于①，可以继续验证其他性质，或者所有性质均已验证，则验证过程结束. 对于②，可能有多种情况：模型错误，即模型没有正确地表示系统的设计模型，需要修改模型并再次运行随机模型检验器；设计错误，验证过程结束，需要修改设计模型并重新启动验证过程；性质规约错误，即性质规约没有正确描述需求，需要修改性质规约并再次运行随机模型检验器. 对于③，可以采用其他一些支持符号化或运行时验证的随机模型检验器，或者对模型进行抽象等化简模型的操作.

4.2　优缺点分析

基于随机模型检验的验证主要具有以下优点：(1) 以严格的数学理论为基础，如逻辑、数据结构和算法理论等；(2) 可以广泛应用于协议分析、嵌入式系统、软件系统和硬件设计等领域；(3) 支持部分验证，可以先验证最重要的性质；(4) 提供反例，即诊断信息；(5) 可以容易地集成到已有的系统开发过程. 其主要缺点有：(1) 随机模型检验验证属于可判定领域范围，不能有效地应用于无限状态空间或抽象数据类型(需要不可判定或半可判定逻辑)的验证与分析；(2) 验证的是系统模型而非系统，所得结果仅对系统模型的层面上有效；(3) 容易产生状态空间爆炸问题；(4) 验证过程中的建模需要一定的专业知识.

4.3　工具支持

随机模型检验工具的实现，推动了随机模型检验验证的广泛应用. 目前已有一些随机模型检验工具，本小节先介绍两种常用的随机模型检验器 PRISM 和 MRMC，然后对其余工具按验证的系统模型类型对其分类并做简单介绍.

(1) PRISM

PRISM[113-114] 是 Kwiatkowska 教授课题组用 Java/C(++)语言实现的最为广泛使用的一种图形化随机模型检验工具，可运行于 Windows、Linux 和 Mac OS X 等操作系统. PRISM 可以验证包括 DTMC、CTMC、MDP、PA、PTA 及其回报扩展模型等多种类型的随机系统模型，其性质规约囊括了 PCTL、CSL、概率的 LTL、PCTL* 及其回报扩展等定量性质规约语言，是功能最为齐全的随机模型检验工具. PRISM 使用基于 BDDs(Binary Decision Diagrams)和MTBDDs(Multi-Terminal Biary Decision Diagrams)[115-116] 的符号数据结构和算法，提供了一个离散事件模拟引擎，实现了包含定量抽象精化和系统归约等分析验证技术，对于规模较大的随机系统模型验证效率较高，一定程度地支持多目标定量性质随机模型检验和统计随机模型检验.

(2) MRMC

MRMC(Markov Reward Model Checker)[117]，是 Katoen 教授课题组用 C 语言实现的较为广泛使用的一种命令行随机模型检验工具，可运行于 Windows、Linux 和 Mac OS X 等操作系统. MRMC 可以验证 DTMC、CTMC、DMRM(Discrete Time Markov Reward Model)、CMRM(Continuous Time Markov Reward Model)和 CTMDP 等随机系统模型，其性质规约包括 PCTL、CSL、PRCTL(Probabilistic Reward Computation Tree Lobic)和 CSRL(Continuous Stochastic Reward Logic). MRMC 采用基于稀疏矩阵的数据结构和算法，支持精确的 on-the-fly 稳态检查和互模拟最小化等验证技术.

在验证连续时间随机系统模型的稳态性质方面具有一定的优势,对于规模较小的随机系统模型验证效率较高.

(3)其他工具

①DTMC、CTMC 及其扩展模型的检验工具:PARAM[101],支持参数随机模型检验,可验证带有参数的 DTMC 及其回报扩展的随机系统模型的可达性质;INFAMY[118],支持无限状态模型的随机模型检验,可验证具有无限状态的 CTMC 类随机系统模型;②MDP 模型检验工具:LiQuor[119],显示状态模型检验 MDP 的 LTL 性质,对于规模较大的随机系统模型,其运行效率要比 PRISM 低;ProbDiVinE-MC[120],支持并行分布式随机模型检验,可验证 MDP 类随机系统模型的概率的 LTL 性质规约;PASS[121],支持谓词抽象精化的随机模型检验,可验证具有无限状态的 MDP 和 DTMC 类随机系统模型;③PTA 模型检验工具:Fortuna[122],可计算 PTA 的最大可达概率及代价 PTA 的回报值;mcpta[123],以 PRISM 为基础支持建模语言 Modest 的 PTA 验证;UPPAAL PRO[124],可计算 PTA 的最大概率可达性质.

5 随机模型检验的应用挑战

随着随机模型检验理论和工具的发展,随机模型检验已经在有些领域得到了初步应用,如:随机分布式算法的正确性验证和性能分析[125]、多媒体通信协议的服务质量分析[126-127]、生物化学过程的建模与分析[128]、系统性能和可靠性分析[129]、随机优化调度与规划[130]和电源管理系统的性能分析与优化[130-131]等.目前随机模型检验主要面临以下挑战:

(1)随机混成系统模型验证

随机模型检验目前验证的随机系统模型是时间同构的,即状态迁移和时间驱动的行为独立于全局时间,但是这类模型不足以建模诸如表现为浴盆曲线的硬件故障、由于重启内存降低和增加的软件可靠性和功率削减与剩余能量成非线性关系的电源消耗等随机现象.为了分析验证这类随机现象,就要涉及异构的马尔可夫链或马尔可夫决策过程,甚至是非马尔可夫性质的随机系统的建模与验证.随机混成系统模型的验证是随机模型检验的一个挑战,文献[132-133]做了一定的尝试工作.

(2)无限状态空间的随机系统模型验证

对于诸如无限大缓冲区的通信网络系统或含有递归调用过程的系统,其系统模型的状态空间是无限的,分析与验证这类系统需要不可判定或半可判定逻辑等理论,是随机模型检验的一个挑战.

(3)合成系统模型

随机模型检验可以用于分析与验证系统的设计模型,而将其用于系统的设计过程本身是随机模型检验的一个挑战.这里的设计过程是指在保证某些性质的前提下,合成系统参数或合成诸如控制器之类的系统构件,涉及此类问题是否属于可判定的范畴.

(4)普适计算系统模型验证

普适计算强调的是与环境融为一体的计算概念,使无数的计算机服务无缝地集成为用户提供所需的服务,这就需要保证它们的正确性,这类带有动态特征的系统建模与验证是随机模型检验的一个挑战.

6 结束语

随机模型检验是用模型检验的方法对展现随机行为的复杂系统进行定量验证和分析,是一种基于模型的自动的形式验证随机系统方法.本文从渊源、最基本原理、研究方向及现状、优势/劣势及工具支持和应用挑战等方面对随机模型检验方法进行分析、总结和论述.其实,如图 9 所示,随机模型检验的研究方向是对其原理研究的深化和优化,使其更真实地验证系统性质和提高验证效率;随机模型检验的应用挑战是其原理应用的推广和扩展,使其可验证系统范围扩大;而随机模型检验研究方向的进展,可以有效应对其应用挑战;解决目前随机模型检验的应用挑战,也促使其研究方向的进展.随着目前计算机系统运行环境开放程度的增大和系统本身复杂性的提高,随机模型检验作为对该类系统进行定量分析验证的新方法,将会有更为广阔的应用前景和发展空间.

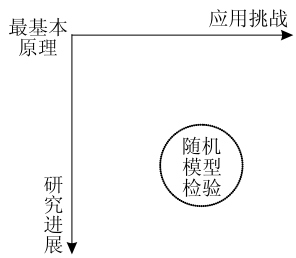

图 9 随机模型检验原理、进展和挑战间的关系

致 谢 感谢德国 RWTH Aachen University 的 Joost-Pieter Katoen 教授,他热情地给了我们很多关于随机模型检验研究的帮助,从他的书籍论文,尤

其是与他的直接交谈中受到很多启发.本文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!

参 考 文 献

[1] Baier C, Katoen J P. Principles of Model Checking. Cambridge Massachusetts: MIT Press, 2008

[2] Clarke E M. The birth of model checking//Grumberg O, Veith H eds. 25 Years of Model Checking: History, Achievements, Perspectives. Berlin: Springer, 2008: 1-26

[3] Turing A M. On computable numbers with an application to the Entscheidungs problem//Proceedings of the London Mathematical Society. London, UK, 1936: 230-265

[4] Turing A M. Checking a large routine//Proceedings of the Inaugural Conference of the EDSAC Computer. Cambridge, UK, 1949: 67-69

[5] Floyd R W. Assigning meanings to programs//Proceedings of the Symposium on Applied Mathematics (Mathematical Aspects of Computer Science). Providence, USA, 1967: 9-32

[6] Hoare C A R. An axiomatic basis for computer programming. Communications of the ACM, 1969, 12(10): 576-580

[7] Pnueli A. The temporal logic of programs//Proceedings of the 18th IEEE Symposium on Foundations of Computer Science. Providence, USA, 1977: 46-67

[8] Clarke E M, Emerson E A. Design and synthesis of synchronization skeletons using branching time temporal logic//Proceedings of the Workshop on Logic of Programs. New York, USA, 1981: 52-71

[9] Queille J P, Sifakis J. Specification and verification of concurrent systems in CESAR//Proceedings of the 5th Colloquium on International Symposium on Programming. Turin, Italy, 1982: 337-351

[10] Lin Hui-Min, Zhang Wen-Hui. Model checking: Theories, techniques and applications. Acta Electronica Sinica, 2002, 30(12A): 1907-1912(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): 1907-1912)

[11] Clarke E M, Emerson E A, Sifakis J. Model checking: Algorithmic verification and debugging. Communications of the ACM, 2009, 52(11): 74-84

[12] Kwiatkowska M, Norman G, Parker D. Stochastic model checking//Proceedings of the 7th International Conference on Formal Methods for Performance Evaluation. Bertinoro, Italy, 2007: 220-270

[13] Hart S, Sharir M, Pnueli A. Termination of probabilistic concurrent programs//Proceedings of the 9th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. Albuquerque, Mexico, 1982: 1-6

[14] Hart S, Sharir M, Pnueli A. Termination of probabilistic concurrent program. ACM Transactions on Programming Languages and Systems, 1983, 5(3): 356-380

[15] Sharir M, Pnueli A, Hart S. Verification of probabilistic programs. SIAM Journal on Computing, 1984, 13(2): 292-314

[16] Vardi M Y. Automatic verification of probabilistic concurrent finite state programs//Proceedings of the 26th Annual Symposium on Foundations of Computer Science. Portland, USA, 1985: 327-338

[17] Vardi M Y, Wolper P. An automata-theoretic approach to automatic program verification//Proceedings of the 1st Symposium on Logic in Computer Science. Cambridge, UK, 1986: 322-331

[18] Pnueli A, Zuck L. Probabilistic verification. Information and Computation, 1993, 103(1): 1-29

[19] Courcoubetis C, Yannakakis M. Verifying temporal properties of finite state probabilistic programs//Proceedings of the 29th Annual Symposium on Foundations of Computer Science. New York, USA, 1988: 338-345

[20] Courcoubetis C, Yannakakis M. The complexity of probabilistic verification. Journal of the ACM, 1995, 42(4): 857-907

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

[22] Bianco A, de Alfaro L. Model checking of probabilistic and nondeterministic systems//Proceedings of the Foundations of Software Technology and Theoretical Computer Science. Bangalore, India, 1995: 499-513

[23] Aziz A, Sanwal K, Singhal V, Brayton R. Verifying continuous time Markov chains//Proceedings of the 8th International Conference on Computer Aided Verification. New Jersey, USA, 1996: 269-276

[24] Aziz A, Sanwal K, Singhal V, Brayton R. Model-checking continuous time Markov chains. ACM Transactions on Computational Logic, 2000, 1(1): 162-170

[25] Baier C, Katoen J P, Hermanns H. Approximate symbolic model checking of continuous-time Markov chains//Proceedings of the 10th International Conference on Concurrency Theory. Eindhoven, Netherlands, 1999: 146-161

[26] Baier C, Haverkort B, Hermanns H, Katoen J P. Model-checking algorithms for continuous-time Markov chains. IEEE Transactions on Software Engineering, 2003, 29(6): 524-541

[27] Ash R B, Doleans-Dade C A. Probability and Measure Theory. Waltham, Massachusetts, USA: Academic Press, 1999

[28] Kemeny J, Snell J, Knapp A. Denumerable Markov Chains. 2nd Edition. Berlin: Springer, 1976

[29] Aziz A, Singhal V, Balarin F, et al. It usually works: The temporal logic of stochastic systems//Proceedings of the 7th International Conference on Computer Aided Verification. Liege, Belgium, 1995: 155-165

[30] Vardi M, Wolper P. Reasoning about infinite computations. Information and Computation, 1994, 115(1): 1-37

[31] Safra S. On the complexity of omega-automata//Proceedings of the 29th IEEE Symposium on Foundations of Computer Science. Los Alamitos, USA, 1988: 319-327

[32] Katoen J P, Kwiatkowska M, Norman G, Parker D. Faster and symbolic CTMC model checking//Proceedings of the 1st Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification. Aachen, Germany, 2001: 23-38

[33] Hermanns H, Katoen J P, Meyer-Kayser J, Siegle M. A Markov chain model checker//Proceedings of the 6th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Germany, 2000: 347-362

[34] Baier C, Haverkort B, Hermanns H, Katoen J P. Model checking continuous-time Markov chains by transient analysis//Proceedings of the 12th International Conference on Computer Aided Verification. Chicago, USA, 2000: 358-372

[35] Kwiatkowska M, Norman G, Parker D. Advances and challenges of probabilistic model checking//Proceedings of the 48th Annual Allerton Conference on Communication, Control and Computing. Allerton, USA, 2010: 1691-1698

[36] Liu Y, Miao H K, Zeng H W, et al. 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

[37] Petri C A. Introduction to general net theory//Brauer W. Lecture Notes in Computer Science 84. Berlin: Springer, 1980: 1-19

[38] Beauquier D. On probabilistic timed automata. Theoretical Computer Science, 2003, 292(1): 65-84

[39] 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

[40] Alur R, Dill D L. A theory of timed automata. Theoretical Computer Science, 1994, 126(2): 183-235

[41] Kwiatkowska M, Norman G, Parker D, Norman G. Performance analysis of probabilistic timed automata using digital clocks. Formal Methods in System Design, 2006, 29(1): 33-78

[42] Kwiatkowska M, Norman G, Sproston J. Symbolic model checking of probabilistic timed automata using backwards reachability. School of Computer Science, University of Birmingham, Birmingham, United Kingdom: Technical Report CSR-00-1, 2000

[43] Howard R A. Dynamic Programming and Markov Processes. New Jersey, USA: John Wiley and Sons, 1960

[44] Bertsekas D P. Dynamic Programming and Optimal Control. 3rd Edition. Cambridge Massachusetts: MIT Press, 2011

[45] Neuhäußer M R, Zhang L. Time-bounded reachability probabilities in continuous-time Markov decision processes//Proceedings of the 7th International Conference on the Quantitative Evaluation of Systems. Williamsburg, USA, 2010: 209-218

[46] Buchholz P, Hahn E M, Hermanns H, Zhang L. Model checking algorithms for CTMDPs//Proceedings of the 23rd International Conference on Computer Aided Verification. Snowbird, USA, 2011: 225-242

[47] Baier C, Hermanns H, Katoen J P, Haverkort B R. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. Theoretical Computer Science, 2005, 345(1): 2-26

[48] Hermanns H. Interactive Markov Chains and the Quest for Quantified Quality. Berlin: Springer, 2002

[49] Zhang L, Neuhäußer M R. Model checking interactive Markov chains//Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Paphos, Cyprus, 2010: 53-68

[50] Chen T, Forejt V, Kwiatkowska M, et al. Playing stochastic games precisely//Proceedings of the 23rd International Conference on Concurrency Theory. Newcastle upon Tyne, UK, 2012: 348-363

[51] Kwiatkowska M, Norman G, Sproston J, Wang F. Symbolic model checking for probabilistic timed automata. Information and Computation, 2007, 205(7): 1027-1077

[52] Alur R, Courcoubetis C, Dill D. Model checking in dense real time. Information and Computation, 1993, 104(1): 2-34

[53] Henzinger T, Nicollin X, Sifakis J, Yovine S. Symbolic model checking for real-time systems. Information and Computation, 1994, 111(2): 193-244

[54] Forejt V, Kwiatkowska M, Norman G, et al. Quantitative multi-objective verification for probabilistic systems//Proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Sarbrücken, Germany, 2011: 112-127

[55] de Alfaro L. Formal Verification of Probabilistic Systems [Ph. D. dissertation]. Stanford University, California, USA, 1997

[56] Hermanns H, Katoen J P, Meyer-Kayser J, Siegle M. Towards model checking stochastic process algebra//Proceedings of the 2nd International Conference on Integrated Formal Methods. Dagstuhl Castle, Germany, 2000: 420-439

[57] Hermanns H, Katoen J P, Meyer-Kayser J, Siegle M. Implementing a model checker for performability behaviour//Proceedings of the 5th International Workshop on Performability Modeling of Computer and Communication Systems. Erlangen, Germany, 2001: 110-115

[58] Obal W, Sanders W. State-space support for path-based reward variables. Performance Evaluation, 1999, 35(3/4): 233-251

[59] Fischer M, Ladner R. Propositional dynamic logic of regular programs. Journal of Computer and System Sciences, 1979, 8(2): 194-211

[60] Baier C, Cloth L, Haverkort B, et al. Model checking pathCSL//Proceedings of the 6th International Workshop Performability Modeling of Computer and Communication Systems. Illinois, USA, 2003: 19-22

[61] Kuntz M, Siegle M. A stochastic extension of the logic PDL//Proceedings of the 6th International Workshop Performability Modeling of Computer and Communication Systems. Illinois, USA, 2003: 58-61

[62] Baier C, Cloth L, Haverkort B R, et al. Model checking Markov chains with actions and state labels. IEEE Transactions on Software Engineering, 2007, 33(4): 209-224

[63] Clarke E M, Jha S, Lu Y, Veith H. Tree-like counterexamples in model checking//Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science. Copenhagen, Denmark, 2002: 19-29

[64] Han T, Katoen J P. Counterexamples in probabilistic model checking//Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Braga, Portugal, 2007: 72-86

[65] Han T, Katoen J P, Damman B. Counterexample generation in probabilistic model checking. IEEE Transactions on Software Engineering, 2009, 35(2): 241-257

[66] Aljazzar H, Leue S. Counterexamples for model checking of Markov decision processes. University of Konstanz, Baden-Württemberg, Germany: Technical Report Soft-08-01, 2007

[67] Han T, Katoen J P. Providing evidence of likely being on time-Counterexample generation for CTMC model checking //Proceedings of the 5th International Symposium on Automated Technology for Verification and Analysis. Tokyo, Japan, 2007: 331-346

[68] Chatterjee K, Henzinger T, Jhala R, Majumdar R. Counterexample guided planning//Proceedings of the 21st International Conference on Uncertainty in Artificial Intelligence. Edinburgh, Scotland, 2005: 104-111

[69] Hermanns H, Wachter B, Zhang L. Probabilistic CEGAR// Proceedings of the 20th International Conference on Computer Aided Verification. Princeton, USA, 2008: 162-175

[70] Chadha R, Viswanathan M. A counterexample guided abstraction-refinement framework for Markov decision processes. ACM Transactions on Computational Logic, 2010, 12(1): 1-45

[71] Zhang Jun-Hua, Huang Zhi-Qiu, Cao Zi-Ning. Counterexample generation for probabilistic timed automata model checking. Journal of Computer Research and Development, 2008, 45(10): 1638-1645(in Chinese)
(张君华，黄志球，曹子宁. 模型检测基于概率时间自动机的反例产生研究. 计算机研究与发展，2008，45(10)：1638-1645)

[72] Clarke E M, Grumberg O, Jha S, et al. Counterexample-guided abstraction refinement//Proceedings of the 12th International Conference on Computer Aided Verification. Chicago, USA, 2000: 154-169

[73] Fecher H, Leucker M, Wolf V. Don't know in probabilistic systems//Proceedings of the 13th International Conference on Model Checking Software. Vienna, Austria, 2006: 71-88

[74] Jonsson B, Larsen K. Specification and refinement of probabilistic processes//Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science. Amsterdam, Netherlands, 1991: 266-277

[75] Katoen J P, Klink D, Leucker M, Wolf V. Three-valued abstraction for probabilistic systems. Journal on Logic and Algebraic Programming, 2012, 81(4): 356-389

[76] D'Argenio P, Jeannet B, Jensen H, Larsen K. Reachability analysis of probabilistic systems by successive refinements// Proceedings of the Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification. Aachen, Germany, 2001: 39-56

[77] 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

[78] Kattenbelt M, Kwiatkowska M, Norman G, Parker D. Abstraction refinement for probabilistic software//Proceedings of the 10th International Conference on Verification, Model Checking, and Abstract Interpretation. Savannah, USA, 2009: 182-197

[79] Kwiatkowska M, Norman G, Parker D. Stochastic games for verification of probabilistic timed automata//Proceedings of the 7th International Conference on Formal Modeling and Analysis of Timed Systems. Budapest, Hungary, 2009: 212-227

[80] Kattenbelt M, Kwiatkowska M, Norman G, Parker D. A game-based abstraction-refinement framework for Markov decision processes. Oxford University Computing Laboratory, Oxford, UK: Technical Report RR-08-06, 2008

[81] Katoen J P, Kemna T, Zapreev I S, Jansen D N. Bisimulation minimisation mostly speeds up probabilistic model checking// Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Braga, Portugal, 2007: 87-101

[82] Kwiatkowska M, Norman G, Parker D. Symmetry reduction for probabilistic model checking//Proceedings of the 18th International Conference on Computer Aided Verification. Seattle, USA, 2006: 234-248

[83] Baier C, Größer M, Ciesinski F. Partial order reduction for probabilistic systems//Proceedings of the 1st International Conference on Quantitative Evaluation of Systems. Enschede, Netherlands, 2004: 230-239

[84] Segala R. Modeling and Verification of Randomized Distributed Real-Time Systems [Ph. D. dissertation]. Department of Electrical Engineering and Computer Science, MIT, USA, 1995

[85] Kwiatkowska M, Norman G, Parker D, Qu H. Assume guarantee verification for probabilistic systems//Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Paphos, Cyprus, 2010: 23-37

[86] Cobleigh J M, Giannakopoulou D, Păsăreanu C S. Learning assumptions for compositional verification//Proceedings of the 9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Warsaw, Poland, 2003: 331-346

[87] Feng L, Kwiatkowska M, Parker D. Compositional verification of probabilistic systems using learning//Proceedings of the 7th International Conference on Quantitative Evaluation of Systems. Williamsburg, USA, 2010: 133-142

[88] Komuravelli A, Păsăreanu C S, Clarke E M. Assume-guarantee abstraction refinement for probabilistic systems//Proceedings of the 24th International Conference on Computer Aided Verification. Berkeley, USA, 2012: 310-326

[89] Buchholz P. A new approach combining simulation and randomization for the analysis of large continuous time Markov chains. ACM Transactions on Modeling and Computer Simulation, 1998, 8(2): 194-222

[90] Tiechroew D, Lubin J F. Computer simulation—discussion of the techniques and comparison of languages. Communications of the ACM, 1966, 9(10): 723-741

[91] Legay A, Delahaye B, Bensalem S. Statistical model checking: An overview//Proceedings of the 1st International Conference on Runtime Verification. St. Julians, Malta, 2010: 122-135

[92] Younes H L S, Simmons R G. Probabilistic verification of discrete event systems using acceptance sampling//Proceedings of the 14th International Conference on Computer Aided Verification. Copenhagen, Denmark, 2002: 223-235

[93] Sen K, Viswanathan M, Agha G. On statistical model checking of stochastic systems//Proceedings of the 14th International Conference on Computer Aided Verification. Edinburgh, UK, 2005: 266-280

[94] Zapreev I S. Model Checking Markov Chains: Techniques and Tools [Ph. D. dissertation]. University of Twente, Netherlands, 2008

[95] Basu S, Ghosh A P, He R. Approximate model checking of PCTL involving unbounded path properties//Proceedings of the 11th International Conference on Formal Engineering Methods: Formal Methods and Software Engineering. Rio de Janeiro, Brazil, 2009: 326-346

[96] Sen K, Viswanathan M, Agha G. Statistical model checking of black-box probabilistic systems//Proceedings of the 16th conference on Computer Aided Verification. Boston, USA, 2004: 202-215

[97] Younes H L S. Probabilistic verification for "black-box" systems//Proceedings of the 17th Conference on Computer Aided Verification. Edinburgh, UK, 2005: 253-265

[98] Younes H L S, Kwiatkowska M, Norman G, Parker D. Numerical vs. statistical probabilistic model checking: An empirical study. International Journal on Software Tools for Technology Transfer, 2006, 8(3): 216-228

[99] Grosu R, Smolka S A. Monte Carlo model checking//Proceedings of the 11th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Edinburgh, UK, 2005: 271-286

[100] Laplante S, Lassaigne R, Magniez F, et al. Probabilistic abstraction for model checking: An approach based on property testing. ACM Transactions on Computational Logic, 2007, 8(4): Article No. 20

[101] Hahn E M, Hermanns H, Wachter B, Zhang L. PARAM: A model checker for parametric Markov models//Proceedings of the 22th International Conference on Computer Aided Verification. Edinburgh, UK, 2010: 660-664

[102] Alur R, Henzinger T A, Vardi M Y. Parametric real-time reasoning//Proceedings of the 25th Annual Symposium on Theory of Computing. San Diego, USA, 1993: 592-601

[103] Tanimoto T, Nakata A, Hashimoto H, Higashino T. Double depth first search based parametric analysis for parametric time-interval automata. IEICE Transaction on Fundamentals, 2005, E88-A(11): 3007-3021

[104] Daws C. Symbolic and parametric model checking of discrete-time Markov Chains//Proceedings of the 1st International Colloquium on Theoretical Aspects of Computing. Guiyang, China, 2004: 280-294

[105] Lanotte R, Maggiolo-Schettini A, Troina A. Decidability results for parametric probabilistic transition systems with an application to security//Proceedings of the 2nd International Conference on Software Engineering and Formal Methods. Beijing, China, 2004: 114-121

[106] Lanotte R, Maggiolo-Schettini A, Troina A. Parametric probabilistic transition systems for system design and analysis. Formal Aspects of Computing, 2007, 19(1): 93-109

[107] Hahn E M, Hermanns H, Zhang L. Probabilistic reachability for parametric Markov models. International Journal on Software Tools for Technology Transfer, 2011, 13(1): 3-19

[108] Hahn E M, Han T, Mereacre A, Zhang L. Synthesis for PCTL in parametric Markov decision processes//Proceedings of the 3rd International Conference on NASA Formal Methods. Pasadena, USA, 2011: 146-161

[109] Miner A, Parker D. Symbolic representations and analysis of large probabilistic systems//Miner A, Parker D eds. Validation of Stochastic Systems: A Guide To Current Research. Berlin: Springer, 2004: 296-338

[110] Della Penna G, Intrigila B, Melatti I, et al. Bounded probabilistic model checking with the murphi verifier//Proceedings of the 5th International Conference on Formal Methods in Computer-Aided Design. Austin, USA, 2004: 15-17

[111] 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)

[112] Filieri A, Ghezzi C, Tamburrelli G. Run-time efficient probabilistic model checking//Proceedings of the 33rd ACM/IEEE International Conference on Software Engineering. Honolulu, USA, 2011: 341-350

[113] Hinton A, Kwiatkowska M, Norman G, Parker D. PRISM: A tool for automatic verification of probabilistic systems//Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Vienna, Austria, 2006: 441-444

[114] Kwiatkowska M, Norman G, Parker D. PRISM 4.0: Verification of probabilistic real-time systems//Proceedings of the 23rd International Conference on Computer Aided Verification. Snowbird, USA, 2011: 585-591

[115] 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

[116] Parker D. Implementation of Symbolic Model Checking for Probabilistic Systems [Ph. D. dissertatopm]. University of Birmingham, Birmingham, UK, 2002

[117] Katoen J P, Hahn E, Hermanns H, et al. The ins and outs of the probabilistic model checker MRMC//Proceedings of the 6th International Conference on Quantitative Evaluation of Systems. Budapest, Hungary, 2009: 167-176

[118] Hahn E M, Hermanns H, Wachter B, Zhang L. INFAMY: An infinite-state Markov model checker//Proceedings of the 21st International Conference on Computer Aided Verification. Grenoble, France, 2009: 641-647

[119] Ciesinski F, Baier C. LiQuor: A tool for qualitative and quantitative linear time analysis of reactive systems//Proceedings of the 3rd International Conference on Quantitative Evaluation of Systems. Riverside, USA, 2006: 131-132

[120] Barnat J, Brim L, Cerna I, et al. ProbDiVinE-MC: Multi-core LTL model checker for probabilistic systems//Proceedings of the 5th International Conference on Quantitative Evaluation of Systems. St Malo, France, 2008: 77-78

[121] Hahn E M, Hermanns H, Wachter B, Zhang L. PASS: Abstraction refinement for infinite probabilistic models//Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Paphos, Cyprus, 2010: 353-357

[122] Berendsen J, Jansen D, Vaandrager F. Fortuna: Model checking priced probabilistic timed automata//Proceedings of the 7th International Conference on Quantitative Evaluation of Systems. Williamsburg, USA, 2010: 273-281

[123] Hartmanns A, Hermanns H. A modest approach to checking probabilistic timed automata//Proceedings of the 6th International Conference on Quantitative Evaluation of Systems. Budapest, Hungary, 2009: 187-196

[124] Gerd B, Alexandre D, Kim G L, et al. Developing UPPAAL over 15 years. Software-Practice and Experience, 2011, 41(2): 133-142

[125] Ndukwu U, McIver A. An expectation transformer approach to predicate abstraction and data independence for probabilistic programs//Proceedings of the 8th Workshop on Quantitative Aspects of Programming Languages. Paphos, Cyprus, 2010: 129-143

[126] Kwiatkowska M, Norman G, Parker D. Analysis of a Gossip protocol in PRISM. ACM SIGMETRICS Performance Evaluation Review, 2008, 36(3): 17-22

[127] Duflot M, Kwiatkowska M, Norman G, et al. Practical Applications of probabilistic model checking to communication protocols//Gnesi S, Margaria T eds. Formal Methods for Industrial Critical Systems: A Survey of Applications. Los Alamitos, CA, USA: IEEE Computer Society, 2013: 133-150

[128] Heath J, Kwiatkowska M, Norman G, et al. Probabilistic model checking of complex biological pathways. Theoretical Computer Science, 2008, 391(3): 239-257

[129] Kwiatkowska M, Norman G, Parker D. Controller dependability analysis by probabilistic model checking. Control Engineering Practice, 2007, 15(11): 1427-1434

[130] Katoen J P. Model checking: One can do much more than you think!//Proceedings of the 4th International Conference on Fundamentals of Software Engineering. Tehran, Iran, 2011: 1-14

[131] Katoen J P. Perspectives in probabilistic verification//Proceedings of the 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering. Nanjing, China, 2008: 3-10

[132] Han T, Katoen J P, Mereacre A. Compositional modeling and minimization of time-inhomogeneous Markov chains//Proceedings of the 11th International Workshop on Hybrid Systems: Computation and Control. St. Louis, USA, 2008: 244-258

[133] Chen T, Han T, Katoen J P, Mereacre A. LTL model checking of time-inhomogeneous Markov chains//Proceedings of the 7th International Symposium on Automated Technology for Verification and Analysis. Macao SAR, China, 2009: 104-119

LIU Yang, born in 1981, Ph. D. His research interests include software engineering, and formal verification.

LI Xuan-Dong, born in 1963, Ph. D., professor. His research interests include software modeling and analysis, and software testing and verification.

MA Yan, born in 1981, lecturer. His research interests include software engineering, and hybrid system verification.

WANG Lin-Zhang, born in 1973, Ph. D., associate professor. His research interests include software engineering, and software testing.

Background

In the recent decade, stochastic model checking caused widespread concern in the formal verification area, and has achieved great development. Around this topic there are some well-known research groups, such as the quantitative analysis and verification research group at the University of Oxford, reliability systems and software research group at the University of Saarland in Germany, and Software Modeling and Verification research group at RWTH Aachen University in Germany. Stochastic model checking is a recent extension and generalization of the classical model checking, which can verify and analyze system model qualitatively and quantitatively, and has been applied to the random distributed algorithm verification, communication protocol performance analysis and even interdisciplinary areas such as systems biology. In this paper, we summarize the main research direction and progress of stochastic model checking, analyze the advantages/disadvantages of stochastic model checking deeply, classify and list tools for stochastic model checking, and point out its future challenge. This work was supported by the National Natural Science Foundation of China under Grant Nos. 61303022 and 91318301, the China Postdoctoral Science Foundation under Grant No. 2013M531328, the Natural Science Foundation of Shandong Province of China under Grant No. ZR2012FQ013, the Project of Shandong Province Higher Educational Science and Technology Program under Grant No. J13LN10, and the Science and Technology Program of Taian under Grant No. 201330629. The above projects study the complex software modeling, analysis and verification. Stochastic model checking can verify complex software quantitatively, such as reliability or resource utilization. In this direction, we have presented a high-level method to model and verify complex software with nondeterminism and probability, which is called the NPPN; and we also proposed a kind of method for verifying trustworthy service flow.