DOI:10.11992/tis.201706008
网络出版地址:http://kns.cnki.net/kcms/detail/23.1538.TP.20170728.1901.012.html

实时并发系统的 PTSL 模型检测

王晓燕[1],韩啸[1,2],彭君[1],刘淑芬[1]
(1.吉林大学 计算机科学与技术学院,吉林 长春 130012; 2. 吉林大学 学报编辑部,吉林 长春 130012)

摘　要:随着实时并发系统的软件规模越来越大、复杂性日趋增加,如何保证并发实时系统正确性和可靠性成为日益紧迫的问题。模型检测技术采用自动化的验证算法判断系统是否具有某一性质,它不仅包括对系统模型的遍历以及基于图形的分析方法,而且还需要大量的数值计算。本文把实时并发模型看成对并发博弈模型(CGS)的扩展,在此基础上添加了概率与时间性质,提出了概率时间并发博弈结构(PTCGS)。同时本文还提出了新的逻辑语言-概率时间策略逻辑(PTSL),它显式地把策略作为一阶逻辑中的对象,从而使我们能够以简单而自然的方式指定 PTCGS 系统中的非零和属性。PTSL 模型检测方法能够让设计者准确知道模型是否满足用户的需求,从而提高系统的可靠性。最后,本文以 ZeroConf 协议为例来说明 PTSL 模型检测方法的正确性。

关键词:模型检测;概率时间并发博弈结构;概率时间策略逻辑;概率时间自动机;区域图;实时并发系统;博弈模型
中图分类号:TP311　**文献标志码**:A　**文章编号**:1673-4785(2017)05-0694-08

中文引用格式:王晓燕,韩啸,彭君,等.实时并发系统的 PTSL 模型检测[J]. 智能系统学报, 2017, 12(5): 694-701.
英文引用格式: WANG Xiaoyan, HAN Xiao, PENG Jun, et al. PTSL model checking of timed concurrent system[J]. CAAI transactions on intelligent systems, 2017, 12(5): 694-701.

PTSL model checking of timed concurrent system

WANG Xiaoyan[1], HAN Xiao[1,2], PENG Jun[1], LIU Shufen[1]
(1.College of Computer Science and Technology, Jilin University, Changchun 130012, China; 2. Editorial Office on Journal, Jilin University, Changchun 130012, China)

Abstract: With the increased scale and complexity of real-time concurrent software systems, ensuringtheir correctness and reliability has become a matter of urgency. Current model-checking technology uses an automated demonstration algorithm to judgesystem properties, which must include the traversal in the system model and graph-based analysis techniques as well asextensive numerical computations. In this paper, we consider the timed concurrency model as an extension of the concurrent game model (CGS) and addprobability and time properties to propose a new probabilistic timed concurrent game structure (PTCGS). We also propose a new logic language called probabilistic timed strategy logic (PTSL), which uses strategy as the object in the first-order logic to specify the nonzero-sum attributes in a simple and natural way. Lastly, we give an example usingthe ZeroConf protocol to demonstrate the correctness of the PTSL model checking method.

Keywords: model checking; probabilistic timed concurrent game structure; probabilistic timed strategy logic; probabilistic timed automata; region graph; timed concurrent system; game model

实时并发系统被广泛应用在各领域中,例如并发和分布式实时软件、离散事件模拟、通信网络的建模等。这些系统中通常包含若干个并发构件,这些构件之间使用实时信号通信,并具有随机性与不确定性。随着计算机技术的发展,实时并发系统的软件规模越来越大、复杂性日趋增加,如何保证其正确性和可靠性成为日益紧迫的问题,而且由于其内在的非确定性,这个问题难度更大。

在为此提出的诸多理论和方法中,模型检测

收稿日期:2017-06-05.　网络出版日期:2017-07-28.
基金项目:国家自然科学基金项目(61502196).
通信作者:王晓燕. E-mail:wangxy@jlu.edu.cn.

(model checking)以其简洁明了和自动化程度高而引人注目[1]。模型检测作为形式化验证的一种主流技术,它可以克服传统软件测试用例生成不完备的不足,同时具有验证自动化的优点,并且当验证的性质不满足时,能给出违背性质的反例。模型检测采用了严格的形式化方法对系统进行验证,因此比测试和仿真更能保证系统的正确性。模型检测中常使用数学抽象模型对系统进行建模,概率时间自动机(probabilistic timed automata, PTA)就是其中的一种[2-3],由于它能同时表示概率性、不确定性和实时性,因此是模型检测实时并发系统的有效工具。

模型检测通过对状态空间的穷举搜索来判定系统是否具有所关心的性质 ϕ,但是随着实时并发系统越来越多地应用在一些对安全性、可靠性要求非常高的领域,如航空系统、电力系统、智能交通系统等,系统一旦发生故障所带来的后果不堪设想,因此设计者需要确切知道在什么情况下系统会出现最坏或者最好的结果,这时候就需要通过模型检测技术找到满足特定性质 ϕ 的某个策略或者最佳策略。

目前模型检测主要用时序逻辑来刻画所关心模型的性质,常用的逻辑包括线性时态逻辑 LTL(linear temporal logic)、概率计算树逻辑 PCTL(probabilistic computation tree logic)、概率时间计算树逻辑 PTCTL(probabilistic timed computation tree logic)、交替时间时态逻辑 ATL(alternating-time temporal logic)等,当系统中存在多个 agent 时,LTL、PCTL、PTCTL 等不能够明确描述每个 agent 对象所使用的策略[4-6],而 ATL 又不能够描述并发系统中的非零和逻辑[7-8],于是 Henzinger 等提出了策略逻辑(strategy logic,SL)[9],但是 SL 中缺少对不确定性以及时间的逻辑,因此本文提出概率时间策略逻辑(probabilistic timed strategy logic,PTSL),把策略作为第一实体对象,能够针对每个模型所使用的策略进行描述,从而使我们能够以简单而自然的方式指定博弈系统中的非零和逻辑属性。

本文的目的是在概率时间并发博弈结构模型基础上,为 PTSL 逻辑语言提供验证方法。本文的主要贡献如下:

1)在并发博弈结构基础上提出了概率时间并发博弈结构模型;

2)提出的 PTSL 逻辑语言,它将策略和时态与概率度量相结合;

3)提出基于区域图的 PTSL 模型检测算法,并证明了区域图中的路径与 PTA 中的路径的等价性。

1　相关工作

对实时并发系统的模型检测可以分为 2 类:对并发模型的研究;对并发模型中使用的逻辑语言的研究。在实时并发系统中,通常使用一些控制策略,该策略模型与系统并发模型交互作用,因此完全可以把实时并发模型看成对博弈模型的扩展。而目前博弈模型通常分为两种类型:一种是轮流(turn-based)博弈模型[10],在整个系统中,可以存在多个玩家,但是在每个状态只能有一个玩家做出选择,从而系统转换到下一个状态;还有一种是并发(concurrent)博弈模型[11],同样在整个系统中,可以存在多个玩家,但是在每个状态多个玩家可以同时且独立地做出选择。Alfaro 等[12]提出了时间博弈自动机,每个游戏者不仅可以选择可能的转换,同时还可以选择转换发生前的等待时间。Thomas Brihaye 等[13]提出了时间并发博弈结构 TCGS,并使用新的逻辑语言 TALTL 进行验证。目前在并发模型中使用的逻辑语言的研究更丰富多彩。ATL 是博弈模型中的一种典型,用来描述零和逻辑的语言[14]。Taolue Chen 等[15]在 ATL 基础上添加了概率算子,提出了 PATL 逻辑语言,Henzinger 等提出了以策略为第一对象的非零和逻辑语言 SL 并给出了模型检测算法,Christel Baier 等[16]也在 ATL 基础上提出了一种新的逻辑语言 Stochastic Game Logic。

2　概率时间自动机与概率时间并发博弈结构

2.1　时钟与概率时间自动机

时钟 x 表示记录时间的非负实数变量,设 $X=\{x_1,x_2,\cdots,x_n\}$ 是有限的时钟变量集合,则时钟赋值函数 $u:X\to R_{\geqslant 0}$ 表示对所有 $x_i\in X$ 分配一个非负实数。设 $t\geqslant 0$,$u+t$ 表示对所有 $x_i\in X$ 增加 t,即 $u+t(x)=u(x)+t$,其中 $x\in X$。通常使用 $\bar{0}$ 表示对 X 中的所有时钟赋值 0。时钟约束是一组有关时间变量的原子公式的合取,记为 $C(X)$,语法定义为 $Z::=x\sim d\,|\,x-y\sim d\,|\,Z\wedge Z\,|\,\text{true}$,其中 $x,y\in X$,$d\in N$,$\sim\in\{<,\leqslant\}$。当时钟赋值 v 满足时钟约束 X 时,记为 $v\models x$。所有满足时钟约束的时钟赋值构成的集合称为时区(zone)。

定义 1　概率时间自动机(probabilistic timed automata, PTA)。PTA 是一个八元组 $\mathbb{P}=(L,l_0,\text{Act},X,\text{prob},\text{inv},\text{enab},\mathscr{L})$,其中 L 表示自动机的位置集合;初始位置 $l_0\in S$;Act 表示有限的动作集合;X 表示 $\mathbb{P}$ 中使用的时钟集合;$\text{inv}:L\to C(X)$ 是环境函数,它为每个位置指定一个不变式;$\text{prob}:L\times\text{Act}\to$

$\mathrm{Dist}(2^X\times L)$表示概率转移函数，$\mathrm{Dist}(L)$表示状态 L 上的条件转移的概率，且 L 上的所有条件转移的概率和为 1，即 $\sum_{l\in L}\theta(l)=1,\theta:L\to[0,1]$；$\mathrm{enab}:L\times\mathrm{Act}\to C(X)$ 表示动作使能条件；标签函数 $\mathscr{L}:L\to 2^{\mathrm{AP}}$ 表示每个位置上的原子命题。

PTA 中的符号状态定义为二元组(l,v)，其中 l 是符号状态的位置信息，η 是时钟赋值且 $\eta\models\mathrm{inv}(l)$。因此在符号状态$(l,\eta)$上，要么有一定的时间流逝，要么某个动作 $m\in\mathrm{Act}$ 被执行，即

1）时间迁移：$(l,\eta)\xrightarrow{d}(l',\eta')$，当且仅当 $l=l',\eta'=\eta+d$，并且 $\eta'\models\mathrm{inv}(l)$。

2）位置迁移：$(l,\eta)\xrightarrow{m}(l',\eta')$，当且仅当 $\eta\models\mathrm{enab}(l,m)$ 并且

$$\mu(l',\eta')=\sum\{|\mathrm{prob}(l,\eta)(X,l')\,|X\in 2^X\wedge\eta'=\eta[X:=0]|\}$$

$(X,l')\in 2^X\times L$ 表示支持动作 m 转换的一条边，在位置(l,η)上，所有的转换边记为$\langle e_1,e_2,\cdots,e_m\rangle$。从位置迁移的定义可以看出，一旦一个动作 m 被选择，则时钟会重置并随机选择后继位置。

定义 2 PTA 组合（PTA composition）。两个 PTA（$\mathbb{P}_1$和 $\mathbb{P}_2$）的组合定义为：$\mathbb{P}_1\parallel\mathbb{P}_2=(L_1\times L_2,(\bar{l}_1,\bar{l}_2),\mathrm{Act}_1\cup\mathrm{Act}_2,X_1\cup X_2,\mathrm{prob},\mathrm{inv},\mathrm{enab},\mathscr{L})$，其中对于位置 $(l_1,l_2)\in L_1\times L_2$ 及动作 $m\in\mathrm{Act}_1\cup\mathrm{Act}_2$：

$$\mathrm{inv}(l_1,l_2)=\mathrm{inv}_1(l_1)\wedge\mathrm{inv}_2(l_2)$$

$$\mathrm{enab}((l_1,l_2),m)=\begin{cases}\mathrm{enab}_1(l_1,m)\wedge\mathrm{enab}_2(l_2,m),m\in\mathrm{Act}_1\cup\mathrm{Act}_2\\ \mathrm{enab}_1(l_1,m),m\in\mathrm{Act}_1\setminus\mathrm{Act}_2\\ \mathrm{enab}_2(l_2,m),m\in\mathrm{Act}_2\setminus\mathrm{Act}_1\end{cases}$$

$$\mathrm{prob}((l_1,l_2),m)=\begin{cases}\mathrm{prob}_1(l_1,m)\otimes\mathrm{prob}_2(l_2,m),m\in\mathrm{Act}_1\cup\mathrm{Act}_2\\ \mathrm{prob}_1(l_1,m)\otimes\mu_{(\phi,l_2)},m\in\mathrm{Act}_1\setminus\mathrm{Act}_2\\ \mu_{(\phi,l_1)}\otimes\mathrm{prob}_2(l_2,m),m\in\mathrm{Act}_2\setminus\mathrm{Act}_1\end{cases}$$

$$\mathscr{L}(l_1,l_2)=\mathscr{L}_1\cup\mathscr{L}_2$$

2.2 概率时间并发博弈结构

由于可以把实时并发模型看成对博弈模型的扩展，本文首先给出由 Henzinger 等提出的并发博弈 CGS 模型的定义，如下所示。

定义 3 并发博弈结构（concurrent game structure，CGS）。CGS 是一个 8 元组 $G=(Q,Q_0,\Sigma,\tau,\mathrm{Agt},M,\Gamma,\mathrm{Edg})$，其中 Q 是 CGS 的有限状态集合，Q_0 是初始状态，Σ 表示输入与输出动作集合，$\tau:Q\times M\to\mu(M)$ 是概率转换函数，其中 $\mu(M)$ 是概率分布函数，$\mathrm{Agt}=\{a_1,a_2,\cdots,a_k\}$ 是系统中 k 个 agent 的集合，M 是系统中所有 agent 可能的动作集合，而 Γ 定义了每个 agent 的动作，$\mathrm{Edg}:Q\times M^k\to\tau$ 表示状态转换表，它包含所有 agent 的每个状态的状态转移函数。本文使用 $\mathrm{Edg}(q,m_{a_1},m_{a_2},\cdots,m_{a_k})$ 表示状态 q 上的所有 agent 的动作。

从上面的定义可以看出，在 CGS 中，由状态 q_i 转换成状态 q_{i+1} 是由所有的 agent 共同作用的结果，这也就是说，每个 agent $a_j\in\mathrm{Agt}$ 都会根据当前的状态 q_i 选择一个动作 $m_j\in\mathrm{Mv}(q_i,a_j)$，因此状态 q_i 转换成状态 q_{i+1} 需使用 $\mathrm{Edg}(q_i,a_1,a_2,\cdots,a_k)$ 表示。

在 CGS 模型中缺少时钟概念，因此本文对 CGS 模型进行扩展，在模型中添加时钟，其定义如下所示。

定义 4 概率时间并发博弈结构（probabilistic timed concurrent game structure，PTCGS）。PTCGS 是一个十元组 $T=(Q,Q_0,\Sigma,X,\mathrm{Inv},\tau,\mathrm{Agt},M,\Gamma,\mathrm{Edg})$，其中$(Q,Q_0,\Sigma,X,\mathrm{Inv},\tau)$就是 PTA，而 Agt、$M$、$\Gamma$、Edg 的含义与 CGS 中一样。

因此在 PTCGS 系统中，每个 agent 都可以使用一个 PTA 来表示，整个 PTCGS 系统就是多个 PTA 的集合。

定义 5 路径（path）。PTCGS 的一条路径 ρ 是一条有限或者无限序列：$\rho=(q_0,\eta_0)\xrightarrow{t_0,m_0,p_0}(q_1,\eta_1)\xrightarrow{t_1,m_1,p_1}\cdots$。本文使用 $s(\rho,k)$ 表示路径中的第 $k+1$ 个状态。而 $m(\rho,k)$ 表示路径中的第 $k+1$ 个动作，$\mathrm{last}(\rho)$ 表示最后一个状态。

定义 6 路径持续时间（duration of a path）。给定 PTA 模型 $\mathbb{P}$及其路径 ω，则 $D_\omega(i)=\sum_{j=0}^{i-1}t_j$ 表示路径 ω 上的节点 i 发生位置迁移时流逝的时间和，当 $i=0$ 时，$D_\omega(0)=0$。

定义 7 策略与联合策略（strategy and coalition strategy）。策略通常表示系统中的 agent 在不确定性环境下如何在每个状态选择动作，从而找到满足某个性质的解决方案。agent a_i 的策略 σ_i 是一个偏序函数，它将有限的路径映射到概率分布函数，即 $\sigma_i:\rho\to\mathfrak{D}(M)$，且满足当 $m\in M(\mathrm{last}(\rho))$ 时，$\sigma(\rho)(m)>0$。联合策略 σ_A 是指包含多个 agent 的策略集合。如果 σ_A 中包含所有的 agent 的策略，即 $A=\mathrm{Agt}$，则称其为完全联合策略，否则为不完全联合策略。

3 概率时间策略逻辑语言

为了描述博弈系统的非零和逻辑，Chatterjee 和 Henzinger 提出了策略逻辑（stragegy logic，SL），该

语言把策略作为第一实体对象,但缺少概率与时间特性。本文在 SL 基础上提出概率时间策略逻辑(probabilistic timed strategy logic, PTSL),其语法如下所示:

$$\varphi ::= p \mid \zeta \mid \neg\ \varphi \mid \varphi_1\ \wedge\ \varphi_2 \mid$$
$$\Phi ::= \varphi_1\ \cup^{\leqslant k} \varphi_2 \mid z.\Phi$$
$$\Lambda ::= \forall x(a,x)\Phi \mid \exists x(a,x)\Phi \mid \Lambda_1\ \wedge\ \Lambda_2 \mid \neg\ \Lambda$$
$$\psi ::= P_{\sim\lambda}\Lambda$$

为了更好地描述时间性质,在 PTSL 中引入了一个新的时钟符号$\mathbb{Z}$,称之为公式时钟,且$\chi \cap \mathbb{Z} = \varnothing$,$\chi$ 是 PTA 模型中的时钟,称为系统时钟。$\zeta \in \mathbb{Z}_{(\chi \cup \mathbb{Z})}$是一个时间区域 Zone。$z.\phi$ 表示从时钟 $z=0$ 的状态开始搜索满足 ϕ 的路径,且 $z \in \mathbb{Z}$。$\forall x \cdot \varphi$ 含义是任意策略 x 都满足 φ,而 $\exists x.\varphi$ 表示存在一个策略 x 满足 φ,$(a,x)\varphi$ 是策略赋值公式,表示模型 a 使用策略 x 且满足 φ,$\sim \in \{<, \ll, >, \gg\}$,$\lambda \in [0,1]$,$k \in \mathbf{R}^+$,概率算子 P 表示满足公式 φ 和界限符 $\sim\lambda$ 的路径的概率。在本文中,我们只考虑有两个 agent 的并发实时系统,因此使用 $x_1, x_2, \cdots$,表示一个模型使用的策略,用 $y_1, y_2, \cdots$表示另外一个模型使用的不同策略。

因此使用 PTSL 语言可以表达如下的属性,例如在 ZeroConf 协议中,发送者在发送信息后要求接收者要在 1s 内接收到消息的概率大于等于 99%,

$$P_{\geqslant 0.99}[\ \forall x_1 \forall y_1(s,x_1)(r,y_1)z.[\text{s.l.} = 12 \cup (r.l = 4\ \wedge\ z \leqslant 1)]$$

定义 8　扩展的时间区域。由于在 PTSL 中存在公式时钟变量 z,因此本文将 $[\eta, \Im]$ 称为扩展的时间区域,其中 η 是系统时钟,$\Im$ 是公式时钟。从而本文使用$(l,[\eta,\Im])$表示 PTA 中的状态,称为扩展的符号状态。

4　基于 region graph 的模型检测算法

模型检测 PTSL 的问题就是,给定概率时间并发博弈结构 PTCGS $\mathbb{T}$及 PTSL 的表达式$\varnothing$,能否找到 σ_A 使之满足$\varnothing$。在 TCGS 系统中,通常使用的属性验证公式是概率公式 $P_{\sim\lambda}\Lambda$,并且在公式中通常也含有公式时钟 z,因此求解满足 $\text{Prob}\left(\left\{\omega \middle| \begin{matrix}\omega \in \text{Path}\langle l,[\eta,\Im]\rangle \\ (l,[\eta,\Im]),\sigma_A \models \Lambda\end{matrix}\right\}\right) \sim \lambda$ 公式的 σ_A 是算法的关键。

4.1　基本知识

对于时钟变量 x,用 k_x 表示时钟 x 的上界值。对于实数 t,用 $\text{frac}(t)$表示 t 的小数部分,用$\lfloor t \rfloor$表示 t 的整数部分。

定义 9　时钟等价(clock equivalence)。两个时钟赋值 v 和 v'是等价的,记为 $v \approx v'$,当且仅当满足下列所有条件:

1)对于每一个时钟变量 x,要么$\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$,要么 $v(x) > k_x$ 并且 $v'(x) > k_x$;

2)对于所有的时钟变量 x 和 x',如果 $v(x) \leqslant k_x$ 并且 $v'(x) \leqslant k_x$,那么

①$\text{frac}(v(x)) = 0$ 当且仅当 $\text{frac}(v'(x)) = 0$;

②$\text{frac}(v(x)) \leqslant \text{frac}(v(x'))$ 当且仅当 $\text{frac}(v'(x)) \leqslant \text{frac}(v'(x'))$。

本文使用$[v]$表示时钟 v 所属的等价类,同样可以将$\approx$扩展到符号状态,使用$\langle l,[v]\rangle$表示状态相同,时间等价的等价类,称其为区域(region)。由于在 PTSL 有公式时钟,因此在 TCGS 系统中使用扩展的符号状态$(l,[\eta,\Im])$,而使用$\langle l,[\eta,\Im]\rangle$表示扩展区域。在扩展区域 α 中,使用 $\text{zone}(\alpha)$ 表示 α 中的等价时间类。

在 PTA 模型 G 中存在多个状态,有的状态存在后继状态,与之对应的区域也会有后继区域,而由多个区域组成的模型称为区域图。

定义 10　后继区域(successor region)。扩展区域$\beta = \langle l,[\eta',\Im']\rangle$是$\alpha = \langle l,[\eta,\Im]\rangle$的后继区域,当且仅当存在正数 $t \in \mathbb{R}, t' \in \mathbb{R}$,当$\langle l,[\eta'+t', \Im'+t']\rangle \in \beta$,则对任意 $t \leqslant t'$,都有$\langle l,[\eta+t,\Im+t]\rangle \in \alpha \cup \beta$,记为 $\text{succ}(\alpha) = \beta$。

定义 11　区域图(region graph)。与 PTA $\mathbb{P}$对应的区域图 $\mathbb{R}$是一个三元组 $\mathbb{R}(\mathbb{P},\Lambda) = (R^*, \text{Steps}^*, L^*)$,其中 R^* 表示扩展区域的集合,对于任意扩展区域$\alpha = \langle l,[\eta,\Im]\rangle \in R^*$转换函数 Steps^* 包含以下 3 种类型的转换。

1)时间迁移:当 $\text{succ}([\eta,\Im]) \models \text{inv}(l)$,则

$$p^{\alpha}_{\text{succ}}\beta = \begin{cases}1, & \beta = \text{succ}(\alpha)\\ 0, & \text{其他}\end{cases}$$

2)位置迁移:如果存在概率 $p' \in \text{prob}(l)$ 而且$[\eta,\Im]$满足 PTA 中的位置转换条件 $\tau_l(p')$,则

$$p^{\alpha}_{\text{succ}}\beta = \sum_{\substack{X \subseteq \chi \\ \text{zone}(\alpha)[X_0] = \text{zone}(\beta)}} p'(l',X)$$

3)自身循环

$$p^{\alpha}_{\text{succ}}\beta = \begin{cases}1, & \beta = \alpha\\ 0, & \text{其他}\end{cases}$$

标签函数 L^* 的定义如下所示:

$$L^*\langle l,[\eta,\Im]\rangle = L(l) \cup \{p_\zeta \mid [\eta,\Im] \text{ 满足 } \zeta, \zeta \in Z_{(\chi \cup \mathbb{Z})}(\Lambda)\}$$

由于在验证过程中需要使用区域图上的策略,其定义如下。

定义 12　区域图上路径(path on the region graph)。$\omega^* = \langle l_0,[\eta_0,\Im_0]\rangle \xrightarrow{p_0} \langle l_1,[\eta_1,\Im_1]\rangle$

$\xrightarrow{p_1}\langle l_2,[\eta_2,\Im_2]\rangle\xrightarrow{p_2}\cdots$，其中 $p_i\in \text{Step}\ s^*\langle l_i,[\eta_i,\Im_i]\rangle$，且 $p_i>0$。

定义 13 区域图上的策略(adversary)。在区域图中的策略 σ_R 将有限路径 ω^* 映射到概率 p 且 $p\in \text{Steps}^*(\text{last}(\omega^*))$。

下面介绍在区域图上的 PTSL 满足关系。

定义 14 PTSL 的满足关系。给定区域图 $\mathbb{R}(\mathbb{P},\Lambda)$ 以及 PTSL 公式 ϕ，则 PTSL 在 $\mathbb{R}$ 上的满足关系定义如下：

$(l,[\eta,\Im])\models$true，对于所有 $l,\eta,\Im$ 均成立

$(l,[\eta,\Im])\models p\Leftrightarrow p\in L^*(l,[\eta,\Im])$

$(l,[\eta,\Im])\models\phi_1\wedge\phi_2\Leftrightarrow$

$(l,[\eta,\Im])\models\phi_1$ and $(l,[\eta,\Im])\models\phi_2$

$(l,[\eta,\Im])\models\neg\phi\Leftrightarrow(l,[\eta,\Im])\models\phi$

$(l,[\eta,\Im])\models\Lambda_1\wedge\Lambda_2\Leftrightarrow$

$(l,[\eta,\Im])\models\Lambda_1$ and $(l,[\eta,\Im])\models\Lambda_2$

$(l,[\eta,\Im])\models\neg\Lambda\Leftrightarrow(l,[\eta,\Im])\models\Lambda$

$(l,[\eta,\Im])\models\phi_1\cup^{\leqslant k}\phi_2\Leftrightarrow$存在整数 i,j 及路径 ω^*，且 $\omega^*(i)\models\phi_1$，且对于任意 $i\leqslant j\leqslant i+k$，都有 $\omega^*(j)\models\phi_2$。

以上定义的满足关系与 PCTL 的满足关系的定义基本相同，而 PTSL 中独有的 3 个操作符——时钟重置 z、任意算子 $\forall$ 以及存在算子 $\exists$ 满足关系定义如下。

$$(l,[\eta,\Im])\models z.\phi\Leftrightarrow$$
$$(l,[\eta,\Im[z:=0]])\models\sigma_A\phi$$
$$(l,[\eta,\Im])\models\forall x(a,x)\Phi\Leftrightarrow\forall x\in\sigma\text{ and }a\in A,(l,[\eta,\Im])\models\Phi$$
$$(l,[\eta,\Im])\models\exists x(a,x)\Phi\Leftrightarrow\exists x\in\sigma\text{ and }a\in A,(l,[\eta,\Im])\models\Phi$$

将带有概率算子公式 $P_{\sim\lambda}\Lambda$ 展开，可以得到公式(1)、(2)、(3)：

$$(l,[\eta,\Im])\models P_{\geqslant\lambda}\forall x\forall y(a,x)(b,y)\Phi\quad(1)$$
$$(l,[\eta,\Im])\models P_{\geqslant\lambda}\forall x\exists y(a,x)(b,y)\Phi\quad(2)$$
$$(l,[\eta,\Im])\models P_{\geqslant\lambda}\exists x\exists y(a,x)(b,y)\Phi\quad(3)$$

公式(1)~(3)的概率值以及 σ_A 的求解过程是本文验证算法的核心。

4.2 验证算法

本节将介绍基于区域图的 PTSL 验证算法。本文将该算法分为 3 步：1)构建 TCGS 系统的区域图；2)在区域图上解析 PTSL 公式；3)在 TCGS 系统中找到满足公式的路径。

从 PTA 构造区域图的方法按照区域图的定义就可以得到，本文不再给出转换算法。本文重点介绍在区域图上解析 PTSL 公式的过程，首先仍旧是构建 PTSL 的语法树。在语法树上，叶节点代表原子命题，而内部节点标识 PTSL 中的操作符标识包括 $\wedge$，$\neg$，$\cup^{\leqslant k}$，P 等，PTSL 的验证算法如下所示。

算法 1 PTSL 验证算法

foreach φ' in sub(φ) do

case $\varphi'=p$: $[\varphi']:=\{(l,[\eta,\Im])\mid l\in L\&l\in\mathcal{L}(p)\}$

case $\varphi'=\neg\theta$: $[\varphi']:=\{(l,[\eta,\Im])\mid(l,[\eta,\Im])\not\models\theta\}$

case $\varphi'=\theta_1\wedge\theta_2$: $[\varphi']:=\{(l,[\eta,\Im])\mid(l,[\eta,\Im])\models\theta_1\cap(l,[\eta,\Im])\models\theta_2\}$

case $\varphi'=\zeta$: $[\varphi']:=\{(l,[\eta,\Im])\mid l\in L\wedge\Im\in p_\zeta\}$

case $\varphi'=z.\theta$:

$[\varphi']:=\{(l,[\eta,\zeta[z:=0]])\mid(l,[\eta,\zeta])\models\theta\}$

case $\varphi'=\theta_1\cup^{\leqslant k}\theta_2$: Until$(\theta_1,\theta_2,\leqslant k)$

case $\varphi'=P_{\sim\lambda}\forall x\forall y(a,x)(b,y)\theta$: $[\varphi']:=P_1\theta$

case $\varphi'=P_{\sim\lambda}\forall x\exists y(a,x)(b,y)\theta$: $[\varphi']:=P_2\theta$

case $\varphi'=P_{\sim\lambda}\exists x\exists y(a,x)(b,y)\theta$: $[\varphi']:=P_3\theta$

…

PTSL 与 PCTL 的不同之处在于存在以下几个操作：时钟重置 $z.\phi$、任意 $\forall x(a,x)\cdot\Phi$ 以及存在 $\exists x(a,x)\cdot\Phi$，因此对于 PTSL 中其他的操作符，例如 $\neg$，$\wedge$，$\vee$，$\cup^{\leqslant k}$ 等，都可以采用 PCTL 中原有的算法，这里不作详细介绍。ζ 是公式时钟，p_ζ 的定义在区域图中已经给出，表示满足时间赋值的命题。依据 PCTL 语义，$P_{\sim\lambda}\Lambda$ 的计算方法如下所示：

$$P_{\sim\lambda}\Lambda=\begin{cases}p_{\max}\Lambda, & \sim\in\{<,\leqslant\}\\ p_{\min}\Lambda, & \sim\in\{>,\geqslant\}\end{cases}$$

从定义可以看出，求解 $P_{\sim\lambda}\Lambda$ 的值的问题其实是求解极值的问题。依据这个思路，我们来叙述式(1)~(3)的计算过程。

为了叙述的方便，下面将 $\sim\lambda$ 直接定义为 $\geqslant\lambda$。从 PTSL 的满足关系可知，式(1)的含义是，无论 agent a 和 agent b 使用任何策略，所找到的路径的目标状态都满足 Φ 且概率大于等于 λ，因此 λ 是 agent a 和 agent b 使用联合策略 $\sigma_A=(x,y)$ 下所得的最小值。从而我们可以将式(1)转换为求解最小值问题。式(2)中 $P_{\geqslant\lambda}\exists x\exists y(a,x)(b,y)\Phi$ 的含义是，agent a 和 agent b 存在某个策略，所找到的路径的目标状态满足 Φ 且概率 $\geqslant\lambda$，这也就是说，找到一条满足条件的路径即可，因此可以将式(2)转换为求解最大概率问题。式(3)的含义是无论 agent a 使用任何策略，agent b 都会找到某个策略 y 使所找到的路径的目标状态满足 Φ 且概率 $\geqslant\lambda$。因为 a 可以使用任意策略，所以式(3)其实是求解最小概率问题。以上将式(1)~(3)求解概率 $\geqslant\lambda$ 的情形进行了分析，而 $\leqslant\lambda$ 恰好是相反的情形，在这里不再赘述。有关在区域图中寻找概率最优解的方法与在 MDP 中

的方法一样,可以使用 value iteration 的方法查找整个模型中的概率最大最小值[17]。而公式时钟 z 值的计算则是将找到的满足 θ 的路径中的各个区域的时间最大值相加起来即可。

当在区域图中找到了满足条件的路径 ω^* 后,这并不是最终解,还需要在 TCGS 系统中构建与 ω^* 相对应的路径 ω,另外还需要证明 ω^* 与 ω 的概率是一样的,这样才能保证最终找到的路径 ω 是正确的。

构建 ω 过程采用递归方法,算法如下所示。

算法 2　构建 w 过程

BuildPathFromRG($\mathcal{R}$,ω^*)

length := ($|\omega^*|$; i := 0; ω := $\varnothing$; Trans := $\varnothing$;

While i<length

if(i=0) &(η_0,$\Im$) $\in$ [η_0,$\Im_0$]

ω:$\omega \cup (l_0,\eta_0)$;

else

if (η_i,$\Im+D_\omega(i)$) $\in$ [η_i,$\Im_i$]

ω: = $\omega \cup (l_i,\eta_i)$;

Trans: = Trans $\cup$ { (l_{i-1},η_{i-1}),(m_i,p_i),(l_i,η_i) }

return (ω,Trans)

证明　ω^* 与 ω 的概率是一样的,即Prob(ω^*)= Prob(ω)。设在区域 r_i 发生转换前概率为 p,区域 r_i 中包含 m 个时间转换,n 个位置转换,k 个自身循环。从区域图的定义可以看出,当发生时间迁移和自身循环时概率均为 1,且这两个迁移不会改变区域位置,转换后的概率为 $p\times1\times1\times1\cdots=p$,故而概率值没有任何改变。而发生位置迁移时, ω^* 中的概率为 $p_{i*}=\{\Sigma p_i(X,l_{i+1})\mid X\subseteq\chi,\text{zone}(\alpha)[X:=0]=\text{zone}(\beta)\}$,$\omega$ 中的概率为 $\mu=\{\Sigma p_i(X,l_{i+1})\mid X\in 2^\chi \wedge (\eta_i+t_i)[X:=0]=\eta_{i+1}\}$,由于区域 r_i 中有 m 个时间转换,因此 $\text{zone}(\alpha)=[\eta_i+d_m,\varepsilon+D_\omega(i)+d_m]$,$\text{zone}(\beta)=[\eta_{i+1},\varepsilon+D_\omega(i+1)+d_{i+1}]$,而$(\eta_i+t_i)[X:=0]\in[\eta_i+d_m][X:=0]$,$\eta_{i+1}\in[\eta_{i+1}]$,因此 $p_i^*=\mu$。

5　实验与结果

本文以 ZeroConf 协议为例说明 PTSL 的模型检测方法。ZeroConf 协议是一种用于局域网内自动生成可用 IP 地址的网络技术。设备 S 连接网络后,首先随机地选择一个 IP 地址,然后设备 S 会向网络中的其他设备连续 K 次发送“这个 IP 地址是否已经被占用”的询问信息, 如果设备 S 没有收到任何应答,则它会使用该 IP 地址,否则只要收到过 1 个“IP 地址被占用”的应答,设备 S 就会重新选择 IP 地址。本文使用的模型是由 Cheshire 等建立的 MDP 模型修改而来[18],为 environment 模型添加了消息丢失的概率,表示网络中的其他设备在应答设备 S 的询问时,发送的信息有可能发生丢失,本文使用参数 plost 表示。则 sender 模型和 environment 模型如图 1、2 所示。

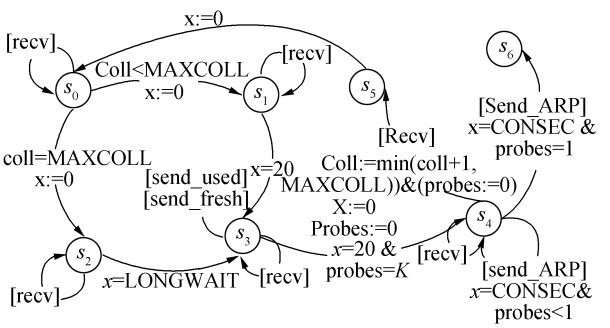


图 1　sender 模型

Fig.1　sender model

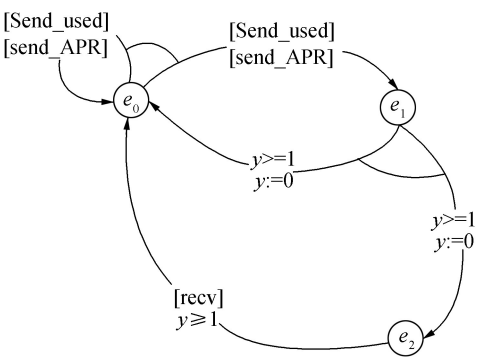


图 2　environment 模型

Fig.2　environment model

为了验证模型,在 Prism 中重新设计实现了 PTSL 逻辑语言,并使用 PTA 模型表示并发实时系统中的 agent,使用 PTA 的组合模型表示整个 PTCGS 模型。组合后的 PTA 模型转换成区域图后,状态节点一共有 631 个,在本文中不能全部给出,本文只给出几个关键节点的图,如图 3 所示。

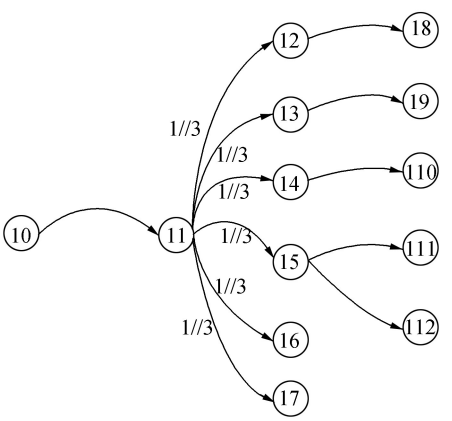


图 3　区域图模型片段

Fig.3　Fragment of region graph model

图 3 中每个节点的值如下所示：

$l[0]=((s=0,\text{probes}=0,\text{ip}=0,\text{coll}=0,e=0),\{x=0,y=0,x=y\})$

$l[1]=((s=1,\text{probes}=0,\text{ip}=0,\text{coll}=0,e=0),\{x=0,y=0,\text{x=y}\})$

$l[2]=((s=3,\text{probes}=0,\text{ip}=1,\text{coll}=0,e=0),\{x=0,y=20,\text{x-y}=-20\})$

$l[3]=((s=3,\text{probes}=0,\text{ip}=1,\text{coll}=0,e=0),\{x=1,y=20,x-y=-19\})$

$l[4]=((s=3,\text{probes}=0,\text{ip}=1,\text{coll}=0,e=0),\{x=2,y=20,x-y=-18\})$

$l[5]=((s=3,\text{probes}=0,\text{ip}=2,\text{coll}=0,e=0),\{x=0,y=20,x-y=-20\})$

$l[6]=((s=3,\text{probes}=0,\text{ip}=2,\text{coll}=0,e=0),\{x=1,y=20,x-y=-19\})$

$l[7]=((s=3,\text{probes}=0,\text{ip}=2,\text{coll}=0,e=0),\{x=2,y=20,x-y=-18\})$

$l[8]=((s=3,\text{probes}=1,\text{ip}=1,\text{coll}=0,e=0),\{x=0,y=40,x-y=-40\})$

$l[9]=((s=3,\text{probes}=1,\text{ip}=1,\text{coll}=0,e=0),\{x=0,y=39,x-y=-39\})$

$l[10]=((s=3,\text{probes}=1,\text{ip}=1,\text{coll}=0,e=0),\{x=0,y=38,x-y=-38\})$

$l[11]=((s=3,\text{probes}=1,\text{ip}=2,\text{coll}=0,e=0),\{x=0,y=0,x=y\})$

$l[12]=((s=3,\text{probes}=1,\text{ip}=2,\text{coll}=0,e=1),\{x=0,y=0,x=y\})$

首先求新入网的设备可以成功分配到未使用的 IP 的概率，使用 PTSL 的属性公式为

$$P_{\geqslant\lambda}\,\forall x\,\forall y(a,x)(b,y)(\text{true}\ \cup\ (s=6\&\text{ip}=2))$$

表 1 为测试时的数据取值，从结果可以看出，当网络中节点数 N 值变化较大时，最后成功获得 IP 地址的概率不会发生很大的变化。

表 1 ZeroConf 的实验结果 1

Table 1 Result 1 of ZeroConf experiment

N 节点数	发包次数 K	丢失率 Plost	概率 Probability
10	2	0.1	0.999 8
100	2	0.1	0.998 7
1 000	2	0.1	0.987 5
10	2	0.01	0.999 8
50	2	0.01	0.999 2
100	2	0.01	0.998 4
1 000	2	0.01	0.984 9

使用 PTSL 求解在大于等于 T 时间内设备仍未成功分配到 IP 的概率的属性公式为

$$P_{\geqslant\lambda}\,\forall x\,\exists y(a,x)(b,y)z.(!\ (s=6\&\text{ip}=2)\ \wedge\ z\geqslant T)$$

表 2 为测试时的数据取值，从结果可以看出，当时间限界 T 较小时（≤10），发生不能配置合适的 IP 的概率大一些，而时间限界 $T\geqslant 20$ 时不能成功获得 IP 地址的概率几乎为 0，完全符合 ZeroConf 协议的标准。

表 2 ZeroConf 的实验结果 2

Table 2 Result 2 of ZeroConf experiment

N 节点数	发包次数 K	丢失率 Plost	时间 T	概率 Probability
100	2	0.1	10	0.001 4
100	2	0.1	20	0.000 100 73
100	2	0.1	30	0.000 100 63
100	2	0.1	40	0.000 100 63
100	2	0.01	10	0.001 4
100	2	0.01	20	0.000 001 222 1
100	2	0.01	30	0.000 001 207 7
100	2	0.01	40	0.000 001 207 7

6 结束语

本文在 CGS 模型基础上添加了概率及时间约束，提出了一种新的并发模型 PTCGS。并根据 PTCGS 特点，提出了新的逻辑语言 PTSL，它在 SL 逻辑的基础上添加了时间与概率特性，把策略作为第一实体对象，能够针对每个模型所使用的策略进行描述，从而使我们能够以简单而自然的方式指定 PTCGS 系统中的非零和逻辑属性，并给出了基于区域图的模型检测算法。最后以 ZeroConf 协议为例来说明 PTSL 的模型检测方法的正确性。

参考文献：

[1] CLARKE E, GRUMBERG O, PELED D. Model Checking [M]. Cambridge : MIT press, 1999: 5-60.

[2] BEAUQUIER D. On probabilistic timed automata [J]. Theoretical computer science, 2003, 292(1): 65-84.

[3] KWIATKOWSKA M, NORMAN G, SEGALA R, et al. Automatic verification of real-time systems with discrete probability distributions[J]. Theoretical computer science , 2002, 282: 101-150.

[4] CLARKE E, EMERSON A. Design and synthesis of

synchronization skeletons using branching time temporal logic[C]//Proceedinns of the Workshop Logic of Programs. Newyork, US, 1981: 52-71.

[5] HANSSON H, JONSSON B. A logic for reasoning about time and reliability[J]. Formal aspects of computing, 1994, 6(5): 512-535.

[6] ALFARO de L. Temporal Logics for the Specification of Performance and Reliability [C]//Proc STACS' 97. New York, US, 1997: 165-176.

[7] ALUR R, HENZINGER T A, KUPFERMAN O. Alternating-time temporal logic[J]. Journal of the ACM, 2002, 49: 672-713.

[8] CHATTERJEE K, HENZINGER T A. Strategy improvement for stochastic Rabin and Strett Games[C]// Proc DBLP 2006. Bonn, Germany, 2006: 375-389.

[9] CHATTERJEE K, HENZINGER T A, PITERMAN N. Strategy logic[J]. Information and computation, 2007, 208 (6): 677-693.

[10] Basset N, Kwiatkowska M, Topcu U, et al. Strategy synthesis for stochastic games with multiple long-run objectives [C]//International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Germany, 2015:256-271.

[11] KRISHNENDU Chatterjee, LUCA de Alfaro, THOMAS A, et al. Strategy improvement for concurrent reachability and turn-based stochastic safety games[J]. Journal of computer and system sciences, 2013, 79: 640-657.

[12] ALFARO L de, FAELLA M, HENZINGER T, et al. The element of surprise in timed games[C]//14th International Conference on Concurrency Theory. Marceille, France , 2003: 144-158.

[13] THOMAS Brihaye, FRAN Cois, LAROUSSINIE, et al. Timed Concurrent Game Structures[C]// Proceedings of the 18th international conference on Concurrency Theory. Lisbon, Portugal, 2007: 445-459.

[14] RAJEEV A, THOMAS A, HENZINGER, et al. Alternating-time temporal logic[J]. Journal of the ACM, 2002 ,49(5): 672-713 .

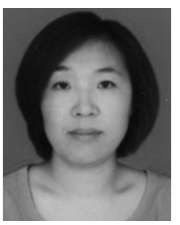
[15] CHEN Taolue, LU Jian. Probabilistic alternating-time temporal logic and model checking algorithm [C]// Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2007. Haikou, China, 2007: 35-39.

[16] CHRISTEL Baier, TOMÁŠ Brázdil, MARCUS Größer, et al. Stochastic game logic[C]// Conference: Quantitative Evaluation of Systems. Edinburgh, UK ,2007: 227-236.

[17] MARTA Kwiatkowska A , GETHIN Norman A , JEREMY Sproston B, et al. Symbolic model checking for probabilistic timed automata[J]. Information and computation, 2007, 205(2): 1027-1077.

[18] MARTA Kwiatkowska, GETHIN Norman, DAVID Parker, et al. Performance analysis of probabilistic timed automata using digital clocks [J]. Formal methods in system design, 2006, 29(1): 33-78.

作者简介:



王晓燕,女,1977 年生,讲师,博士,主要研究方向为软件建模与验证技术、软件开发新方法。申请发明专利 2 项,并获得 2010 年中国国家专利优秀奖,2008 年吉林省科技进步一等奖,中国商业科技进步二等奖。发表论文 20 余篇,其中被 SCI 收录 5 篇。

韩啸,男, 1981 年生,编辑,博士研究生,主要研究方向为网络协同、软件建模和网络技术。

彭君,男,1981 年生,讲师,博士,主要研究方向为模型驱动技术、构件技术、软件体系结构。