描述逻辑的动态时序扩展 *

孙永新[1,2], 赵希顺[1], 符志强[2]
(1. 中山大学 逻辑与认知研究所, 广州 510275; 2. 仲恺农业工程学院 计算机科学与工程学院, 广州 510225)

摘 要: 在一些基于本体的动态应用中,需要描述组合动作和变化域的时间特性。为了对这类应用建模,通过整合动态时序逻辑和描述逻辑,提出一类描述逻辑扩展。分析了该类扩展的基本形式 $DLTL_{ALC}$ 的语法和语义,并提出一种可终止的 tableau 算法判别 $DLTL_{ALC}$ 公式可满足性。利用该类扩展,可以表达组合动作执行过程中域变化的时间特性,该类扩展为语义 Web 服务等动态应用建模和推理提供了一条有效途径。

关键词: 动态时序描述逻辑; 动作推理; 表判定算法; 语义 Web 服务
中图分类号: TP301 **文献标志码**: A **文章编号**: 1001-3695(2012)02-0536-06
doi:10.3969/j.issn.1001-3695.2012.02.036

Dynamic linear temporal extensions of description logics

SUN Yong-xin[1,2], ZHAO Xi-shun[1], FU Zhi-qiang[2]
(1. *Institute of Logic & Cognition, Sun Yat-sen University, Guangzhou* 510275, *China*; 2. *College of Computer Science & Engineering, Zhongkai University of Agriculture & Engineering, Guangzhou* 510225, *China*)

Abstract: In some ontology based dynamic applications, there are needs for describing composite actions and time properties about changing domains. This paper proposed a family of extensions of description logics, integrating description logics into dynamic linear time temporal logic, for modeling these applications. First analyzed the syntax, semantics of $DLTL_{ALC}$, the basic formalism of the extensions, and presented a terminating tableau algorithm for determining the $DLTL_{ALC}$-formulas' satisfiability. With the extensions, time properties about changing domains according to the execution of composite actions can be expressed, so the extensions provide good support for modeling and reasoning about dynamic applications, such as semantic Web services etc.

Key words: dynamic linear temporal description logics; reasoning about actions; tableau decision algorithm; semantic Web service

0 引言

动态知识,如动作、时间以及状态变化等知识的表示与推理,在人工智能领域向来受到极大关注,许多基于逻辑的形式系统也因之而设计和发展。对于这类形式系统,表达能力和可计算性等特性被看做是评价其可应用性的关键。近年来,由于描述逻辑在这两个方面均有良好表现,基于描述逻辑扩展的动态知识表示系统逐渐成为知识表示和推理领域研究的热点。

描述逻辑(description logics, DLs)[1]是一族基于逻辑的知识表示形式系统。描述逻辑对应于可判定的一阶逻辑子集,具有较强的表达能力,不过一般要对其进行扩展才适合表达动态知识。常见的用于表达动态知识的描述扩展形式包括时序描述逻辑和动态描述逻辑等。时序描述逻辑在描述逻辑的基础上引入时态算子以描述和推理变化域的时间特性[2,3],而动态描述逻辑在描述逻辑的基础上引入动态算子以支持描述动作知识和对组合动作相关的效果问题进行推理[4,5]。

笔者发现,有些动态应用需要同时描述组合动作和变化域的时间特性,用单纯时态或动态扩展的描述逻辑难以对其建模。比方说,有这样一个语义 Web 服务推理的例子,已知有 n 个航班的信息,并为每个航班建立一个语义 Web 服务,假设从甲地到乙地的航班服务 i 用动作 a_i 表示,本文采用文献[5]中的方法对航班服务建模。为了便于说明问题,本文简化了服务表示(省略了离港时间和抵港时间等),把航班服务 i 建模为 $a_i \equiv$((旅客 AtCity 离港地 i), (旅客 AtCity 抵港地 i, 旅客 ArrivedByPlane空客)),表示执行 a_i 的前提条件是旅客在离港地 i,执行 a_i 的后置条件(结果)是旅客在抵港地 i 且抵达时乘坐的是空客公司生产的飞机。人们希望了解通过这些航班服务能否规划出一个乘机计划,使得旅客可以乘机从广州出发,途径武汉、北京,最后回到广州,且途中乘坐的都是空客公司的飞机。人们关注其规划目标,显然可以很方便地用动态描述逻辑公式 (旅客 AtCity 广州)$(\wedge\langle a_1 + a_2 + \cdots + a_n)^*\rangle$(旅客 AtCity 广州) 表示从广州出发,存在航班服务系列使得旅客可以最终返回广州,但如果还希望表达该旅客乘坐该系列航班的途中信息,用动态描述逻辑公式就很难表达了,用时序描述逻辑公式表达倒是很方便: ((○(旅客 ArrivedByPlane 空客))U(旅客 AtCity 广州))∧(○((旅客 AtCity 武汉)∧○○(旅客 AtCity 北京)∧○○○(旅客 AtCity 广州) (○为 next 算子, U 为 until 算子))。

为了能对上例类型的应用进行形式建模,在文献[6,7]的

收稿日期: 2011-08-08; **修回日期**: 2011-09-16 **基金项目**: 国家自然科学基金资助项目(60970040)

作者简介: 孙永新(1972-),男,江西吉安人,工程师,博士研究生,主要研究方向为描述逻辑、语义 Web、软件工程、人工智能逻辑(syx2000@tom.com);赵希顺(1964-),男,教授,博导,主要研究方向为可计算性与计算复杂性理论、数理逻辑、人工智能逻辑;符志强(1979-),男,讲师,主要研究方向为人工智能计算.

启发下,结合动态时序逻辑和描述逻辑提出一类形式系统,称做动态时序描述逻辑(dynamic linear temporal description logic, $DLTL_{DL}$)。本文将介绍该类逻辑的基本形式 $DLTL_{ALC}$,包括它的语法和语义,它的变体 $DLTL^{A}_{ALC}$ 和判定其公式可满足性的 tableau 算法。

1　基本定义

1.1　$DLTL_{ALC}(\Sigma)$语法和语义

$DLTL_{ALC}$用正规程序[8]来描述动作组合,本文先回顾一些与正规程序相关的基本定义。

定义 1　正规程序。假设有限非空字母表 $\Sigma=\{a_1,a_2,\cdots,a_n\}$是一个原子动作集合。$\Sigma$ 上的正规程序定义如下:所有的 $a\in(\Sigma\cup\{\varepsilon\})$为原子正规程序;如果 π_0, π_1 是正规程序,则 π_0 与 π_1 的选择 $\pi_0+\pi_1$ 也是正规程序;如果 π_0, π_1 是正规程序,则 π_0 与 π_1 的连接 $\pi_0;\pi_1$ 也是正规程序;如果 π 是正规程序,则 π 的闭包 π^* 也是正规程序。

文中用 π 或 π 加下标表示正规程序,用 π^i 表示 π 的 i 次连接,π 的 0 次连接等于空串 ε;文中还用 $prg(\Sigma)$ 表示 Σ 上的所有正规程序集合。

定义 2　正规集。π 的正规集 $RS(\pi)$的定义如下:如 $\pi=a\in(\Sigma\cup\{\varepsilon\})$,则 $RS(\pi)=\{a\}$;$RS(\pi_0+\pi_1)=RS(\pi_0)\cup RS(\pi_1)$;$RS(\pi_0;\pi_1)=\{\tau_0\tau_1\mid\tau_0(RS(\pi_0),\tau_1(RS(\pi_1)\}$;$RS(\pi^*)=\cup_{i\in\omega}RS(\pi^i)$。

文中用 Σ^ω 表示 Σ 上无穷长的动作序列集合,用$|\sigma|$表示动作序列 σ 的字符串长度,用 $prf(\sigma)$表示 σ 的前缀的集合。$prf(\sigma)$集合上有前缀关系$\leqslant$和严格前缀关系$<$,$\tau_0\leqslant\tau_1$ 当且仅当存在 τ_2 使得 $\tau_0\tau_2=\tau_1$;$\tau_1<\tau_2$ 当且仅当 $\tau_0\leqslant\tau_1$ 且 $\tau_0\neq\tau_1$。

定义 3　$DLTL_{ALC}(\Sigma)$语法。$DLTL_{ALC}(\Sigma)$引入 Σ 上的正规程序标记的 U(until)算子对基本描述逻辑 ALC 进行动态时序扩展。$DLTL_{ALC}(\Sigma)$符号包括一个概念名集合 $N_C=\{C_1,C_2,\cdots,C_m\}$,一个角色名集合 $N_R=\{R_1,R_2,\cdots,R_n\}$和一个对象名集合 $N_O=\{a_1,a_2,\cdots,a_k\}$。$DLTL_{ALC}(\Sigma)$概念归纳定义如下:$T$ 和所有的 $C_i\in N_C$ 是原子概念;如果 C 是概念,则$\neg C$ 是概念;如果 C 和 D 是概念,则 $C\quad D$ 是概念;如果 $R\in N_R$,C 是概念,则 $\exists R.C$ 是概念;如果 $\pi\in prg(\Sigma)$,C 和 D 是概念,则 $CU^\pi D$ 是概念。

$DLTL_{ALC}(\Sigma)$公式归纳定义如下:$\top$,$C=D$ 和 $a:C$ 和 aRb 是原子公式,其中 C 和 D 是概念,$a,b\in N_O$;如果 α 是公式,则 $\neg\alpha$ 是公式;如果 α 和 β 是公式,则 $\alpha\wedge\beta$ 是公式;如果 $\pi\in prg(\Sigma)$,α 和 β 是公式,则 $\alpha U^\pi\beta$ 是公式。

文中,分别用 $\bot$,$C\quad D$,$C\quad D$,$\alpha\vee\beta$ 和 $\alpha\rightarrow\beta$ 分别表示$\neg\top$,$\neg(\neg C\quad\neg D)$,$\neg C\quad D=\top$,$\neg(\neg\alpha\wedge\neg\beta)$,$\neg\alpha\vee\beta$。

定义 4　$DLTL_{ALC}(\Sigma)$语义。$DLTL_{ALC}(\Sigma)$模型 M 是一个三元组$\langle\sigma,<,I\rangle$,其中$\langle\sigma,<\rangle$是一个关于无穷动作序列 $\sigma\in\Sigma^\omega$ 的线性框架,$<$是 $prf(\sigma)$上的严格前缀关系。I 是一个解释函数,将每个 $\tau\in prf(\sigma)$映射到一个 ALC 模型 $I(\tau)=\langle\Delta^{I(\tau)},{R_0}^{I(\tau)},\cdots,{C_0}^{I(\tau)},\cdots,{a_0}^{I(\tau)}\rangle$,其中,$\Delta^{I(\tau)}$是 K 在 τ 点的域,非空且满足以下条件:对任意 $\tau_1<\tau_2$,$\Delta^{I(\tau1)}\subseteq\Delta^{I(\tau2)}$;$R_i\in N_R$ 在 τ 点的解释 ${R_i}^{I(\tau)}$ 是 $\Delta^{I(\tau)}$ 上的二元关系;$C_i\in N_C$ 在 τ 点的解释 ${C_i}^{I(\tau)}$是 $\Delta^{I(\tau)}$的子集;$a_i\in N_O$ 在 τ 点的解释 ${a_i}^{I(\tau)}\in\Delta^{I(\tau)}$ 且对任意 τ_1,τ_2,${a_i}^{I(\tau1)}={a_i}^{I(\tau2)}$。显然,上述解释是基于扩张域模型假设,且概念名、角色名均为局部名,而对象名为全局名。扩展以上解释,任意概念的值可定义如下:

$T^{I(\tau)}=\Delta^{I(\tau)}$;

$(C\quad D)^{I(\tau)}=C^{I(\tau)}\cap D^{I(\tau)}$;

$(\neg C)^{I(\tau)}=\Delta^{I(\tau)}\setminus C^{I(\tau)}$;

$(\exists R.C)^{I(\tau)}=\{d\mid\exists d_1\in C^{I(\tau)},dR^{I(\tau)}d_1\}$;

$(CU^\pi D)^{I(\tau)}=\{d\in\Delta^{I(\tau)}\mid\exists\tau_1\in RS(\pi)(\tau\tau_1\in prf(\sigma),d\in D^{I(\tau\tau1)},\forall\tau_2(\varepsilon\leqslant\tau_2<\tau_1\Rightarrow d\in C^{I(\tau\tau2)}))\}$。

定义 5　$DLTL_{ALC}(\Sigma)$公式真值和可满足性。给定一个 $DLTL_{ALC}(\Sigma)$模型 M 和一个时间点 $\tau\in prf(\sigma)$,公式 φ 在 τ 点的真值关系 $M,\tau\models\varphi$ 定义如下:

$M,\tau\models\top$;

$M,\tau\models C=D$ 当且仅当 $C^{I(\tau)}=D^{I(\tau)}$;

$M,\tau\models a:C$ 当且仅当 $a^{I(\tau)}\in C^{I(\tau)}$;

$M,\tau\models aRb$ 当且仅当 $a^{I(\tau)}R^{I(\tau)}b^{I(\tau)}$;

$M,\tau\models\neg\alpha$ 当且仅当 非 $M,\tau\models\alpha$;

$M,\tau\models\alpha\wedge\beta$ 当且仅当 $M,\tau\models\alpha$, $M,\tau\models\beta$;

$M,\tau=\alpha U^\pi\beta$ 当且仅当 $\exists\tau_1\in RS(\pi)(\tau\tau_1\in prf(\sigma)$,

$M,\tau\tau_1\models\beta,\forall\tau_2(\varepsilon\leqslant\tau_2<\tau_1\Rightarrow M,\tau\tau_2\models\alpha))$。

如果存在一个模型 $M=\langle prf(\sigma),<,I\rangle$和一个时间点 τ 满足 $M,\tau\models\alpha$,则称公式 α 在基于 σ 的模型 M 中可满足。

本文只考虑开放世界假设下的 $DLTL_{ALC}(\Sigma)$公式可满足性推理问题,参照描述逻辑中的一般做法,概念可满足性推理以及其他推理问题可归结到 $DLTL_{ALC}(\Sigma)$公式可满足性推理问题。

为了便于表述且不失一般性,后文假设形如$\neg\neg\theta$的表达式均自动转换为 θ(θ 为任意公式或概念),形如$\neg(x:C)$的公式自动转换为$(x:\neg C)$。并假设所有的概念等式的形式均为 $C=\top$($\neg\neg\theta$ 与 θ 等价,$\neg(x:C)$与$(x:\neg C)$等价,$C=D$ 也可以等价地转换为$\neg(C\quad\neg D)\quad\neg(D\quad\neg C)$)。

$DLTL_{ALC}(\Sigma)$公式可以表达动态和时序约束。给定动作集合 $\Sigma=\{a_1,a_2,\cdots,a_n\}$,动态算子$\langle\pi\rangle$,$[\pi]$和时态算子 U(until)、$\bigcirc$(next)、$\Diamond$(sometimes)、$\Box$(always)可以分别定义为:$\langle\pi\rangle\theta\equiv\top U^\pi\theta$;$[\pi]\theta\equiv\neg\langle\pi\rangle\neg\theta$;$\theta U\gamma\equiv\theta U^{\Pi^*}\gamma$,其中 $\Pi=(a_0+\cdots+a_n)$;$\bigcirc\theta\equiv\vee_{a\in\Sigma}\langle a\rangle\theta$($\theta$ 为公式)或$\bigcirc\theta\equiv\quad_{a\in\Sigma}\langle a\rangle\theta$($\theta$ 为概念);$\Diamond\theta\equiv\top U\theta$;$\Box\theta\equiv\neg\Diamond\neg\theta$。

1.2　$DLTL^{A}_{ALC}(\Sigma)$

直接给出一个 $DLTL_{ALC}(\Sigma)$的 tableau 算法很难,因此本文提出 $DLTL_{ALC}(\Sigma)$的变体 $DLTL^{A}_{ALC}(\Sigma)$,$DLTL_{ALC}(\Sigma)$公式可满足性问题可归约到 $DLTL^{A}_{ALC}(\Sigma)$公式可满足性问题。

语法上,$DLTL^{A}_{ALC}(\Sigma)$和 $DLTL_{ALC}(\Sigma)$的区别仅在于前者 U 算子用不含 ε 弧的非确定性有限自动机标记,而后者的 U 算子用正规程序标记。除了 U 算子的解释之外,$DLTL^{A}_{ALC}(\Sigma)$的语义也基本与 $DLTL_{ALC}(\Sigma)$的一样。对 $DLTL^{A}_{ALC}(\Sigma)$来说,给定一个模型 $M=\langle prf(\sigma),<,I\rangle$和一个时间点 $\tau\in prf(\sigma)$,含 U 算子的概念的值定义如下:$(CU^AD)^{I(\tau)}=\{d\in\Delta^{I(\tau)}\mid\exists\tau_1\in L(A)(\tau\tau_1\in prf(\sigma),d\in D^{I(\tau\tau1)},\forall\tau_2(\varepsilon\leqslant\tau_2<\tau_1\Rightarrow d\in C^{I(\tau\tau2)}))\}$。文中用 $L(A)$表示自动机 A 所接收的语言,含 U 算子的公式的真值关系定义如下:$M,\tau\models\alpha U^A\beta$ 当且仅当 $\exists\tau_1\in L(A)(\tau\tau_1\in prf(\sigma),M,\tau\tau_1\models\beta,\forall\tau_2(\varepsilon\leqslant\tau_2<\tau_1\Rightarrow M,\tau\tau_2\models\alpha))$。

正规表达式和有限自动机等价,存在多项式时间算法将一

个正规表达式转换为等价的不含 ε 边的非确定有限自动机[9]。显然,利用以上结论,可以将 $DLTL_{ALC}(\Sigma)$ 公式的可满足性问题归结为 $DLTL^{A}_{ALC}(\Sigma)$ 公式可满足性问题:给定一个 $DLTL_{ALC}(\Sigma)$ 公式 α,把 α 中出现的每个正规程序 π 替代为一个不含 ε 边的、等价的非确定有限自动机 A(即 $L(A)=RS(\pi)$),则可将 α 翻译为一个 $DLTL^{A}_{ALC}(\Sigma)$ 公式 α′,α 可满足当且仅当 α′可满足。

文中用五元组 (Σ,Q,ρ,Q_0,F) 描述非确定有限自动机,其中 Σ 为字母表,Q 为状态集合,ρ 为转移函数,Q_0 为初始状态集合,F 为接受状态集合。给定自动机 A,定义 A 相关自动机 $A(q)=(\Sigma,Q,\rho,\{q\},F)$,其中 $q\in Q$。A 和 $A(q)$ 区别仅在于初态不同,定义 $A(q)$ 有助于跟踪 tableau 算法执行过程中动—时态约束扩展。另外,使用 $\langle a\rangle\theta$ 来表示 $TU^{A_a}\theta$,其中 A_a 为一个接收语言 $\{a\}$ 的两状态自动机,$\langle a\rangle\theta$ 在 τ 点可满足当且仅当在 τa 点 θ 可满足。下文将 $\langle a\rangle\theta$ 当做一类特殊的公式或概念,以区别于一般的形如 $\theta U^{A}\gamma$ 的公式或概念。

2 tableau 算法

2.1 tableau 概述

$DLTL^{A}_{ALC}(\Sigma)$ 的 tableau 算法为输入公式 φ 构造出一个 tableau。Tableau 是一个有向图,图中每个节点 N 都标记有一个约束系统 $l(N)$,每条有向边 (N_1,N_2) 都标记有一个原子动作。约束系统是约束的集合,文中有时用 S 表示,一个约束或者是一个 $DLTL^{A}_{ALC}(\Sigma)$ 公式,或者是一个形似 $x:C$ 或 yRx 的公式,其中,C 是 $DLTL^{A}_{ALC}(\Sigma)$ 概念,R 为关系名,x 是变元,y 为变元或对象名,同时又把形如 $\alpha U^{A}\beta$ 或 $y:CU^{A}D$ 的约束称为预测。

变元在约束系统中代表一类概念,称为概念类型,例如一个约束系统中包含公式 $x:C$ 和 $x:D$,则 x 的概念类型为 $\{C,D\}$,对于 $DLTL^{A}_{ALC}(\Sigma)$ 的公式可满足性推理问题来说,同一约束系统中同一概念类型只需用一个变元表示即可。Tableau 算法中,假设所有变元按其引入的先后顺序排序,当 x_1 先于 x_2 引入时,称 $x_1<x_2$。将满足相同概念类型的不同变元用其中最小变元替代的过程称为合并同类变元,合并同类变元不影响约束系统的可满足性,却可以防止在 tableau 算法执行过程中无止尽地引入变元。

直觉上,一个约束系统包含在某时间点描述逻辑域要满足的所有条件,一个可满足的约束系统是某些 ALC 模型的抽象,而根据 φ 的 tableau 构造出的一个可满足的约束系统系列就是 φ 的某些 $DLTL^{A}_{ALC}(\Sigma)$ 模型的抽象。

2.2 tableau 算法描述

Tableau 算法执行过程是约束扩展过程,算法按后面给出的 tableau 规则扩展约束。Tableau 规则分为两大类,一类为局部规则,tableau 算法对一个约束系统应用局部规则后,总是将其输出加入当前节点的约束系统中;另一类规则为全局规则,只有一条。Tableau 算法对一个约束系统应用全局规则后,构造出新的 tableau 节点作为当前 tableau 节点的后续节点。在 tableau 规则描述中,自动机 $A=(\Sigma,Q,\rho,Q_0,F)$,S 表示 tableau 当前节点的约束系统。

$DLTL^{A}_{ALC}(\Sigma)$ 的 tableau 算法 sat(φ) 如下所示,类似文献[6,10]中的 tableau 算法,算法执行过程分两个阶段。第一阶段的主要任务是为输入公式 φ 构造一个完全 tableau G;第二个阶段的主要任务是删除 G 中约束系统不可满足的节点,如最终 G 非空,则 φ 可满足。

输入公式 φ 的 tableau 的初始节点 N_0 标记有初始约束系统 S_0。为了满足全局对象名和扩张域假设,S_0 设为 $\{\varphi,\Box(x_0:\top)\}$($ob(\varphi)$ 为空集)或 $\{\varphi\}\cup\{\Box(a:\top)\mid a\in ob(\varphi)\}$($ob(\varphi)$ 非空)。其中 x_0 为算法引入的第一个变元,$ob(\varphi)$ 表示 φ 中出现的对象名的集合。

```
tableau 算法
define procedure sat(φ)
{ 创建初始节点 N0;
    if (ob(φ)为空集) l(N0) = {φ, □(x0:T)};
    else l(N0) = {φ} ∪ {□(a:T) | a ∈ ob(φ)};
    G = {}; cons(N0, NIL, -);  elim ( );
    if (G 非空) return satisfiable;
    return unsatisfiable;
}
define procedure cons(N, pre, a)
{ SatN = saturate(N);
    for (SatN 中的每一个节点 Ni)
    { 合并 l(N)中的同类变元; ext(Ni, pre, a);}
  return;
}
define procedure saturate(N)
{ i = 2;
    while (i < 6)
    { if (图 i 中规则 r 可应用到 l(N)中的某约束)
        { for (r 在该约束上的每一个可能输出 X)
            {创建 N 的拷贝 NX;
             l(NX) = l(NX) ∪ X; SatX = saturate(NX);}
          return ∪X为r的一个可能输出(SatX);}
    i = i + 1;} return {N};
}
define procedure ext(N, pre, a)
{ if (N 被 G 中 N1 全局阻塞)
      创建标志为 a, pre 到 N1 的有向边;
  else
    { G = G ∪ {N};
      if (l(N)有冲突) return;
      if (pre 非 NIL)
        创建标志为 a, pre 到 N 的有向边;
      if (全局规则 r 可应用到 l(N))
        {令(S1, a)为 r 的输出;构造新节点 N2;
         l(N2) = S1;合并 l(N2)中的同类变元;
         cons(N2, N, a); } }
  return;
}
define procedure elim( )
{ for (G 的每个节点 N)
   if (l(N)有冲突) G = G/{N};
while (G 非空)
{ changed = false;
    for (G 的每个节点 N)
   { if (S 中含有不能实现的预测)
     {G = G/{N}; changed = true;} }
  if (changed = = false) return;
}
```

在第一阶段,tableau 算法从初始节点 N_0 开始,不断按中 cons(N, pre, a) 描述的策略应用以下 tableau 规则 1)5),直到没有规则可以应用,最终获得一个完全 tableau。cons(N,pre, a) 子程序执行又可以分为两个过程,第一个过程是对节点 N 的约束系统施加局部规则的过程,称之为浸透过程,当没有局部规则可应用到节点 N 的约束系统上时,称节点 N 的约束系统已饱和,称此时的 N 节点为饱和节点。可以观察到,有些局部规则是析取规则,如 $\rightarrow_{\neg\wedge}$,$\rightarrow_{UA(q)}$ 等规则,有多个可能输出,

因此,节点 N 经过浸透过程最终可能生成多个饱和节点。第二个过程称为动—时态扩展过程,在该过程中,算法判断饱和节点 N 是否被其他节点全局阻塞,它的约束系统是否有冲突,如果都没有且全局规则可以作用到 N 的约束系统上,则扩展出 N 的后续节点。

1)局部公式规则

$\rightarrow_{\wedge}$ 规则:条件:$\alpha\wedge\beta\in S,\alpha\notin S$ 或 $\beta\notin S$。

动作:输出 $\{\alpha,\beta\}$。

$\rightarrow_{\neg\wedge}$ 规则:条件:$\neg(\alpha\wedge\beta)\in S,\neg\alpha\notin S,\neg\beta\notin S$。

动作:输出 $\{\neg\alpha\}$ 或 $\{\neg\beta\}$。

$\rightarrow_{UA}$ 规则:条件:$\alpha U^A\beta\in S$ 且不存在 $q\in Q_0$ 使得 $\alpha U^{A(q)}\beta\in S$。

动作:输出 $\{\alpha U^{A(q)}\beta\}$,其中 $q\in Q_0$。

$\rightarrow_{\neg UA}$ 规则:条件:$\neg(\alpha U^A\beta)\in S$ 且存在 $q\in Q_0$ 满足 $\neg(\alpha U^{A(q)}\beta)\notin S$。

动作:输出 $\{\neg(\alpha U^{A(q)}\beta)\mid q\in Q_0\}$。

$\rightarrow_{UA(q)}$ 规则:条件:$\alpha U^{A(q)}\beta\in S$,对任意 $q'\in\rho(q,a)$,不存在 $\{\alpha,\langle a\rangle\alpha U^{A(q')}\beta\}\subseteq S$,且如果 $q\in F,\beta\notin S$。

动作:如果 $q\in F$,输出 $\{\beta\}$ 或 $\{\alpha,\langle a\rangle\alpha U^{A(q')}\beta\}$;否则,输出 $\{\alpha,\langle a\rangle\alpha U^{A(q')}\beta\}$,其中 $q'\in\rho(q,a)$。

$\rightarrow_{\neg UA(q)}$ 规则:条件:a)$\neg(\alpha U^{A(q)}\beta)\in S$,b)如果 $q\in F,\neg\beta\notin S$,或者 $\neg\alpha\notin S$ 且 $\{\neg\langle a\rangle(\alpha U^{A(q')}\beta)\mid q'\in\rho(q,a)\}$ 不包含于 S;否则,$\neg\alpha\notin S$ 且 $\{\neg\langle a\rangle(\alpha U^{A(q')}\beta)\mid q'\in\rho(q,a)\}$ 不包含于 S。

动作:如果 $q\notin F$,输出 $\{\neg\alpha\}$ 或 $\{\neg\langle a\rangle(\alpha UA^{(q')}\beta)\mid q'\in\rho(q,a)\}$;否则,输出 $\{\neg\alpha,\neg\beta\}$ 或 $\{\neg\langle a\rangle(\alpha U^{A(q')}\beta)\mid q'\in\rho(q,a)\}\cup\{\neg\beta\}$。

2)局部非生成概念规则

$\rightarrow_{=}$ 规则:条件:$C=\top\in S,x$ 在 S 中出现且 $x:C\notin S$。

动作:输出 $\{x:C\}$。

$\rightarrow_{\cap}$ 规则:条件:$x:C\quad D\in S,x:C\notin S$ 或 $x:D\notin S$。

动作:输出 $\{x:C,x:D\}$。

$\rightarrow_{\neg\cap}$ 规则:条件:$x:\neg(C\quad D)\in S,x:\neg C\notin S,x:\neg D\notin S$。

动作:输出 $\{x:\neg C\}$ 或 $\{x:\neg D\}$。

$\rightarrow_{\neg\exists}$ 规则:条件:$x:\neg\exists R.C\in S,xRy\in S,y:\neg C\notin S$。

动作:输出 $\{y:\neg C\}$。

$\rightarrow_{UAC}$ 规则:条件:$x:CU^AD\in S$ 且不存在 $q\in Q_0$ 使得 $x:CU^{A(q)}D\in S$。

动作:输出 $\{x:CU^{A(q)}D\}$,其中 $q\in Q_0$。

$\rightarrow_{\neg UAC}$ 规则:条件:$x:\neg(CU^AD)\in S$ 且存在 $q\in Q_0$ 满足 $x:\neg(CU^{A(q)}D)\notin S$。

动作:输出 $\{x:\neg(CU^{A(q)}D)\mid q\in Q_0\}$。

$\rightarrow_{UA(q)C}$ 规则:条件:$x:CU^{A(q)}D\in S$,对任意 $q'\in\rho(q,a)$,不存在 $\{x:C,x:\langle a\rangle CUA^{(q')}D\}\subseteq S$,且如果 $q\in F,x:D\notin S$。

动作:如果 $q\in F$,输出 $\{x:D\}$ 或 $\{x:C,x:\langle a\rangle CU^{A(q')}D\}$;否则,输出 $\{x:C,x:\langle a\rangle CU^{A(q')}D\}$,其中 $q'\in\rho(q,a)$。

$\rightarrow_{\neg UA(q)C}$ 规则:条件:a)$x:\neg(CU^{A(q)}D)\in S$,b)如果 $q\in F,x:\neg D\notin S$,或者,$x:\neg C\notin S$ 且 $\{x:\neg\langle a\rangle(CU^{A(q')}D)\mid q'\in\rho(q,a)\}$ 不包含于 S;否则,$x:\neg C\notin S$ 且 $\{x:\neg\langle a\rangle(CU^{A(q')}D)\mid q'\in\rho(q,a)\}$ 不包含于 S。

动作:如果 $q\notin F$,输出 $\{x:\neg C\}$ 或 $\{x:\neg\langle a\rangle(CU^{A(q')}D)\mid q'\in\rho(q,a)\}$;否则,输出 $\{x:\neg C,x:\neg D\}$ 或 $\{x:\neg\langle a\rangle(CU^{A(q')}D)\mid q'\in\rho(q,a)\}\cup\{x:\neg D\}$。

3)局部生成概念规则

$\rightarrow_{\neg=}$ 规则:条件:$\neg(C=\top)\in S$ 且不存在 x 满足 $x:\neg C\in S$。

动作:输出 $\{x:\neg C\}$,其中 x 是新引入的变元。

$\rightarrow_{\exists}$ 规则:条件:$x:\exists R.C\in S$,不存在 y 满足 $\{xRy,y:C\}\in S$。

动作:如果 x 被变元 x' 局部阻塞,输出 $\{xRy\}$,其中变元 y 满足 $x'Ry\in S$;如果 x 未被阻塞,输出 $\{y:C,xRy\}$,其中 y 是新引入的变元。

4)$\mathrm{DLTL}^A_{\mathrm{ALC}}(\Sigma)$ 的 $\rightarrow\neg\langle\rangle$ 规则(局部规则)

$\rightarrow_{\neg\langle\rangle}$ 规则:条件:a)不存在 $\langle a\rangle\alpha\in S$ 或 $x:\langle a\rangle C\in S,a\in\Sigma$,b)$\neg\langle\alpha\rangle C\in S$ 或 $x:\neg\langle a\rangle C\in S,a\in\Sigma$。

动作:输出 $\{\varphi\mid\varphi$ 形如 $\langle a\rangle\neg\alpha,\langle a\rangle\neg\alpha\in S$ 或 φ 形如 $x:\langle a\rangle\neg C,x:\neg\langle a\rangle C\in S\}$,其中 $a\in\{a\mid\neg\langle a\rangle\alpha\in S$ 或 $x:\neg\langle a\rangle C\in S\}$。

5)$\mathrm{DLTL}^A_{\mathrm{ALC}}(\Sigma)$ 的全局规则

$\rightarrow_{\langle\rangle}$ 规则:条件:$\langle a\rangle\alpha\in S$ 或 $x:\langle a\rangle C\in S$。

动作:输出 $(\{\alpha\mid\langle a\rangle\alpha\in S\}\cup\{x:C\mid x:\langle a\rangle C\in S\},a)$。

由于 $\mathrm{DLTL}^A_{\mathrm{ALC}}(\Sigma)$ 允许一般概念相等形式的描述逻辑公式,在浸透过程中,可能在反复应用 $\rightarrow_{\exists}$ 规则时引入无穷多的变元,为此应用了 Baader 提出的变元阻塞技术,并称之为局部阻塞(参见规则 3)中的 $\rightarrow_{\exists}$ 规则):如果约束系统 S 中的变元 x、x' 满足 $x<x'$(且 $\{C\mid(x:C)\in S\}\subseteq\{C\mid(x':C)\in S\}$,则称 x 被 x' 所阻塞[11]。

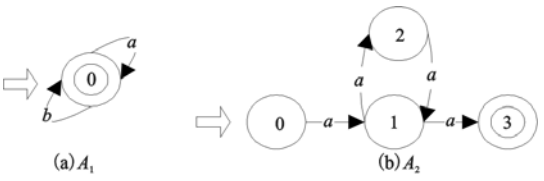
在动—时态扩展过程中,使用了一种类似于文献[6]中应用的全局阻塞技术以避免因不停生成新的 tableau 节点使得算法不得终止。如果一个新节点 N 的约束系统 $l(N)$ 约束等价于 tableau G 中的一个已存在的节点 N' 的约束系统 $l(N')$,则 N 被 N' 阻塞。所谓约束系统 S 约束等价于 S',指的是存在一个从 S 中所有变元到 S' 中所有变元的双射 f,使得如果用 $f(x)$ 去替代 S 中的每个变元 x,则 S 变为 S'。

在第二阶段,按 tableau 算法中 elim( )所描述,迭代遍历 φ 的完全 tableau G,删除那些没有后续节点、标记约束系统中有冲突或有不可实现的预测的节点。直到在某次遍历 G 后,G 为空或没有变化。

称约束系统有冲突当且仅当约束系统包含有形如 $\neg\top$ 或 $x:\neg\top$ 的公式,或者如下形式的公式对:$(x:C,x:\neg C)$,$(\alpha,\neg\alpha)$,$(\langle a_i\rangle\alpha,\langle a_j\rangle\beta)$,$(\langle a_i\rangle\alpha,x:\langle a_j\rangle C)$,$(\langle a_i\rangle x:C,y:\langle a_j\rangle D)$,其中 a_i、a_j 为不同的原子动作。

n_0 中一个含 U^A 算子的预测 φ 是可实现的当且仅当 G 中存在一条从该预测所在节点 n_0 开始且无回路的路径 $n_0,\cdots,n_m$,使得以下两个条件成立:a)从 n_0 到 n_m 有向边上的动作序列 $\sigma\in L(A)$;b)假设 S_i 是 n_i 的约束系统。如果 φ 形如 $\alpha U^A\beta$,则 $\beta\in S_m$ 且对任意 $i\neq m,\alpha\in S_i$;如果 φ 形如 $a:CU^AD$,则 $(a:D)\in S_m$ 且对任意 $i\neq m,(a:C)\in S_i$;如果 φ 形如 $x:CU^AD,x$ 为变元,则 S_0,S_1,…,S_m 中分别存在变元 $x_0,\cdots,x_m$,这些变元满足 $x_0=x,(x_m:D)\in S_m$,$\{CU^AD\mid(x_{i-1}:\langle a\rangle CU^AD)\in S_{i-1}\}\subseteq\{CU^AD\mid(x_i:CU^AD)\in S_i\}$,其中 $1\leqslant i\leqslant m$。

下面是一个构造 tableau 的例子。令 $\Sigma=\{a,b\}$,要判断 $\mathrm{DLTL}_{\mathrm{ALC}}(\Sigma)$ 公式 $d:\langle a;a^*;a\rangle C\wedge\Box(d:\neg C)$ 是否可满足。首先,展开 $d:\langle a;a^*;a\rangle C\wedge\Box(d:\neg C)$ 并构造它的等价 $\mathrm{DLTL}^A_{\mathrm{ALC}}(\Sigma)$ 公式 $\varphi=d:\top U^{A_2}C\wedge\neg(\top U^{A1}d:C)$,其中自动机 A_1 和 A_2 如图 1(a)(b)所示。φ 的完全 tableau 如图 2 所示。本文简化了一些不重要的细节,根据 tableau 算法,φ 的完全 tableau 的节点都将在算法第二阶段被删除,φ 不可满足。



(a)A_1　　(b)A_2

图1　非确定有限自动机

2.3　tableau 算法性质

给定一个 $\mathrm{DLTL}^A_{\mathrm{ALC}}(\Sigma)$ 公式 φ,用 $ob(\varphi)$ 表示 φ 中的对象名集合,用 $\mathrm{rol}(\varphi)$ 表示 φ 中的角色名集合。φ 的概念闭包 $\mathrm{con}(\varphi)$ 定义如下:

$\top$ 和出现在 φ 中的所有 $\mathrm{DLTL}^A_{\mathrm{ALC}}(\Sigma)$ 子概念都属于 $\mathrm{con}(\varphi)$;

如果 $CU^AD\in\mathrm{con}(\varphi)$,$A=(\Sigma,Q,\rho,Q_0,F)$,则对任意 $q\in Q$,$CU^{A(q)}D\in\mathrm{con}(\varphi)$ 且对任意满足 $q'\in\rho(q,a)$ 的 (q',a),$\langle a\rangle CU^{A(q')}D\in\mathrm{con}(\varphi)$;

如果 $C\in\mathrm{con}(\varphi)$,则 $\neg C\in\mathrm{con}(\varphi)$。

φ的公式闭包 sub(φ)定义如下：

⊤和 φ 中出现的所有 $DLTL^{A}_{ALC}(\Sigma)$子公式都属于 sub(φ)；

如果 $a \in ob(\varphi)$，则$(a:C) \in sub(\varphi)$，其中 $C \in con(\varphi)$；

如果 $\alpha U^{A} \beta \in sub(\varphi)$，$A=(\Sigma, Q, \rho, Q_0, F)$，则对任意 $q \in Q$，$\alpha U^{A(q)} \beta \in sub(\varphi)$且对任意满足 $q' \in \rho(q,a)$的(q',a)，$\langle a \rangle \alpha U^{A(q')} \beta \in sub(\varphi)$；

如果 $\alpha \in sub(\varphi)$，则$\neg \alpha \in sub(\varphi)$。

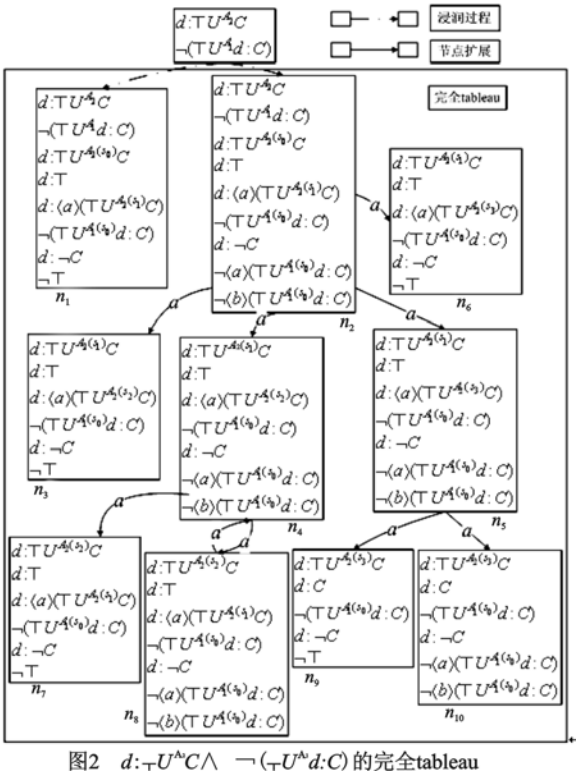


图2　$d:\top U^{A_0} C \wedge \neg(\top U^{A_0} d:C)$的完全tableau

命题 1　在 φ 的 tableau 中，一个饱和节点的约束系统仅包含有限多个约束。

证明　自动机 $A=(\Sigma, Q, \rho, Q_0, F)$的长度用$|A|$表示，$|A|$大于 A 中有向边和节点的数量，算子 U^{A} 的长度等于$|A|+1$。假设输入 φ 的长度为 n，则根据 con(φ)和 sub(φ)定义，依公式结构归纳可证明$|con(\varphi)|$和$|sub(\varphi)|$均属于 $O(n)$（文中用$|\cdot|$表示集合· 中元素的个数）。

根据约束的三种不同形式——$DLTL^{A}_{ALC}(\Sigma)$公式，yRx，$x:C$（x 为变元，y 为变元或对象名）分别分析 tableau 中约束的个数。a）由 tableau 规则可知，φ 的 tableau 中出现的 $DLTL^{A}_{ALC}(\Sigma)$公式必然都属于 sub(φ)，其个数属于 $O(n)$；b）由 tableau 规则可知，φ 的 tableau 中出现的概念必然都属于 con(φ)，φ 的所有概念类型的个数等于$2^{|con(\varphi)|}$。由于同类变元被合并，一个 tableau 节点 n 在浸透前其约束系统中的对象和变元的总数 p_n 少于 $v(\varphi)=2^{|con(\varphi)|}+ob(\varphi)$。浸透过程中由于采用了局部阻塞技术，使得对任意一个概念类型 T，节点 n 在饱和时其约束系统中最多有(p_n+1)变元代表 T。因此，饱和节点的约束系统中；形如 yRx 的约束个数少于$(v(\varphi)(2^{|con(\varphi)|}+ob(\varphi))\times|rol(\varphi)|\times(v(\varphi)\times 2^{|con(\varphi)|})$，属于 $2^{O(n)}$，形如 $x:C$ 的约束个数少于 $(v(\varphi)\times 2^{|con(\varphi)|})\times|con(\varphi)|$，也属于 $2^{O(n)}$（x 为变元，y 为变元或对象名）。由 a）b）可知，在 φ 的 tableau 节点中，一个饱和节点的约束系统中约束的个数属于 $2^{O(n)}$，是有限的。

命题 2　在构造 φ 的完全 tableau 期间，产生的饱和节点的个数有限。

证明　φ 的极大约束系统 S_φ 满足以下条件：

S_φ 中有 $2^{con(\varphi)}-1$ 个彼此概念类型不同的变元；

$\{a \mid a$ 为在 S_φ 中出现的对象名$\}=ob(\varphi)$，且对任意 $a \in ob(\varphi)$，$\{C \mid a:C \in S_\varphi\}=con(\varphi)$；

$\{R \mid R$ 为在 S_φ 中出现的角色名$\}=rol(\varphi)$，且对任意 $R \in rol(\varphi)$，任意 S_φ 中出现的变元 x，和任意 $y \in ob(\varphi) \cup \{S_\varphi$ 中出现的变元$\}$，均有 $yRx \in S_\varphi$。

a）φ 的完全 tableau 中，有后续的节点没有被阻塞，彼此间约束系统约束不等价，但均约束等价于 S_φ 的某个子集，因此有后续的节点总数不超过 $2^{|S_\varphi|}$，$|S_\varphi|=|sub(\varphi)|+2^{|con(\varphi)|}\times|con(\varphi)|+(2^{|con(\varphi)|}+ob(\varphi))\times|rol(\varphi)|\times 2^{|con(\varphi)|}$ 属于 $2^{O(n)}$。

b）由命题 1 和 tableau 算法的浸透过程可以看出，给定一个标记有初始约束系统的节点，最多只需经过 $2^{O(n)}$ 运用 tableau 局部规则即可使得其约束系统饱和，而且，由于每个 tableau 局部规则的可能的输出个数都是有限的，所以，一个节点的后续饱和节点的个数属于 $2^{(2^{(O(n))})}$（表示幂运算）。

由 a）b）可知，一个完全 tableau 中的节点的总数属于 $2^{(2^{(O(n))})}$，是有限的。

定理 1　$DLTL^{A}_{ALC}(\Sigma)$的 tableau 算法可终止。

证明　由命题 1 和 2 可推得，给定 φ，构造 φ 的完全 tableau 只需要有限多次运用 tableau 规则，因此，tableau 算法执行的第一阶段是可终止的。在 tableau 算法执行的第二阶段，每次迭代过程最多遍历有限多个 tableau 节点，除了最后一次迭代，每次至少删除一个节点，因此，迭代次数有限，tableau 算法执行的第二阶段也是可终止的。

定理 2　$DLTL^{A}_{ALC}(\Sigma)$的 tableau 算法是完全的和合理的。

证明　略，证明思路可参考文献[6]的定理 7.1 和定理 8.1 的证明。

3　结束语

本文提出了一类描述逻辑的动态线性扩展 $DLTL_{DL}$。$DLTL_{DL}$是一类动态知识表示系统，文中提出了判别 $DLTL_{DL}$公式可满足的 tableau 算法并分析了 $DLTL_{DL}$的可计算特性。

由于 $DLTL_{DL}$可以同时表达变化域的时间特性和组合动作，因此应用 $DLTL_{DL}$对动态系统建模很直观，比方说，在动作建模方面，最一般的动作建模方式通常基于情景演算中[13]的状态变化的观念，通过给定动作的前提条件和后置条件（有时又称效果），将动作定义为从一个状态到另一个状态的转换函数。利用 $DLTL_{DL}$可以方便地基于以上观念对动作建模，可以用$\Box(\langle a \rangle \top \rightarrow \alpha)$形式的公式表达执行动作 a 的前提条件为 α，用$\Box(\alpha \rightarrow [a]\beta)$或$\Box(C \rightarrow [a]D)$形式的公式表达动作 a 的后置条件。如对引言中旅客乘机规划的例子，对某航班服务 a_i，用$\Box(\langle a_i \rangle \top \rightarrow$（旅客 AtCity 广州））表示搭乘航班 a_i 的前提是旅客从广州出发，用$\Box([a_i]$（旅客 AtCity 武汉）$\wedge$（旅客 ArrivedByPlane 空客））表示旅客搭乘航班 a_i 后，旅客抵达武汉且途中乘坐的是空客的飞机。

描述逻辑的表达能力远强于采用命题逻辑，$DLTL_{DL}$的表达能力因而远强于基于命题逻辑的动态知识表示系统，如命题 STRIPS 系统[12]。而且，由于一般的动态算子和时态算子都可由 U^{π} 算子定义出来，因此，一般时态描述逻辑和动态描述逻辑都是基于相同描述逻辑的 $DLTL_{DL}$的片段，从而可知 $DLTL_{DL}$

的表达能力要强于以相同 DL 为基础的时态或动态描述逻辑扩展。

有些形式系统也许比 $DLTL_{DL}$ 具有更强的表达能力,例如,情景演算[13]和一些基于一阶逻辑扩展的时序逻辑[14]或动态逻辑[8],遗憾的是,它们表达能力虽强,其基本推理问题却一般不是可判定的。这也是近年来,包括本文在内的不少动态知识表示系统研究基于描述逻辑的一大动机,主要目的就是利用描述逻辑的良好可计算性和表达能力,对其扩展有很大可能既保持其基本推理问题可判定,又保证具有良好的表达能力。

笔者打算将 $DLTL_{DL}$ 应用于语义 Web 服务的组合研究。语义 Web 服务的标准语言 OWL-S[15] 基于描述逻辑 SHOIQ(D),而本文研究的 $DLTL_{DL}$ 基于基本描述逻辑 ALC,在域描述方面还有所不足,实际应用中必然还有不少限制。因此,在后续研究中,笔者重点将研究基于表达能力更强的描述逻辑的,具有良好可应用性的 $DLTL_{DL}$。

参考文献:

[1] BAADER F, CALVANSES D, McGUINNESS D, *et al*. The description logic handbook: theory, implementation, and applications[M]. Cambridge: Cambridge University Press, 2003.

[2] WOLTER F, ZAKHARYASCHEV M. Temporalizing description logics[M]//GABBAY D, De RIJKE M. Frontiers of Combining Systems 2. [S.l.]: Studies Press/Wiley, 1999:379-402.

[3] ARTALE A, FRANCONI E. Temporal description logics[M]//GABBAY D, FISHER M, VILA L. Handbook of Time and Temporal Reasoning in Artificial Intelligence. [S.l.]: Elsevier, 2005: 375-388.

[4] WOLTER F, ZAKHARYASCHEV M. Dynamic description logics [C]//ZAKHARYASCHEV M, SEGERBERG K, RIJKE M, *et al*. Advances in Modal Logic. Stanford: CSLI Publications, 2000:449-463.

[5] 常亮,史忠植,陈立民,等. 一类扩展的动态描述逻辑[J]. 软件学报,2010,21(1):1-13.

[6] STURM H, WOLTER F. A tableau calculus for temporal description logic: the expanding domain case[J]. Journal of Logic and Computation,2002,12(5): 809-838.

[7] HENRIKSEN J G, THIGARAJAN P S. Dynamic linear time temporal logic[J]. Annals of Pure and Applied logic,1999,96(1-3):187-207.

[8] HAREL D, KOZEN D, TIURYN J. Dynamic logic[M]. Cambridge: MIT Press, 2000.

[9] HOPCROFT J E, MOTWANI R, ULLMAN J D. Introduction to automata theory, languages and computation[M]. MA: Addison Wesley Reading,1979.

[10] WOLPER P. The tableau method for temporal logic: an overview[J]. Logique et Analyse,1985,28(110-111): 119-136.

[11] BAADER F, SATTLER U. Tableau algorithms for description logics [J]. Studia Logica,2001,69(1):5-40.

[12] BYLANDER T. The computational complexity of propositional STRIPS planning[J]. Artificial Intelligence,1994,69(1-2):165-204.

[13] REITER R. Knowledge in action: logical foundations for specifying and implementing dynamical systems[M]. Cambridge: MIT Press, 2001.

[14] KRÖGER F, MERZ S. Temporal logic and state systems[M]. New York: Springer-Verlag,2008.

[15] The W3C Consortium. OWL-S: semantic markup for Web services [EB/OL]. (2004-11-22)[2011-08-01]. http://www.w3.org/Submission/OWL-S/.

(上接第 535 页)

类似地,可求

$r^{12}=r_1^{12}\wedge r_2^{12}\wedge r_3^{12}\wedge r_4^{12}\wedge r_5^{12}=(0.4,0.5)$

$r^{13}=r_1^{13}\wedge r_2^{13}\wedge r_3^{13}\wedge r_4^{13}\wedge r_5^{13}=(0.3,0.5)$

这里 r^{ij} 表示方案 A_i 综合购买者和专家 e_j 评价得到的评价结果。利用 $r^i=\omega_1\cdot r^{i1}\vee\omega_2\cdot r^{i2}\vee\omega_2\cdot r^{i3}=(0.87,0.07)$ 得到方案 A_i 的最后评价结果。

类似地,可求 $r^2=(0.83,0.13)$,$r^3=(0.82,0.10)$。

由于三个方案的最终评价结果均不相同,因此不需要计算其可靠性程度。

因此,方案 A_1 是最佳方案。

4　结束语

针对方案有偏好的直觉模糊数多属性群决策问题,通过建立推理决策模型,从而综合考虑了专家和决策者的偏好,进而给出了一种新的决策方法。与 AHP 方法相比,AHP 方法只考虑了决策者(专家)的意见,而本文所提出的决策方法不仅考虑了专家的专业意见,同时也参考了决策者的意愿,即同时考虑了主观意愿和客观评价,使得决策者可以获得更合理、更有帮助的决策结果。此方法可以推广到对方案有偏好的区间模糊多属性群决策问题。

参考文献:

[1] 田志刚,卢兴华,宋一中. 关于直觉决策几个问题的研究[C]//决策科学理论与方法——中国系统工程学会决策科学专业委员会第四届学术年会论文集. 2001:245-251.

[2] 黄孟藩,王凤彬. 决策行为和决策心理[M]. 北京:机械工业出版社,1995.

[3] 王毅, 雷英杰. 一种直觉模糊熵的构造方法[J]. 控制与决策,2007,22(12):1390-1394.

[4] 徐泽水. 直觉模糊信息集成理论与应用[M]. 北京:科学出版社,2008.

[5] BISWAS B. On fuzzy sets and intuitionistic fuzzy sets[J]. N IFS, 1997,3: 3-11.

[6] 雷英杰,王宝树, 路艳丽. 基于直觉模糊逻辑的近似推理方法[J]. 控制与决策,2006,21(3):305-309.

[7] 王毅, 雷英杰, 路艳丽. 基于直觉模糊集的多属性模糊决策方法[J]. 系统工程与电子技术,2007,29(12):2060-2063.

[8] 林琳. 直觉模糊集在近似推理与决策中的应用[D]. 大连:大连理工大学,2006.

[9] 施丽娟,黄天民. 基于投影法的直觉模糊多属性决策两阶段优化模型[C]//第十一届中国管理科学技术年会论文集. 2009:147-150.

[10] 莫生红,吕宏芳,李明伟. 层次分析法在市民购房决策中的应用[J]. 经济论坛,2007(19):49-51.

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

[12] ATANASSOVE K. Intuitionistic fuzzy sets[J]. Fuzzy Sets and Systems,1986,20(1):87-96.

[13] ATANASSOVE K. More on intuitionistic fuzzy sets[J]. Fuzzy Sets and Systems,1989,33(1): 37-46.

[14] ATANASSOVE K. New operations defined over the intuitionistic fuzzy sets[J]. Fuzzy Sets and Systems,1994,61(2):137-142.

[15] 胡于进,凌玲. 决策支持系统的开发与应用[M]. 北京:机械工业出版社,2006.