

# 广义可能性计算树逻辑的模型检测问题

梁常建<sup>1,2</sup>, 李永明<sup>1</sup>

(1. 陕西师范大学数学与信息科学学院, 陕西西安 710119; 2. 商丘师范学院数学与统计学院, 河南商丘 476000)

**摘要:** 本文首先分别给出了“约束可达”, “总是可达”这两个公式在广义可能性计算树逻辑(GPoCTL)中的另外两种等价形式;其次讨论了基于广义可能性测度的计算树逻辑的模型检测问题,将 GPoCTL 的模型检测问题规约为经典的 CTL 模型检测问题,利用截集的方法,给出了计算 GPoCTL 的模型检测问题的算法及其复杂度,并通过实例分析说明了这种算法的可行性;最后,研究了具有公平性假设的 GPoCTL 模型检测问题的计算复杂度,得到了与上面相似的结论.

**关键词:** 可能性理论; 计算树逻辑; 模型检测; 时间复杂性; 规约

**中图分类号:** TP301.2      **文献标识码:** A      **文章编号:** 0372-2112 (2017)11-2641-08

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2017.11.010

## The Model Checking Problem of Computing Tree Logic Based on Generalized Possibility Measures

LIANG Chang-jian<sup>1,2</sup>, LI Yong-ming<sup>1</sup>

(1. School of Mathematics and Information Science, Shaanxi Normal University, Xi'an, Shaanxi 710119, China;

2. School of Mathematics and Statistics, Shangqiu Normal University, Shangqiu, Henan, 476000, China)

**Abstract:** Firstly, two alternative equivalent forms of GPoCTL state formulas, “until”, “always”, are given respectively. Secondly, it shows that the model checking problem of GPoCTL can be reduced to which of CTL, its algorithm is given through the method of cut set, and whose availability is explained with an example analysis, as a result, the time complexity of the algorithm is obtained. Finally, the properties of the GPoCTL model checking problem with fairness assumptions, which are similar to the GPoCTL, are studied by the similar method with GPoCTL.

**Key words:** possibility theory; computation tree logic; model checking; time complexity; reduction

### 1 引言

经典的模型检测<sup>[1]</sup>已应用于确定性系统的验证,但由于系统日益复杂,在验证过程中不可避免地会出现一些不确定的信息,在正确性要求较高的系统中这种不确定的信息将会产生非常大的影响,并且模型中状态转移之间常常存在很多需要量化的信息,这些信息仅用经典的模型检测已无法处理.为了有效的处理、刻画这些不确定性,人们研究了基于概率测度的随机模型检测<sup>[2]</sup>、基于量子理论的量子<sup>[3]</sup>模型检测、基于多值理论的多值模型检测<sup>[4-6]</sup>、基于模糊逻辑的模糊模型检测<sup>[7]</sup>.特别地,为了处理不满足概率测度可加性的不确定现象,李永明等基于可能性理论<sup>[8]</sup>,定义了

广义可能性 Kripke 结构(GPKS)<sup>[9-12]</sup>,在其上引入了可能性测度,在 GPKS 下研究了线性时序逻辑(GPoLTL)和计算树逻辑(GPoCTL),说明了基于广义可能性测度的时序逻辑比经典的时序逻辑具有更强的表达能力.在此基础上,文献[13]在 GPKS 下研究了基于决策过程的模型检测.相对于多值、模糊时序逻辑,文献[9~12]引入广义可能性测度,使时态性质可度量;相对于随机模型检测,文献[9~12]处理的是满足可能性测度的现象而不是满足概率测度可加性的现象.

文献[11]研究了 GPoCTL 的模型检测问题,给出了 GPoCTL 模型检测问题的基于模糊矩阵的算法,其时间复杂度为  $O(|M| \cdot |S|^3 \cdot |\Phi|)$ . 本文进一步讨论基于广义可能性测度的计算树逻辑的模型检测问题,将

GPoCTL 的模型检测问题规约为经典的 CTL 模型检测问题,并通过相应的算法过程计算  $\|\Phi\|(s)$ ,相对于文献[11],降低了 GPoCTL 模型检测问题的计算复杂度.

## 2 预备知识

**定义 1**<sup>[11,12]</sup> 一个广义的可能性 Kripke 结构(Generalized Possibilistic Kripke Structure, GPKS)  $M = (S, P, I, AP, L)$  是一个五元组,其中:(1)  $S$  是一个可数的非空状态集;(2)  $P: S \times S \rightarrow [0, 1]$  是可能性状态转移分布函数;(3)  $I: S \rightarrow [0, 1]$  是可能性初始分布函数;(4)  $AP$  是一个原子命题集合;(5)  $L: S \times AP \rightarrow [0, 1]$  是可能性标签函数,也可写作  $\forall a \in AP, s \in S, \text{有 } L(s, a) = L(s)(a)$ ,即为每个状态赋一个原子命题的模糊集,表示状态  $s$  满足命题  $a$  的可能性.

**注 1** 对于一个 GPKS  $M$  而言,从状态  $s$  出发的一条路径  $\pi$  用一组无限的状态序列表示,即:  $s_0 s_1 s_2 \dots$ , 其中  $s_0 = s$ , 且对于任意的  $i \geq 0$ , 都有  $P(s_i, s_{i+1}) > 0$ , 记  $\pi_i = s_i, \pi^i = s_i s_{i+1} \dots$ .  $Paths(s)$  表示从状态  $s$  出发的所有路径组成的集合.  $Paths(M)$  表示  $M$  中从初始状态  $s_0 \in I$  出发的所有路径组成的集合. 记  $M_s = (S, P, s, AP, L)$ , 其中初始状态为单点集  $s$ .  $M$  的值为  $M$  中的状态数与转移个数之和, 记为  $|M|$ , 即:  $|M| = |Supp(P)| + |S|$ . 如果  $S$  和  $AP$  都是有限的, 则称  $M$  是有限 GPKS. 本文讨论的 GPKS 都是有限的(本文在逻辑公式的运算中分别用符号  $\vee, \wedge$  和  $\neg$  表示逻辑析取, 逻辑合取和逻辑非; 在逻辑公式的赋值运算中分别用符号  $\vee, \wedge$  和  $\neg$  表示模糊集理论中的实数区间  $[0, 1]$  上的取大运算, 取小运算和否定运算).

**定义 2**<sup>[8]</sup> 设  $X$  是一个非空集合,  $\Omega = 2^X$ , 若函数  $Me: \Omega \rightarrow [0, 1]$  满足条件:(1)  $Me(\phi) = 0$ ; (2) 对任意的  $E_i \in \Omega, i \in I, Me(\bigcup_{i \in I} E_i) = \bigvee_{i \in I} Me(E_i)$ , 则称  $Me$  是  $\Omega$  上的广义可能性测度.

**定义 3**<sup>[11,12]</sup> 设  $M$  是一个 GPKS, 定义映射  $Po^M: Paths(M) \rightarrow [0, 1]$ , 对任意的路径  $\pi = s_0 s_1 \dots$ , 有:  $Po^M(\pi) = I(s_0) \wedge \bigwedge_{i=0}^{\infty} P(s_i, s_{i+1})$ . 若对  $E \subseteq Paths(M)$ , 定义  $Po^M(E) = \bigvee \{Po^M(\pi) \mid \pi \in E\}$ , 则可得扩张映射  $Po^M: 2^{Paths(M)} \rightarrow [0, 1]$ , 易知  $Po^M$  是  $\Omega$  上的广义可能性测度.

**命题 1**<sup>[11,12]</sup> 设  $M$  是一个 GPKS, 映射  $r_p: S \rightarrow [0, 1]$  定义为:  $r_p(s) = \bigvee \{P(s, s_1) \wedge P(s_1, s_2) \wedge \dots \wedge P(s_{i-1}, s_i) \wedge P(s_i, s_{i+1}) \in Paths(M)\}$ , 其中  $s \in S$ , 则有  $r_p(s) = \bigvee \{P^+(s, t) \wedge P^+(t, t) \mid t \in S\}$ .

**定义 4**<sup>[11]</sup> 广义可能性计算树逻辑(Generalized Possibilistic Computation Tree Logic, GPoCTL) 的语法递归定义如下:

GPoCTL 的状态公式  $\Phi$  按如下规则定义:  $\Phi ::=$

$true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid Po(\varphi)$ , 其中  $\varphi$  是 GPoCTL 路径公式,  $a \in AP$ .

GPoCTL 的路径公式  $\varphi$  按如下规则定义:  $\varphi ::= \bigcirc \Phi \mid \Phi_1 \sqcup \Phi_2 \mid \Phi_1 \sqcup^{\leq n} \Phi_2 \mid \square \Phi$ , 其中  $\Phi, \Phi_1, \Phi_2$  是 GPoCTL 状态公式,  $n \in \mathbb{N}$ .

**定义 5**<sup>[11]</sup> GPoCTL 的语义定义如下:

设  $M = (S, P, I, AP, L)$  是一个 GPKS, 对于状态公式  $\Phi$  和任意状态  $s$ , 它的语义解释是一个模糊集  $\|\Phi\|: S \rightarrow [0, 1]$ , 递归定义如下:  $\|true\|(s) = 1$ ;  $\|a\|(s) = L(s, a)$ .

$\|\Phi_1 \wedge \Phi_2\|(s) = \|\Phi_1\|(s) \wedge \|\Phi_2\|(s)$ ;  
 $\|\neg \Phi\|(s) = 1 - \|\Phi\|(s)$ ;  $\|Po(\varphi)\|(s) = Po(s \models \varphi)$ .

对于路径公式  $\varphi$  和任意路径  $\pi$ , 它的语义解释是一个模糊集  $\|\varphi\|: Paths(M) \rightarrow [0, 1]$ , 递归定义如下:

$\|\bigcirc \Phi\|(\pi) = P(s_0, s_1) \wedge \|\Phi(s_1)\|$ .  
 $\|\Phi_1 \sqcup \Phi_2\|(\pi) = \|\Phi_2\|(s_0) \vee \bigvee_{j \geq 1} (\|\Phi_1\|(s_0) \wedge \bigwedge_{0 \leq k < j-1} P(s_k, s_{k+1}) \wedge \|\Phi_1\|(s_{k+1})) \wedge P(s_{j-1}, s_j) \wedge \|\Phi_2\|(s_j)$ ;  
 $\|\Phi_1 \sqcup^{\leq n} \Phi_2\|(\pi) = \|\Phi_2\|(s_0) \vee \bigvee_{1 \leq j \leq n} (\|\Phi_1\|(s_0) \wedge \bigwedge_{0 \leq k < j-1} P(s_k, s_{k+1}) \wedge \|\Phi_1\|(s_{k+1})) \wedge P(s_{j-1}, s_j) \wedge \|\Phi_2\|(s_j)$ ;  
 $\|\square \Phi\|(\pi) = \bigwedge_{j \geq 0} \bigwedge_{0 \leq k < j} P(s_k, s_{k+1}) \wedge \|\Phi\|(s_j)$ . 特别的, 路径公式  $\diamond \Phi = true \sqcup \Phi$  的语义为:  $\|\diamond \Phi\|(\pi) = \bigvee_{j \geq 0} \bigwedge_{0 \leq k < j} P(s_k, s_{k+1}) \wedge \|\Phi\|(s_j)$ .

$Po(s \models \varphi)$  定义如下:  $Po(s \models \varphi) = \bigvee_{\pi \in Paths(M)} Po^M(\pi) \wedge \|\varphi\|(\pi)$ , 直观地,  $Po(s \models \varphi)$  表示从状态  $s$  出发, 且满足公式  $\varphi$  的路径的最大可能性. 若  $M$  是明确的, 则  $M$  是可以省略的.

**命题 2**<sup>[11]</sup> 设  $M$  是一个有限的 GPKS,  $\Phi$  是一个 GPoCTL 状态公式, 则  $\Phi$  的像集  $Im(\Phi)$  是一个有限集, 且  $|Im(\Phi)| \leq |Im(P) \sqcup Im(L)| + 2$ .

## 3 GPoCTL 模型检测问题

在 CTL 中  $\Phi_1 \sqcup \Phi_2$  (约束可达) 和  $\square \Phi$  (总是可达) 的扩张律是成立的, 在 GPoCTL 中  $Po(\Phi_1 \sqcup \Phi_2)$  和  $Po(\square \Phi)$  也存在对应的扩张律. 设  $M$  是一个 GPKS, 对任意状态  $s \in M$  及任意的 GPoCTL 状态公式  $\Phi, \Phi_1, \Phi_2$ ,

**命题 3** 以下结论成立:

(1)  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s) = \|Po(\Phi_1 \sqcup^{\leq |S|} \Phi_2)\|(s)$ .

(2)  $\|Po(\square \Phi)\|(s) = \|Po(\square^{\leq |S|} \Phi)\|(s)$ .

**证明** (1) 对任意的状态公式  $\Phi_1, \Phi_2$  及任意状态  $s \in M$ , 由定义可得,  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s) \geq \|Po(\Phi_1 \sqcup^{\leq |S|} \Phi_2)\|(s)$ ; 因此只要证明  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s) \leq \|Po(\Phi_1 \sqcup^{\leq |S|} \Phi_2)\|(s)$ , 即只需要证明: 对任意的

状态  $s \in M, k \geq |S|$ , 有  $\|Po(\Phi_1 \sqcup^{\leq k} \Phi_2)\|(s) \leq \|Po(\Phi_1 \sqcup^{\leq |S|} \Phi_2)\|(s)$ . 可以利用归纳法证明:

当  $k = |S|$  时, 结论显然成立; 假设当  $k = m > |S|$  时, 有  $\|Po(\Phi_1 \sqcup^{\leq k} \Phi_2)\|(s) \leq \|Po(\Phi_1 \sqcup^{\leq |S|} \Phi_2)\|(s)$ ; 当  $k = m + 1$  时,  $\|Po(\Phi_1 \sqcup^{\leq m+1} \Phi_2)\|(s) = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\bigwedge_{0 \leq k < m+1} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}) \wedge \|\Phi_2\|(s_{m+1})) = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\|\Phi_1 \sqcup^{\leq m} \Phi_2\|(\pi) \vee \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\bigwedge_{0 \leq k < m+1} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}) \wedge \|\Phi_2\|(s_{m+1})))$ , 因为  $m + 1 > |S|$ , 路径  $\pi = s_0 \cdots s_i \cdots s_m s_{m+1} \cdots$  至少包含一个循环, 即存在  $i$  和  $j, 0 \leq i < j \leq m$  使得  $s_i = s_j$ , 令  $\pi' = s_0 \cdots s_i s_j \cdots s_m s_{m+1} \cdots \in Paths(s)$ , 有  $\bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\bigwedge_{0 \leq k < m+1} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}) \wedge \|\Phi_2\|(s_{m+1})) \leq \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\|\Phi_1\|(s_0) \wedge P(s_0, s_1) \wedge \cdots \wedge \|\Phi_1\|(s_i) \wedge P(s_i, s_{j+1}) \wedge \|\Phi_1\|(s_{j+1}) \wedge \cdots \wedge P(s_m, s_{m+1}) \wedge \|\Phi_2\|(s_{m+1})) \leq \bigvee_{\pi' \in Paths(s)} Po(\pi') \wedge (\|\Phi_1 \sqcup^{\leq m} \Phi_2\|(\pi')) \leq \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\|\Phi_1 \sqcup^{\leq m} \Phi_2\|(\pi))$ , 所以  $Po(\Phi_1 \sqcup^{\leq m+1} \Phi_2)(s) \leq \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\|\Phi_1 \sqcup^{\leq m} \Phi_2\|(\pi)) = Po(\Phi_1 \sqcup^{\leq m} \Phi_2)(s) \leq Po(\Phi_1 \sqcup^{\leq |S|} \Phi_2)(s)$ .

(2) 对任意的状态公式  $\Phi$  及任意的状态  $s \in M$ , 由定义易知  $\|Po(\Box\Phi)\|(s) \leq \|Po(\Box^{\leq |S|}\Phi)\|(s)$ , 因此只需证明  $\|Po(\Box\Phi)\|(s) \geq \|Po(\Box^{\leq |S|}\Phi)\|(s)$ . 由于任意路径  $\pi = s_0 \cdots s_i \cdots s_{|S|} \cdots$  至少包含一个循环, 即存在  $i$  和  $t, 0 \leq i < t \leq |S|$  使得  $s_i = s_t$ , 令  $\pi' = s_0 \cdots (s_i \cdots s_t)^w \in Paths(s)$ . 由定义  $\|Po(\Box^{\leq |S|}\Phi)\|(s) = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge \|\Box^{\leq |S|}\Phi\|(\pi) = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge \bigwedge_{0 \leq j \leq |S|} \bigwedge_{0 \leq k < j} P(s_k, s_{k+1}) \wedge \|\Phi\|(s_j) \leq \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge \bigwedge_{0 \leq j \leq |S|} \bigwedge_{0 \leq k < j} P(s_k, s_{k+1}) \wedge \|\Phi\|(s_j) \leq \bigvee_{\pi' \in Paths(s)} Po(\pi') \wedge \bigwedge_{0 \leq j \leq |S|} \bigwedge_{0 \leq k < j} P(s_k, s_{k+1}) \wedge \|\Phi\|(s_j) = \bigvee_{\pi' \in Paths(s)} Po(\pi') \wedge \|\Box\Phi\|(\pi') \leq \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge \|\Box\Phi\|(\pi) = \|Po(\Box\Phi)\|(s)$ .

**注 2** 对一个经典的 Kripke 结构 (KS)  $M$ , 任意的状态  $s \in M$  及任意的 CTL 的状态公式  $\Phi$ , 有  $s \in Sat(\exists(\Box\Phi)) \Leftrightarrow$  在  $M$  中, 存在一条从状态  $s$  可达的循环路径  $\pi = s_0 \cdots (s_i \cdots s_n)^w$ , 其中  $s_i = s_n, n \leq |S|$ , 且对任意的  $0 \leq k \leq n$ , 有  $s_k \vDash \Phi$ . 因此命题 3 可看作经典的 CTL 性质在 GPoCTL 上的广义体现.

**命题 4** 状态公式  $Po(\Phi_1 \sqcup \Phi_2), Po(\Box\Phi)$  的扩张律成立:

$$(1) \|Po(\Phi_1 \sqcup \Phi_2)\|(s) = \|\Phi_2 \wedge r_p\|(s) \vee (\|\Phi_1\|(s) \wedge \|Po(\Box(Po(\Phi_1 \sqcup \Phi_2)))\|(s));$$

$$(2) \|Po(\Box\Phi)\|(s) = \|\Phi \wedge r_p\|(s) \wedge \|Po(\Box(Po(\Box\Phi)))\|(s).$$

**证明** (1)  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s) = \bigvee_{\pi \in Paths(s)} Po(\pi)$

$$\begin{aligned} & \wedge (\bigvee_{j \geq 0} \bigwedge_{0 \leq k < j} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}) \wedge \|\Phi_2\|(s_j)) \\ & = \bigvee_{\pi \in Paths(s)} \bigvee_{j \geq 0} (\|\Phi_1\|(s_0) \wedge P(s_0, s_1) \wedge \cdots \wedge P(s_{j-1}, s_j) \wedge \|\Phi_2\|(s_j) \wedge Po(\pi)) \\ & = \bigvee_{\pi \in Paths(s)} \bigvee_{j \geq 0} (\|\Phi_1\|(s_0) \wedge P(s_0, s_1) \wedge \cdots \wedge P(s_{j-1}, s_j) \wedge \|\Phi_2\|(s_j) \wedge \bigvee_{s_j \in M} P(s_j, s_{j+1}) \wedge \cdots) \\ & = \bigvee_{\pi \in Paths(s)} \bigvee_{j \geq 0} (\bigwedge_{0 \leq k < j} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}) \wedge \|\Phi_2\|(s_j) \wedge r_p(s_j)) \\ & = \|\Phi_2 \wedge r_p\|(s) \vee \bigvee_{\pi' \in Paths(s)} \bigvee_{j \geq 0} (\bigwedge_{0 \leq k < j} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}) \wedge \|\Phi_2\|(s_j) \wedge r_p(s_j)) \\ & = \|\Phi_2 \wedge r_p\|(s) \vee \bigvee_{\pi' \in Paths(s)} \bigvee_{j \geq 1} (\|\Phi_1\|(s_0) \wedge P(s_0, s_1) \wedge \cdots \wedge \|\Phi_1\|(s_{j-1}) \wedge P(s_{j-1}, s_j) \wedge \|\Phi_2\|(s_j) \wedge r_p(s_j)) \\ & = \|\Phi_2 \wedge r_p\|(s) \vee (\|\Phi_1\|(s_0) \wedge \bigvee_{\pi' \in Paths(s)} (P(s_0, s_1) \wedge \bigwedge_{j \geq 1} (\|\Phi_1\|(s_1) \wedge P(s_1, s_2) \wedge \cdots \wedge \|\Phi_1\|(s_{j-1}) \wedge P(s_{j-1}, s_j) \wedge \|\Phi_2\|(s_j) \wedge r_p(s_j)))) \\ & = \|\Phi_2 \wedge r_p\|(s) \vee (\|\Phi_1\|(s_0) \wedge \bigvee_{\pi' \in Paths(s)} (P(s_0, s_1) \wedge \bigwedge_{j \geq 1} (\bigwedge_{1 \leq k < j} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}) \wedge \|\Phi_2\|(s_j) \wedge r_p(s_j)))) \\ & = \|\Phi_2 \wedge r_p\|(s) \vee (\|\Phi_1\|(s_0) \wedge \bigvee_{\pi' \in Paths(s)} (P(s_0, s_1) \wedge \bigwedge_{j \geq 1} (\bigwedge_{1 \leq k < j} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}) \wedge \|\Phi_2\|(s_j) \wedge r_p(s_j)))) \\ & = \|\Phi_2 \wedge r_p\|(s) \vee (\|\Phi_1\|(s_0) \wedge \bigvee_{\pi \in Paths(s)} (Po(\pi) \wedge P(s_0, s_1) \wedge \|Po(\Phi_1 \sqcup \Phi_2)\|(s_1))) \\ & = \|\Phi_2 \wedge r_p\|(s) \vee (\|\Phi_1\|(s_0) \wedge \bigvee_{\pi \in Paths(s)} (Po(\pi) \wedge \|\Box(Po(\Phi_1 \sqcup \Phi_2))\|(s_1))) \\ & = \|\Phi_2 \wedge r_p\|(s) \vee (\|\Phi_1\|(s) \wedge \|Po(\Box(Po(\Phi_1 \sqcup \Phi_2)))\|(s)); \end{aligned}$$

$$\begin{aligned} (2) \|Po(\Box\Phi)\|(s) & = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge \bigwedge_{j \geq 0} \bigwedge_{0 \leq k < j} P(s_k, s_{k+1}) \wedge \|\Phi\|(s_j) \\ & = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge \|\Phi\|(s) \wedge \bigwedge_{j \geq 1} \bigwedge_{1 \leq k < j} P(s_k, s_{k+1}) \wedge \|\Phi\|(s_j) \\ & = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge \|\Phi\|(s) \wedge \|\Box\Phi\|(\pi^1) = (\|\Phi\|(s) \wedge \bigvee_{\pi \in Paths(s)} Po(\pi)) \wedge (\bigvee_{\pi = s_0 s_1 \cdots \in Paths(s)} P(s_0, s_1) \wedge Po(\pi^1) \wedge \|\Box\Phi\|(\pi^1)) \\ & = \|\Phi \wedge r_p\|(s) \wedge (\bigvee_{\pi = s_0 s_1 \cdots \in Paths(s)} Po(\pi) \wedge P(s_0, s_1) \wedge Po(\pi^1) \wedge \|\Box\Phi\|(\pi^1)) \\ & = \|\Phi \wedge r_p\|(s) \wedge (\bigvee_{\pi = s_0 s_1 \cdots \in Paths(s)} Po(\pi) \wedge P(s_0, s_1) \wedge (\bigvee_{\pi = s_1 s_2 \cdots \in Paths(s)} Po(\pi^1) \wedge \|\Box\Phi\|(\pi^1))) \\ & = \|\Phi \wedge r_p\|(s) \wedge (\bigvee_{\pi = s_0 s_1 \cdots \in Paths(s)} Po(\pi) \wedge P(s_0, s_1) \wedge \|Po(\Box\Phi)\|(s_1)) \\ & = \|\Phi \wedge r_p\|(s) \wedge (\bigvee_{\pi = s_0 s_1 \cdots \in Paths(s)} Po(\pi)) \end{aligned}$$

$$\begin{aligned} & \wedge \|\circ(Po(\square\Phi))\|(s_1)) \\ = & \|\Phi \wedge r_p\|(s) \wedge \|Po(\circ(Po(\square\Phi)))\|(s_1). \end{aligned}$$

GPoCTL 模型检测问题:对给定的广义可能性 Kripke 结构  $M$ , 状态  $s \in M$  及 GPoCTL 状态公式  $\Phi$ , 计算  $\|\Phi\|(s)$  的值, 简记为  $M, s \models \Phi$ .  $\|\Phi\|(s)$  可根据  $\Phi$  的长度递归地计算, 其中  $|\Phi|$  ( $\Phi$  的子公式的个数) 递归定义如下:  $|\Phi| = 1$ , 如果  $\Phi \in AP \sqcup \{true\}$ ;  $|\Phi_1 \wedge \Phi_2| = |\Phi_1| + |\Phi_2| + 1$ ;  $|\neg \Phi| = |\Phi| + 1$ ;  $|Po(\circ\Phi)| = |Po(\square\Phi)| = |\Phi| + 1$ ;  $|Po(\Phi_1 \sqcup \Phi_2)| = |Po(\Phi_1 \sqcup^{\leq n} \Phi_2)| = |\Phi_1| + |\Phi_2| + 1$ .

设  $G = (V, E)$  是一个加权的定向图, 其中  $V$  表示顶点的集合,  $E$  表示有向边的集合. 设加权函数  $w: E \rightarrow [0, 1]$ , 则最短路径问题<sup>[14]</sup>可描述为:从某顶点出发沿边到达另一顶点所经的路径中各边权值之和最小的一条路径. 在文献[7]中经过适当的修改, 得到基于  $\wedge$  运算的最优路径问题: 设  $w(\pi) = \bigwedge_{0 \leq i \leq k-1} w(v_i, v_{i+1})$  表示路径  $\pi = v_0 v_1 \dots v_k$  的(加)权值, 从顶点  $u$  到顶点  $v$  的最优路径为顶点  $u$  到顶点  $v$  的路径中加权值最大的那条路径. 若加权函数  $\delta: V \times V \rightarrow [0, 1]$  定义为:  $\delta(u, v) = \begin{cases} \max\{w(\pi) \mid \pi \text{ 是从 } u \text{ 到 } v \text{ 的一条路径}\}, & \text{当存在从 } \\ & u \text{ 到 } v \text{ 的一条路径时,} \\ 0, & \text{其他} \end{cases}$

则从顶点  $u$  到顶点  $v$  的最优路径的(加权)值为  $\delta(u, v)$ . 基于  $\wedge$  运算的最优路径问题就是求从任意顶点  $u \in V \setminus \{t\}$  到给定的目标顶点  $t$  的最优路径(加权)值. 其算法主要步骤如下: 设  $Q$  表示已经求得的到给定目标顶点  $t$  的最优路径的顶点,  $d(v)$  表示从顶点  $v$  到顶点  $t$  的最优路径的加权值. (1) 初始地, 设  $Q = \{t\}$ , 令  $d(v) = \begin{cases} w(v, t), & \text{当 } (v, t) \in E \text{ 时} \\ 0, & \text{其他} \end{cases}$ ; (2) 从  $V \setminus Q$  中选取使得  $d(v)$  最大的  $v$ , 令  $Q := Q \sqcup \{v\}$ , 对任意的  $u \in V \setminus Q$ , 如果  $(u, v) \in E$ , 且  $d(u) \leq d(v) \wedge w(u, v)$ , 令  $d(u) = d(v) \wedge w(u, v)$ ; (3) 重复(2)和(3)直到所有顶点添加到  $Q$ , 即  $Q = V \sqcup \{t\}$ .

文献[7]证明了算法的可行性, 因此当算法结束时对任意  $v \in V \setminus Q$ , 有  $d(v) = \delta(u, v)$ , 即证明了  $d(v)$  是顶点  $v$  到目标顶点  $v$  的最优路径(加权)值. 命题5 利用该算法给出了计算  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s)$  的时间复杂度.

**命题5** 设  $M$  是一个 GPKS,  $s$  是  $M$  中的任意状态,  $\Phi_1, \Phi_2$  是任意的 GPoCTL 状态公式, 则计算  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s)$  的时间复杂度是  $O(|Supp(P)| + |S| \cdot \log |S|)$ .

**证明** 给定一个 GPKS  $M = (S, P, I, AP, L)$  及状态公式  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s)$ , 构造加权定向图  $G = (V, E)$ , 其中  $V = S \sqcup \{t\}$ ,  $t \notin S$  且  $t$  是目标顶点,  $E = \{(s, s') \mid P(s, s') > 0\} \sqcup \{(s, t) \mid s \in S\}$ , 函数  $w: E \rightarrow [0, 1]$  定义

$$\text{为: } w(s, s') = \begin{cases} \|\Phi_1\|(s) \wedge P(s, s'), & \text{当 } s, s' \in S \text{ 时} \\ \|\Phi_2\|(s) \wedge r_p(s), & \text{其他} \end{cases},$$

由命题3, 对任意的  $s \in M$  可得

$$\begin{aligned} & \|Po(\Phi_1 \sqcup \Phi_2)\|(s) = \|Po(\Phi_1 \sqcup^{\leq |S|} \Phi_2)\|(s) \\ = & \bigvee_{\pi \in Paths(s)} \bigvee_{0 \leq j < |S|} (\bigwedge_{0 \leq k < j} \|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1})) \\ & \wedge \|\Phi_2\|(s_j) \wedge r_p(s_j) \\ = & \bigvee_{\pi \in Paths(s)} \bigvee_{0 \leq j < |S|} (\bigwedge_{0 \leq k < j} w(s_k, s_{k+1}) \wedge w(s_j, t)) = \delta(s, t), \end{aligned}$$

即  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s)$  等于图  $G$  中顶点  $s$  到目标顶点  $t$  的最优路径(加权)值. 由于计算  $\delta(s, t)$  的时间复杂度为  $O(|Supp(P)| + |S| \cdot \log |S|)$  (类似 Dijkstra 算法<sup>[7, 14]</sup>), 进而计算  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s)$  的时间复杂度是  $O(|Supp(P)| + |S| \cdot \log |S|)$ .

下面主要通过模糊集的分解定理<sup>[15]</sup>讨论 GPoCTL 的模型检测问题, 首先给出与 GPKS 对应的类似于截集的 Kripke 结构  $M_\alpha$  定义.

**定义6** 设  $M = (S, P, I, AP, L)$  是一个 GPKS, 对任意的  $\alpha \in [0, 1]$ , 构造一个 Kripke 结构  $M_\alpha$  为:

$$M_\alpha = (S_\alpha, P_\alpha, I, AP, L_\alpha),$$

$$\text{其中 } L_\alpha(s) = \begin{cases} \{p \mid L(s, p) = \|p\|(s) \geq \alpha\}, & s \in S \\ \emptyset, & s \notin S \end{cases},$$

$$S_\alpha = S \sqcup \{t \mid t \notin S\}, P_\alpha = \{(s_1, s_2) \mid P(s_1, s_2) \geq \alpha\} \sqcup \{(s, t) \mid s \in S\} \sqcup \{(t, t)\}. \text{对任意的状态 } s \in M \text{ 及 CTL 公式 } \Phi, \text{记 } s \models_\alpha \Phi \Leftrightarrow \text{在 Kripke 结构 } M_\alpha \text{ 下, } s \models \Phi; \text{记 } s \in Sat_\alpha(\Phi) \Leftrightarrow \text{在 Kripke 结构 } M_\alpha \text{ 下, } s \in Sat(\Phi).$$

引理1 将要 GPoCTL 中的可能性状态公式  $\Phi$  与 CTL 中的经典的状态公式  $\Phi'$  对应起来, 进而可利用引理1 和分解定理将 GPoCTL 的模型检测问题(计算  $\|\Phi\|(s)$ ) 转化为经典的 CTL 问题(判定  $s \models_\alpha \Phi'$ ).

**引理1** 设  $\Phi$  是任意的 GPoCTL 状态公式, 且  $\Phi$  不包含  $\neg$  连接词,  $\Phi'$  是与  $\Phi$  对应的 CTL 状态公式, 其中若  $\Phi$  中含有  $Po$  时, 则在  $\Phi'$  中对应为  $\exists$ . 则对任意的  $\alpha \in [0, 1]$  以及任意的 GPKSM,  $M$  中的任意状态  $s$ ,  $\|\Phi\|(s) \geq \alpha \Leftrightarrow s \models_\alpha \Phi'$ .

**证明** 利用归纳法证明: 设  $\Phi, \Phi_1, \Phi_2$  是 GPoCTL 状态公式,  $\alpha \in [0, 1]$ ,  $M = (S, P, I, AP, L)$  是一个 GPKS,  $M_\alpha = (S_\alpha, P_\alpha, I, AP, L_\alpha)$  是一个 Kripke 结构,  $s$  是  $M$  中的状态. 当  $|\Phi| = 1$  时, 结论显然成立. 假设当  $|\Phi| \leq n$  时, 有  $\|\Phi\|(s) \geq \alpha \Leftrightarrow s \models_\alpha \Phi'$ .

当  $|\Phi| = n + 1$  时, 若  $\Phi = \Phi_1 \wedge \Phi_2$ , 此时  $\Phi' = \Phi'_1 \wedge \Phi'_2$ ,  $\|\Phi\|(s) \geq \alpha \Leftrightarrow \|\Phi_1\|(s) \geq \alpha$  和  $\|\Phi_2\|(s) \geq \alpha \Leftrightarrow$  在 Kripke 结构  $M_\alpha$  下,  $s \models_\alpha \Phi'_1$  和  $s \models_\alpha \Phi'_2 \Leftrightarrow s \models_\alpha \Phi'_1 \wedge \Phi'_2 \Leftrightarrow s \models_\alpha \Phi'$ ; 若  $\Phi = Po(\Phi_1)$ , 此时  $\Phi' = \exists(\circ\Phi'_1)$ ,  $\|\Phi\|(s) = \|Po(\Phi_1)\|(s) = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge P(s, s_1) \wedge \|\Phi_1\|(s) \geq \alpha \Leftrightarrow$  存在一条路径  $\pi = ss_1 \dots \in Paths(s)$  和状态  $s_1$ , 使得  $Po(\pi) \geq \alpha, P(s, s_1) \geq \alpha, \|\Phi_1\|(s)$

$\geq \alpha \Leftrightarrow$  在 Kripke 结构  $M_\alpha$  下, 存在一条路径  $\pi = ss_1 \cdots$  和状态  $s_1$ , 使得  $(s, s_1) \in P_\alpha, s_1 \Vdash_\alpha \Phi'_1 \Leftrightarrow$  在 Kripke 结构  $M_\alpha$  下,  $s \Vdash_\alpha \exists (\bigcirc \Phi'_1) \Leftrightarrow s \Vdash_\alpha \Phi'$ ; 若  $\Phi = Po(\Phi_1 \sqcup \Phi_2)$ , 此时  $\Phi' = \exists (\Phi'_1 \sqcup \Phi'_2)$ ,  $\|\Phi\|(s) = \|Po(\Phi_1 \sqcup \Phi_2)\|(s) = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\bigvee_{j \geq 0} (\bigwedge_{0 \leq k < j} (\|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1}))) \wedge \|\Phi_2\|(s_j)) \geq \alpha \Leftrightarrow$  存在整数  $j$  和一条路径  $\pi = ss_1 \cdots s_j \cdots$ , 使得  $Po(\pi) \geq \alpha, \bigwedge_{0 \leq k < j} (\|\Phi_1\|(s_k) \wedge P(s_k, s_{k+1})) \wedge \|\Phi_2\|(s_j) \geq \alpha \Leftrightarrow$  存在整数  $j$  和一条路径  $\pi = ss_1 \cdots s_j \cdots$ , 使得  $Po(\pi) \geq \alpha, \|\Phi_2\|(s_j) \geq \alpha$ , 且对任意的  $0 \leq k < j, P(s_k, s_{k+1}) \geq \alpha, \|\Phi_1\|(s_k) \geq \alpha \Leftrightarrow$  在 Kripke 结构  $M_\alpha$  下, 存在整数  $j$  和一条路径  $\pi = ss_1 \cdots s_j \cdots$ , 使得  $s_j \Vdash_\alpha \Phi'_2$ , 且对任意的  $0 \leq k < j, (s_k, s_{k+1}) \in P_\alpha, s_k \Vdash_\alpha \Phi'_1 \Leftrightarrow$  在 Kripke 结构  $M_\alpha$  下,  $s \Vdash_\alpha \exists (\Phi'_1 \sqcup \Phi'_2) \Leftrightarrow s \Vdash_\alpha \exists (\Phi')$ ; 若  $\Phi = Po(\Phi_1 \sqcup^{\leq n} \Phi_2)$ , 此时  $\Phi' = \exists (\Phi'_1 \sqcup^{\leq n} \Phi'_2)$ , 证法同  $\Phi = Po(\Phi_1 \sqcup \Phi_2)$ ; 若  $\Phi = Po(\Box \Phi_1)$ , 此时,  $\Phi' = \exists (\Box \Phi'_1) \|\Phi\|(s) = \|Po(\Box \Phi_1)\|(s) = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge \|\Box \Phi_1\|(s) = \bigvee_{\pi \in Paths(s)} Po(\pi) \wedge (\bigwedge_{j \geq 0} \bigwedge_{0 \leq k < j} P(s_k, s_{k+1})) \wedge (\bigwedge_{k \geq 0} (\|\Phi_1\|(s_k) \geq \alpha) \Leftrightarrow$  存在一条路径  $\pi \in Paths(s)$ , 使得对任意的  $k \geq 0, P(s_k, s_{k+1}) \geq \alpha, \|\Phi_1\|(s_k) \geq \alpha \Leftrightarrow$  在 Kripke 结构  $M_\alpha$  下, 存在一条路径  $\pi \in Paths(s)$ , 使得对任意的  $k \geq 0, (s_k, s_{k+1}) \in P_\alpha, s_k \Vdash_\alpha \Phi'_1 \Leftrightarrow$  在 Kripke 结构  $M_\alpha$  下,  $s \Vdash_\alpha \exists (\Box \Phi'_1) \Leftrightarrow s \Vdash_\alpha \Phi'$

综上, 当  $|\Phi| = n + 1$  时,  $\|\Phi\|(s) \geq \alpha \Leftrightarrow s \Vdash_\alpha \Phi'$ . 由归纳假设, 结论成立.

**定理 1** 设  $\Phi$  是不包含  $\neg$  连接词的任意 GPoCTL 状态公式,  $M$  是一个 GPKS, 则对任意的状态  $s \in M$ , 有  $\|\Phi\|(s) = Sup \{ \alpha \mid s \Vdash_\alpha \Phi', \alpha \in Im(\Phi) \}$ , 其中  $\Vdash_\alpha, \Phi'$  如引理 1 中.

**证明** 设  $\Phi_\alpha = \{ s \mid \|\Phi\|(s) \geq \alpha, s \in S \}$ ,  $\alpha \cdot \Phi_\alpha(s) = \begin{cases} \alpha, & \text{当 } \alpha \in \Phi_\alpha \text{ 时} \\ 0, & \text{其他} \end{cases}$ , 由分解定理  $\|\Phi\|(s) = Sup \{ \alpha \cdot \Phi_\alpha(s) \mid \alpha \in (0, 1] \} = Sup \{ \alpha \mid s \in \Phi_\alpha, \alpha \in (0, 1] \} = Sup \{ \alpha \mid \|\Phi\|(s) \geq \alpha, \alpha \in (0, 1] \} = Sup \{ \alpha \mid s \Vdash_\alpha \Phi', \alpha \in Im(\Phi) \}$ . 由于  $M$  是有限的, 最后一个等式由命题 2 可得.

**注 3** 定理 1 说明了 GPoCTL 模型检测问题(计算  $\|\Phi\|(s)$ ) 可规约为对任意的  $\alpha \in Im(\Phi)$ , 是否有  $\|\Phi\|(s) \geq \alpha$  的问题, 而  $\|\Phi\|(s) \geq \alpha$  问题又可规约为经典的 CTL 模型检测  $s \Vdash \Phi'$  问题.

定理 2 将给出利用分解定理把 GPoCTL 模型检测问题转化为经典的 CTL 模型检测问题的算法及相应的复杂度.

**定理 2** 设  $\Phi$  是任意的 GPoCTL 状态公式,  $M$  是一个 GPKS, 则对  $M$  中的任意状态  $s$ , 计算  $\|\Phi\|(s)$  的时间复杂度是  $O(|M| \cdot |Im(\Phi)| \cdot |\Phi|)$ .

**证明** 首先, 若  $\Phi = a, true, \Phi_1 \wedge \Phi_2, Po(\bigcirc \Phi_1), \Phi_1 \sqcup \Phi_2, \Phi_1 \sqcup^{\leq n} \Phi_2, \Box \Phi_1$ , 其中  $a \in AP, True, \Phi_1, \Phi_2$  为 GPoCTL 状态公式, 对任意的状态  $s \in M$ , 由定理 1 知, 要求  $\|\Phi\|(s)$  的值, 只需要对任意的  $\alpha \in Im(\Phi)$ , 到使  $s \in \Phi_\alpha$  的所有  $\Phi_\alpha$ , 然后求出这些  $\alpha$  中的最大值, 这个最大值就等于  $\|\Phi\|(s)$ ; 由引理 1, 要求  $\Phi_\alpha$ , 可通过构建的经典的 Kripke 结构  $M_\alpha$ , 即对任意状态  $s \in M, s \in \Phi_\alpha \Leftrightarrow \|\Phi\|(s) \geq \alpha \Leftrightarrow s \Vdash_\alpha \Phi' \Leftrightarrow s \in Sat_\alpha(\Phi')$ , 其中  $\Vdash_\alpha, \Phi'$  如引理 1 中, 因此要求  $s \in \Phi_\alpha$ , 只要在经典的 Kripke 结构中求得  $s \in Sat_\alpha(\Phi')$ .

具体地, 可按下面的算法步骤求得  $\|\Phi\|(s)$ : 设  $M$  是一个 GPKS,  $s$  是  $M$  中任意的状态,  $\Phi$  是任意的 GPoCTL 状态公式, (1) 对给定的 GPoCTL 状态公式  $\Phi$ , 求出其对应的 CTL 状态公式  $\Phi'$ ; (2) 对任意的  $\alpha \in Im(\Phi)$ , 求出相应的 Kripke 结构  $M_\alpha$ , 在  $M_\alpha$  下求出  $Sat_\alpha(\Phi')$ , 令  $\Phi_\alpha = Sat_\alpha(\Phi')$ ; (3) 对任意的状态  $s \in M$ , 找到包含状态  $s$  的  $\Phi_\alpha$ , 令  $\|\Phi\|(s) = Sup \{ \alpha \mid s \in \Phi_\alpha \}$ . 由引理 1 及定理 1 可知, 第(3)步求得结果是正确的.

由经典 CTL 模型检测知, 在第(2)步中, 对任意的  $\alpha \in Im(\Phi)$ , 计算  $Sat_\alpha(\Phi')$  需花费时间  $O(|M| \cdot |\Phi'|) = O(|M| \cdot |\Phi|)$ , 故而计算  $\|\Phi\|(s)$  的时间复杂度为  $O(|M| \cdot |Im(\Phi)| \cdot |\Phi|)$ .

其次, 当  $\Phi = \neg \Phi_1$  时, 可先通过计算  $\|\Phi_1\|(s)$ , 而  $\|\Phi\|(s) = 1 - \|\Phi_1\|(s)$ .

综上所述, 对任意的 GPoCTL 状态公式  $\Phi$ , 任意的状态  $s \in M$ , 计算  $\|\Phi\|(s)$  的时间复杂度是  $O(|M| \cdot |Im(\Phi)| \cdot |\Phi|)$ .

通过下面的例子<sup>[6,11]</sup> 解释如何将 GPoCTL 模型检测问题规约为 CTL 模型检测问题, 并利用定理 2 中的算法步骤求  $\|\Phi\|(s)$ .

**例 1** 某个动物感染了一种新的疾病, 医生对这种新的疾病并不十分了解, 但是根据经验觉得用某种药物(青霉素)会对这种疾病有较好的疗效. 下面利用广义可能性 Kripke 结构对该疾病的治疗进程进行建模.

医生把动物的身体状况分为三种情况: 身体差, 身体正常, 身体好, 这三种身体状况可分别用  $s_0, s_1, s_2$  表示, 即得  $S = \{s_0, s_1, s_2\}$ . 设原子命题  $AP = \{p, f, g\}$ , 其中  $p, f, g$  分别表示动物在某种状态的身体是‘差的’, ‘正常的’, ‘好的’. 由于  $p, f, g$  本身是不确定的概念, 医生也不能完全确定动物的身体状况, 因此根据医生的经验, 可得到它们在动物的每个身体状况发生的可能性程度, 利用标签函数表示如下:

$$L(s_0)(p) = 1, L(s_0)(f) = 0.5, L(s_0)(g) = 0.3,$$

$L(s_1)(p) = 0.5, L(s_1)(f) = 1, L(s_1)(g) = 0.5, L(s_2)(p) = 0.4, L(s_2)(f) = 0.6, L(s_2)(g) = 1$ , 通过药物治疗后,动物的身体状况会一定程度地从一个状态转向另一个状态,根据医生的经验,这些转移可用可能性分布函数  $P$  表示: $P(s_0, s_0) = 0.3, P(s_0, s_1) = 1, P(s_0, s_2) = 0.6, P(s_1, s_0) = 0.3, P(s_1, s_1) = 0.6, P(s_1, s_2) = 1, P(s_2, s_0) = 0.4, P(s_2, s_1) = 0.5, P(s_2, s_2) = 0.5$ , 初始地,假设动物处在差( $s_0$ )的身体状况.

因此,动物的疾病治疗进程可用广义可能性 Kripke 结构  $M = (S, P, s_0, L, AP)$  表示,其对应的转移图如图 1.

首先对  $s \in M, \Phi = Po(p \sqcup g)$ , 给出  $\|Po(p \sqcup g)\|(s)$  (在状态  $s$  经过药物治疗后,‘身体是好的’以前一直是‘身体是弱的’的可能性)的计算:(1) GPoCTL 状态公式  $\Phi = Po(p \sqcup g)$  在 CTL 中对应的公式为  $\Phi' = \exists (p \sqcup g)$ . (2) 对任意的  $\alpha \in \text{Im}(\Phi)$ , 当  $\alpha = 0.3$  时,其对应的 Kripke 结构  $M_{0.3}$  可用如图 2 表示,易得  $\Phi_{0.3} = \{s_0, s_1, s_2\}$ ; 当  $\alpha = 0.4$  时,其对应的 Kripke 结构  $M_{0.4}$  可用如图 3 表示,易得  $\Phi_{0.4} = \{s_0, s_1, s_2\}$ ; 当  $\alpha = 0.5$  时,其对应的 Kripke 结构  $M_{0.5}$  可用如图 4 表示,易得  $\Phi_{0.5} = \{s_0, s_1, s_2\}$ ; 当  $\alpha = 0.6$  时,其对应的 Kripke 结构  $M_{0.6}$  可用如图 5 表示,易得  $\Phi_{0.6} = \{s_0, s_2\}$ ; 当  $\alpha = 1$  时,其对应的 Kripke 结构  $M_1$  可用如图 6 表示,易得  $\Phi_1 = \{s_2\}$ . (3) 由  $\| \Phi \| (s) = \text{Sup} \{ \alpha | s \in \Phi_\alpha \}$  得,  $\|Po(p \sqcup g)\|(s_0) = 0.6, \|Po(p \sqcup g)\|(s_1) = 0.5, \|Po(p \sqcup g)\|(s_2) = 1$ .

类似地,可以得到下面的结论:

$\|Po(\diamond f)\|(s_0) = 1$ , 表示在  $s_0$  状态,经过药物治疗后,身体最终是‘正常的’可能性是 1.

$\|Po(\diamond g)\|(s_0) = 1$ , 表示在  $s_0$  状态,经过药物治疗后,身体最终是‘好的’可能性是 1.

虽然  $\|Po(\diamond p)\|(s_0) = 1$ , 但其取值 1 是在第 0 步实现的(无药物治疗),因此表示在  $s_0$  状态,经过药物治疗后(至少经过一步),身体最终仍是‘差的’可能性是 0.5.

$\|Po(\square p)\|(s_0) = 0.5$ , 表示在  $s_0$  状态,经过药物治疗后,身体一直是‘差的’可能性是 0.5.

$\|Po(\square f)\|(s_1) = 0.6$ , 表示在  $s_1$  状态,经过药物治疗后,身体一直是‘正常的’可能性是 0.6.

$\|Po(\square g)\|(s_2) = 0.5$ , 表示在  $s_2$  状态,经过药物治疗后,身体一直是‘好的’可能性是 0.5.

**注 4** (1) 图 2 ~ 图 6 不一定是 Kripke 结构  $M_\alpha$  对应的转移图,但可以在图中增加顶点  $t$ , 使得在顶点  $t$  对任意原子命题赋值为 0, 并对图中的任意顶点  $s$  增加到顶点  $t$  的边  $(s, t)$  及边  $(t, t)$ , 这样得到的图就可以表示  $M_\alpha$  对应的转移图,但为了便于理解,仍用图 2 ~ 图 6 表示;(2) 三元组  $x, y, z$  表示在状态  $s$  命题  $p, f, g$  分别发生的可能性,例如,在  $s_1$  处的三元组 0.5, 1, 0.5 表示在状态  $s_1$  命题  $p, f, g$  发生的可能性分别是 0.5, 1, 0.5;(3) 由上面的运算结果也可得到:药物青霉素的疗效是好的,构建的模型比较合适的.

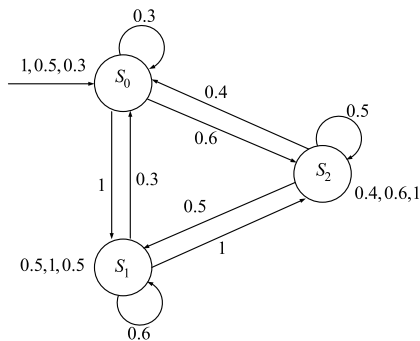


图1 GPKS结构M对应的转移图

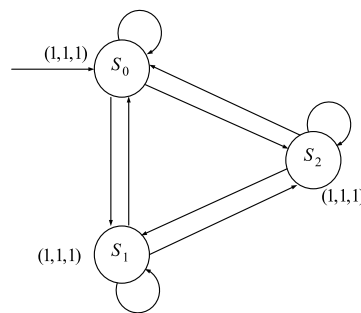


图2 Kripke结构M<sub>0.3</sub>对应的转移图

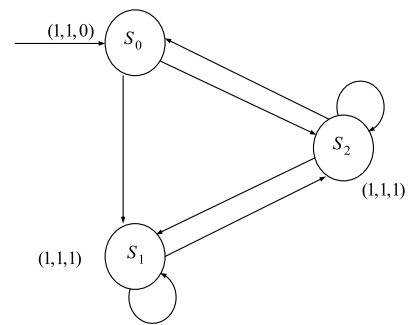


图3 Kripke结构M<sub>0.4</sub>对应的转移图

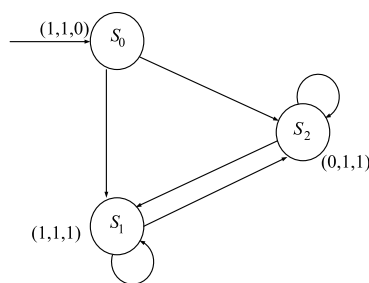


图4 Kripke结构M<sub>0.5</sub>对应的转移图

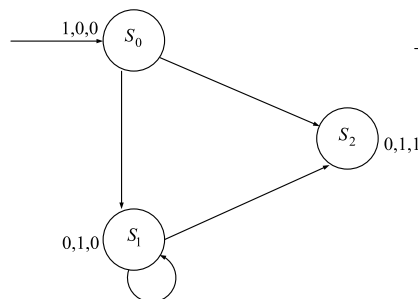


图5 Kripke结构M<sub>0.6</sub>对应的转移图

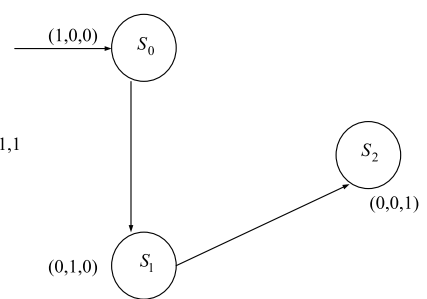


图6 Kripke结构M<sub>1</sub>对应的转移图

## 4 GPoCTL<sub>f</sub>模型检测问题

具有公平性假设的计算树逻辑 (Computation Tree Logic with fairness, CTL<sub>f</sub><sup>[2]</sup>) 中的任一条路径都满足 CTL 公平性假设 *fair*, 其语法和 CTL 相同, 其语义与 CTL 不同的是要求状态所在的路径满足公平性假设, 满足关系用  $\models_{fair}$  表示. 用  $FairPaths(s)$  表示从状态  $s$  出发的所有公平性路径组成的集合,  $FairPaths(s) = \{\pi \in Paths(s) \mid \pi \models_{fair}\}$ ; 用  $FairPaths(M)$  表示  $M$  中从初始状态  $s_0 \in I$  出发的所有公平性路径组成的集合,  $FairPaths(M) = \bigcup_{s_0 \in I} FairPaths(s_0)$ . 例如:  $s \models_{fair} \exists \varphi$  当且仅当存在  $\pi \in FairPaths(s)$ ,  $\pi \models_{fair} \varphi$ ;  $s \models_{fair} \forall \varphi$  当且仅当对任意的  $\pi \in FairPaths(s)$ ,  $\pi \models_{fair} \varphi$ .

类似地, 若要求 GPoCTL 中的任一条路径都满足 CTL 公平性假设 *fair*, 则称其为具有公平性假设的广义可能性计算树逻辑 (GPoCTL with fairness, GPoCTL<sub>f</sub>), 其语法和 GPoCTL 相同, 其语义与 GPoCTL 不同的也是要求状态所在的路径满足公平性假设. 例如:  $\|Po(\varphi)\|(s) = Po(s \models_{fair} \varphi) = \bigvee_{\pi \in FairPaths(s)} Po(\pi) \wedge \|\varphi\|(\pi)$ .

**定理 3** 设  $\Phi$  是不包含  $\neg$  连接词的任意 GPoCTL<sub>f</sub> 状态公式,  $M$  是一个 GPKS, 则对任意的状态  $s \in M$ , 有  $\|\Phi\|(s) = Sup\{\alpha \mid s \models_{\alpha} \Phi', \alpha \in Im(\Phi)\}$ , 其中  $\models_{\alpha}$ ,  $\Phi'$  如引理 1 中, 且  $\models_{\alpha}$  也是公平性满足关系.

**定理 4** 设  $\Phi$  是任意的 GPoCTL<sub>f</sub> 状态公式, 且 CTL 公平性假设 *fair* 中连接词的个数是  $k$ ,  $M$  是一个 GPKS, 则对任意状态  $s \in M$ , 计算  $\|\Phi\|(s)$  的时间复杂度是  $O(|M| \cdot |Im(\Phi)| \cdot |\Phi| \cdot k)$ .

**证明** 文献[2]中, 当 *fair* 中连接词的个数是  $k$ , 计算具有公平性假设 CTL 的模型检测问题的时间复杂度是  $O(|M| \cdot |\Phi| \cdot k)$ . 结合定理 2 和定理 3 可知, 具有公平性假设的 GPoCTL 的模型检测问题的时间复杂度是  $O(|M| \cdot |Im(\Phi)| \cdot |\Phi| \cdot k)$ .

## 5 小结

本文证明了 GPoCTL 状态公式  $Po(\Phi_1 \sqcup \Phi_2)$ ,  $Po(\Box\Phi)$  分别等价于  $Po(\Phi_1 \sqcup^{\leq |S|} \Phi_2)$  和  $Po(\Box^{\leq |S|} \Phi)$ , 并给出了它们对应的扩张律; 把 GPoCTL 状态公式  $\Phi$  的模型检测问题规约到经典的 CTL 模型检测问题, 并通过给定的算法过程, 证明了 GPoCTL 状态公式  $\Phi$  的模型检测问题的时间复杂度是  $O(|M| \cdot |Im(\Phi)| \cdot |\Phi|)$ ; 特别地, 通过改进了的算法, 把计算  $\|Po(\Phi_1 \sqcup \Phi_2)\|(s)$  的时间复杂度提高到了  $O(|Supp(P)| + |S| \cdot \log |S|)$ . 而文献[11]给出的 GPoCTL 状态公式  $\Phi$  模型检测问题的时间复杂性仅是  $O(|Supp(P)| + |S|) \cdot |S|^3 \cdot |\Phi|$ , 与其相比, 在本文中,  $\|\Phi\|(s)$  的计算时间复杂度得到了降低. 进一步地, 通过相同的方法, 把

GPoCTL<sub>f</sub> 状态公式的模型检测问题约简到 CTL<sub>f</sub> 模型检测问题, 证明了其计算时间复杂度是  $O(|M| \cdot |Im(\Phi)| \cdot |\Phi| \cdot k)$ . 最后通过实例分析了如何利用规约方法计算  $\|\Phi\|(s)$ , 计算的数据结果也表明了 GPKS 模型的合理性.

文献[12]给出了 GPoLTL 公式的可达性分析, 自动化验证等性质, 而 GPoLTL 公式模型检测问题的规约及复杂性还未讨论, 可作为下一步研究的内容.

## 参考文献

- [1] 林惠民, 张文辉. 模型检测: 理论, 方法与应用 [J]. 电子学报, 2002, 30(12): 1907 - 1912.  
Lin H M, Zhang W H. Model checking: theories, Techniques and Applications [J]. Acta Electronica Sinica, 2002, 30(12): 1907 - 1912. (in Chinese)
- [2] C Baier, J P Katoen. Principles of Model Checking [M]. Cambridge, Massachusetts: MIT Press, 2008.
- [3] 林运国, 雷红轩, 李永明. 量子马尔可夫链安全性模型检测 [J]. 电子学报, 2014, 42(11): 2191 - 2197.  
Lin Y G, Lei H X, Li Y M. Model checking of safety property over quantum markov chain [J]. Acta Electronica Sinica, 2014, 42(11): 2191 - 2197. (in Chinese)
- [4] M Chechik, B Devereux, A Gurfinkel, et al. Multi-valued symbolic model-checking [J]. ACM Transactions on Software Engineering Methodology, 2004, 12(4): 1 - 38.
- [5] M Chechik, A Gurfinkel, B Devereux, et al. Data structures for symbolic multi-valued model-checking [J]. Formal Methods in System Design, 2006, 29(3): 295 - 344.
- [6] Pan H Y, Li Y M, Cao Y Z, Ma Z Y. Model checking computation tree logic over finite lattices [J]. Theoretical Computer Science, 2016, 612: 45 - 62.
- [7] Pan H Y, Li Y M, Cao Y Z, Ma Z Y. Model checking fuzzy computation tree logic [J]. Fuzzy Sets and Systems, 2015, 2612: 60 - 77.
- [8] D Dubois. Possibility theory and statistical reasoning [J]. Computational Statistics and Data Analysis, 2006, 51: 47 - 69.
- [9] Li Y M, Li L. Model checking of linear-time properties based on possibility measure [J]. IEEE Transactions on Fuzzy Systems, 2013, 21(5): 842 - 854.
- [10] Li Y M, Li Y N. Computation tree logic model checking based on possibility measures [J]. Fuzzy Sets and Systems, 2015, 26(2): 44 - 59.
- [11] Li Y M, Ma Z Y. Quantitative computation tree logic model checking based on generalized possibility measures [J]. IEEE Transactions on Fuzzy Systems, 2015, 23(6): 2034 - 2047.
- [12] Li Y M. Quantitative model checking of linear-time prop-

erties based on generalized possibility measures [DB/OL]. <http://arxiv.org/abs/1601.06504>.

- [13] 马占有, 李永明. 基于决策过程的广义可能性计算树逻辑模型检测[J]. 中国科学: 信息科学, 2016, 46(11): 1591 – 1607.

Ma Z Y, Li Y M. Model checking generalized possibilistic computation tree logic based on decision processes [J]. Science China Information Sciences, 2016, 46(11): 1591 – 1607. (in Chinese)

- [14] J Kleinberg, E Tardos. Algorithm Design [M]. Addison-Wesley, MA, 2005.

- [15] 李永明. 模糊计算理论 [M]. 北京: 科学出版社, 2016.  
Li Y M. Theories of Computations [M]. Beijing: Science Press, 2016.

#### 作者简介



**梁常建** 男, 1981 年出生于河南鹿邑, 博士研究生, 讲师, 主要研究领域为定量模型检测.

E-mail: liangcj37@snnu.edu.cn



**李永明** 男, 1966 年出生于陕西大荔, 博士, 教授, 博士生导师, 研究方向为计算智能、量子逻辑、量子计算、模型检测.

E-mail: liyongm@snnu.edu.cn