模糊计算树逻辑的符号模型检测*

聂朋展[1]，姜久雷[2†]，马占有[1]
(1. 北方民族大学 计算机科学与工程学院，银川 750021；2. 常熟理工学院 计算机科学与工程学院，江苏 苏州 215500)

摘 要：对含有模糊不确定性信息的系统进行模型检测时，状态空间爆炸问题成为了亟待解决的主要问题。将形式化的系统模型用拟布尔公式表示，用多终端二叉决策图来对拟布尔公式进行存储。对模糊计算树逻辑的不动点语义给出了解释和证明，然后给出模糊计算树逻辑的符号化模型检测算法，最后通过一个实例验证算法的正确性。该算法可有效缓解对模糊模型检测验证时的状态空间爆炸问题，并扩展了模型检测的应用范围。

关键词：模糊计算树逻辑；不动点算法；多终端二叉决策图；符号模型检测
中图分类号：TP305 **文献标志码**：A **文章编号**：1001-3695(2021)08-024-2381-05
doi:10.19734/j.issn.1001-3695.2020.12.0405

Symbolic model checking of fuzzy computation tree logic

Nie Pengzhan[1], Jiang Jiulei[2†], Ma Zhanyou[1]
(1. *College of Computer Science & Engineering, North Minzu University, Yinchuan* 750021, *China*; 2. *School of Computer Science & Engineering, Changshu Institute of Technology, Suzhou Jiangsu* 215500, *China*)

Abstract: In the process of model checking for systems with fuzzy and uncertain information, the problem of state space explosion becomes the main problem. This paper used Quasi-boolean formula to represent the formal model of the system and multi-terminal binary decision graph to store the quasi-boolean formula. The paper provided explanation and proof of the fixed point semantics of fuzzy computation tree logic and gave the symbolic model checking algorithm of fuzzy computation tree logic. Finally, this paper verified the correctness of the algorithm through an example. This algorithm can effectively alleviate the problem of state space explosion in the verification of fuzzy model checking and expands the application scope of model checking.

Key words: fuzzy computation tree logic; fixed point algorithm; multi-terminal binary decision diagrams; symbolic model checking

0 引言

随着计算机系统的快速发展，如何保障其安全性和可靠性成为一个焦点问题，采用形式化方法[1]对系统进行验证和分析，是保障系统安全性和可靠性的一个重要途径。模型检测[2~4]作为一种可以自动验证的形式化技术，被广泛地应用在软硬件设计、通信协议、航空航天、医疗设备、铁路调度等领域。它的基本思想是：用抽象模型对系统进行形式化描述，用时序逻辑表示系统的属性规约，通过遍历有穷的状态空间来判断模型是否系统以及是否具有时许逻辑所表达的属性规约，如果系统满足属性规约，则返回 true，否则返回 false 且给出反例。模型检测的一般方法是求出给定的抽象模型 M 满足时序逻辑公式 Φ 的所有状态集合 sat(Φ)，然后判断初始状态是否在 sat(Φ) 中，如果存在，则模型 M 满足时序逻辑公式 Φ，否则模型 M 不满足时序逻辑公式 Φ。

传统的模型检测技术侧重于对模型进行定性验证，无法对含有不确定性或者不一致性信息[5]的系统进行验证。不确定性信息有多种表现形式，可以分为随机性、可能性、模糊性等。事件的随机性表示该事件发生的频繁程度，主要考虑事件本身，可以采用概率论进行研究；事件的可能性表示该事件实现的难易程度，不仅要考虑事件本身，还要考虑相关事物对它的影响，可以用可能性理论进行研究；事件的模糊性表示人们对客观事物的主观感知，可以用模糊理论进行研究。不确定性信息的来源可能是系统信息不完整或者被删除；不一致性信息指信息来源于不同的用户，从而导致信息内容相互冲突，或者当集成由不同的人开发的组件时导致的不一致性。为了对这样的系统进行验证，研究人员提出了相应的定量模型检测方法。定量模型检测是在传统模型检测的基础上进行扩展，融入了多种数学理论，如格理论、多值逻辑、模糊逻辑、可能性理论、概率论等。由于这些数学理论的加入，可以将定量模型检测大体分为多值模型检测[6,7]、概率模型检测[8~10]、可能性模型检测[11~17]、模糊模型检测[18~20]等。

模型检测中最突出的一个问题是状态空间爆炸[21]，缓解这个问题的方法有多种，比如符号模型检测[22]、有界模型检测、抽象、互模拟等。在符号模型检测中，最重要的计算是通过不动点函数来进行的。同时，通过不动点函数来进行计算的不动点算法也是非符号模型检测常用的算法。由此可以看出，在模型检测中，对不动点语义的证明具有重要意义。2013 年，Li 等人[11]提出了基于可能性测度的 LTL 模型检测。在之后的几年时间里，又研究了基于可能性测度的 CTL 模型检测问题和基于广义可能性测度的 LTL 和 CTL 模型检测，从而对模型检测技术实现了扩展。同年 5 月，文献[19]讨论了基于模糊逻辑的几类 Kripke 结构之间的关系，这为选取合理的模型提供了理论依据。之后在 2015 年，Pan 等人[18]研究了基于模糊逻辑的

收稿日期：2020-12-02；**修回日期**：2021-01-13 **基金项目**：国家自然科学基金资助项目(61762002,61962001)；宁夏自然科学基金资助项目(2018AAC03127)；北方民族大学研究生创新项目(YCX20068)

作者简介：聂朋展(1995-)，男，河南汝州人，硕士，主要研究方向为形式化方法；姜久雷(1972-)，男(通信作者)，山东嘉祥人，教授，硕导，博士，主要研究方向为形式化方法、社交网络分析、智能推荐(wjjl2005@126.com)；马占有(1979-)，男(回族)，宁夏固原人，副教授，硕导，博士，主要研究方向为模型检测.

CTL 模型检测问题,进一步扩展了模型检测技术。2018 年,文献[14]在可能性测度的基础上,对 CTL 模型检测的符号化进行了研究,这对复杂系统的模型检测具有重要的意义。

本文对基于模糊计算树逻辑的模型检测进行符号化的研究,用拟布尔公式表示模糊 Kripke 结构;然后通过多终端二叉决策图来对拟布尔公式进行存储和运算;接下来对模糊计算树逻辑的不动点语义给出解释和证明,并给出其符号模型检测算法;最后通过一个实例来验证。

1 模糊计算树逻辑

本章介绍模糊计算树逻辑(fuzzy computation tree logic, FCTL)。FCTL 的语义是在模糊 Kripke 结构的基础上进行解释的。接下来本文首先给出模糊 Kripke 结构的定义,然后给出 FCTL 的语法和语义。

1.1 模糊 Kripke 结构

定义 1[18] 模糊 Kripke 结构(fuzzy kripke structures, FKS)是一个基于原子命题集 AP 的四元组 $K=(S,s_0,P,L)$。其中:S 表示非空可数的状态集合;s_0 表示初始状态,$s_0\in S$;P 表示模糊迁移函数,$S\times S\to[0,1]$;L 表示模糊标记函数,$S\times 2^{AP}\to[0,1]$。

本文用 $|S|$ 和 $|AP|$ 表示状态集合和命题集合中元素的个数。如果一个 FKS 中的 S 和 AP 的个数是有限的,那么这个 FKS 被称为有限的模糊 Kripke 结构。对于两个状态 s 和 s',$P(s,s')$ 表示从状态 s 迁移到状态 s'的可能性。FKS 中的一条路径 π 是一个状态序列,如果路径中状态的个数为无限时,该路径是一条无穷路径,如 $\pi=s_0s_1s_2\cdots$;如果路径中状态的个数是有限时,该路径是一条有穷路径,如 $\pi=s_0s_1s_2\cdots s_n$。本文用 paths(s) 表示在 K 中,所有以状态 s 开始的无穷路径的集合,paths(K) 表示在 K 中,所有无穷路径的集合。

1.2 模糊计算树逻辑

FCTL 的语法包括状态公式和路径公式两部分。状态公式表示了特定状态的性质,路径公式表示了特定路径的性质。FCTL 和 CTL 在定义上是十分类似的,它们的区别在于语义的不同,FCTL 的语义是基于模糊逻辑,CTL 的语义是基于二值逻辑。

定义 2[18] FCTL 的状态公式按如下递归定义:$true$,$false$,a 是状态公式;$\neg\varphi$,$\varphi_1\vee\varphi_2$,$\varphi_1\wedge\varphi_2$,$\varphi_1\to\varphi_2$ 是状态公式;$\exists\Psi$,$\forall\Psi$ 是状态公式。其中,$a\in AP$ 是原子命题,Ψ 是路径公式。FCTL 的路径公式按如下递归定义:$\bigcirc\varphi$ 或 $\varphi_1\cup\varphi_2$,其中 φ,φ_1,φ_2 是状态公式。

定义 3[18] 设 K 是一个 FKS,$s\in S$,$\pi=s_0s_1s_2s_3\cdots$,$a\in AP$,φ,φ_1,φ_2 是状态公式,Ψ 是路径公式,$\|\varphi\|(s)$ 表示状态 s 到[0,1]的一个映射,即 $\|\varphi\|(s):s\to[0,1]$。FCTL 状态语义定义如下:

$$\|true\|(s)=1\qquad \|false\|(s)=0$$
$$\|a\|(s)=L(s)(a)\qquad \|\neg\varphi\|(s)=1-\|\varphi\|(s)$$
$$\|\varphi_1\wedge\varphi_2\|(s)=\|\varphi_1\|(s)\wedge\|\varphi_2\|(s)$$
$$\|\varphi_1\vee\varphi_2\|(s)=\|\varphi_1\|(s)\vee\|\varphi_2\|(s)$$
$$\|\varphi_1\to\varphi_2\|(s)=\|\varphi_1\|(s)\to\|\varphi_2\|(s)$$
$$\|\exists\Psi\|(s)=\bigvee_{\pi\in\text{paths}(s)}\|\Psi\|(\pi)$$
$$\|\forall\Psi\|(s)=\bigwedge_{\pi\in\text{paths}(s)}\|\Psi\|(\pi)$$

FCTL 的路径语义定义如下:

$$\|\bigcirc\varphi\|(\pi)=P(s_0,s_1)\wedge\|\varphi\|(s_1)$$
$$\|\varphi_1\cup\varphi_2\|(\pi)=\bigvee_{i\geqslant 0}(\wedge(\|\varphi_1\|(s_j)\wedge P(s_j,s_{j+1}))\wedge\|\varphi_2\|(s_i))$$

其中:对于公式 $\|\varphi_1\to\varphi_2\|(s)$,可以用如下等价公式求解:

$$\|\varphi_1\to\varphi_2\|(s)=\|\neg\varphi_1\|(s)\vee\|\varphi_2\|(s)\tag{1}$$

对于路径运算符◇,□运算符的求解可以用式(2) ~(5)转换求解。

$$\exists\Diamond\varphi=\exists(true\cup\varphi)\tag{2}$$
$$\forall\Diamond\varphi=\forall(true\cup\varphi)\tag{3}$$
$$\exists\Box\varphi=\neg\forall\Diamond\neg\varphi\tag{4}$$
$$\forall\Box\varphi=\neg\exists\Diamond\neg\varphi\tag{5}$$

在 FCTL 路径语义中,运算符○、∪、◇、□表示的含义分别是“下一个”“直到”“存在”“总是”。其中,$\|\bigcirc\varphi\|(\pi)$ 表示在一条以 s_0 开始的路径 π 中,下一个状态满足状态公式 φ 的值;$\|\varphi_1\cup\varphi_2\|(\pi)$ 表示在一条以 s_0 开始的路径 π 中,状态 s_i 满足 φ_2 且状态 s_j 满足 φ_1 的值,其中 $0\leqslant j<i$;$\|\Diamond\varphi\|(\pi)$ 表示在一条以 s_0 开始的路径 π 中,存在状态满足状态公式 φ 的值;$\|\Box\varphi\|(\pi)$ 表示在一条以 s_0 开始的路径 π 中,路径中每一个状态都满足状态公式 φ 的值。

2 符号化模型

根据模糊 Kripke 结构的定义可知,要对其进行符号化处理,就是要解决对其状态、迁移关系和标记函数的符号化。分析经典的符号模型检测,本文可以将这些元素用多终端二叉决策图表示,从而实现对模糊 Kripke 结构的符号化。

2.1 多终端二叉决策图

多终端二叉决策图(multi-terminal binary decision diagram, MTBDD)是一个类似与二叉决策图(binary decision diagram, BDD)的图结构,它可以用来表示 $f:B^n\to D$,即将 n 个布尔变量映射到实数集 D 上。在这里可以将实数集 D 限定在[0,1],这样就可以用来描述本文的形式化模型。

MTBDD 和 BDD 一样,都是只有一个根节点的有向无环图。它的节点集包含终端节点和非终端节点两类。每一个非终端节点 v 都有两个子节点,分别用 low(v) 和 high(v) 来表示,终端节点则用[0,1]的数值来表示。MTBDD 和 BDD 的不同之处在于,MTBDD 的叶子节点取值是[0,1],而 BDD 的取值是{0,1}。

假设 $a_1,a_2,a_3,\cdots,a_m\in[0,1]$ 是函数 f 的可能取值,将空间 B^n 划分为 m 个集合 $\{S_1,S_2,S_3,\cdots,S_m\}$,其中 $S_i=\{\boldsymbol{x}\mid f(\boldsymbol{x})=n_i\}$。本文令 f_i 作为 S_i 的特征函数,那么就可以得到 f 的规范式 $f=\sum_{i=1}^{m}f_i(\boldsymbol{x})\,n_i$,函数 f 被称为拟布尔函数。这个公式可以用来表示将取值于[0,1]的数作为叶子节点的二叉决策图,本文把这类图称为多终端二叉决策图。MTBDD 之间的算术运算如下:

$$h(\boldsymbol{x})=f(\boldsymbol{x})\,\Theta\,g(\boldsymbol{x})=\sum_{i=1}^{m}f_i(\boldsymbol{x})\,n_i\,\Theta\sum_{j=1}^{m'}g_j(\boldsymbol{x})\,n'_j=$$
$$\sum_{i=1}^{m}\sum_{j=1}^{m'}f_i(\boldsymbol{x})\,g_j(\boldsymbol{x})\,(n_i\Theta n'_j)=\sum_{k=1}^{m''}\bigvee_{n_i\Theta n'_j=n''_k}f_i(\boldsymbol{x})\,g_j(\boldsymbol{x})\,n''_k\tag{6}$$

2.2 状态的符号化

模型中的状态采用布尔向量表示,向量中变量个数由状态的个数决定。用 $\|V\|$ 表示变量的个数,$\|S\|$ 表示状态的个数,则 $\|V\|=\lceil\log_2\|S\|\rceil$。假设有 n 个变量 $x_1,x_2,x_3,\cdots,x_n$,本文用布尔表达式 $f(x_1,x_2,x_3,\cdots,x_n)$ 来表示状态。如图 1 所示,其中的状态可以分别用布尔表达式表示为

$$f_{s_0}=\bar{x}_1\bar{x}_2,\quad f_{s_1}=\bar{x}_1x_2,\quad f_{s_2}=x_1\bar{x}_2;\quad f_{s_3}=x_1x_2$$

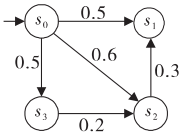

图 1 模糊 Kripke 结构

Fig. 1 Fuzzy Kripke structure

2.3　迁移关系的符号化

由迁移关系的定义 $P: S\times S\rightarrow[0,1]$ 可知,本文需要用迁移关系的前驱状态和后继状态一起来表示。本文用向量 $\boldsymbol{X}=(x_1,x_2,x_3,\cdots,x_n)$ 表示前驱状态,向量 $\boldsymbol{Y}=(x_1,x_2,x_3,\cdots,x_n)$ 表示后继状态。每一条迁移关系之间用“+”连接。图 1 的迁移关系可以表示为

$$f_P=0.5\bar{x}_1\bar{x}_2\bar{y}_1y_2+0.5\bar{x}_1\bar{x}_2y_1y_2+0.6\bar{x}_1\bar{x}_2y_1\bar{y}_2+0.2x_1x_2y_1\bar{y}_2+0.3x_1\bar{x}_2\bar{y}_1y_2$$

2.4　标记函数的符号化

对标记函数进行符号化,就是为每一个原子命题建立 MTBDD。在 MTBDD 中,非终端节点表示变量,终端节点赋值为命题在该布尔向量对应状态上的数值。

3　FCTL 不动点证明

通过不动点算法对 FTCL 模型检测进行计算时,需要解决状态公式 $\|\exists\varphi_1\cup\varphi_2\|(s)$ 和 $\|\forall\varphi_1\cup\varphi_2\|(s)$ 的求解问题。本文通过构造这两个状态公式的不动点函数,使用归纳法证明它们是不动点函数的最小不动点,从而解决该问题。在对 FCTL 模型检测进行符号化求解的过程中,也需要通过不动点函数来计算这两个状态公式,由此说明对这两个状态公式的不动点语义的证明是十分必要的。

设 K 是一个 FKS, $\|\exists\varphi_1\cup\varphi_2\|(s)$ 和 $\|\forall\varphi_1\cup\varphi_2\|(s)$ 是 K 上的状态公式,如果函数 $f:[0,1]^{|S|}\rightarrow[0,1]^{|S|}$,有

$$f_1(Z)=\|\varphi_2\|\vee(\|\varphi_1\|\wedge\|\exists\bigcirc Z\|)$$
$$f_2(Z)=\|\varphi_2\|\vee(\|\varphi_1\|\wedge\|\forall\bigcirc Z\|)$$

则函数 $f_1(Z)$ 和 $f_2(Z)$ 有最小不动点,且最小不动点分别是 $\|\exists\varphi_1\cup\varphi_2\|(s)$ 和 $\|\forall\varphi_1\cup\varphi_2\|(s)$,记为

$$\|\exists\varphi_1\cup\varphi_2\|(s)=\mu Z.(\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\|\exists\bigcirc Z\|(s)))$$
$$\|\forall\varphi_1\cup\varphi_2\|(s)=\mu Z.(\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\|\forall\bigcirc Z\|(s)))$$

其中: $\mu Z.f(Z)$ 表示函数 f 的最小不动点。

证明　设 $\boldsymbol{Z}_0=\{0,0,0,\cdots,0\}^{|S|}$ 是元素全为 0 的最小向量, $Z_{i+1}=f(Z_i)$。因为函数 f 是单调的,所以当 $Z'\geqslant Z''$ 时,有 $f(Z')\geqslant f(Z'')$,所以有上升链 $Z_0\leqslant Z_1\leqslant Z_2\leqslant\cdots\leqslant Z_i\leqslant Z_{i+1}\leqslant\cdots$。

因为状态公式 φ 的象集是有穷的,函数 f 在取大、取小运算的过程中是不会产生新元素的,所以 Z_i 中的元素个数也是有穷的。因此,存在 K 值使得 $Z_{k+1}=Z_k$ 成立,即 $f(Z_k)=Z_k$。此外,如果 Z 是 f 的一个不动点,则有 $Z\geqslant Z_0$。因为 f 是单调的,所以 $Z=f(Z)\geqslant Z_1$,通过归纳可得 $Z\geqslant Z_k$。综上所述, Z_k 是函数 f 的最小不动点。

由 $\|\exists\varphi_1\cup\varphi_2\|(s)$ 的语义可知:

$$\|\exists\varphi_1\cup\varphi_2\|(s)=\bigvee_{\pi\in\text{paths}(s)}\bigvee_{i\geqslant0}(\bigwedge_{0\leqslant j<i}(\|\varphi_1\|(s_j)\wedge P(s_j,s_{j+1}))\wedge\|\varphi_2\|(s_i))=\bigvee_{\pi\in\text{paths}(s)}(\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\bigvee_{j>0}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s)))=\|\varphi_2\|(s)\vee\bigvee_{\pi\in\text{paths}(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s))$$

令 $\|\exists\varphi_1\cup\varphi_2\|(s)=A_1(s)$,首先证明 $A_1(s)$ 是 $f_1(Z)$ 的不动点。

$$f_1(A_1(s))=\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\|\exists\bigcirc A_1\|(s))=\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\bigvee_{s_1\in S}P(s,s_1)\wedge\|A_1\|(s_1))=\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\bigvee_{s_1\in S}P(s,s_1)\wedge\bigvee_{\pi\in\text{paths}(s_1)}(\|\varphi_2\|(s_1)\vee(\|\varphi_1\|(s_1)\wedge\bigvee_{j>1}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))))=\|\varphi_2\|(s)\vee(\bigvee_{s_1\in S}\|\varphi_1\|(s)\wedge P(s,s_1)\wedge\bigvee_{\pi\in\text{paths}(s_1)}(\|\varphi_2\|(s_1)\vee(\|\varphi_1\|(s_1)\wedge\bigvee_{j>1}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))))=\|\varphi_2\|(s)\vee\bigvee_{s_1\in S}\|\varphi_1\|(s)\wedge P(s,s_1)\wedge\|\varphi_2\|(s_1)\vee\bigvee_{\pi\in\text{paths}(s_1)}(\|\varphi_1\|(s_1)\wedge\bigvee_{j>1}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j)))=\|\varphi_2\|(s)\vee\bigvee_{\pi\in\text{paths}(s)}(\|\varphi_1\|(s)\wedge(\bigvee_{j>0}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j)))=A_1(s)$$

接下来通过归纳法证明 $A_1(s)$ 是 $f_1(Z)$ 的最小不动点。

令 Z 是 $f_1(Z)$ 的不动点,则:

$$Z(s)=\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\|\exists\bigcirc Z\|(s))$$

再令

$$A_j(s)=\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j)$$

其中: $j\geqslant0$,则 $A_1(s)=\bigvee_{\pi\in\text{paths}(s)}\bigvee_{j=0}A_j(s)$。当 $j=0$ 时, $Z(s)=\|\varphi_2\|(s)\vee\|\varphi_1\|(s)$, $A_1(s)=\|\varphi_2\|(s)$,此时 $Z(s)\geqslant A_1(s)$。

当 $j=n$ 时,

$$Z(s)=\|\varphi_2\|(s)\vee\bigvee_{\pi\in\text{paths}(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}^{n}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))\vee\bigvee_{\pi\in\text{paths}(s)}(\bigwedge_{i=0}^{n-1}P(s_i,s_{i+1})\wedge\|\varphi_1\|(s_{i+1})\wedge P(s_n,s_{n+1})\wedge Z(s_{n+1}))\geqslant\|\varphi_2\|(s)\vee\bigvee_{\pi\in\text{paths}(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}^{n}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))=A_1(s)$$

当 $j=n+1$ 时,

$$Z(s)=\|\varphi_2\|(s)\vee\bigvee_{\pi\in\text{paths}(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}^{n+1}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{k-1},s_k)\wedge\|\varphi_2\|(s_j))\vee\bigvee_{\pi\in\text{paths}(s)}(\bigwedge_{i=0}^{n}P(s_i,s_{i+1})\wedge\|\varphi_1\|(s_{i+1})\wedge P(s_{n+1},s_{n+2})\wedge Z(s_{n+2}))\geqslant\|\varphi_2\|(s)\vee\bigvee_{\pi\in\text{paths}(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}^{n}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))=A_1(s)$$

通过归纳法最终证得 $Z(s)\geqslant A_1(s)$,所以 $A_1(s)$ 是 $f_1(Z)$ 的最小不动点。

由 $\|\forall\varphi_1\cup\varphi_2\|(s)$ 的语义可知:

$$\|\forall\varphi_1\cup\varphi_2\|(s)=\bigwedge_{\pi\in\text{paths}(s)}\bigvee_{i\geqslant0}(\bigwedge_{0\leqslant j<i}(\|\varphi_1\|(s_j)\wedge P(s_j,s_{j+1}))\wedge\|\varphi_2\|(s_i))=\bigwedge_{\pi\in\text{paths}(s)}(\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\bigvee_{j>0}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s)))=\|\varphi_2\|(s)\vee\bigwedge_{\pi\in\text{paths}(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0k<j}\bigwedge P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))$$

令 $\|\forall\varphi_1\cup\varphi_2\|(s)=A_2(s)$,首先证明 $A_2(s)$ 是 $f_2(Z)$ 的不动点。

$$f_2(A_2(s))=\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\|\forall\bigcirc A_2\|(s))=\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\bigwedge_{s_1\in S}P(s,s_1)\wedge\|A_2\|(s_1))=\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\bigwedge_{s_1\in S}P(s,s_1)\wedge(\|\varphi_2\|(s_1)\vee\bigwedge_{\pi\in\text{paths}(s_1)}(\|\varphi_1\|(s_1)\wedge\bigvee_{j>1k<j}\bigwedge P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))))=\|\varphi_2\|(s)\vee(\bigwedge_{s_1\in S}\|\varphi_1\|(s)\wedge P(s,s_1)\wedge(\|\varphi_2\|(s_1)\vee\bigwedge_{\pi\in\text{paths}(s_1)}(\|\varphi_1\|(s_1)\wedge\bigvee_{j>1k<j}\bigwedge P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))))=\|\varphi_2\|(s)\vee(\bigwedge_{s_1\in S}\|\varphi_1\|(s)\wedge P(s,s_1)\wedge\|\varphi_2\|(s_1)\vee\|\varphi_1\|(s)\wedge P(s,s_1)\wedge\bigwedge_{\pi\in\text{paths}(s_1)}(\|\varphi_1\|(s_1)\wedge\bigvee_{j>1}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j)))=\|\varphi_2\|(s)\vee\bigwedge_{\pi\in\text{paths}(s)}(\|\varphi_1\|(s)\wedge(\bigvee_{j>0}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j)))=A_2(s)$$

此时 $f_2(A_2(s))=A_2(s)$,证得 $A_2(s)$ 是 $f_2(Z)$ 的不动点。

接下来通过归纳法证明 $A_2(s)$ 是 $f_2(Z)$ 的最小不动点。

令 Z 是 $f_2(Z)$ 的不动点,即

$$Z(s)=\|\varphi_2\|(s)\vee(\|\varphi_1\|(s)\wedge\|\forall\bigcirc Z\|(s))$$

再令

$$A_j(s)=\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j)$$

其中: $j\geqslant0$,则 $A_2(s)=\bigwedge_{\pi\in\text{paths}(s)}\bigvee_{j=0}A_j(s)$。

当 $j=0$ 时，$Z(s)=\|\varphi_2\|(s)\vee\|\varphi_1\|(s)$，$A_2(s)=\|\varphi_2\|(s)$，此时 $Z(s)\geqslant A_2(s)$。

当 $j=n$ 时，

$$Z(s)=\|\varphi_2\|(s)\vee\bigwedge_{\pi\in\mathrm{path}(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}^{n}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))\vee\bigwedge_{\pi\in path(s)}(\bigwedge_{i=0}^{n-1}P(s_i,s_{i+1})\wedge\|\varphi_1\|(s_{i+1})\wedge P(s_n,s_{n+1})\wedge Z(s_{n+1}))\geqslant\|\varphi_2\|(s)\vee\bigwedge_{\pi\in path(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}^{n}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))=A_2(s)$$

当 $j=n+1$ 时，

$$Z(s)=\|\varphi_2\|(s)\vee\bigwedge_{\pi\in path(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}^{n+1}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))\vee\bigwedge_{\pi\in\mathrm{path}(s)}(\bigwedge_{i=0}^{n}P(s_i,s_{i+1})\wedge\|\varphi_1\|(s_{i+1})\wedge P(s_{n+1},s_{n+2})\wedge Z(s_{n+2}))\geqslant\|\varphi_2\|(s)\vee\bigwedge_{\pi\in\mathrm{path}(s)}(\|\varphi_1\|(s)\wedge\bigvee_{j>0}^{n+1}\bigwedge_{k<j}P(s_{k-1},s_k)\wedge\|\varphi_1\|(s_k)\wedge P(s_{j-1},s_j)\wedge\|\varphi_2\|(s_j))=A_2(s)$$

通过归纳法最终证得 $Z(s)\geqslant A_2(s)$，所以 $A_2(s)$ 是 $f_2(Z)$ 的最小不动点。

不动点函数求解的过程，就是求从状态 s 出发的长度为 n 的有穷路径 π 满足公式 φ 的最小模糊隶属度。由于状态空间的状态个数 $|S|$ 是有穷的，所以对于任意长度大于 $|S|+1$ 的路径 π'，它的值都大于 π 满足 φ 的值，因此不动点的迭代计算在 $|S|+1$ 次之后收敛。不动点的计算只涉及到取大、取小运算，所以一次迭代的时间复杂度是 $O(|S|^2)$，从而整个不动点计算的时间复杂度为 $O(|S|^3)$。

4 FCTL 符号模型检测算法

通过以上分析，结合经典的符号模型检测方法，本文在本章给出 FCTL 符号模型检测的计算方法。由 FCTL 的语义可知，要解决 FCTL 符号模型检测的计算问题，主要是求解公式 $\exists\bigcirc\varphi$、$\forall\bigcirc\varphi$、$\exists(\varphi_1\cup\varphi_2)$、$\forall(\varphi_1\cup\varphi_2)$ 的计算方法。

$\exists\bigcirc\varphi$ 的计算公式为

$$f_{\exists\bigcirc\varphi}(X)=\exists(f_P(X,Y),f_\varphi(Y))=\vee(f_P(X,Y)\wedge f_\varphi(Y)) \tag{7}$$

$\forall\bigcirc\varphi$ 的计算公式为

$$f_{\forall\bigcirc\varphi}(X)=\forall(f_P(X,Y)\rightarrow f_\varphi(Y))=\wedge(\bar{f}_P(X,Y)\vee f_\varphi(Y)) \tag{8}$$

$\exists(\varphi_1\cup\varphi_2)$ 的计算方式为

$$f_{\exists(\varphi_1\cup\varphi_2)}(X)=f_{\mu Z.(\varphi_2\vee(\varphi_1\wedge\exists\bigcirc Z))}(X) \tag{9}$$

$\forall(\varphi_1\cup\varphi_2)$ 的计算方式为

$$f_{\forall(\varphi_1\cup\varphi_2)}(X)=f_{\mu Z.(\varphi_2\vee(\varphi_1\wedge\forall\bigcirc Z))}(X) \tag{10}$$

算法 1 不动点算法

输入：不动点函数 f。
输出：不动点 x。
procedure Fix_poin(x,f)
  $x'=f(x)$
  while $x=x'$ do
    $x=x'$
    $x'=f(x)$
  end while
  return x
end procedure

算法 2 最小不动点算法

输入：$f(Z)$。
输出：$f(Z)$ 的最小不动点。
procedure Lfp($f(Z)$)
  $Q=false$
  $Q'=f(Q)$
  while $Q=Q'$ do
    $Q=Q'$
    $Q'=f(Q')$
  end while
  return Q
end procedure

本文用 $\bar{s}$ 来表示状态的编码，FCTL 符号模型检测算法如下：

算法 3 FCTL 模型检测算法

输入：FKS K 和 FCTL 公式 φ。
输出：K 中的任意状态 s 满足 φ 的程度值。
procedure FCTLCheck($\|\varphi\|(\bar{s})$)
  case: φ
    true return 1
    false return 0
    a return $f_L(\bar{s})$
    $\neg\varphi$ return $1-\|\varphi\|(\bar{s})$
    $\varphi_1\vee\varphi_2$ return $\|\varphi_1\|(\bar{s})\vee\|\varphi_2\|(\bar{s})$
    $\varphi_1\wedge\varphi_2$ return $\|\varphi_1\|(\bar{s})\wedge\|\varphi_2\|(\bar{s})$
    $\|\exists\bigcirc\varphi\|(\bar{s})$ return $\vee(f_P(X,Y)\wedge f_\varphi(Y))$
    $\|\forall\bigcirc\varphi\|(\bar{s})$ return $\wedge(\bar{f}_P(X,Y)\vee f_\varphi(Y))$
    $\|\exists(\varphi_1\cup\varphi_2)\|(\bar{s})$ return $\mathrm{Lfp}(f_1(Z))$
    $\|\forall(\varphi_1\cup\varphi_2)\|(\bar{s})$ return $\mathrm{Lfp}(f_2(Z))$
  end case
end procedure

5 实例

本章将通过一个类似于文献[18]的实例来验证本文算法。假设在一个地区出现了一种新型传染病，医生们对这种疾病没有完全的了解。根据他们的经验，医生认为抗菌素盘尼西林可能对治疗这种疾病有效，可以用模糊 Kripke 结构来描述这个模型。医生根据自己的经验，将病人的身体状况设置为 poor、fair 和 excellent 三种状态。经过医生的治疗，病人身体状况的变化情况可以采用模糊 Kripke 结构来描述。

根据医生设置的三种状态，将原子命题集 AP 设为{poor, fair, excellent}。分别用 P、F、E 来简写这三种身体状态，则原子命题集 $AP=\{P,F,E\}$。如图 2 所示，状态集合 $S=\{s_0,s_1,s_2\}$，根据医生的经验可以得出标记函数如下：

$$L(s_0)(P)=0.9,L(s_0)(F)=0.3,L(s_0)(E)=0;$$
$$L(s_1)(P)=0.3,L(s_1)(F)=0.8,L(s_1)(E)=0.6;$$
$$L(s_2)(P)=0,L(s_2)(F)=0.6,L(s_2)(E)=1$$

其中：$L(s_0)(P)=0.9$ 的含义是在状态 s_0 下，病人身体状态为 poor 的模糊度为 0.9。迁移关系 $P(s_0,s_1)=0.3$ 表示病人从状态 s_0 转变到 s_1 的模糊度为 0.3。

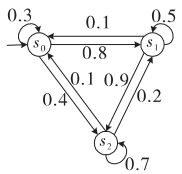

图 2 病人治疗过程的模糊 Kripke 结构

Fig. 2 Fuzzy Kripke structure of patient treatment process

根据算法，计算下面的属性值：

$$\|\exists\bigcirc E\|(s_0)\text{ 和 }\|\exists(true\cup E)\|(s_0)$$

其中：$\|\exists\bigcirc E\|(s_0)$ 表示在初始状态 s_0 下，经过一次治疗之后，病人的身体状态变为 excellent 的最大可能性；$\|\exists(true\cup E)\|(s_0)$ 表示在初始状态 s_0 下，经过治疗之后，病人的身体状态最终变为 excellent 的最大可能性。

首先，对状态进行符号化。设两个变量 x_1 和 x_2，状态 s_0、s_1、s_2 可以依次编码为(0,0)、(0,1)、(1,0)，则状态可符号化表示为

$$f_{s_0}=\bar{x}_1\bar{x}_2,f_{s_1}=\bar{x}_1x_2,f_{s_2}=x_1\bar{x}_2$$

状态集合 S 表示为

$$f_S = \bar{x}_1\bar{x}_2 + \bar{x}_1x_2 + x_1\bar{x}_2$$

接下来对迁移关系符号化,结果为

$$f_P(X,Y) = 0.3\bar{x}_1\bar{x}_2\bar{y}_1\bar{y}_2 + 0.8\bar{x}_1\bar{x}_2\bar{y}_1y_2 + 0.4\bar{x}_1\bar{x}_2y_1\bar{y}_2 + 0.1\bar{x}_1x_2\bar{y}_1\bar{y}_2 + 0.5\bar{x}_1x_2\bar{y}_1y_2 + 0.9\bar{x}_1x_2y_1\bar{y}_2 + 0.1x_1\bar{x}_2\bar{y}_1\bar{y}_2 + 0.2x_1\bar{x}_2\bar{y}_1y_2 + 0.7x_1\bar{x}_2y_1\bar{y}_2$$

然后对标记函数进行符号化,结果为

$$f_P = 0.9\bar{x}_1\bar{x}_2 + 0.3\bar{x}_1x_2 + 0x_1\bar{x}_2$$
$$f_F = 0.3\bar{x}_1\bar{x}_2 + 0.8\bar{x}_1x_2 + 0.6x_1\bar{x}_2$$
$$f_E = 0\bar{x}_1\bar{x}_2 + 0.6\bar{x}_1x_2 + 1x_1\bar{x}_2$$

最后计算结果:

$$f_{\exists\circ E}(X) = 0.6\bar{x}_1\bar{x}_2 + 0.9\bar{x}_1x_2 + 0.7x_1\bar{x}_2$$

将三个状态的编码代入求得的拟布尔函数:

$$f_{\exists\circ E}(0,0) = 0.6,\ f_{\exists\circ E}(0,1) = 0.9,\ f_{\exists\circ E}(1,0) = 0.7$$

结果说明,在初始状态分别为 s_0、s_1、s_2 的情况下,病人接受一次治疗之后,身体状态变为 excellent 的最大可能性值分别为 0.6、0.9、0.7。

$$f_{\exists true\cup E}(X) = 0\bar{x}_1\bar{x}_2 + 0.6\bar{x}_1x_2 + x_1\bar{x}_2$$

将三个状态的编码代入求得的拟布尔函数,得

$$f_{\exists true\cup E}(0,0) = 0,\ f_{\exists true\cup E}(0,1) = 0.6,\ f_{\exists true\cup E}(1,0) = 1$$

结果说明,在初始状态分别为 s_0、s_1、s_2 的情况下,病人接受治疗之后,身体状态最终变为 excellent 的最大可能性值分别为 0、0.6、1。

6　结束语

本文将符号模型检测和基于模糊计算树逻辑的模型检测方法相结合,有效地缓解了在对含有模糊不确定性信息的系统进行模型检测时遇到的状态空间爆炸的问题。此外也给出了 FCTL 的不动点语义,并对不动点语义进行解释和证明。本文在模糊模型检测的基础上进行了进一步拓展,将符号化技术应用在模糊模型检测中。采用符号化技术可以扩大验证系统的规模,对于一些异步系统来说,符号化技术并没有突显出它的优势,这是它的一个局限性。未来的工作可以参考经典的符号模型检测器来设计实现模糊符号模型检测器,结合实际的应用来对其进行改进。

参考文献:

[1] 张广泉. 形式化方法导论[M]. 北京: 清华大学出版社,2015. (Zhang Guangquan. Introduction to formal methods [M]. Beijing: Tsinghua University Press,2015.)

[2] Clarke E M,Grumberg O,Peled D. 模型检测[M]. 吴尽昭,何安平,高新岩,译. 北京: 电子工业出版社,2018. (Clarke E M,Grumberg O,Peled D. Model checking[M]. Wu Jinzhao,He Anping,Gao Xinyan,Trans. Beijing: Publishing House of Electronics Industry,2018.)

[3] Baier C,Katoen J P. Principles of model checking[M]. Massachusetts: MIT Press,2008.

[4] Lam V S W. Handbook of model checking[J]. Computing Reviews,2019,60(8):308.

[5] 杨风暴,吉琳娜,王肖霞. 可能性理论及应用[M]. 北京: 科学出版社,2019. (Yang Fengbao,Ji Linna,Wang Xiaoxia. Possibility theory and its application[M]. Beijing: Science Press,2019.)

[6] Chechik M,Devereux B,Easterbrook S,*et al*. Multi-valued symbolic model-checking[J]. ACM Trans on Software Engineering and Methodology,2003,12(4):371-408.

[7] 袁申,魏杰林,李永明. 具有多值决策过程的广义可能性计算树逻辑模型检测[J]. 计算机工程与科学,2019,41(1):88-97. (Yuan Shen,Wei Jielin,Li Yongming. Logic model checking of generalized possibility computation tree with multi valued decision process[J]. Computer Engineering and Science,2019,41(1):88-97.)

[8] Field T,Harrison P G,Bradley J,*et al*. PRISM: probabilistic symbolic model checker[C]//Proc of International Conference on Computer Performance Evaluation. London: Springer-Verlag,2002:200-204.

[9] 周从华. 随机模型检测理论与应用[M]. 北京: 科学出版社,2014. (Zhou Conghua. Theory and application of stochastic model checking[M]. Beijing: Science Press,2014.)

[10] 刘阳,李宣东,马艳,等. 随机模型检验研究[J]. 计算机学报,2015,38(11):2145-2162. (Liu Yang,Li Xuandong,Ma Yan,*et al*. Study on stochastic model checking[J]. Chinese Journal of Computer,2015,38(11):2145-2162.)

[11] Li Yongming,Li Lijun. Model checking of linear-time properties based on possibility measure[J]. IEEE Trans on Fuzzy Systems,2013,21(5):842-854.

[12] Li Yongming. Quantitative model checking of linear-time properties based on generalized possibility measures[J]. Fuzzy Sets and Systems,2017,320:17-39.

[13] Li Yongming,Ma Zhanyou. Quantitative computation tree logic model checking based on generalized possibility measures[J]. IEEE Trans on Fuzzy Systems,2015,23(6):2034-2047.

[14] 雷丽晖,郭越,张延波. 可能性测度下的 CTL 符号化模型检测[J]. 计算机工程与科学,2018,40(11):106-112. (Lei Lihui,Guo Yue,Zhang Yanbo. Symbolic CTL model checking based on possibility measure[J]. Computer Engineering and Science,2018,40(11):106-112.)

[15] 马占有. 基于决策过程的广义可能性时态逻辑模型检测[D]. 西安: 陕西师范大学,2017. (Ma Zhanyou. Generalized possibility temporal logic model checking based on decision process[D]. Xi'an: Shaanxi Normal University,2017.)

[16] 邓楠轶,张兴兴,李永明. 广义可能性计算树逻辑的不动点语义[J]. 陕西师范大学学报: 自然科学版,2015,43(4):22-27. (Deng Nanyi,Zhang Xingxing,Li Yongming. Fixed point semantics of generalized possibility computation tree logic [J]. Journal of Shaanxi Normal University: Natural Science Edition,2015,43(4):22-27.)

[17] Jiang Jiulei,Zhang Panqing,Ma Zhanyou. The μ-calculus model-checking algorithm for generalized possibilistic decision process[J]. Applied Sciences,2020,10(7):article No. 2594.

[18] Pan Haiyu,Li Yongming,Cao Yongzhi,*et al*. Model checking fuzzy computation tree logic[J]. Fuzzy Sets and Systems,2015,262:60-77.

[19] 范艳焕,李永明,潘海玉. 不确定型模糊 Kripke 结构的计算树逻辑模型检测[J]. 电子学报,2018,46(1):152-159. (Fan Yanhuan,Li Yongming,Pan Haiyu. Computation tree logic model checking of uncertain fuzzy Kripke structure[J]. Acta Electronica Sinica,2018,46(1):152-159.)

[20] Nie Pengzhan,Jiang Jiulei,Ma Zhanyou. CTL symbolic model checking based on fuzzy logic[C]//Proc of IEEE International Conference on Dependable,Autonomic and Secure Computing,International Conference on Pervasive Intelligence and Computing,International Conference on Cloud and Big Data Computing,International Conference on Cyber Science and Technology Congress. Piscataway,NJ: IEEE Press,2020:380-385.

[21] 侯刚,周宽久,勇嘉伟,等. 模型检测中状态爆炸问题研究综述[J]. 计算机科学,2013,40(Z6):77-86. (Hou Gang,Zhou Kuanjiu,Yong Jiawei,*et al*. Review of state explosion in model checking[J]. Computer Science,2013,40(Z6):77-86.)

[22] Clarke E,Mcmillan K,Campos S,*et al*. Symbolic model checking[M]//Alur R,Henzinger T A. Computer Aided Verification. Berlin: Springer,2005:419-422.

[23] Zadeh L A. Fuzzy sets[J]. Information and Control,1965,8(3):338-353.

[24] Zadeh L A. Fuzzy sets as a basis for a theory of possibility[J]. Fuzzy Sets and Systems,1978,1(1):3-28.

[25] 李永明,李平. 模糊计算理论[M]. 北京: 科学出版社,2016. (Li Yongming,Li Ping. Fuzzy computing theory[M]. Beijing: Science Press,2016.)