

不确定型模糊 Kripke 结构的计算 树逻辑模型检测

范艳焕^{1,2}, 李永明¹, 潘海玉^{1,3}

(1. 陕西师范大学计算机科学学院, 陕西西安 710062; 2. 青海师范大学民师院数学系,
青海西宁 810008; 3. 泰州学院计算机科学与技术学院, 江苏泰州 225300)

摘要: 本文研究了不确定型模糊 Kripke 结构的计算树逻辑的模型检测问题, 并说明了该问题可以在对数多形式时间内解决. 首先给出了不确定型模糊 Kripke 结构的定义, 引入了模糊计算树逻辑的语法和语义. 为了刻画存在量词 \exists 和任意量词 \forall 在不确定型模糊 Kripke 结构中的两种语义解释, 在模糊计算树逻辑语法中引入了路径量词 \exists_{sup} , \exists_{inf} 和 \forall_{sup} , \forall_{inf} , 分别用于替换存在量词 \exists 和任意量词 \forall . 其次讨论了基于不确定型模糊 Kripke 结构的计算树逻辑模型检测算法, 特别地对于模糊计算树逻辑公式 $\exists_{\text{sup}}pUq$, $\forall_{\text{sup}}pUq$, $\exists_{\text{inf}}pUq$ 和 $\forall_{\text{inf}}pUq$ 分别给出时间复杂度为对数多项式时间的改进算法.

关键词: 模型检测; 计算树逻辑; 模糊逻辑; Kripke 结构; 时态逻辑

中图分类号: TP301.2 **文献标识码:** A **文章编号:** 0372-2112 (2018)01-0152-08

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2018.01.021

Computation Tree Logic Model Checking for Nondeterministic Fuzzy Kripke Structure

FAN Yan-huan^{1,2}, LI Yong-ming¹, PAN Hai-yu^{1,3}

(1. College of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi 710062, China;

2. Department of Mathematics in National Normal College, Qinghai Normal University, Xining, Qinghai 810008, China;

3. College of Computer Science and Technology, Taizhou University, Taizhou, Jiangsu 225300, China)

Abstract: This paper studies model-checking problem for FCTL (fuzzy computation tree logic) over nondeterministic fuzzy system and shows that it can be solved in linear-logarithmic running time in the size of the system. Firstly, we introduce NFKS (nondeterministic fuzzy kripke structure) which is adapted to model nondeterministic fuzzy system. The syntax and semantics of fuzzy computation tree logic over NFKS are presented. For describing two kinds of semantic explanation, we use path quantifiers \exists_{sup} , \exists_{inf} and \forall_{sup} , \forall_{inf} as substitutes for an existential path quantifier \exists and a universal path quantifier \forall in the syntax of the CTL. Then, we study the model checking algorithm for fuzzy computation tree logic over NFKS. Furthermore, the improvement algorithms for FCTL formulæ $\exists_{\text{sup}}pUq$, $\forall_{\text{sup}}pUq$, $\exists_{\text{inf}}pUq$ and $\forall_{\text{inf}}pUq$ are given, whose time complexities are linear-logarithmic running time in the size of the system.

Key words: model checking; computation tree logic; fuzzy logic; Kripke structure; temporal logic

1 引言

模型检测^[1,2] (model checking) 的基本思想是: 给定系统模型和刻画系统性质的时序逻辑^[3,4], 验证系统模型是否满足该公式. 如果满足, 则输出 True; 否则输出

False 并给出反例. 其中系统模型可以由 Kripke 结构或迁移系统刻画. 模型检测已广泛应用于计算机软硬件系统、集成电路、通信协议等方面的分析验证中.

经典的模型检测强调的是系统行为的绝对正确, 回答的是系统是否满足所需的性质. 由于定量系统的

收稿日期: 2017-05-19; 修回日期: 2017-08-14; 责任编辑: 孙瑶

基金项目: 国家自然科学基金 (No. 11671244, No. 61261047, No. 61672023, No. 61673352); 教育部博士点基金 (No. 20130202110001); 青海省自然科学基金 (No. 2014-ZJ-908); 江苏高校“青蓝工程”

存在,定量模型检测是利用模型检测的方法对定量系统进行自动的形式验证,定量分析其满足用户需求的程度.最近十几年里,定量模型检测受到了很多学者的关注,比如概率模型检测^[1],多值模型检测^[5-7],广义可能性模型检测^[8-11],和模糊模型检测^[12,13].其中概率模型检测针对的是随机系统,基本的随机系统模型是马尔可夫链(markov chain).马尔可夫决策过程(markov decision process)是带有不确定性选择的马尔可夫链,其概率空间与其不确定性选择有关.

广义可能性模型检测是由李永明^[10,11]等提出,他们将模糊理论中的可能性测度与模型检测技术相结合,其基本的系统模型是广义可能性 Kripke 结构.潘海玉等研究了基于任意模糊逻辑代数结构的 FCTL 模型检测^[14]和基于任意有限格的 CTL 的模型检测问题^[15],其基本的系统模型是模糊 Kripke 结构.这两者的模型结构特点是当前状态对应于一个模糊状态集.然而在实际系统建模时存在这样的系统,在某一状态上后继状态的选择是基于不确定的模糊选择.因此这样的系统同时具有不确定性和模糊性,本文称这类系统为不确定型模糊 Kripke 结构.例如模糊有限自动机^[16-18].对于具有这类特点的系统结构,文献[19,20]分别研究了基于广义可能性结构和多值结构的模型检测,并分别给出了多项式时间的模型检测算法.在本文中研究基于任意模糊逻辑结构的不确定型 Kripke 结构的 CTL 模型检测问题,并给出解决该问题的对数多项式时间的改进算法,该研究扩展了模型检测技术在模糊系统中的应用范围.

2 基础知识

模糊集^[21]用于刻画隶属范围不是很清楚的集合,形式化定义为:设 X 是一个有限论域, $A: X \rightarrow [0,1]$ 是论域 X 上的模糊集.对任意 $x \in X, A(x)$ 表示 x 属于 A 的真值.若对任意 $x \in X, A(x) \in \{0,1\}$,则称 A 为分明集. $\text{Supp}(A)$ 表示模糊集 A 的支集,定义为 $\text{Supp}(A) = \{x \in X: A(x) > 0\}$.若 $\text{Supp}(A)$ 是有限集,并设 $\text{Supp}(A) = \{x_1, x_2, \dots, x_n\}$,则 $A = A(x_1)/x_1 + A(x_2)/x_2 + \dots + A(x_n)/x_n$.用 $F(X)$ 表示 X 上的所有模糊集.任意 $A, B \in F(X), A \subseteq B$ 表示对任意的 $x \in X, A(x) \leq B(x)$. $A = B$ 当且仅当 $A \subseteq B$ 且 $B \subseteq A$.为叙述方便,在这里先约定一些记号. \mathbf{N} 表示包含 0 的自然数集. \mathbf{I} 表示指标集.设 X 是有限完备格, $f: X \rightarrow X$ 是单调函数,根据 Tarski 定理^[22], f 存在最小不动点和最大不动点,分别用 $\mu.f$ 和 $\nu.f$ 表示.当 X 是非空有限集时,用 $|X|$ 表示集合 X 中元素的个数.

在模糊逻辑中运算符 t -模 \otimes 、 t -余模 \oplus 、蕴涵 \Rightarrow 和补 \neg 分别是经典逻辑中的合取、析取、蕴涵和否定连接

词的推广,其中 t -模 \otimes 是 $[0,1]$ 上的单调递增,满足交换律,结合律,且以 1 为单位元的二元运算.给定有限序列 $x_1, x_2, \dots, x_n \in [0,1], \otimes_{0 \leq i \leq n} x_i$ 表示为 $x_1 \otimes x_2 \otimes \dots \otimes x_n$.

几类常见的模糊逻辑^[14]为 Zadeh 逻辑、Product 逻辑、Lukasiewicz 逻辑和 Gödel 逻辑.

3 不确定型模糊 Kripke 结构和模糊计算树逻辑

3.1 不确定型模糊 Kripke 结构

模糊 Kripke 结构^[14]是一个四元组 $(S, \text{AP}, \delta, L)$,其中 S 是可数非空状态集合, $\delta: S \rightarrow F(X)$ 是一个模糊状态转换函数, AP 是原子命题集, $L: S \rightarrow F(\text{AP})$ 是模糊标签函数.

定义 1 不确定型模糊 Kripke 结构 (NFKS) 是一个五元组 $M = (S, \text{Act}, \delta, \text{AP}, L)$,其中:

- (1) S 是一个可数非空状态集合;
- (2) Act 是动作集合;
- (3) $\delta: S \times \text{Act} \rightarrow F(X)$ 是模糊转移函数, $\delta(s, a)(t)$ 表示状态 s 在动作 a 作用下转移到状态 t 的真值,其中 $s, t \in S$ 和 $a \in \text{Act}$;
- (4) AP 是一组原子命题集合;
- (5) $L: S \rightarrow F(\text{AP})$ 是模糊标签函数, $L(s)(p)$ 表示原子命题 p 在状态 s 上的真值.

若 S, Act 和 AP 是有限集,则称 M 是有限的.对任意 $s \in S, s$ 的可用动作集定义为 $\Gamma(s) = \{a: \text{存在 } t \in S, \text{使得 } \delta(s, a)(t) > 0\}$.若对于任意 $s \in S, |\Gamma(s)| = 1$,则 NFKS 退化为模糊 Kripke 结构.因此模糊 Kripke 结构是 NFKS 的一种特殊情形.若对任意 $s \in S, \Gamma(s) \neq \emptyset$,则称 M 是完全的.如无特别说明,本文中研究的 NFKS 均是有限和完全的.有限 NFKS M 的大小定义为: $|M| = |S| + \sum_{s \in S} \sum_{a \in \Gamma(s)} |\text{Supp}(\delta(s, a))|$.

在 NFKS M 中,对任意 $s \in S$ 和 $a \in \Gamma(s)$,若存在 $t \in S$ 使得 $\delta(s, a)(t) > 0$,则称 t 是 s 在动作 a 作用下的后继状态.用 $\text{Post}(s, a) = \{t \in S: \delta(s, a)(t) > 0\}$ 表示状态 s 在动作 a 作用下的后继状态集. $\text{Pre}(s) = \{t \in S: \text{存在 } a \in \text{Act} \text{ 使得 } \delta(s, a)(t) > 0\}$ 表示状态 s 的前驱状态集.一条从状态 s 出发的路径是一个无限序列 $\rho = s_0 a_0 s_1 a_1 \dots$,其中 $s_0 = s$,且对所有 $i \geq 0, s_i \in S$ 和 $a_i \in \Gamma(s_i)$,使得 $\delta(s_i, a_i)(s_{i+1}) > 0$.用 $\rho(i)$ 表示路径 ρ 上的第 $i+1$ 个状态.路径 ρ 的有限前缀 $\rho' = s_0 a_0 s_1 a_1 \dots s_n$ 称作有限路径片段.用 $|\rho|$ 表示有限路径 ρ 的长度,即 $|\rho[0 \dots i]| = i$,其中 $\rho[0 \dots i]$ 表示路径 ρ 以 $\rho(i)$ 为最后状态的有限路径片段. $\text{Paths}(M, s)$ 和 $\text{Paths}(M)$ 分别表示 M 中所有从 s 出发的路径集合和 M 中所有路径集合.

在 NFKS M 中,策略 $\sigma: S^+ \rightarrow \text{Act}$ 定义为:对任意的 $s_0 s_1 \dots s_n \in S^+$,使得 $\sigma(s_0 s_1 \dots s_n) \in \Gamma(s_n)$,其中 S^+ 表示 S

上的所有长度大于0的有限序列. 用 Π 表示 NFKS M 中所有策略之集. 在策略 σ 作用下 M 中从 s 出发的一条路径是 $\rho = s_0 a_0 s_1 a_1 \cdots \in \text{Paths}(M, s)$, 其中对所有的 $i \geq 0$, 满足 $\sigma(s_0 s_1 \cdots s_i) = a_i$. 用 $\text{Paths}_\sigma(M, s)$ 和 $\text{Paths}_\sigma(M)$ 分别表示在策略 σ 作用下 M 中从 s 出发的所有路径集合和 M 中的所有路径集合. 当 M 上下文明确时, 省去记号 M .

给定 NFKS $M = (S, \text{Act}, \delta, \text{AP}, L)$ 和 $\sigma \in \Pi$. $M_\sigma = (S^+, \text{Act}, \delta_\sigma, L_\sigma)$ 是 M 由策略 σ 生成的模糊 Kripke 结构, 其中对于任意 $\rho = s_0 s_1 \cdots s_n \in S^+$, $\delta_\sigma(\rho, \rho s_{n+1}) = \delta(s_n, \sigma(\rho))(s_{n+1})$, $L_\sigma(\rho) = L(s_n)$. 在 σ 作用下 M 中的路径与 M_σ 中的路径一一对应.

3.2 模糊计算树逻辑

模糊计算树逻辑可以用于刻画 NFKS 的分支时序性质. 与文献[14]不同的是: 为了区别存在量词“ \exists ”和任意量词“ \forall ”基于 NFKS 的两种不同语义解释, 在 FCTL 语法中引入记号 \exists_{inf} , \forall_{inf} , \exists_{sup} 和 \forall_{sup} . 下面给出 FCTL 的语法和基于 NFKS 的语义解释.

定义 2 设 AP 是一个有限原子命题集. 基于 AP 的模糊计算树逻辑 (简称 FCTL) 公式定义如下:

$$\begin{aligned} \Phi &::= \text{true} \mid p \mid \neg \Phi \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \rightarrow \\ &\quad \Phi_2 \mid \exists_{\text{inf}} \varphi \mid \exists_{\text{sup}} \varphi \mid \forall_{\text{inf}} \varphi \mid \forall_{\text{sup}} \varphi, \\ \varphi &::= X\Phi \mid \Phi_1 U \Phi_2, \end{aligned}$$

其中, $p \in \text{AP}$, Φ, Φ_1, Φ_2 是状态公式, φ 是路径公式.

设 Φ 是 FCTL 状态公式, 则公式 Φ 的长度记作 $|\Phi|$, 递归定义为

$$\begin{aligned} |p| &= |\text{true}| = 1, \\ |\Phi_1 \wedge \Phi_2| &= |\Phi_1 \vee \Phi_2| = |\Phi_1 \rightarrow \Phi_2| = |\Phi_1| + |\Phi_2| + 1, \\ |\neg \Phi| &= |\exists_{\text{inf}} X\Phi| = |\exists_{\text{sup}} X\Phi| = |\forall_{\text{inf}} X\Phi| = |\forall_{\text{sup}} X\Phi| \\ &= |\Phi| + 1, \\ |\exists_{\text{inf}} \Phi_1 U \Phi_2| &= |\exists_{\text{sup}} \Phi_1 U \Phi_2| = |\forall_{\text{inf}} \Phi_1 U \Phi_2| \\ &= |\forall_{\text{sup}} \Phi_1 U \Phi_2| = |\Phi_1| + |\Phi_2| + 1. \end{aligned}$$

给定 NFKS M 和模糊逻辑 FL, 对任意 $s \in S$, $\|\Phi\|_{\text{FL}}^M(s)$ 表示在模糊逻辑 FL 下, 状态 s 满足公式 Φ 的真值. 当 M 和 FL 上下文意义明确时, $\|\Phi\|_{\text{FL}}^M(s)$ 简写为 $\|\Phi\|(s)$.

定义 3 给定 NFKS $M = (S, \text{Act}, \delta, \text{AP}, L)$, $p \in \text{AP}$, $s \in S$, Φ, Φ_1, Φ_2 是 FCTL 的状态公式, φ 是 FCTL 的路径公式. 基于 M 的公式 Φ 的语义是 S 上的一个模糊集合, 即 $\|\Phi\| \in F(S)$. 对公式 Φ 的语义归纳定义如下: 对于任意 $s \in S$,

$$\begin{aligned} \|\text{true}\|(s) &= 1; \\ \|p\|(s) &= L(p)(s); \\ \|\neg \Phi\|(s) &= \neg \|\Phi\|(s); \\ \|\Phi_1 \wedge \Phi_2\|(s) &= \|\Phi_1\|(s) \otimes \|\Phi_2\|(s); \\ \|\Phi_1 \vee \Phi_2\|(s) &= \|\Phi_1\|(s) \oplus \|\Phi_2\|(s); \\ \|\Phi_1 \rightarrow \Phi_2\|(s) &= \|\Phi_1\|(s) \Rightarrow \|\Phi_2\|(s); \end{aligned}$$

$$\begin{aligned} \|\exists_{\text{sup}} \varphi\|(s) &= \sup_{\sigma \in \Pi} \sup_{\rho \in \text{Paths}_\sigma(M, s)} \|\varphi\|_\sigma(\rho); \\ \|\exists_{\text{inf}} \varphi\|(s) &= \sup_{\sigma \in \Pi} \inf_{\rho \in \text{Paths}_\sigma(M, s)} \|\varphi\|_\sigma(\rho); \\ \|\forall_{\text{sup}} \varphi\|(s) &= \inf_{\sigma \in \Pi} \sup_{\rho \in \text{Paths}_\sigma(M, s)} \|\varphi\|_\sigma(\rho); \\ \|\forall_{\text{inf}} \varphi\|(s) &= \inf_{\sigma \in \Pi} \inf_{\rho \in \text{Paths}_\sigma(M, s)} \|\varphi\|_\sigma(\rho). \end{aligned}$$

给定策略 $\sigma \in \Pi$, 对任意 $\rho = s_0 a_0 s_1 a_1 \cdots \in \text{Paths}_\sigma(M, s)$, 公式 φ 的语义是 $\|\varphi\|_\sigma: \text{Paths}_\sigma(M, s) \rightarrow [0, 1]$, 表示在策略 σ 作用下, 路径 ρ 满足公式 φ 的真值. 对公式 φ 的语义归纳定义如下:

$$\begin{aligned} \|X\Phi\|_\sigma(\rho) &= \delta(s_0, a_0)(s_1) \otimes \|\Phi\|(s_1); \\ \|\Phi_1 U \Phi_2\|_\sigma(\rho) &= \sup_{i \geq 0} (\otimes_{0 \leq j \leq i} (\|\Phi_1\|(s_j) \otimes \delta(s_j, a_j) \\ &\quad (s_{j+1})) \otimes \|\Phi_2\|(s_i)). \end{aligned}$$

与 CTL 的语法^[1]相比, FCTL 的语法中有路径量词 \exists_{sup} , \exists_{inf} , \forall_{sup} 和 \forall_{inf} . 但是, 由于在 Kripke 结构中, 策略和路径存在 1-1 对应的关系. 因此当不确定型模糊 Kripke 结构退化为 Kripke 结构时, FCTL 的语义和 CTL 的语义是一致的. 特别地, FCTL 公式 $\exists_{\text{sup}} \varphi$ 和 $\exists_{\text{inf}} \varphi$ 的语义与 CTL 公式 $\exists \varphi$ 的语义一致, FCTL 公式 $\forall_{\text{inf}} \varphi$ 和 $\forall_{\text{sup}} \varphi$ 的语义与 CTL 公式 $\forall \varphi$ 的语义一致.

与文献[14]相比, 定义 1 包含路径量词 \exists_{sup} , \exists_{inf} , \forall_{sup} 和 \forall_{inf} . 由于 FCTL 基于 NFKS 的语义是由状态模糊分布和策略共同决定. 因此, 当不确定型模糊 Kripke 结构退化为模糊 Kripke 结构时, Π 是单点集, 基于 NFKS 的 FCTL 语义和基于模糊 Kripke 结构的 FCTL 语义是一致的. 特别地, $\|\exists_{\text{sup}} \varphi\| = \|\forall_{\text{sup}} \varphi\| = \|\exists \varphi\|$, $\|\exists_{\text{inf}} \varphi\| = \|\forall_{\text{inf}} \varphi\| = \|\forall \varphi\|$, 其中 $\exists \varphi$ 和 $\forall \varphi$ 是文献[14]中基于模糊 Kripke 结构的 FCTL 公式.

4 FCTL 模型检测

给定 NFKS $M = (S, \text{Act}, \delta, \text{AP}, L)$ 和模糊逻辑 FL, 设 Φ 是 FCTL 公式, 基于 NFKS M 的 FCTL 模型检测问题是计算 $\|\Phi\|(s)$ 的值, 其中 $s \in S$. 若已知 Φ, Φ_1 和 Φ_2 在任意状态 S 上的真值, 由定义 2 知, 公式 $\text{true}, p, \neg \Phi, \Phi_1 \wedge \Phi_2, \Phi_1 \vee \Phi_2, \Phi_1 \rightarrow \Phi_2$ 的模型检测问题容易解决. 但对于公式 $\exists_{\text{inf}} \varphi, \exists_{\text{sup}} \varphi, \forall_{\text{inf}} \varphi$ 和 $\forall_{\text{sup}} \varphi$ 的模型检测问题, 需要分 $\varphi = X\Phi$ 和 $\varphi = \Phi_1 U \Phi_2$ 两种情况讨论. 又因为公式 $\exists_{\text{inf}} X\Phi, \exists_{\text{sup}} X\Phi, \forall_{\text{inf}} X\Phi$ 和 $\forall_{\text{sup}} X\Phi$ 的模型检测问题可根据定义 2 解决. 因此本节主要研究公式 $\exists_{\text{inf}} pUq, \exists_{\text{sup}} pUq, \forall_{\text{inf}} pUq$ 和 $\forall_{\text{sup}} pUq$ 的模型检测问题, 其中 $p, q \in \text{AP}$.

首先从 FCTL 语义的不动点刻画入手, 给出一种不动点迭代算法解决 FCTL 公式 $\exists_{\text{inf}} pUq, \exists_{\text{sup}} pUq, \forall_{\text{inf}} pUq$ 和 $\forall_{\text{sup}} pUq$ 的模型检测问题. 然后进一步对 FCTL 公式 $\exists_{\text{inf}} pUq, \exists_{\text{sup}} pUq, \forall_{\text{inf}} pUq$ 和 $\forall_{\text{sup}} pUq$ 的模型检测问题, 给出不同的改进算法. 另外为了分析算法的时间复杂度, 这里设运算符 $\otimes, \oplus, \Rightarrow, \neg$ 在单位区间 $[0, 1]$ 上的计算时间为常数.

4.1 不动点算法

本节用不动点迭代算法计算 $\|\exists_{\sup} pUq\|$, $\|\exists_{\inf} pUq\|$, $\|\forall_{\sup} pUq\|$ 和 $\|\forall_{\inf} pUq\|$ 的值. 首先给出有限路径形式的时态算子 U 的语义定义: 给定 NFKS M , 对任意 $\sigma \in \Pi$ 和 $\rho \in \text{Paths}_\sigma(M)$, $\|\exists pU^n q\|_\sigma(\rho) = \sup_{0 \leq i \leq n} (\otimes_{0 \leq j \leq i} (\|p\|(s_j) \otimes \delta(s_j, a_j)(s_{j+1}))) \otimes \|q\|(s_i)$, 其中 $i, j, n \in \mathbf{N}$.

下面的引理 1 说明: $\|\exists_{\sup} pUq\|(s)$, $\|\exists_{\inf} pUq\|(s)$, $\|\forall_{\sup} pUq\|(s)$ 和 $\|\forall_{\inf} pUq\|(s)$ 的值只与从状态 s 出发长度小于等于 $|S|$ 的有限路径片段相关. 证明方法与文献[14]类似.

引理 1 设 $M = (S, \text{Act}, \delta, \text{AP}, L)$ 是 NFKS, 对任意 $s \in S$, 则下列等式成立:

$$\begin{aligned} \|\exists_{\sup} pUq\|(s) &= \|\exists_{\sup} pU^{|S|} q\|(s); \\ \|\exists_{\inf} pUq\|(s) &= \|\exists_{\inf} pU^{|S|} q\|(s); \\ \|\forall_{\sup} pUq\|(s) &= \|\forall_{\sup} pU^{|S|} q\|(s); \\ \|\forall_{\inf} pUq\|(s) &= \|\forall_{\inf} pU^{|S|} q\|(s). \end{aligned}$$

引理 1 说明: FCTL 公式 $\exists_{\inf} pUq$, $\exists_{\sup} pUq$, $\forall_{\inf} pUq$ 和 $\forall_{\sup} pUq$ 的语义值只与有限路径片段有关. 当只考虑有限路径片段时, 策略集合为有限集. 因此公式 $\exists_{\inf} pUq$, $\exists_{\sup} pUq$, $\forall_{\inf} pUq$ 和 $\forall_{\sup} pUq$ 的语义分别可以写为

$$\begin{aligned} \|\exists_{\sup} \varphi\|(s) &= \max_{\sigma \in \Pi} \max_{\rho \in \text{Paths}_\sigma(M, s)} \|\varphi\|_\sigma(\rho); \\ \|\exists_{\inf} \varphi\|(s) &= \max_{\sigma \in \Pi} \min_{\rho \in \text{Paths}_\sigma(M, s)} \|\varphi\|_\sigma(\rho); \\ \|\forall_{\sup} \varphi\|(s) &= \min_{\sigma \in \Pi} \max_{\rho \in \text{Paths}_\sigma(M, s)} \|\varphi\|_\sigma(\rho); \\ \|\forall_{\inf} \varphi\|(s) &= \min_{\sigma \in \Pi} \min_{\rho \in \text{Paths}_\sigma(M, s)} \|\varphi\|_\sigma(\rho). \end{aligned}$$

另外, t -模 \otimes 满足如下性质: 对任意 $x, y, z \in [0, 1]$, $x \otimes \max\{y, z\} = \max\{x \otimes y, x \otimes z\}$, $x \otimes \min\{y, z\} = \min\{x \otimes y, x \otimes z\}$. 由以上分析可得: FCTL 公式 $\exists_{\inf} pUq$, $\exists_{\sup} pUq$, $\forall_{\inf} pUq$ 和 $\forall_{\sup} pUq$ 的语义值可以采用不动点理论求解.

定理 1 设 $M = (S, \text{Act}, \delta, \text{AP}, L)$ 是 NFKS, 函数 $F_i: F(S) \rightarrow F(S)$, $i = 1, 2, 3, 4$, 定义为: 对任意 $A \in F(S)$, $s \in S$, $p, q \in \text{AP}$,

$$\begin{aligned} F_1(A)(s) &= \|q\|(s) \vee (\|p\|(s) \otimes \sup_{a \in \Gamma(s)} \sup_{t \in S} (\delta(s, a)(t) \otimes A(t))); \\ F_2(A)(s) &= \|q\|(s) \vee (\|p\|(s) \otimes \inf_{a \in \Gamma(s)} \sup_{t \in S} (\delta(s, a)(t) \otimes A(t))); \\ F_3(A)(s) &= \|q\|(s) \vee (\|p\|(s) \otimes \sup_{a \in \Gamma(s)} \inf_{t \in S} (\delta(s, a)(t) \otimes A(t))); \\ F_4(A)(s) &= \|q\|(s) \vee (\|p\|(s) \otimes \inf_{a \in \Gamma(s)} \inf_{t \in S} (\delta(s, a)(t) \otimes A(t))). \end{aligned}$$

则 $\|\exists_{\sup} pUq\|$, $\|\forall_{\sup} pUq\|$, $\|\exists_{\inf} pUq\|$ 和 $\|\forall_{\inf} pUq\|$ 分别是函数 F_1, F_2, F_3 和 F_4 的最小不动点.

证明 这里只证明 $\|\exists_{\sup} pUq\|$ 是函数 F_1 的最小不动点. 函数 F_2, F_3 和 F_4 的不动点特性的证明方法类似, 所以省略它们的证明过程.

首先证明函数 F_1 存在最小不动点. 设 $A, B \in F(S)$ 且 $A \subseteq B$, 则对任意的 $s \in S$, $F_1(A)(s) \leq F_1(B)(s)$, 从而可以得到 $F_1(A)(s) = \|q\|(s) \vee (\|p\|(s) \otimes \sup_{a \in \Gamma(s)} \sup_{t \in S} (\delta(s, a)(t) \otimes A(t))) \leq \|q\|(s) \vee (\|p\|(s) \otimes \sup_{a \in \Gamma(s)} \sup_{t \in S} (\delta(s, a)(t) \otimes B(t))) = F_1(B)(s)$. 故 F_1 是 $F(S)$ 上单调递增函数. 由 Tarski 不动点理论^[22]可知, F_1 存在最小不动点. 下面证明 $\|\exists_{\sup} pUq\|$ 是函数 F_1 的最小不动点.

根据 F_1 的定义和 t -模的性质, 对任意 $s \in S$, $F_1(\|\exists_{\sup} pUq\|)(s) = \|q\|(s) \vee (\|p\|(s) \otimes \sup_{a \in \Gamma(s)} \sup_{t \in S} (\delta(s, a)(t) \otimes \|\exists_{\sup} pU^{|S|} q\|(t))) = \|q\|(s) \vee (\|p\|(s) \otimes \max_{a \in \Gamma(s)} \max_{t \in S} (\delta(s, a)(t) \otimes \|\exists_{\sup} pU^{|S|} q\|(t))) = (\max_{a \in \Gamma(s)} \max_{t \in S} (\|q\|(s) \vee \|p\|(s) \otimes (\delta(s, a)(t) \otimes \sup_{\sigma \in \Pi} \sup_{\rho \in \text{Paths}_\sigma(t)} \|pU^{|S|} q\|(\rho)))) = \sup_{\sigma \in \Pi} \sup_{\rho \in \text{Paths}_\sigma(s)} \|pU^{|S|+1} q\|(\rho^s) = \|\exists_{\sup} pUq\|(s)$, 其中 $\rho^s = \rho$. 因此 $\|\exists_{\sup} pUq\|$ 是 F_1 的一个不动点. 接下来证明: $\|\exists_{\sup} pUq\|$ 是 F_1 的最小不动点, 即 F_1 的所有不动点 $Z \in F(S)$, 都满足 $\|\exists_{\sup} pUq\| \subseteq Z$.

假设存在 $s \in S$ 和函数 F_1 的一个不动点 Z , 使得 $Z(s) < \|\exists_{\sup} pUq\|(s)$. 根据引理 1 和 M 的有限性, 必定存在一条从 s 出发的有限路径 $s_0 a_0 s_1 a_1 \cdots s_{n-1} a_{n-1} s_n$, 使得 $s_0 = s$ 且对任意 $0 \leq i < n$, $a_i \in \Gamma(s)$, $\delta(s_i, a_i)(s_{i+1}) > 0$, 满足 $\|\exists_{\sup} pUq\|(s) = \|p\|(s) \otimes \delta(s, a_0)(s_1) \otimes \cdots \otimes \|p\|(s_{n-1}) \otimes \delta(s_{n-1}, a_{n-1})(s_n) \otimes \|q\|(s_n)$. 又因为 $F_1(Z) = Z$, 故 $Z(s) = \|q\|(s) \vee (\|p\|(s) \otimes \sup_{a \in \Gamma(s)} \sup_{t \in S} (\delta(s, a)(t) \otimes Z(t))) \geq \|p\|(s) \otimes \delta(s, a_0)(s_1) \otimes Z(s_1) \cdots \geq \|p\|(s) \otimes \delta(s, a_0)(s_1) \otimes \cdots \otimes \|p\|(s_{n-1}) \otimes \delta(s_{n-1}, a_{n-1})(s_n) \otimes \|q\|(s_n)$, 与假设矛盾. 因此 $\|\exists_{\sup} pUq\| = \mu. F_1$.

计算单调增函数 F 的最小不动点的迭代算法伪代码如算法 1 所示.

算法 1 迭代算法

输入: NFKS $M = (S, \text{Act}, \delta, \text{AP}, L)$ 和单调递增函数 $F: F(S) \rightarrow F(S)$;
输出: $\mu. F$;

```

1:  $i = 0$ ;  $X_0 = \emptyset$ ;
2: do {
3:  $X_{i+1} = F(X_i)$ ;
4:  $i = i + 1$ ;
5: } while  $X_i \neq X_{i+1}$ ;
6: end while;
7: return  $X_i$ .

```

由定理 1 可知, 当函数 F 分别为定理 1 中函数 F_1, F_2, F_3 和 F_4 时, 通过算法 1 可以求得 $\|\exists_{\sup} pUq\|$, $\|\forall_{\sup} pUq\|$, $\|\exists_{\inf} pUq\|$ 和 $\|\forall_{\inf} pUq\|$ 的值. 算法 1 的时间复杂度为 $O(|S| |M|)$.

4.2 改进算法

本节研究如何花费更少的时间解决 FCTL 公式 $\exists_{\text{inf}} pUq$, $\exists_{\text{sup}} pUq$, $\forall_{\text{inf}} pUq$ 和 $\forall_{\text{sup}} pUq$ 的模型检测问题, 并且给出不同的改进算法.

4.2.1 $\exists_{\text{sup}} pUq$ 的模型检测问题

受文献[23]和文献[14]中的启发, 本文采用优先队列这类数据结构解决 FCTL 公式 $\exists_{\text{sup}} pUq$ 的模型检测问题, 其中优先队列的实现方法为二叉堆. 计算 $\|\exists_{\text{sup}} pUq\|$ 值的伪代码如算法 2 所示. 下面通过三个引理证明: 当算法 2 终止时, $v = \|\exists_{\text{sup}} pUq\|$. 引理 2, 3, 4 的证明细节见文献[14, 23].

算法 2 计算 $\|\exists_{\text{sup}} pUq\|$

输入: NFKS $M = (S, \text{Act}, \delta, \text{AP}, L)$ 和公式 $\exists_{\text{sup}} pUq$;

输出: $\|\exists_{\text{sup}} pUq\|$;

- 1: for 任意 $s \in S$ do
- 2: INSERT-KEY($Q, s, \|q\|(s)$); $\quad \backslash$ 将赋予优先权 $v(s)$ 的元素 s 插入到队列 Q 中.
- 3: while $Q \neq \emptyset$ do
- 4: $(t, v(t)) \leftarrow \text{EXTRACT-MAX}(Q)$ $\quad \backslash$ 输出 $(t, v(t))$ 并且把元素 t 从队列 Q 中删除, 其中 t 是队列 Q 中拥有最高优先权的元素且其优先权为 $v(t)$.
- 5: for 任意 $s \in S$ do
- 6: if 存在动作 $a \in \Gamma(s)$ 满足 $\delta(s, a)(t) > 0$ then
- 7: INCREASE-KEY($Q, s, \|q\|(s) \vee (\|p\|(s) \otimes \max_{a \in \Gamma(s)} (\delta(s, a)(t) \otimes v(t)))$); $\quad \backslash$ 将队列 Q 中元素 s 的优先权赋予新值 x . 假定新值 x 不小于当前的优先权. 其中 $x = \|p\|(s) \otimes \max_{a \in \Gamma(s)} (\delta(s, a)(t) \otimes v(t))$.

引理 2 在执行算法 2 的过程中, 如果 s 已经从队列 Q 中删除, 则 $v(s)$ 不小于队列 Q 中的最高优先权.

引理 3 当算法 2 终止时, 对任意的 $s \in S$, $v(s) \leq \|\exists_{\text{sup}} pUq\|(s)$.

引理 4 当算法 2 终止时, 对于任意的 $s \in S$, $v(s) \geq \|\exists_{\text{sup}} pUq\|(s)$.

综合引理 2 和 3, 算法 2 可以计算 $\|\exists_{\text{sup}} pUq\|$ 的值.

定理 2 当算法 2 终止时, 对任意 $s \in S$, $v(s) = \|\exists_{\text{sup}} pUq\|(s)$.

定理 3 算法 2 的时间复杂度为 $O(|M| + |S| \log |S|)$.

证明 初始时, 待排序的结点以无序形式连续存放在一个数组中, 对其进行堆排序. 调整成最大堆后, 各结点即是以完全二叉树的顺序存储结构形式存储. 调整堆的过程中时间复杂度为 $O(\log |S|)$, 该过程共需迭代执行 $|S|$ 次. 每次删除堆顶结点后, 逆向搜索该结点的前驱结点并且对每个前驱结点调用一次 INCREASE-KEY 操作, 总共需要迭代该过程 $\sum_{s \in S} \sum_{a \in \Gamma(s)} |\text{Supp}(\delta(s, a))|$ 次 (每条边和每个状态最多被访问一次, 但是 s 的优先权只改变一次). 故算法 2 最长运行时间为 $O(|S|$

$\log |S| + \sum_{s \in S} \sum_{a \in \Gamma(s)} |\text{Supp}(\delta(s, a))|) = O(|M| + |S| \log |S|)$.

4.2.2 $\forall_{\text{sup}} pUq$ 的模型检测问题

FCTL 公式 $\forall_{\text{sup}} pUq$ 的模型检测问题可以在线性时间内归纳为基于 t -模的加权有向图单目的地最长路径问题^[14].

给定加权有向图 $G = (V, E)$ 和目的地结点 $t \in V$, 其加权函数 $w: E \rightarrow [0, 1]$, 该加权函数将每条边映射到 $[0, 1]$ 的权重上. G 上的一条路径 $\pi = v_0 v_1 \cdots v_k$ 的权值 $w(\pi) = \otimes_{0 \leq i \leq k} w(v_i, v_{i+1})$.

从结点 u 到 v 的最长路径权值 $d(u, v)$ 定义为:

$$d(u, v) = \begin{cases} \max \{ w(\pi) : \pi \in \text{Paths}_{G(u, v)} \}, & \text{Paths}_{G(u, v)} \neq \emptyset \\ 0, & \text{Paths}_{G(u, v)} = \emptyset \end{cases}$$

其中, $\text{Paths}_{G(u, v)}$ 表示在加权有向图 $G = (V, E)$ 中, 从结点 u 到结点 v 的所有路径.

单目的地最长路径问题是: 给定加权有向图 G 和目的地结点 t , 计算 $d(u, t)$ 的值, 其中 $u \in V$. 即计算加权有向图 G 中的任意结点 u 到目的地结点 t 的最长路径权值.

以下定理表明, 可以用文献[14]中提出的解决单目的地最长路径问题的算法来解决 FCTL 公式 $\forall_{\text{sup}} pUq$ 的模型检测问题.

定理 4 给定 NFKS $M = (S, \text{Act}, \delta, \text{AP}, L)$, FCTL 公式 $\forall_{\text{sup}} pUq$ 的模型检测问题可以在 $O(|S| \log |S| + \sum_{s \in S} \sum_{a \in \Gamma(s)} |\text{Supp}(\delta(s, a))|)$ 时间内解决.

证明 给定 NFKS M 和 FCTL 公式 $\forall_{\text{sup}} pUq$, 加权有向图 $G = (V, E)$ 构造如下: $V = S \cup \{t\}$, 其中 $t \notin S$ 且 t 是目的地结点. $E = \{(s, s') : s, s' \in S, \exists a \in \Gamma(s), \text{s.t.}, \delta(s, a)(s') > 0\} \cup \{(s, t) : s \in S\}$. 加权函数 $w: E \rightarrow [0, 1]$ 定义为: 对任意 $(s, s') \in E$,

$$w(s, s') = \begin{cases} \|p\|(s) \otimes \min_{a \in \Gamma(s)} \delta(s, a)(s'), & s, s' \in S \\ \|q\|(s), & \text{其他情况} \end{cases}$$

下面通过对 l_s 归纳证明: 对任意 $s \in S$, $\|\forall_{\text{sup}} pUq\|(s) = d(s, t)$, 其中 l_s 的意义如定理 6 所示, 且对所有 $t \in \cup_{a \in \Gamma(s)} \text{Post}(s, a)$, 满足 $l_s > l_t$.

当 $l_s = 0$ 时, $\|\forall_{\text{sup}} pUq\|(s) = \|q\|(s) = d(s, t)$ 显然成立. 假设当 $l_s = k \leq |S|$ 时, 等式 $\|\forall_{\text{sup}} pUq\|(s) = d(s, t)$ 成立, 则当 $l_s = k + 1$ 时, 由定理 1 可得,

$$\begin{aligned} \|\forall_{\text{sup}} pUq\|(s) &= \|q\|(s) \vee (\|p\|(s) \otimes \inf_{a \in \Gamma(s)} \sup_{s' \in S} \delta(s, a)(s') \otimes \|\forall_{\text{sup}} pUq\|(s')) \\ &= \|q\|(s) \vee (\|p\|(s) \otimes \sup_{s' \in S} \inf_{a \in \Gamma(s)} \delta(s, a)(s') \otimes \|\forall_{\text{sup}} pUq\|(s')) \\ &= \|q\|(s) \vee \sup_{s' \in S} (\|p\|(s) \otimes \inf_{a \in \Gamma(s)} \delta(s, a)(s') \otimes d(s', t)) \end{aligned}$$

$$\begin{aligned} &= \|q\|(s) \vee \sup_{s' \in S} (w(s, s') \otimes d(s', t)) \\ &= d(s, t). \end{aligned}$$

根据文献[14]中定理 4.1 可得,解决模糊计算树逻辑公式 $\forall_{\text{sup}} pUq$ 模型检测问题的时间复杂度为 $O(\sum_{s \in S} \sum_{a \in \Gamma(s)} |\text{Supp}(\delta(s, a))| + |S| \log |S|)$.

4.2.3 $\exists_{\text{inf}} pUq$ 的模型检测问题

FCTL 公式 $\exists_{\text{inf}} pUq$ 的模型检测问题可以在线性时间内归约为基于 t 模的 semi-模糊博弈的可达目标状态问题^[24].

设 $G = (G, R)$ 是 semi-模糊可达博弈图,其中 $G = (V_1, V_2, \text{Act}, \Gamma_1, \Gamma_2, \delta_1, \delta_2)$ 是含有两个局中人 (Player₁ 和 Player₂) 的 semi-模糊博弈图,且满足对任意 $s \in V_2$ 和 $b \in \Gamma_2(s)$, $|\delta_2(s, b)| = 1$. 其中 V_i 表示 Player _{i} 的状态集, $\Gamma_i: V_i \rightarrow \text{Act}$ 表示动作指派函数, $\delta_i: S_i \times \text{Act} \rightarrow F(S)$ 是 Player _{i} 的模糊转移函数, $i = 1, 2$. R 是目标状态集. Π_1, Π_2 分别表示 Player₁, Player₂ 的策略集. 对任意 $s \in S$, $\sigma_1 \in \Pi_1$ 和 $\sigma_2 \in \Pi_2$, $\pi = s_0 \rightarrow^{x_0|a_0} s_1 \rightarrow^{x_1|a_1} \dots$ (是由策略 σ_1 和 σ_2 生成且从 s 出发的一条路径,则路径 π 到达目标状态集 R 的真值定义为:

$$G_s^{\sigma_1, \sigma_2}(\pi) = \begin{cases} R(s_0), & s_0 = s \\ x_0 \otimes x_1 \otimes \dots \otimes x_n \otimes R(x_n), & n = \min \{i \in I, R(x_i) > 0\} \\ 0, & \text{其他情况} \end{cases}$$

状态 s 到达目标状态集 R 的真值定义为:

$$\begin{aligned} T(s) &= \inf_{\sigma_2 \in \Pi_2} \sup_{\sigma_1 \in \Pi_1} \sup \{G_s^{\sigma_1, \sigma_2}(\pi) : \pi \in G_s^{\sigma_1, \sigma_2}\} \\ &= \sup_{\sigma_1 \in \Pi_1} \inf_{\sigma_2 \in \Pi_2} \sup \{G_s^{\sigma_1, \sigma_2}(\pi) : \pi \in G_s^{\sigma_1, \sigma_2}\} \end{aligned}$$

以下定理表明,可以用文献[23]中提出的解决 semi-模糊博弈图的可达目标状态集问题的算法来解决 FCTL 公式 $\exists_{\text{inf}} pUq$ 模型检测问题.

定理 5 给定 NFKS $M = (S, \text{Act}, \delta, \text{AP}, L)$, 则 FCTL 公式 $\exists_{\text{inf}} pUq$ 的模型检测问题可以在线性时间内归约为 semi-模糊博弈图的可达目标状态集问题.

证明 给定 NFKS M 和 FCTL 公式 $\exists_{\text{inf}} pUq$, 可以构造一个 semi-模糊可达博弈图 $G = (G, R)$, 在这里 $G = (V_1, V_2, \text{Act}_c, \Gamma_1, \Gamma_2, \delta_1, \delta_2)$ 和 R 是目标状态集, 其中 $V_1 = \{s_i : s_i \in S\}$ 表示 Player₁ 的状态, $V_2 = \{s_i^a : s_i \in S, a \in \Gamma(s_i)\}$ 是 Player₂ 的状态集; $\text{Act}_c = \text{Act}_1 \cup \text{Act}_2$, 其中 $\text{Act}_1 = \text{Act}$ 是 Player₁ 对应的动作集, $\text{Act}_2 = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ 是 Player₂ 对应的动作集; 对任意 $s_i \in V_1$, $\Gamma_1(s_i) = \Gamma(s_i)$, 对任意 $s \in V_2$, $\Gamma_2(s) = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$, 对任意的 $s \in V_1, a \in \text{Act}_1$,

$$\delta_1(s_i, a)(s) = \begin{cases} \|q\|(s_i), & s = s_i^a \text{ 且 } \text{Supp}(\delta(s_i, a)) \neq \emptyset \\ 0, & \text{其他情况} \end{cases}$$

对任意 $\alpha_j \in \text{Act}_2, s_i^a \in V_2$

$$\delta_2(s_i^a, \alpha_j)(s) = \begin{cases} \delta(s_i, a)(s_j), & s \in V \text{ 且 } \delta(s_i, a)(s_j) > 0 \\ 0, & \text{其他情况} \end{cases}$$

$$\text{目标状态集 } R(s) = \begin{cases} \|q\|(s), & s \in V_1 \\ 0, & s \in V_2 \end{cases}$$

如上所示: 给定 NFKS M 和公式 $\exists_{\text{inf}} pUq$, 可以在线性时间内构造一个 semi-模糊博弈图 G , 且 G 的状态个数 $|V| = |S| + |S| \cdot |\text{Act}|$, 大小 $|G| = |M| + |S| \cdot |\text{Act}|$. 则有

$$\begin{aligned} \|\exists_{\text{inf}} pUq\|(s) &= \sup_{\sigma \in \Pi_{\rho \in \text{Paths}_\rho(s)}} \inf_{\rho \in \text{Paths}_\rho(s)} \|pUq\|(s) \\ &= \sup_{\sigma \in \Pi_{\rho \in \text{Paths}_\rho(s)}} \inf_{i \geq 0} (\sup_{0 \leq j \leq i} (\|q\|(s_j) \otimes \delta(s_j, \tau_j)(s_{j+1}))) \otimes \|q\|(s_i) \\ &= \sup_{\sigma \in \Pi_{\rho \in \text{Paths}_\rho(s)}} \inf_{i \geq 0} (\sup_{0 \leq j \leq i} (\delta_1(s_j, \tau_j)(s_{j+1}^{\tau_j}) \otimes \delta_2(s_{j+1}^{\tau_j}, \alpha_{j+1})(s_{j+1}^{\tau_j}))) \otimes R(s_i) \\ &= T(s) \end{aligned}$$

其中, $\tau_j = \sigma(\rho[0 \dots j])$.

根据文献[23]中用于解决 semi-模糊博弈图可达目标状态集问题的算法的时间复杂性分析, FCTL 公式 $\exists_{\text{inf}} pUq$ 模型检测问题可以在 $O((|M| + |S| \cdot |\text{Act}|) \log(|S| \cdot |\text{Act}|))$ 时间内解决.

4.2.4 $\forall_{\text{inf}} pUq$ 的模型检测问题

受文献[14, 23]的启发, FCTL 公式 $\forall_{\text{inf}} pUq$ 的模型检测问题可以采用优先队列这种数据结构来解决, 其中优先队列的实现方法为最大堆. 计算 $\|\forall_{\text{inf}} pUq\|$ 值的伪代码如算法 3 所示. 下证: 当算法 3 终止时, $v = \|\forall_{\text{inf}} pUq\|$.

算法 3 计算 $\|\forall_{\text{inf}} pUq\|$

输入: NFKS $M = (S, \text{Act}, \delta, \text{AP}, L)$ 和公式 $\forall_{\text{inf}} pUq$;

输出: $v = \|\forall_{\text{inf}} pUq\|$;

- 1: for 任意 $s \in S$ do
- 2: $\text{count}[s] \leftarrow \sum_{a \in \Gamma(s)} |\text{Supp}(\delta(s, a))|$; $w[s] \leftarrow 1$;
- 3: INSERT-KEY($Q, s, \|q\|(s)$); \backslash 将赋予优先权 $v(s)$ 的元素 s 插入到队列 Q 中.
- 4: while $Q \neq \emptyset$ do;
- 5: $(t, v(t)) \leftarrow \text{EXTRACT-MAX}(Q)$ \backslash 输出 $(t, v(t))$ 并且把元素 t 从队列 Q 中删除, 其中 t 是队列 Q 中拥有最高优先权的元素且其优先权为 $v(t)$.
- 6: for 任意 $s \in S$, 存在动作 $a \in \Gamma(s)$ 使得 $t \in \text{Post}(s, a)$ do;
- 7: $\text{count}[s] \leftarrow \text{count}[s] - |\{a \in \Gamma(s) : t \in \text{Post}(s, a)\}|$;
- 8: $w[s] \leftarrow w[s] \wedge (\|q\|(s) \otimes \min_{a \in \Gamma(s)} (\delta(s, a)(t) \otimes v(t)))$;
- 9: if $\text{count}[s] = 0$ then
- 10: INCREASE-KEY($Q, s, \|q\|(s) \vee w[s]$); \backslash 将队列 Q 中元素 s 的优先权赋予新值 x . 假定新值 x 不小于当前的优先权. 其中 $x = \|q\|(s) \vee w[s]$.

定理 6 当算法 3 终止时, 对任意 $s \in S, v(s) =$

$\| \forall_{\text{inf}} p U q \| (s)$.

证明 要证该定理的结论成立, 只需证明以下两条断言成立:

(1) 对任意 $s \in S, v(s) \leq \| \forall_{\text{inf}} p U q \| (s)$;

(2) 对任意 $s \in S, v(s) \geq \| \forall_{\text{inf}} p U q \| (s)$.

通过对结点从队列 Q 中删除的先后次序归纳假设证明: 对任意 $s \in S, v(s) \leq \| \forall_{\text{inf}} p U q \| (s)$.

当 s 是第一个从 Q 中删除的结点时, 由算法 3, $v(s) = \| q \| (s)$. 根据定理 1, $v(s) \leq \| \forall_{\text{inf}} p U q \| (s)$ 成立.

假设先于结点 s 从 Q 中删除的任意结点 t , 满足 $v(s) \leq \| \forall_{\text{inf}} p U q \| (t)$. 下证对 s 执行操作 EXTRACT-MAX 后, $v(s) \leq \| \forall_{\text{inf}} p U q \| (s)$. 当对 s 执行操作 EXTRACT-MAX 时, 如果 $\text{count}[s] \neq 0$, 则由算法 3 知, $v(s) = \| q \| (s)$. 根据定理 1, $\| \forall_{\text{inf}} p U q \| (s) = \| q \| (s) \vee (\| p \| (s) \otimes \min_{a \in \Gamma(s)} \min_{t \in S} \delta(s, a)(t) \otimes \| \forall_{\text{inf}} p U q \| (t)) \geq \| q \| (s) = v(s)$. 若 $\text{count}[s] = 0$, 则由算法 3 和定理 1 可以得到, $\| \forall_{\text{inf}} p U q \| (s) = \| q \| (s) \vee (\| p \| (s) \otimes \min_{a \in \Gamma(s)} \min_{t \in S} (\delta(s, a)(t) \otimes \| q \| (s))) \geq \| q \| (s) \vee (\| p \| (s) \otimes \min_{a \in \Gamma(s)} \min_{t \in S} (\delta(s, a)(t) \otimes \| q \| (s))) = v(s)$.

综上所述, 结论成立.

在 NFKS M 中, 设 $\eta_s^\sigma = \{ \rho_s : \rho_s \text{ 是 } \rho \text{ 的最短有限路径片段且满足 } \| p U q \|_\sigma(\rho) = \| p U q \|_\sigma(\rho_s), \rho \in \text{Paths}_\sigma(s) \}$, $l_s^\sigma = \max \{ |\eta^\sigma| : \eta^\sigma \in \eta_s^\sigma \}$, $l_s = \max \{ l_s^\sigma : \sigma \in \Pi \}$, 其中 $s \in S$ 和 $\sigma \in \Pi$. 下面对 l_s 归纳假设证明: 对任意 $s \in S, v(s) \geq \| \forall_{\text{inf}} p U q \| (s)$.

当 $l_s = 0$ 时, $\| \forall_{\text{inf}} p U q \| (s) = \| q \| (s) \leq v(s)$ 显然成立. 假设当 $0 < l_s \leq k \leq |S|$ 时, $\| \forall_{\text{inf}} p U q \| (s) \leq v(s)$ 成立. 下证当 $l_s = k + 1$ 时, $\| \forall_{\text{inf}} p U q \| (s) \leq v(s)$ 成立.

证明当 $l_s = k + 1$ 时, 对任意 $t \in \cup_{a \in \Gamma(s)} \text{Post}(s, a)$, $l_t \leq l_s - 1$. 若该断言不成立, 则必定存在 $a \in \Gamma(s)$ 和 $s \in S, \delta(s, a)(t) > 0$ 使得 $l_t > l_s = k$. 进而必定存在一个策略 σ 和路径 $\rho \in \text{Paths}_\sigma(t)$, $\rho^* \in \eta_t^\sigma$ 且 $|\rho^*| > k$. 从而 $s \rho \in \text{Paths}_\sigma(s)$ 且 $s \rho^* \in \eta_s^\sigma$, 其中 σ^a 表示策略 $\sigma^a(s) = a$, 其他情况下与策略 σ 动作选择一样. 则 $l_t^\sigma = |s \rho^*| > k + 1 > l_s$, 与已知矛盾. 因此断言成立.

下面通过反证法证明: 当对 s 执行操作 EXTRACT-MAX 时, $\text{count}[s] = 0$. 设 $\text{count}[s] \neq 0$, 则存在 $t \in \cup_{a \in \Gamma(s)} \text{Post}(s, a)$, 且 t 仍在队列 Q 中, 因而由算法 3 第 9 ~ 10 行, s 在队列 Q 中的优先权没有改变, 则有 $\| q \| (s) = v(s)$ 和 $v(s) \geq v(t)$. 再根据归纳假设, $\| q \| (s) < \| \forall_{\text{inf}} p U q \| (s) \leq \| p \| (s) \otimes (\delta(s, a)(t) \otimes \| \forall_{\text{inf}} p U q \| (t)) \leq \| p \| (s) \otimes (\delta(s, a)(t) \otimes v(t)) \leq v(t)$, 与 $v(s) \geq v(t)$ 矛盾. 故当 s 从 Q 中删除时, $\text{count}[s] = 0$.

根据归纳假设和算法 3 第 9 ~ 10 行, 则 $\| \forall_{\text{inf}} p U q \| (s) = \| q \| (s) \vee (\| p \| (s) \otimes \min_{a \in \Gamma(s)} \min_{t \in S} (\delta(s, a)(t) \otimes \| q \| (s))) \leq \| q \| (s) \vee (\| p \| (s) \otimes \min_{a \in \Gamma(s)} \min_{t \in S} (\delta(s, a)(t) \otimes \| q \| (s))) = v(s)$.

定理 7 算法 3 的时间复杂度为 $O(|M| + |S| \log |S|)$.

证明 算法 3 的复杂性分析类似与定理 3.

综合以上分析, 基于 NFKS 的 FCTL 模型检测问题可以在对数多项式时间内完成.

定理 8 设 M 是不确定型模糊 Kripke 结构和 Φ 是 FCTL 公式, 则公式 Φ 的模型检测问题的时间复杂度为 $O(|M| + |S| |\text{Act}| \log(|S| |\text{Act}| |\Phi|))$.

5 结论

本文研究了基于不确定型模糊 Kripke 结构的模糊计算树逻辑模型检测问题, 扩大了模型检测的应用范围, 使得模型检测技术可以适用于具有不确定型模糊系统的形式化验证和分析. 首先给出了 NFKS 的形式化定义. 其次给出基于 NFKS 的 FCTL 语法和语义, 特别地为了刻画基于 NFKS 的 FCTL 中存在量词 \exists 和任意量词 \forall 两种语义解释, 在 FCTL 语法中引入记号 \exists_{sup} , \exists_{inf} 和 \forall_{sup} , \forall_{inf} . 最后研究了 FCTL 公式的模型检测算法.

参考文献

- [1] Christe B, Joost P K. Principles of Model Checking [M]. Cambridge: The MIT Press, 2008.
- [2] 林慧民, 张文辉. 模型检测: 理论、方法与应用 [J]. 电子学报, 2002, 36(12A): 1907 - 1912.
LIN Hui-min, ZHANG Wen-hui. Model checking: Theories, techniques and applications [J]. Acta Electronica Sinica, 2002, 36(12A): 1907 - 1912. (in Chinese)
- [3] Pnueli A. The temporal logic of programs [A]. Proceedings of the 18th IEEE Symposium on Foundations of Computer Science [C]. USA: IEEE, 2017. 46 - 67.
- [4] 朱维军, 周清雷, 李永亮. 以 DNA 为载体的线性时序逻辑模型检测 [J]. 电子学报, 2016, 44(6): 1265 - 1271.
Zhu Wei-jun, Zhou Qing-lei, Li Yong-liang. Conduct linear temporal logic model checking via DNA molecules [J]. Acta Electronica Sinica, 2016, 44(6): 1265 - 1271. (in Chinese)
- [5] Glenn B, Patrice G. Model checking with multi-valued modal logic [A]. Proceedings of the 31st International Colloquium Automata, Languages and Programming [C]. Turku, Finland, 2004. 281 - 293.
- [6] Marsha C, Benet D, Steve E, Arie G. Multi-valued symbolic model-checking [J]. ACM Transactions on Software Engineering and Methodology, 2003, 12(C): 371 - 408.
- [7] Yong M Li, Manfred D, Li H L. Model checking of linear-time properties based on multi-valued systems [J]. Informa-

- tion Sciences, 2017, 377: 51 – 74.
- [8] Yong M Li, Li J Li. Model checking of linear-time properties based on possibility measures [J]. IEEE Transactions on Fuzzy Systems, 2013, 21(5): 842 – 854.
- [9] Yong M Li, Ya L Li, Zhan Y Ma. Computation tree logic model checking based on possibility measures [J]. Fuzzy Sets and Systems, 2015, 262(C): 44 – 59.
- [10] Yong M Li, Zhan Y Ma. Quantitative computation tree logic model checking based on generalized possibility measures [J]. IEEE Transactions on Fuzzy Systems, 2015, 23(C): 2034 – 2047.
- [11] Li Y M. Quantitative Model checking of linear-time properties based on generalized possibility measures [J]. Fuzzy Sets and Systems, 2017, 320(C): 17 – 39.
- [12] Seong-ick M, Ko-Hsin L, Doheo L. Fuzzy branching temporal logic [J]. IEEE Transactions on System, Man and Cybernetics, Part B (Cybernetics), 2004, 34(2): 1045 – 1055.
- [13] Subhankar M, Pallab D. A fuzzy real-time temporal logic [J]. International Journal of Approximate Reasoning, 2013, 54(9): 1452 – 1470.
- [14] Pan H Y, Li Y M, Cao Y Z C, Ma Z Y. Model checking fuzzy computation tree logic [J]. Fuzzy Sets and Systems, 2015, 262(C): 60 – 77.
- [15] 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(C): 45 – 62.
- [16] Li Y M. Finite automata theory with membership valued in lattice [J]. Information Science, 2011, 181(C): 1003 – 1017.
- [17] Cao Y Z, Yoshinoru E. Nondeterministic fuzzy automata [J]. Information Sciences, 2012, 191(C): 86 – 97.
- [18] 李永明, 李平. 模糊计算理论 [M]. 北京: 科学出版社, 2016.
- [19] 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 Zhan-you, LI Yong-ming. Computation tree logic model checking for generalized possibilistic decision processes [J]. Scientia Sinica (Informationis), 2016, 46(11): 1591 – 1607. (in Chinese)
- [20] 郇丽, 沈应兄, 潘海玉. 不确定多值 Kripke 结构的模型检测 [J]. 模糊系统与数学, 2016, 30(5): 60 – 70.
- Li L, Shen Y X, Pan H Y. Nondeterministic multi-valued Kripke structure model checking [J]. Fuzzy Systems and Sets, 2016, 30(5): 60 – 70. (in Chinese)
- [21] Lotfi A Z. Fuzzy sets [J]. Information and Control, 1965, 8(C): 338 – 353.
- [22] Alfred T. A lattice-theoretical fixpoint theorem and its applications [J]. Pacific Journal of Mathematics, 1955, 5(2): 285 – 309.
- [23] Luca D A, Marco F, Thomas A H, et al. Model checking discounted temporal properties [J]. Theoretical Computer Science, 2005, 345(C): 139 – 170.
- [24] Pan H Y, Li Y M, Cao Y Z, et. al. Reachability in fuzzy game graphs [J]. IEEE Transactions on Systems, 2017, 25(4): 972 – 984.

作者简介



范艳焕 女, 1987 年生于河南商丘. 现为陕西师范大学计算机科学学院博士研究生. 主要研究方向为定量模型检测.
E-mail: fan. fan1222@163. com



李永明 (通讯作者) 男, 1966 年生于陕西西安. 现为陕西师范大学教授、博士生导师. 主要研究方向为定量模型检测、非经典计算理论、模糊系统分析、量子逻辑与量子计算.
E-mail: liyongm@snnu. edu. cn