基于不动点逻辑的混成系统性能评价语言

李 晴[1],曹子宁[1,2,3],黄 涛[1]
(1. 南京航空航天大学 计算机科学与技术学院,江苏 南京 211106;
2. 光电控制技术重点实验室,河南 洛阳 471023;
3. 软件新技术与产业化协同创新中心,江苏 南京 210023)

摘 要:混成系统是一类连续与离散行为紧密结合的复杂动态系统,目前广泛地应用在医疗和国防等安全关键领域。安全关键系统要求自身具有较高的安全性与可靠性,以减少系统故障引起的生命和财产方面的灾难性后果。而形式化方法是保障系统可靠性的一种常用方法,其中模型检测应用最为广泛。由于模型检测只能给出系统是否满足某个性质的真或假逻辑值,通过将其与性能评价相结合,以描述系统与实值计算相关的一些性质。现有的性能评价语言 CTML 可以描述系统与概率和平均期望相关的性质,μ 演算则可以通过最小和最大不动点运算符描述迁移系统的某些性质。在基于 μ 演算的模型检测和 CTML 的基础上,提出一种面向混成系统的基于不动点的新的性能评价语言 MLBoF 以及 MLBoF 公式的性能评价算法。针对 CTML 的子逻辑,给出与其语义等价的 MLBoF 公式表示以及二者等价的证明过程。通过飞机起飞系统实例说明,提出的性能评价语言 MLBoF 不仅将基于 μ 演算的模型检测结果从{0,1}扩展到实数区间,具有验证系统概率等实值性质的能力;而且通过扩展经典的计算不动点的改进算法,保证了 MLBoF 的性能评价算法的效率。

关键词:混成系统;不动点;CTML;μ 演算;性能评价
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2022)12-0069-05
doi:10.3969/j.issn.1673-629X.2022.12.011

Hybrid System Performance Evaluation Language Based on Fixed Point Logic

LI Qing[1],CAO Zi-ning[1,2,3],HUANG Tao[1]
(1. School of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 211106,China;
2. Science and Technology on Electro-optic Control Laboratory,Luoyang 471023,China;
3. Collaborative Innovation Center of Novel Software Technology and Industrialization,Nanjing 210023,China)

Abstract:Hybrid system is a complex dynamic system with continuous and discrete behavior,which is widely used in various safety critical fields,such as medical treatment and national defense. Such safety critical system requires high safety and reliability to reduce the catastrophic consequences of life and property caused by system failures. Formal method is a common method to guarantee system reliability,and model checking is the most widely used. Since model checking can only give the true or false logical value whether the system satisfies a certain property,it is combined with performance evaluation to describe some properties related to real value calculation of the system. The existing performance evaluation language CTML can describe the properties of the system related to probability and average expectation,and μ-calculus can describe some properties of the migration system through the minimum and maximum fixed point operators. Based on μ-calculus and CTML,a new performance evaluation language MLBoF for hybrid system based on fixed point and an performance evaluation algorithm are proposed. For the sublogic of CTML,the MLBoF formula representation equivalent to its semantics and the proof process of their equivalence are given. The example of aircraft taking off control system shows that the MLBoF extends the verification results of μ-calculus from {0,1} to real interval,with the ability to verify the probability properties of system,but also ensure the efficiency of MLBoF performance evaluation algorithm by extending the algorithm of traditional fixed points calculation.

Key words:hybrid system;fixed point;CTML;μ-calculus;performance evaluation

收稿日期:2022-01-10 **修回日期**:2022-05-14
基金项目:航空科学基金(20185152035);国家自然科学基金(61572253);中央高校基本科研项目(NJ2020022,NJ2019010)
作者简介:李 晴(1996-),女,硕士研究生,CCF 会员(J7790G),通讯作者,研究方向为形式化方法;曹子宁,教授,博导,研究方向为形式化方法、人工智能。

0 引 言

混成系统[1](Hybrid System,HS)通常由离散组件和连续组件组成,二者互相影响、互相依赖[2]。并且随着混成系统的设计越来越复杂,二者之间的交互越来越紧密。目前,它已广泛应用在智能电网、自动公路系统和医疗机器人等各种安全关键领域[3]。它们对系统的安全性有极高的要求,然而通过模型检测等常用的形式化方法保障其安全性是很有限的。因此,需要将模型检测与性能评价相结合[4],利用各自的优点,在保证验证系统效率的同时,描述更多系统的属性。

文献[5]提出 PLTL 用于描述"系统最终不会到达死锁状态的概率是否小于等于 0.9?"等系统性质。文献[6]提出了计算树度量语言 CTML,用于评价系统与实数计算相关的性质,如概率、平均期望等。文献[7]在 CTML、ZIA 的基础上,提出了 PZIA,用于描述系统与数据相关的性质和行为的概率性质等。

基于 μ 演算[8-9]的模型检测可用于设计与验证并发系统[10],通过使用不动点算子刻画系统的性质。文献[11-12]在不动点算子的基础上分别提出并发加权 μ 演算和广义可能性 μ 演算以增强经典 μ 演算的表达能力。此外,目前已有一些较为高效的算法用于评价 μ 演算的公式[10,13],并且很多时序逻辑可以转换为 μ 演算,比如 CTL。

因此,结合 CTML 和 μ 演算,该文提出了基于不动点的性能评价语言 MLBoF。由于 MLBoF 公式包含不动点,而不动点的存在不是必然的,因此给出其语义存在的合理性证明。为了简化 CTML 子逻辑的计算步骤,给出与其语义等价的 MLBoF 公式及相应的证明过程。最后,给出 MLBoF 公式的性能评价算法。

1 性能评价语言 MLBoF

为了提出能够描述实值计算相关性质的性能评价语言,通过对经典 Kripke 结构中的标签函数进行扩展,将标签函数 L 的值域从 $\{0,1\}$ 扩展为 $[m,n]$,其中 $m<n$,并且 m、n 都是非负实数。下面给出其具体的定义:

定义 1(状态标签函数):状态标签函数 f 表示从状态空间 S 到闭区间 $[m,n]$ 的一个映射,它可以表示为:$f:S\to[m,n]$。其中,m 和 n 的取值取决于具体实例。

状态标签函数也可称为原子函数或状态函数,带有原子函数的 Kripke 结构定义如下:

定义 2($\text{Kripke}_{\text{AF}}$):$\text{Kripke}_{\text{AF}}$ 结构是一个四元组 $M_{\text{AF}}=(S,S_0,\text{AF},R)$,其中,$S$ 是一个有限状态集合;S_0 是一个有限初始状态集合;AF 是原子函数的有限集合;R 是一个迁移关系,$R\subseteq S\times S$,R 满足 $\forall s\in S$,$\exists s'\in S$,使 $(s,s')\in R$。

$\text{Kripke}_{\text{AF}}$ 为提出可以描述系统实值计算相关性质的新的性能评价语言奠定了基础。

1.1 MLBoF 语法

MLBoF 作为 CTML 和 μ 演算部分结合的产物,其语法定义如下所述:

定义 3(MLBoF 语法):AF 为原子函数集合,令 f_p 表示由原子命题转换而来的状态函数,AF 由一般的状态函数 f 和 f_p 组成。令 $\text{VAR}=\{\varphi_1,\varphi_2,\varphi_3,\cdots\}$,表示函数变量集合,其中每个 $\varphi\in\text{VAR}$ 可被赋值为 $G(S)$ 的一个元素,$G(S)$ 是所有可能的状态函数组成的集合。MLBoF 公式按如下规则构造:

(1)若 $f_{\text{AF}}\in\text{AF}$,则 f_{AF} 是一个公式;

(2)一个函数变量 φ 是一个公式;

(3)若 f 和 g 是公式,则 $f\cdot g$,$f\vee g$,$f\wedge g$ 也是公式;

(4)如果 f 是公式,那么 MXf 是公式;

(5)若 $\varphi\in\text{VAR}$ 是一个公式且 f 是一个公式,又 f 在 φ 上是单调的(由于 MLBoF 公式的构造中去除了否定算子,即 f 中出现的所有函数变量 φ,在公式 f 中其外面嵌套的否定算子个数为 0),则 $\mu\varphi\cdot f$ 和 $\nu\varphi\cdot f$ 都是公式。

1.2 MLBoF 语义

经典的 μ 演算公式表示一个状态集合,而 MLBoF 的公式表示一个函数。MLBoF 的语义定义如下所示:

定义 4(MLBoF 语义):公式 f 可以解释为一个状态映射函数 $f:S\to[m,n]$,并将其记为 $[[f]]_Me$,其中 M 是变迁系统,$e:\text{VAR}\to G(S)$ 是环境。函数 $[[f]]_Me$ 的递归定义如下所示:

(1)若 $h=[[f_{\text{AF}}]]_Me$,则存在两种情况:若 $f_{\text{AF}}=f$,则对于 $\forall s\in S$,存在 $h(s)=f(s)$;若 $f_{\text{AF}}=f_p$,则对于 $\forall s\in S$,存在:

$$h(s)=\begin{cases}1, M\text{ 上的状态 }s\text{ 满足原子命题 }p\\0, M\text{ 上的状态 }s\text{ 不满足原子命题 }p\end{cases}$$

(2)$[[\varphi]]_Me=e(\varphi)$。

(3)若 $h=[[f\cdot g]]_Me$,则对 $\forall s\in S$,有 $h(s)=f(s)\cdot g(s)$;需要注意,使用 · 算子的前提是公式 f 和 g 满足 $f:S\to[0,1]$ 和 $g:S\to[0,1]$,并且 $[0,1]\subseteq[m,n]$。

(4)若 $h=[[\text{MX}f]]_Me$,则对 $\forall s\in S$,有:

$$h(s)=\begin{cases}\sum\limits_{t\in T_s}(p_{s,t}\times f(t)), T_s\neq\varnothing\\f(s), T_s=\varnothing\end{cases}$$

其中,$T_s=\{t\mid s\to t,\text{且}s\neq t\}$,$p_{s,t}$ 是由状态 s 到状态 t 的迁移概率。

(5)若 $h=[[f\wedge g]]_Me$,则对 $\forall s\in S$,有 $h(s)=$

$\min\{f(s),g(s)\}$ 。

(6)若 $h=[[f\vee g]]_M e$,则对 $\forall s\in S$,有 $h(s)=\max\{f(s),g(s)\}$ 。

(7)若 $h=[[\mu\varphi.f]]_M e$,则 $[[\mu\varphi.f]]_M e$ 是变换 $\tau:G(S)\to G(S)$ 的最小不动点,其中 $\tau(W)=[[f]]_M e[\varphi\leftarrow W]$ 。

(8)若 $h=[[\nu\varphi.f]]_M e$,则 $[[\nu\varphi.f]]_M e$ 是变换 $\tau:G(S)\to G(S)$ 的最大不动点,其中 $\tau(W)=[[f]]_M e[\varphi\leftarrow W]$ 。

2　MLBoF 语义合理性

由于性能评价语言 MLBoF 的语义涉及到不动点,而不动点是否存在并不是显然的。因此,MLBoF 语义的合理性依赖于不动点的存在,进而下文对不动点的存在性进行论述。

根据 Tarski 不动点定理,若 MLBoF 公式的语义解释环境构成完备格以及映射函数 τ 是单调的,那么可证得不动点是存在的。

下面给出基于 MLBoF 公式语义的 $G(S)$ 上的二元关系定义,以及该二元关系是偏序关系的证明过程,具体如下:

定义 5(二元关系 $\preccurlyeq$):给定一个 $\text{Kripke}_{\text{AF}}$ 结构 $M=(S,S_0,\text{AF},R)$, S 上所有的状态函数构成集合 $G(S)$, $G(S)$ 上存在一个二元关系 $\preccurlyeq$:对于函数 $f_1,f_2\in G(S)$,若对 $\forall s\in S$, $f_1(s)\preccurlyeq f_2(s)$ 成立,则称 $f_1\preccurlyeq f_2$ 成立。

命题 1: $G(S)$ 上的二元关系集合($G(S)$, $\preccurlyeq$)是偏序集。

证明:根据 $\preccurlyeq$ 和偏序的定义,可证得 $\preccurlyeq$ 满足自反性、反对称性和传递性。因此,命题 1 成立,具体的证明过程见文献[14]。

命题 2:偏序集($G(S)$, $\preccurlyeq$)构成一个完备格。

证明:根据完备格[11]和($G(S)$, $\preccurlyeq$)的定义,取 $G(S)$ 中任一子集 $A(S)$ 。考虑 $A(S)$ 的最小上界分别是 $A(S)$ 和 $G(S)\backslash A(S)$ 中的元素两种情况,可证得 $G(S)$ 的任意子集 $A(S)$ 存在最小上界和最大下界。具体的证明过程见文献[14]。

在此基础上,可以利用结构归纳法证明 $G(S)$ 上的映射函数 $\tau:G(S)\to G(S)$ 的单调性。而 τ 函数的单调性证明可转化为 MLBoF 公式的单调性证明,进而可转化为 MLBoF 公式中出现的所有算子的单调性证明。

命题 3:MLBoF 公式中出现的所有算子都是单调的。

证明:根据基础算子 · 、MX、∧和∨的语义及利用其构造的 MLBoF 公式语义,可证得基础算子 · 、MX、∧和∨的是单调的。因此,命题 3 成立,具体的证明过程见文献[14]。

因此,由上述算子构造的 MLBoF 的任何公式也是单调的。由于不动点算子中的公式是 · 、∧、∨和 MX 算子构成的,所以出现在不动点算子中的每一个可能的公式也是单调的。根据结构归纳法,不动点算子构成的 MLBoF 公式也是单调的。而 $G(S)$ 上的映射函数 τ 是由 · 、∧、∨和 MX 和不动点算子构成的,因此,根据结构归纳法,若构成 τ 函数的基础算子是单调的,那么 τ 函数也是单调的。

由 Tarski 不动点定理,可知 MLBoF 存在最小和最大不动点。所以其语义是合理的。

利用结构归纳法的思想,可将 μ 演算中谓词变换满足∪连续和∩连续定义和性质[15]推广到 MLBoF 上的映射函数 $\tau(W)=[[f]]_M e[\varphi\leftarrow W]$,因此 $\tau(W)=[[f]]_M e[\varphi\leftarrow W]$ 也是∪连续和∩连续的。而根据 MLBoF 的语法和语义可知, $\mu\varphi.f$ 和 $\nu\varphi.f$ 的语义(即 $[[\mu\varphi.f]]_M e$ 和 $[[\nu\varphi.f]]_M e$ 分别表示映射函数 $\tau(W)=[[f]]_M e[\varphi\leftarrow W]$ 的最小不动点和最大不动点。

因此,根据 Tarski 不动点定理可知,若 τ 是∪连续的,则 $[[\mu\varphi.f]]_M e=\cup_i(\tau^i(\text{MIN}))$;若 τ 是∩连续的,则 $[[\nu\varphi.f]]_M e=\cap_i(\tau^i(\text{MAX}))$ 。假设 $G(S)$ 中的所有函数映射的值都位于实数区间 $[m,n]$,那么 MIN 表示 $f_{\min}:S\to\{m\}$,MAX 表示 $f_{\max}:S\to\{n\}$ 。

3　CTML 子逻辑的 MLBoF 公式表示

3.1　CTML 简介

CTML 是一个可用于描述系统的概率和期望等实值属性的性能评价语言,它的某些子逻辑可以用 MLBoF 表示,该文将它们命名为 SLoCTML(Sub-Logic of CTML)。它的语法和语义具体如下:

定义 6(SLoCTML 语法):SLoCTML 的语法定义如下:

(1) $\varphi::=\text{one}\mid\text{zero}\mid r\mid\varphi\cdot\varphi\mid M\delta$;

(2) $\delta::=\varphi\text{U}_x\varphi\mid X\varphi$;

由于 SLoCTML 代表 CTML 的某些子逻辑,其更加具体的语法解释和语义定义可见文献[6]。

根据二者的语义,可知 MLBoF 公式可以表示 SLoCTML,如 MX、· 、MU_x,而且 SLoCTML 的原子状态函数可以用 MLBoF 的原子状态函数 f_p 表示。

根据 SLoCTML 的语义可知,SLoCTML 的 MU_x 算子描述的系统属性相对复杂,具有较大的实际意义。所以下文以 MU_x 为例,给出与其语义等价的 MLBoF 公式表示以及证明过程。

3.2　与 MU_x 公式语义等价的 MLBoF 公式

由于性能评价语言 MLBoF 的公式涉及到不动

点,表达形式比较抽象,不利于理解和描述性质。根据二者公式语义,可知 $M(f\mathrm{U}_{\times}g)$ 与 $\mu\varphi.[g \vee (f \cdot \mathrm{MX}\varphi)]$ 语义等价,且 $\tau(\varphi) = g \vee (f \cdot \mathrm{MX}\varphi)$。若状态 s 满足 $\mathrm{M}(f\mathrm{U}_{\times}g)$,则使用上述语义等价的 MLBoF 公式计算;否则,对于不满足路径公式的状态 s,令其函数值为 0。

在证明语义等价前,需要进行一些说明:

(1) 将 SLoCTML 的 $\mathrm{MU}_{\times}$ 公式中出现的原子命题 f 和 g 看作 MLBoF 中的原子函数,并且它们的值域为实数区间 $[m,n]$ 中,其中 $m=0$, $n=1$。

(2)令最大的状态函数和最小的状态函数分别为 $\mathrm{MAX}:S \to \{n\}$ 和 $\mathrm{MIN}:S \to \{m\}$。

下面给出二者语义等价的具体证明过程:

命题 4:$\tau(\varphi) = g \vee (f \cdot \mathrm{MX}\varphi)$ 是单调的。

证明:假设 $f_1 \leqslant f_2$,根据函数 $\tau(\varphi) = g \vee (f \cdot \mathrm{MX}\varphi)$ 的语义,给出 $\tau(f_1)$ 和 $\tau(f_2)$ 各自可能的两种语义。并在此基础上,根据偏序关系 $\leqslant$ 的定义、$\tau(f_1)$ 和 $\tau(f_2)$ 每种可能语义之间的依赖关系,进而证得 $\tau(f_1) \leqslant \tau(f_2)$ 成立。具体的证明过程见文献[14]。

命题 5:$\mathrm{M}(f\mathrm{U}_{\times}g)$ 的语义和 $\tau(\varphi) = g \vee (f \cdot \mathrm{MX}\varphi)$ 的不动点的语义等价。

证明:综合考虑 $\mathrm{M}(f\mathrm{U}_{\times}g)$ 可能的两种语义 $h(s) = g(s)$ 和 $h(s) = (f \cdot \mathrm{MX}(\mathrm{M}f\mathrm{U}_{\times}g))(s)$,利用公式转换和具体的公式语义可证得命题 5 成立,具体的证明过程见文献[14]。

命题 6:CTML 的 $\mathrm{M}(f\mathrm{U}_{\times}g)$ 公式的语义和 MLBoF 的 $\mu\varphi.[g \vee (f \cdot \mathrm{MX}\varphi)]$ 公式语义等价,即 $\mathrm{M}(f\mathrm{U}_{\times}g) = [[\mu\varphi.[g \vee (f \cdot \mathrm{MX}\varphi)]]]_M e$。

证明:在分析公式 $\mu\varphi.[g \vee (f \cdot \mathrm{MX}\varphi)]$ 和 $\mathrm{M}(f\mathrm{U}_{\times}g)$ 的语义之后,利用归纳法,依次证得 $\cup_i(\tau^i(\mathrm{MIN})) \leqslant \mathrm{M}(f\mathrm{U}_{\times}g)$ 和 $\mathrm{M}(f\mathrm{U}_{\times}g) \leqslant \cup_i(\tau^i(\mathrm{MIN}))$ 成立。因此,命题 6 成立。具体的证明过程见文献[14]。

4 MLBoF 公式的性能评价算法

该算法是对 Emerson 和 Lei 提出的用于评价 μ 演算公式的算法[15]的一个扩展,其不同之处在于在计算外层不动点的新的近似值时,不必使用 MIN 或 MAX 重新初始化内层不动点的计算,只需从这个不动点下的任何已知的近似值开始迭代计算即可。此外,通过引入人为设置的误差值 α,避免算法进行无终止地计算。所以,该算法能够在保证效率的同时,尽可能地计算出较为接近真正不动点的近似不动点。

MLBoF 公式的性能评价算法输入 MLBoF 公式、系统模型 $\mathrm{Kripke}_{\mathrm{AF}}$ 和误差值 α,输出一个状态函数 h。该算法的伪代码如下所示:

```
1:Function eval ( h,e )
2:if ( h = f_p )
3:    for s in S
4:if ( p ∈ L(s) )    h(s) = 1;
5:Else               h(s) = 0;
6:    return h ;
7:if ( h = f )
8:    for s in S    h(s) = f(s) ;
9:    return h ;
10:if ( h = φ )    return e(φ) ;
11:if ( h = g1 · g2 )
12:    g1 = eval( g1 , e ); g2 = eval( g2 , e );
13:    for s in S    h(s) = g1(s) · g2(s) ;
14:    return h ;
15:if( h = g1 ∧ g2 )
16:    g1 = eval( g1 , e ); g2 = eval( g2 , e );
17:    for s in S    h(s) = min{ g1(s) , g2(s) };
18:    return h ;
19:if ( h = g1 ∨ g2 )
20:    g1 = eval( g1 , e ); g2 = eval( g2 , e );
21:    for s in S    h(s) = max{ g1(s) , g2(s) };
22:    return h ;
23:if( h = MX g )
24:    g = eval( g , e );
25:    for s in S
26:        if ( ∃t, s → t) h(s) = Σ_{t∈|t|s→t|} p_{s,t} · g(t) ;
27:        Else    h(s) = g(s) ;
28: return h ;
29:if ( h = μφ_i . f(φ_i) )
30:    for top-level greatest fixpoint subformulas νφ_j . f(φ_j) in f
31:        F[j] = MAX;
32:        while ∀ s ∈ S , F[j](s) - φ_old( s ) > α
33:            φ_old = F[i] ;
34:            F[i] = eval( f,e[φ_i ← F[i] ]);
35:    h = F[i] ;
36:    return h ;
37:if ( h = νφ_i. f(φ_i) )
38: for top-level least fixpoint subformulas μφ_j. f(φ_j) in f
39:        F[j] = MIN;
40:        while ∀ s ∈ S , F[i] ( s ) - φ_old( s ) > α
41:            φ_old = F[i] ;
42:            F[i] = eval( g, e[φ_i ← F[i] ]);
43:    h = F[i] ;
44:    return h ;
45:end function eval;
```

5 MLBoF 性能评价公式实例

本节以飞机起飞控制系统[16]为例,使用 MLBoF 和 CTML 公式分别描述与其本身相关的系统性质,并

使用 MLBoF 公式的性能评价算法计算 MLBoF 公式的值。

它的整个起飞过程由停止、滑行、起飞、爬升、平飞等阶段组成,并且前四个阶段有可能进入故障状态。它的自动机模型如图 1 所示。

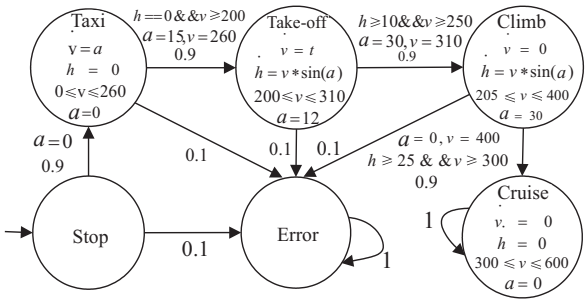

图 1　飞机控制系统

(1)对于性质“飞机从 Stop 开始,最终发生异常即到达 Error 状态的概率是多少”, CTML 描述为:(Mone $U_{\varphi} f_{\text{Error}}$)(Stop), MLBoF 公式描述为:$\mu\varphi.[f_{\text{Error}} \vee (\text{one} \cdot \text{MX}\varphi)]$, $\tau(\varphi) = f_{\text{Error}} \vee (\text{one} \cdot \text{MX}\varphi)$。需要注意的是,下面公式中的 S 的元素依次为状态 Stop、Taxi、Take_off、Climb、Error 和 Cruise。

其中,MLBoF 公式分别由 CTML 中的原子命题 one 和 f_{Error} 转换而来。因此,其映射的实数区间为[0, 1],并且 MIN:S→{0}。根据性能评价算法,令误差 α =0.001,有:

τ(MIN)(S)=(0,0,0,0,1,0);

τ^2(MIN)(S)=(0.1, 0.1, 0.1, 0.1, 1, 0);

τ^3(MIN)(S)=(0.19, 0.19, 0.19, 0.1, 1, 0);

τ^4(MIN)(S)=(0.271, 0.271, 0.19,0.1,1,0);

τ^5(MIN)(S)=(0.343 9,0.271,0.19,0.1,1,0);

τ^6(MIN)(S)= τ^5(MIN)(S)。

此时算法终止,可知飞机从 Stop 开始,最终发生异常即到达 Error 状态的概率是 0.343 9。

(2)对于性质“飞机从 Stop 开始,最终成功巡航即到达 Cruise 状态的概率是多少”,CTML 描述为:(M one$U_{\varphi} f_{\text{Cruise}}$)(Stop), MLBoF 公式描述为:$\mu\varphi.[f_{\text{Cruise}} \vee (\text{one} \cdot \text{MX}\varphi)]$, $\tau(\varphi) = f_{\text{Cruise}} \vee (\text{one} \cdot \text{MX}\varphi)$。

根据性能评价算法,令误差 α =0.001,有:

τ(MIN)(S) = (0, 0, 0, 0, 0, 1);

τ^2(MIN)(S) = (0, 0, 0, 0.9, 0,1);

τ^3(MIN)(S) = (0, 0, 0.81, 0.9, 0, 1);

τ^4(MIN)(S) = (0, 0.729, 0.81, 0.9, 0, 1);

τ^5(MIN)(S)=(0.656 1, 0.729,0.81, 0.9,0, 1);

τ^6(MIN)(S) = τ^5(MIN)(S);

此时算法终止,可知飞机从 Stop 开始,最终成功巡航即到达 Cruise 状态的概率是 0.656 1。

通过上述应用实例可知,MLBoF 公式的性能评价算法的效率体现在以下两个层面:

(1)由于 MU_x 公式的计算依赖于涉及矩阵方程等复杂计算的线性系统求解[6],计算复杂,且时间复杂度为 O(Poly(|S|)。其中 Poly(|S|)表示|S|大小的多项式时间,|S|表示系统模型中的状态个数。而与 MU_x 公式语义等价的 MLBoF 公式的计算依赖于计算不动点的思想,该过程无需进行复杂的数学计算。因此,从这个层面上讲,该算法的效率体现在算法思想和计算步骤的简单。

(2)由于 MLBoF 公式的性能评价算法的提出基于对需要进行 $O(n^d)$ 次迭代的计算不动点的算法的扩展,相对于扩展需要进行 $O(n^k)$ 次迭代的简单直接的递归算法,该算法的效率更高。其中,n 表示系统模型的状态个数,k 表示不动点嵌套深度,d 表示公式的交替深度[15]。因此,从这个层面上,该算法的效率也得到了一定的改善。

6　结束语

由于混成系统广泛应用于各种安全关键领域,通过一般的形式化方法保障其安全性有一定的局限性。因此,提出一种面向混成系统的性能评价语言 MLBoF,用于描述系统行为的概率属性等。同时,为了简化 SLoCTML 的计算步骤以及提高 MLBoF 公式的理解性,给出与 SLoCTML 语义等价的 MLBoF 公式。最后,通过扩展计算不动点的算法,提出一个 MLBoF 公式的性能评价算法。与传统的基于 μ 演算的模型检测[15]相比,MLBoF 可以描述系统行为的概率属性。在以后的工作中,可以尝试往 MLBoF 公式中引入更多的算子如+算子,使其能够描述完整的 CTML 逻辑。

参考文献:

[1] DOYEN L, FREHSE G, PAPPAS G J, et al. Verification of hybrid systems[M]//Handbook of model checking. Berlin: Springer, 2018: 1047-1110.

[2] ROEHM H, OEHLERKING J, HEINZ T, et al. STL model checking of continuous and hybrid systems[C]//International symposium on automated technology for verification and analysis. [s. l.]: Springer, 2016: 412-427.

[3] 黄　超. 混成系统设计与验证的若干问题研究[D]. 南京: 南京大学, 2018.

[4] BAIER C, HAVERKORT B R, HERMANNS H, et al. Performance evaluation and model checking join forces[J]. Communications of the ACM, 2010, 53(9): 76-85.

[5] SHI H X. A quantitative approach for linear temporal logic

(下转第 122 页)