

基于动态描述逻辑 DDL 的动作理论

常 亮¹ 陈立民²

(桂林电子科技大学计算机科学与工程学院 桂林 541004)¹

(中国科学院计算技术研究所智能信息处理重点实验室 北京 100190)²

摘 要 基于一阶谓词逻辑或高阶逻辑的动作理论与采用命题语言的动作理论之间存在一个关于描述和推理能力的鸿沟;作为描述逻辑的动态扩展,动态描述逻辑 DDL 为基于描述逻辑的动作刻画和推理提供了一种途径。系统地研究了基于 DDL 的动作表示和推理问题。首先,在应用描述逻辑对静态领域知识进行刻画的基础上,引入带参数的原子动作定义式和带参数的复杂动作定义式,刻画出关于动作的知识;这两部分知识与关于具体状态的知识一起构成基于 DDL 的知识表示系统。接下来,针对该系统中刻画的各种知识,对相关推理问题进行了严格定义,给出了相应的推理机制和算法。最后探讨了基于 DDL 的动作理论在智能主体建模中的应用。基于 DDL 的动作理论既具有较强的知识表示能力,又提供了丰富的推理服务,适用于语义 Web 环境下对动作的刻画和推理。

关键词 动态描述逻辑,知识表示和推理,动作理论,智能主体,语义 Web

中图分类号 TP301 **文献标识码** A

Action Theory Based on the Dynamic Description Logic DDL

CHANG Liang¹ CHEN Li-min²

(School of Computer Science and Technology, Guilin University of Electronic Technology, Guilin 541004, China)¹

(Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)²

Abstract There is a gap on expressive power and reasoning ability between the action theories which are based on first-order or higher-order logics and the action theories which are only propositional. As a kind of dynamic extensions of description logics, the dynamic description logic DDL provides an approach for describing and reasoning about actions. An action theory based on DDL was presented and studied systematically. Firstly, based on a representation of static domain knowledge with description logics, the parameterized atomic action definitions and the parameterized complex action definitions were introduced for describing the knowledge of actions; both of these knowledge, and together with the knowledge on the state of the world, are unified as a DDL-based knowledge representation system. Secondly, many reasoning tasks on the knowledge represented in this system were formally defined; corresponding reasoning mechanisms were also provided. Finally, the application of this action theory for the modeling of intelligent agents was discussed. The action theory based on DDL offers not only considerable expressive power but also attractive reasoning services; it is suitable for the description and reasoning of actions in the environment of the semantic Web.

Keywords Dynamic description logic, Knowledge representation and reasoning, Action theory, Intelligent agent, Semantic Web

对动作的刻画和推理是知识表示和推理中重要的研究课题,是当前成为研究热点的语义 Web 服务和智能主体的理论基础。目前比较成熟的动作理论可以分为两类:基于一阶谓词逻辑或高阶逻辑的动作理论;采用命题语言的动作理论。前者以情景演算^[1]、流演算^[2]和 STRIPS 系统^[3]为代表,它们的共同点是采用一阶谓词或高阶逻辑中的公式刻画世界状态、动作的前提条件和执行动作后产生的影响,具有很强的描述能力,然而相关推理问题却是不可判定的,限制了关于动作的推理能力。后者以命题 STRIPS 系统、基于命题动态逻辑的动作理论^[4]和基于命题时序逻辑的动作理论^[5]为代表,它

们的共同点是采用命题公式刻画世界状态、动作的前提条件以及执行动作后产生的影响等,保证了相关推理问题的可判定性,但描述能力却受到很大限制。因此,这两类动作理论之间存在一个关于描述能力与推理能力的鸿沟。

描述逻辑是一类用于表示和推理静态领域知识的形式化工具,是当前作为研究热点的语义 Web 的重要支撑。描述逻辑的主要特点在于提供了命题逻辑所无法比拟的知识表示能力,同时保证了相关推理问题的可判定性。

针对描述逻辑只能处理静态领域知识的局限,文献[6]将描述逻辑、动态逻辑和动作理论有机地结合起来,提出了一种

到稿日期:2010-08-15 返修日期:2010-12-06 本文受国家自然科学基金(60903079,60775035,60963010)资助。

常 亮(1980—),男,博士,副教授,CCF 会员,主要研究方向为知识表示与知识推理、形式化方法、语义 Web 等,E-mail:changli@guet.edu.cn;

陈立民(1982—),男,博士生,主要研究方向为描述逻辑、语义 Web。

动态描述逻辑 DDL。文献[7]为 DDL 设计了与描述逻辑中采用的开世界假设相兼容的判定算法。文献[8,9]将 DDL 应用于对语义 Web 服务的建模、推理和组合。

在上述工作的基础上,本文系统地研究了基于动态描述逻辑的动作表示和推理问题。首先,以描述逻辑中 RBox 和 TBox 刻画的领域知识为背景,依次引入带参数的原子动作定义式和带参数的复杂动作定义式,形成关于原子动作的知识模块 AActBox 和关于复杂动作的知识模块 CActBox,然后将 RBox, TBox, AActBox, CActBox 以及描述逻辑中的 ABox 一起构成基于 DDL 的知识库;该知识库在描述逻辑的基础上实现了对静态领域知识、关于动作的知识和应用领域具体状态的统一刻画。接下来,对与知识库相关的各种推理问题进行了全面的考察,给出了在执行动作之后引起的 ABox 更新算法。最后将基于 DDL 的动作理论应用于智能主体建模,给出了智能主体的心智状态模型和基本操作。

1 动态描述逻辑 DDL

动态描述逻辑 DDL 是将描述逻辑、动态逻辑和动作理论结合起来后得到的一类具有二维特征的逻辑系统^[6,7]。对应于描述逻辑家族的一系列成员,可以相应地构造出具有不同描述能力的动态描述逻辑。为了显式地表明这种对应关系,用 $DDL(X)$ 表示在描述逻辑 X 中引入动态逻辑和动作理论后得到的动态描述逻辑系统,其中 X 可以是其中最典型的 ALC 到与本体语言 OWL 相对应的 SHOIQ 等具有不同描述能力的描述逻辑系统^[10]。

从语法上看,DDL(X)在描述逻辑 X 的基础上增加了 3 个方面的内容。首先,引入了原子动作定义式,将每个原子动作 α 刻画为二元组 (P, E) 的形式,其中 P 和 E 分别描述了动作 α 在执行之前必须满足的前提条件以及执行 α 之后产生的影响。其次,引入了动态逻辑中的顺序、测试、选择、迭代等动作构造符,在原子动作和公式的基础上对复杂动作进行刻画。最后,将任一动作 π 作为模态词,用于构造形如 $\langle \pi \rangle \varphi$ 和 $[\pi] \varphi$ 的公式,直观含义分别为“存在动作 π 的某种执行情况使得公式 φ 在执行结束时成立”和“动作 π 执行结束时公式 φ 必然成立”。

从语义上看,DDL(X)在描述逻辑语言 X 的语义解释的基础上引入了一个动态维,使得语义模型从整体上扩展为由多个可能世界构成的可能世界空间;其中的每个可能世界都被赋予描述逻辑语言 X 中的一个语义解释,而各个可能世界之间则通过动作联系起来。

下面以 X 为描述逻辑 SHOIQ 的情况为例,介绍相应的动态描述逻辑系统 $DDL(SHOIQ)$ 。

DDL(SHOIQ)的基本符号包括 4 个部分:描述逻辑 SHOIQ 的基本符号、由原子动作名组成的集合 N_A 、动作构造符“ $\cup$ ”、“ $\cdot$ ”、“ $\cdot^T$ ”和“ $\cdot^*$ ”以及公式构造符“ $\langle \cdot \rangle$ ”和“ $[\cdot]$ ”。从这些符号出发,可以递归地构造出 DDL(SHOIQ)中的角色、概念、公式和动作等 4 类基本成员。

DDL(SHOIQ)中的角色和概念分别与 SHOIQ 中的角色和概念具有相同的构造过程和语法规则。在此基础上,DDL(SHOIQ)中对角色包含公理、RBox、概念定义式、TBox、ABox 断言和 ABox 的定义都与描述逻辑 SHOIQ 中相应的定义相同^[10]。限于篇幅,这里不再详细列出。

DDL(SHOIQ)中的公式由如下产生式生成:

$$\varphi, \psi ::= \text{Ass} \mid \neg \varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \langle \pi \rangle \varphi \mid [\pi] \varphi$$

式中,Ass 代表描述逻辑 SHOIQ 中的任一 ABox 断言, π 为动作。

相对于某个 TBox $\mathcal{T}$,如果 ABox 断言 Ass 中不含有出现在 $\mathcal{T}$ 中概念定义式左边的概念,则将 Ass 称为一个简单公式。

DDL(SHOIQ)中的动作由如下产生式生成:

$$\pi, \pi' ::= \alpha \mid \varphi? \mid \pi \cup \pi' \mid \pi; \pi' \mid \pi^*$$

式中, $\alpha \in N_A$, φ 为公式。将 α 称为原子动作,将其它动作称为复杂动作。

DDL(SHOIQ)中通过原子动作定义式对每个原子动作进行进一步的刻画。每个原子动作定义式形如 $\alpha = (P, E)$,其中:(1) $\alpha \in N_A$,表示被定义的原子动作;(2) P 是由公式组成的有限集合,表示执行该动作之前必须满足的条件;(3) E 是由简单公式组成的有限集合,表示执行该动作之后将会产生的影响。

对于由原子动作定义式组成的有限集合 $\mathcal{A}$,如果每个原子动作名最多在 $\mathcal{A}$ 中原子动作定义式的左边出现一次,则称 $\mathcal{A}$ 为 ActBox。

DDL(SHOIQ)的语义模型为一个三元组 $M = (\Sigma, \Delta_M, I)$,其中:(1) Σ 是形如 $\Sigma = (W, \cdot^T)$ 的框架,其中 W 是由可能世界组成的非空集合,函数 $\cdot^T$ 将 N_A 中每个原子动作名 α ,映射为 W 上的某个二元关系 $\alpha^T \subseteq W \times W$;(2) Δ_M 是由个体组成的非空集合;(3) 函数 I 对 W 中每个可能世界 w 赋予描述逻辑 SHOIQ 中的一个解释 $I(w) = (\Delta_M, \cdot^{I(w)})$,并且要求在各个可能世界下对个体名的解释保持不变。

给定任一模型 $M = (\Sigma, \Delta_M, I)$ (其中 $\Sigma = (W, \cdot^T)$) 以及该模型中的任一可能世界 $w \in W$,DDL(SHOIQ)中对角色、概念、角色包含公理、概念定义式以及 ABox 断言的语义定义都与描述逻辑 SHOIQ 中相应的语义定义相同。

在此基础上,对公式的语义递归定义如下:

- (1) $(M, w) \models \text{Ass}$ iff 描述逻辑 SHOIQ 中的语义解释 $I(w) = (\Delta_M, \cdot^{I(w)})$ 满足 $I(w) \models \text{Ass}$;
- (2) $(M, w) \models \neg \varphi$ iff $(M, w) \not\models \varphi$ 不成立;
- (3) $(M, w) \models \varphi \vee \psi$ iff $(M, w) \models \varphi$ 或者 $(M, w) \models \psi$;
- (4) $(M, w) \models \varphi \wedge \psi$ iff $(M, w) \models \varphi$ 并且 $(M, w) \models \psi$;
- (5) $(M, w) \models \langle \pi \rangle \varphi$ iff 存在某个可能世界 $w' \in W$ 使得 $(w, w') \in \pi^T$ 并且 $(M, w') \models \varphi$;
- (6) $(M, w) \models [\pi] \varphi$ iff 对于 W 中满足 $(w, w') \in \pi^T$ 的任一可能世界 w' 都有 $(M, w') \models \varphi$ 。

DDL(SHOIQ)中对动作的语义递归定义如下:

- (1) $(\varphi?)^T := \{(w, w) \mid (M, w) \models \varphi\}$;
- (2) $(\pi \cup \pi')^T := \pi^T \cup \pi'^T$;
- (3) $(\pi; \pi')^T := \{(w, w') \mid \text{存在某个可能世界 } w'' \in W \text{ 使得 } (w, w'') \in \pi^T \text{ 和 } (w'', w') \in \pi'^T\}$;
- (4) $(\pi^*)^T := \pi^T$ 的自反传递闭包。

最后,对任一 ABox $\mathcal{A}$ 、RBox $\mathcal{R}$ 、TBox $\mathcal{T}$ 和 ActBox $\mathcal{A}$ 的语义定义如下:

- (1) $(M, w) \models \mathcal{A}$ 当且仅当对于任一公式 $\varphi \in \mathcal{A}$ 都有 $(M, w) \models \varphi$;
- (2) $M \models \mathcal{R}$ 当且仅当对于任一可能世界 $w \in W$ 以及任一角色包含公理 $R_1 \sqsubseteq R_2 \in \mathcal{R}$ 都有 $R_1^{I(w)} \subseteq R_2^{I(w)}$;

(3) $M \models \mathcal{T}$ 当且仅当对于任一可能世界 $w \in W$ 以及任一概念定义式 $D \equiv C \in \mathcal{T}$ 都有 $D^{I(w)} = C^{I(w)}$;

(4) $M \models \mathcal{A}$ 当且仅当对于任一原子动作定义式 $\alpha \equiv (P, E) \in \mathcal{A}$ 都有 $\alpha^T = \{(w, w') \mid$ ① 对于任一 $\psi \in P$ 都有 $(M, w) \models \psi$; ② 对于任一 $\varphi \in E$ 都有 $(M, w) \models \neg \varphi$ 和 $(M, w') \models \varphi$; ③ 对于满足 $\varphi \notin E$ 的任一简单公式 φ : 如果 $(M, w) \models \varphi$ 则有 $(M, w') \models \varphi$, 如果 $(M, w) \not\models \varphi$ 则有 $(M, w') \not\models \varphi\}$ 。

DDL(SHOIQ)中最基本的推理问题是判断公式的可满足性。称公式 φ 相对于 $RBox \mathcal{R}$, $TBox \mathcal{T}$ 和 $ActBox \mathcal{A}$ 是可满足的, 当且仅当存在某个模型 $M = (\Sigma, \Delta_M, I)$ (其中 $\Sigma = (W, \cdot^T)$) 以及某个可能世界 $w \in W$ 使得 $M \models \mathcal{R}, M \models \mathcal{T}, M \models \mathcal{A}$ 和 $(M, w) \models \varphi$ 。

2 基于动态描述逻辑 DDL 的知识表示

为了比较紧凑地刻画关于动作的知识, 本文在动态描述逻辑 DDL 的公式和原子动作定义式中引入个体变元。需要指出的是, 这些个体变元只起到占位符的作用; 我们不从逻辑系统的角度研究个体变元的量词修饰和语义指派等内容。

令 N_{IV} 是由个体变元组成的集合。对于动态描述逻辑 DDL 中公式的语法规则来说, 如果在个体名出现的地方都允许个体变元出现, 则称构造出来的式子为伪公式。

在此基础上, 可以引入带参数的原子动作定义式和带参数的复杂动作定义式。

定义 1 带参数的原子动作定义式形如 $\alpha(v_1, \dots, v_n) \equiv (P, E)$, 其中: (1) α 为字符串, 表示该动作的名字; (2) $(v_1, \dots, v_n)$ 是由出现在 P 和 E 中的所有个体变元组成的有限序列, 称为该原子动作的界面; (3) P 是由伪公式组成的有限集合; (4) E 是由简单伪公式组成的有限集合。

对于由带参数的原子动作定义式组成的有限集合 $aAct$, 如果 $aAct$ 中任意两个定义式左边的动作名都不相同, 则称 $aAct$ 为原子动作 Box, 简称为 AActBox。

每个带参数的原子动作定义式 $\alpha(v_1, \dots, v_n) \equiv (P, E)$ 都描述了一类原子动作。对应于界面 $(v_1, \dots, v_n)$, 相应地代入 n 个个体名 $p_1, \dots, p_n$ 之后, 将得到一个原子动作定义式 $\alpha(p_1, \dots, p_n) \equiv (P[v_1/p_1, \dots, v_n/p_n], E[v_1/p_1, \dots, v_n/p_n])$, 其中 $P[v_1/p_1, \dots, v_n/p_n]$ 和 $E[v_1/p_1, \dots, v_n/p_n]$ 是将在 P 和 E 中出现的 $v_1, \dots, v_n$ 分别用 $p_1, \dots, p_n$ 进行替换后得到的公式集合。

给定任意 n 个个体名或个体变元 $u_1, \dots, u_n$, 将 $\alpha(u_1, \dots, u_n)$ 称为以 $aAct$ 为参照的被定义的伪原子动作。

定义 2 带参数的复杂动作定义式形如 $A(v_1, \dots, v_n) \equiv \pi$, 其中: (1) A 为字符串, 表示该动作的名字; (2) $(v_1, \dots, v_n)$ 是由出现在 π 中的所有个体变元组成的有限序列, 称为该原子动作的界面; (3) π 由如下产生式生成

$$\pi, \pi' ::= \alpha(u_1, \dots, u_n) \mid \varphi \mid \pi \cup \pi' \mid \pi; \pi' \mid \pi^*$$

式中, $\alpha(u_1, \dots, u_n)$ 是以 $aAct$ 为参照的被定义的伪原子动作, φ 为伪公式。

对于由带参数的复杂动作定义式组成的有限集合 $cAct$, 如果 $cAct$ 中任意两个定义式左边的动作名都不相同, 则称 $cAct$ 为复杂动作 Box, 简称为 CActBox。

每个带参数的复杂动作定义式 $A(v_1, \dots, v_n) \equiv \pi$ 都描述了一类复杂动作。对应于界面 $(v_1, \dots, v_n)$, 相应地代入 n 个个体

名 $p_1, \dots, p_n$ 之后, 将得到一个复杂动作 $\pi[v_1/p_1, \dots, v_n/p_n]$ 。由于 $cAct$ 中各个定义式左边的动作名都不相同, 因而也可以用 $A(p_1, \dots, p_n)$ 来指代相应的复杂动作 $\pi[v_1/p_1, \dots, v_n/p_n]$ 。

最后, 可以引入基于动态描述逻辑 DDL 的知识库。

定义 3 基于动态描述逻辑 DDL 的知识库是一个五元组 $\mathcal{K} = (\mathcal{R}, \mathcal{T}, aAct, cAct, \mathcal{A})$, 其中的 $\mathcal{R}, \mathcal{T}, aAct, cAct$ 以及 $\mathcal{A}$ 分别为 $RBox, TBox, AActBox, CActBox$ 和 $ABox$ 。

知识库 $\mathcal{K}$ 包含了 3 个方面的知识: 由 $\mathcal{R}$ 和 $\mathcal{T}$ 表示的领域本体、由 $aAct$ 和 $cAct$ 表示的关于动作的知识以及由 $\mathcal{A}$ 刻画的关于应用领域具体状态的知识。各部分知识之间具有图 1 所示的层次关系。其中, 在刻画领域本体时发挥了描述逻辑的刻画能力, 既描述了静态领域知识, 又为动作和状态的刻画提供了词汇; 在刻画原子动作时通过描述逻辑公式分别对动作的前提条件和执行动作之后产生的影响进行了描述; 而在刻画复杂动作时则发挥了动态逻辑的刻画能力, 可以对顺序、测试、选择、迭代等控制结构进行刻画; 在刻画应用领域具体状态时再一次发挥了描述逻辑的刻画能力。

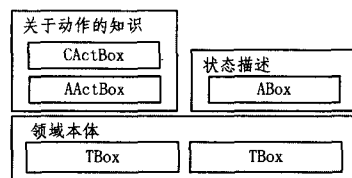


图 1 基于动态描述逻辑 DDL 的知识库结构

3 基于动态描述逻辑 DDL 的知识推理

根据知识库的构成, 将相关推理问题分为 4 大类: 关于领域本体的推理、关于状态的推理、关于动作的推理以及由动作的执行导致的状态更新问题。

关于领域本体的推理问题主要包括 $RBox$ 与 $TBox$ 的一致性问題以及概念的可满足性问题。关于状态的推理问题主要是指 $ABox$ 相对于 $RBox$ 和 $TBox$ 的一致性问题。DDL(例如 SHOIQ)中对这些推理问题的定义与描述逻辑(例如 SHOIQ)中的定义相同, 因此这些推理问题都可以通过描述逻辑中的推理机制来解决。

关于动作的推理问题又可以分为两部分: 第一部分主要包括判断原子动作定义式的一致性, 判断 AActBox 的一致性, 以及判断 CActBox 的一致性。下面分别进行讨论。

定义 4 称原子动作定义式 $\alpha \equiv (P, E)$ 相对于 $RBox \mathcal{R}$ 和 $TBox \mathcal{T}$ 是一致的, 当且仅当存在某个模型 $M = (\Sigma, \Delta_M, I)$ (其中 $\Sigma = (W, \cdot^T)$) 以及该模型中的某两个可能世界 $w, w' \in W$ 使得 $M \models \mathcal{R}, M \models \mathcal{T}, M \models \mathcal{A}$ 以及 $(w, w') \in \alpha^T$; 其中的 $\mathcal{A}$ 是由原子动作定义式 $\alpha \equiv (P, E)$ 组成的 ActBox。

显然, 根据定义, 原子动作定义式的一致性问题可以转换为公式的可满足性问题。

定理 1 原子动作定义式 $\alpha \equiv (P, E)$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是一致的当且仅当公式 $\langle \alpha \rangle \text{true}$ 相对于 $\mathcal{R}, \mathcal{T}$ 以及某个 ActBox $\mathcal{A} = \{\alpha \equiv (P, E)\}$ 是可满足的。

定义 5 称带参数的原子动作定义式 $\alpha(v_1, \dots, v_n) \equiv (P, E)$ 相对于 $RBox \mathcal{R}$ 和 $TBox \mathcal{T}$ 是一致的, 当且仅当存在 n 个未在 $\mathcal{R}, \mathcal{T}, \mathcal{A}$ 和 E 中出现的个体名 $p_1, \dots, p_n$ 使得原子动作定义式 $\alpha(p_1, \dots, p_n) \equiv (P[v_1/p_1, \dots, v_n/p_n], E[v_1/p_1, \dots, v_n/p_n])$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是一致的。称 AActBox $aAct$ 相对于 $\mathcal{R}$ 和

$\mathcal{T}$ 是一致的,当且仅当 $aAct$ 中的每个定义式都是相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 一致的。

定义 6 称带参数的复杂动作定义式 $A(v_1, \dots, v_n) \equiv \pi$ 相对于 $RBox \mathcal{R}$ 和 $TBox \mathcal{T}$ 是一致的,当且仅当存在 n 个未在 $\mathcal{R}, \mathcal{T}$ 和 π 中出现的个体名 $p_1, \dots, p_n$ 以及某个模型 $M = (\Sigma, \Delta_M, D)$ (其中 $\Sigma = (W, \cdot^T)$) 和该模型中的某两个可能世界 $w, w' \in W$ 使得 $M \models \mathcal{R}, M \models \mathcal{T}, M \models \mathcal{A}_{\pi[p/v]}$ 以及 $(w, w') \in (\pi[v_1/p_1, \dots, v_n/p_n])^T$; 其中,令 $\alpha_1, \dots, \alpha_k$ 是出现在动作 $\pi[v_1/p_1, \dots, v_n/p_n]$ 中的所有原子动作,则 $\mathcal{A}_{\pi[p/v]}$ 表示由关于这些原子动作的所有原子动作定义式组成的 $ActBox$ 。称 $CActBox cAct$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是一致的,当且仅当 $cAct$ 中的每个定义式都是相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 一致的。

显然,带参数的复杂动作定义式的一致性可以转换为公式的可满足性问题。

定理 2 带参数的复杂动作定义式 $A(v_1, \dots, v_n) \equiv \pi$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是一致的,当且仅当存在 n 个未在 $\mathcal{R}, \mathcal{T}$ 和 π 中出现的个体名 $p_1, \dots, p_n$ 使得公式 $\langle \pi[v_1/p_1, \dots, v_n/p_n] \rangle \text{true}$ 相对于 $\mathcal{R}, \mathcal{T}$ 和 $\mathcal{A}_{\pi[p/v]}$ 是可满足的。

关于动作的第二部分推理问题主要包括动作的可执行性问题、投影问题和规划问题。直观来说,给定某个动作 π 、某个 $ABox \mathcal{A}$ 、某个公式 φ 以及某个由动作组成的集合 Σ ,动作的可执行性问题是指判断 π 在 $\mathcal{A}$ 刻画的状态下是否可以执行;动作的投影问题是指判断在 $\mathcal{A}$ 刻画的状态下执行 π 之后是否使得 φ 成立;规划问题是指可否应用 Σ 中的元素构造出某个动作序列,使得从 $\mathcal{A}$ 刻画的状态出发可以依次执行该序列中的动作,并且在序列执行结束之后使得目标公式 φ 成立,其中将满足上述条件的动作序列称为一个规划。文献[11]已经基于动态描述逻辑 DDL 给出了对这些推理问题的形式化定义,并且将它们转换为 DDL 中公式的可满足性问题来解决,这里不再重复阐述。

由动作的执行导致的状态更新问题是本文需要解决的关键问题。基于 DDL 的知识库通过 $ABox$ 对应用领域的具体状态进行了刻画;当由于动作的执行使得应用领域的具体状态发生改变时,需要相应地对 $ABox$ 进行更新,使得更新后的 $ABox$ 恰好刻画了更新后的状态。例如,假设当前的 $ABox$ 中含有以下 $ABox$ 断言:

(Father $\sqcap$ hasChild. Student)(John),
hasChild(John, Peter)

这些断言刻画了以下知识:John 是一位父亲并且其所有孩子都还是学生;Peter 是 John 的孩子。假设在当前状态下执行了由如下定义式刻画的原子动作 graduate_Peter:

graduate_Peter $\equiv (\{Student(Peter)\}, \{\neg Student(Peter)\})$

则上述 $ABox$ 需要更新为如下形式:

(Father $\sqcap \forall$ hasChild. (Student $\sqcup \{Peter\}))$ (John),
hasChild(John, Peter),
 $\neg Student(Peter)$

其含义为:John 是一位父亲,并且其所有孩子要么是学生要么是 Peter;Peter 是 John 的孩子,并且 Peter 不是学生。

在设计 $ABox$ 更新算法之前,我们先基于 DDL 给出对该问题的严格定义。

定义 7 令 $\mathcal{R}, \mathcal{T}$ 分别为 $RBox$ 和 $TBox$, $\mathcal{A}, \mathcal{A}'$ 为两个

$ABox, \alpha \equiv (P, E)$ 是包含在某个 $ActBox \mathcal{A}$ 中的原子动作定义式。如果满足以下条件,则称 $\mathcal{A}$ 是相对于 $\mathcal{R}, \mathcal{T}$ 和 $\mathcal{A}$ 来说在 $\mathcal{A}$ 上执行原子动作 α 之后得到的新的 $ABox$:

(1) 动作 α 在 $\mathcal{A}$ 刻画的状态下是可以执行的;

(2) 对于任一模型 $M = (\Sigma, \Delta_M, D)$ (其中 $\Sigma = (W, \cdot^T)$) 和任意两个可能世界 $w, w' \in W$: 如果 $M \models \mathcal{R}, M \models \mathcal{T}, M \models \mathcal{A}$ 以及 $(w, w') \in \alpha^T$, 则 $(M, w) \models \mathcal{A}$ 当且仅当 $(M, w') \models \mathcal{A}'$ 。

为了根据 $ABox \mathcal{A}$ 和原子动作 α 构造出更新后的 $ABox \mathcal{A}'$, 参照 H. Liu 的研究结果^[12], 我们引入形如 $@_u C$ 的概念, 其中 u 为任一主体名, C 为任一概念; 相应的语义定义为: 如果 $u^{I(w)} \in C^{I(w)}$, 则 $(@_u C)^{I(w)} := \Delta_M$, 否则 $(@_u C)^{I(w)} := \emptyset$ 。在此基础上, 以动态描述逻辑 DDL(SHOIQ) 为例, 可以设计出如下构造 $\mathcal{A}'$ 的算法。

算法 1 令 $\mathcal{R}, \mathcal{T}, \mathcal{A}$ 分别为 $RBox, TBox$ 和 $ABox, \alpha \equiv (P, E)$ 为原子动作定义式。通过以下步骤构造在 $\mathcal{A}$ 上执行原子动作 α 之后得到的 $ABox \mathcal{A}'$:

(1) 如果公式 $\text{Conj}(\mathcal{A}) \wedge \neg \langle \alpha \rangle \text{true}$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是可满足的, 则返回结论“原子动作 α 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 来说在 $\mathcal{A}$ 上不可执行”, 算法终止。其中的 $\text{Conj}(\mathcal{A})$ 是将 $ABox \mathcal{A}$ 的各个元素进行合取后得到的合取式。

(2) 对于 $\mathcal{A}$ 中出现的任一概念名, 如果其出现在 $\mathcal{T}$ 中某个概念定义式的左边, 则用该定义式右边的概念对其进行替换; 如此反复进行, 直到没有概念名需要替换。将所得到的 $ABox$ 记为 $\mathcal{A}_T$ 。

(3) 令 $\sqsubseteq^R$ 表示二元关系“ $\sqsubseteq$ ”在集合 $\mathcal{R} \cup \{\text{Inv}(R) \sqsubseteq \text{Inv}(S) \mid R \sqsubseteq S \in \mathcal{R}\}$ 上的自反传递闭包, 其中 $\text{Inv}()$ 是如下定义的函数: 对于任一角色名 r 都有 $\text{Inv}(r) = r^-$ 和 $\text{Inv}(r^-) = r$ 。应用以下规则构造出 E 相对于 $\mathcal{R}$ 的闭包 $E_{\mathcal{R}}^*$:

- ① $E \subseteq E_{\mathcal{R}}^*$;
- ② 如果 $R(u, v) \in E_{\mathcal{R}}^*$, 则 $\text{Inv}(R)(v, u) \in E_{\mathcal{R}}^*$;
- ③ 如果 $\neg R(u, v) \in E_{\mathcal{R}}^*$, 则 $\neg \text{Inv}(R)(v, u) \in E_{\mathcal{R}}^*$;
- ④ 如果 $R(u, v) \in E_{\mathcal{R}}^*$ 并且 $R \sqsubseteq^R R'$, 则 $R'(u, v) \in E_{\mathcal{R}}^*$;
- ⑤ 如果 $\neg R'(u, v) \in E_{\mathcal{R}}^*$ 并且 $R \sqsubseteq^R R'$, 则 $\neg R(u, v) \in E_{\mathcal{R}}^*$ 。

(4) 令 $\text{Obj}(E)$ 是由出现在 E 中的所有个体名组成的集合; 对 $ABox \mathcal{A}'$ 构造如下:

$\mathcal{A}' := \{D^E(u) \mid D(u) \in \mathcal{A}_T\} \cup \{R(u, v) \mid R(u, v) \in \mathcal{A}_T \text{ 并且 } \neg R(u, v) \notin E_{\mathcal{R}}^*\} \cup \{\neg R(u, v) \mid \neg R(u, v) \in \mathcal{A}_T \text{ 并且 } R(u, v) \notin E_{\mathcal{R}}^*\}$, 其中 D^E 是经过如下递归构造之后得到的概念:

- ① 对于概念名 C_i , 有 $C_i^E := (C_i \sqcup \bigwedge_{\neg C_i(u) \in E_{\mathcal{R}}^*} \{u\}) \sqcap_{C_i(u) \in E_{\mathcal{R}}^*} \neg \{u\}$;
- ② $\{u\}^E := \{u\}$;
- ③ $(\neg C)^E := \neg C^E$;
- ④ $(C \sqcup C')^E := C^E \sqcup C'^E$;
- ⑤ $(\leq n S, C)^E := ((\bigwedge_{u \in \text{Obj}(E)} \neg \{u\}) \sqcap (n S, C^E)) \sqcap_{u \in \text{Obj}(E)} (\{u\} \sqcap (\bigwedge_{n_1 + n_2 + n_3 = n} (\leq n_1 S, (\bigwedge_{v \in \text{Obj}(E)} \neg \{v\} \sqcap C^E) \sqcap \leq n_2 S, (\bigwedge_{v \in \text{Obj}(E), S(u, v) \notin E_{\mathcal{R}}^*, \neg S(u, v) \notin E_{\mathcal{R}}^*} \{v\} \sqcap C^E)))$;
- ⑥ $(@_u C)^E := \bigwedge_{O \models \{v \mid \neg S(u, v) \in E_{\mathcal{R}}^*\}, \#O = n_3 + 1} ((@_v C^E))$

式中, n_1, n_2, n_3 为非负整数;

$$\begin{aligned}
& \textcircled{6} (\forall R.C)^E := ((\bigwedge_{u \in \text{Obj}(E)} \{u\}) \sqcup \forall R.C^E) \\
& \sqcap ((\bigwedge_{u \in \text{Obj}(E)} \rightarrow \{u\}) \sqcup \forall R. ((\bigwedge_{v \in \text{Obj}(E)} \{v\}) \sqcap C^E)) \sqcap \\
& \bigwedge_{u,v \in \text{Obj}(E), R(u,v) \in E_{\mathcal{R}}^*, \neg R(u,v) \in E_{\mathcal{R}}^*} ((\rightarrow \{u\}) \sqcup \forall R. ((\rightarrow \{v\}) \sqcup C^E)) \\
& \sqcap \bigwedge_{\neg R(u,v) \in E_{\mathcal{R}}^*} ((\rightarrow \{u\}) \sqcup @_u C^E); \\
& \textcircled{7} (@_u C)^E := @_u C^E.
\end{aligned}$$

(5) 构造 $\text{ABox } \mathcal{A}' := \mathcal{A} \cup E$; 返回 $\mathcal{A}'$, 算法终止。

本算法的直观思想如下: 步骤(1)用于判断原子动作 α 在 $\mathcal{A}$ 刻画的状态下是否可以执行; 步骤(2)用于处理领域本体中的 TBox, 将 TBox 中刻画的关于概念的约束编码到 $\text{ABox } \mathcal{A}_T$ 中; 步骤(3)用于处理角色的逆运算以及领域本体中的 RBox, 将角色之间的逆反关系以及 RBox 中刻画的关于角色的约束都编码到原子动作 α 的效果集 E 上, 从而将 E 扩展为 $E_{\mathcal{R}}^*$; 步骤(4)中对概念 D^E 的构造过程与文献[12]给出的概念更新过程类似; 步骤(5)进一步将效果集 E 显式地添加到更新后的 ABox 中。容易证明, 步骤(4)中对概念 D^E 的构造过程具有如下性质。

引理 1 令 D^E 是从概念 D 出发经过算法 1 中步骤(4)给出的递归构造过程得到的概念。对于任一模型 $M = (\Sigma, \Delta_M, D)$ (其中 $\Sigma = (W, \cdot^T)$) 和任意两个可能世界 $w, w' \in W$; 如果 $M \models \mathcal{R}, M \models \mathcal{T}, M \models \mathcal{A}$ 以及 $(w, w') \in \alpha^T$, 则必然有 $D^{(w)} = (D^E)^{(w')}$ 。

该引理与文献[12]中关于概念更新过程的引理 14 相似; 区别在于文献[12]是在描述逻辑框架下考虑更新问题, 而本文在 DDL 的框架下考察更新问题。因此对引理 1 的证明实际上与文献[12]对引理 14 的证明类似。

在引理 1 的基础上, 结合定义 7 容易得到如下结论。

定理 3 令 $\mathcal{A}$ 是算法 1 返回的 ABox ; 则 $\mathcal{A}$ 是相对于 $\mathcal{R}, \mathcal{T}$ 和 $\mathcal{A}$ 来说在 $\mathcal{A}$ 上执行原子动作 α 之后得到的新的 ABox 。

限于篇幅, 这里省略详细证明过程。

下面对算法 1 的复杂度进行考察。不难看出, 算法 1 中从步骤(2)到步骤(4)的构造过程可以在与 $\mathcal{R}, \mathcal{T}, \mathcal{A}$ 和 E 的大小成多项式关系的时间内完成; 算法 1 的计算复杂度主要取决于步骤(1)。步骤(1)需要在 DDL (SHOIQ) 中判断公式 $\text{Conj}(\mathcal{A}) \wedge \neg \langle \alpha \rangle \text{true}$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是否可满足; 由于其中的 α 是定义为 $\alpha \equiv (P, E)$ 的原子动作, 并且其相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是一致的, 因此, 根据文献[7]给出的判定过程, 对公式 $\text{Conj}(\mathcal{A}) \wedge \neg \langle \alpha \rangle \text{true}$ 的可满足性的判断实际上将转换为 SHOIQ 中对 $\text{ABox } \mathcal{A} \cup P$ 的一致性的判断。由于描述逻辑 SHOIQ 中 ABox 一致性问题的推理复杂度为 NEXPTIME-完全, 因此, 算法 1 在最坏情况下的计算复杂度为 NEXPTIME。

算法 1 考察的是执行原子动作的情况。在 ABox 上执行任一动作 π 之后, 实际发生的动作总可以表示为由原子动作和测试动作构成的某个序列, 其中的测试动作不会对 ABox 产生影响。因此, 对应于任一动作 π , 可以通过多次应用算法 1 而构造出在 ABox 上执行 π 之后得到的新的 ABox 。

最后, 我们对本节考察的推理问题进行总结。给定一个基于 DDL 的知识库 $\mathcal{K} = (\mathcal{R}, \mathcal{T}, a\mathcal{A}ct, c\mathcal{A}ct, \mathcal{A})$ 之后, 首先需要依次考察以下几个一致性问题: RBox $\mathcal{R}$ 与 TBox $\mathcal{T}$ 是否一致; $\text{ABox } \mathcal{A}$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是否一致; $\text{AActBox } a\mathcal{A}ct$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是否一致; $\text{CActBox } c\mathcal{A}ct$ 相对于 $\mathcal{R}$ 和 $\mathcal{T}$ 是否一致。当其中的任何一个一致性不成立时, 表明知识库存在问题, 需要相应地进行修正。如果上述一致性都得到了保证, 则接下来可以从可执行性、投影、规划等角度对动作进行推理, 以及可以在

执行了任一动作之后构造出新的 ABox 。以描述逻辑中对 ABox 一致性进行判定的推理算法、DDL 中对公式可满足性进行判定的推理算法以及本文给出的算法 1 为基础, 可以有效地实现上述各种推理机制, 提供具有可判定性的推理服务。

4 在智能主体建模中的应用

本节将基于 DDL 的动作理论应用于智能主体建模, 给出一种智能主体的心智状态模型以及相应的基本操作。

首先, 将智能主体的心智状态建模为一个六元组 $\text{Agent DDL} = (\text{Bel}, \text{Act}, \text{Stra}, \text{Goals}, \text{Evt}, \text{Intent})$, 其中:

(1) Bel 为主体的信念集, 描述了主体对其所处环境的认识。信念集又表示为一个三元组 $\text{Bel} = (\mathcal{R}, \mathcal{T}, \mathcal{A})$, 其中的 $\mathcal{R}, \mathcal{T}, \mathcal{A}$ 分别是 DDL 中的 RBox, TBox 和 ABox ; $\mathcal{R}$ 和 $\mathcal{T}$ 构成领域本体, 反映了主体所具有的领域知识; $\mathcal{A}$ 反映了主体对当前状态下客观世界的认识。

(2) Act 为主体的行为能力库, 描述了主体能够进行的动作。Act 在语法上是一个 AActBox 。

(3) Stra 为主体的策略库, 描述了主体执行复杂动作的经验。Stra 在语法上是一个 CActBox 。

(4) Goals 为主体的目标集, 表示主体希望实现的目标。Goals 是由 DDL 中的公式组成的集合, 其中每个公式表示一个独立的目标。

(5) Evt 为主体的响应事件集, 表示主体在外界环境的触发下需要执行的动作。Evt 中的每个事件都是可以由 Act 或 Stra 中的定义式确定的原子动作或复杂动作。

(6) Intent 为主体的行为意图, 是由原子动作构成的序列, 等待主体对其中的原子动作依次调度执行。

显然, 该心智状态模型中的 $\mathcal{R}, \mathcal{T}, \mathcal{A}, \text{Act}$ 和 Stra 一起形成了一个基于 DDL 的知识库。我们需要保证该知识库具有上一节讨论的各种一致性。

其次, 基于上述心智状态模型, 智能主体在运行过程中可以执行以下基本操作:

(1) 接受外界信息, 修正信念集。当主体通过观察或与外界的交互感知到环境的变化时, 需要及时地修改自身对环境的信念。一般情况下, 由 $\mathcal{R}$ 和 $\mathcal{T}$ 构成的领域本体是不变的, 变化的是由 $\mathcal{A}$ 刻画的对客观世界具体状态的认识。

对于由简单公式组成的任一修正集 U , 可以先构造出一个原子动作定义式 $\alpha \equiv (\{\}, U)$, 然后通过算法 1 构造出在 $\mathcal{A}$ 上执行 α 后得到的新的 ABox , 实现对信念集的修正。

(2) 接受外界请求, 添加目标。主体可以在运行过程中接受外界请求, 将相应的目标公式添加到目标集 Goals 中。

(3) 接受外界请求, 添加响应事件。主体可以在运行过程中接受来自外界明确指定的事件请求, 将这些请求添加到响应事件集 Evt 中。

(4) 选择目标或事件, 生成行为意图。主体可以从目标集 Goals 以及响应事件集 Evt 中选择一个目标或事件, 并相应地生成行为意图, 然后添加到 Intent 中。

具体来说, 如果选择了某个目标 φ , 则主体将根据当前 Bel, Act 和 Stra 的内容进行规划, 得到一个可以实现目标 φ 的原子动作序列, 然后将其加入 Intent。如果选择的是事件, 则必须保证该事件所对应的动作在 $\mathcal{A}$ 刻画的状态下是可以执行的, 在此基础上再根据当前的信念集以及 Stra 和 Act 中的定义式, 解析出由可以依次执行的原子动作构成的动作序列, 然后将该序列加入 Intent。

(5)执行动作。主体根据 *Intent* 中的内容,依次取出并执行各个原子动作。

(6)根据动作执行情况更新信念集。主体在每执行完一个动作之后,需要根据执行结果对信念集进行相应的更新。更新过程可以基于算法 1 来实现。

由上述心智状态模型和基本操作组成的智能主体为语义 Web 服务的功能自动化提供了支持。首先,语义 Web 上基于描述逻辑的大量领域知识可以被放入主体的信念集 *Bel* 中。其次,在对语义 Web 服务进行建模之后^[8],可以将关于原子 Web 服务和 Web 服务组合方案的知识分别放入主体的行为能力库 *Act* 和策略库 *Stra* 中。最后,在动态描述逻辑推理机的支持下,智能主体可以灵活、自动地处理来自外界的相关请求,实现对语义 Web 服务的自动推理、发现、组合和执行。

5 讨论及相关工作

基于描述逻辑的动作理论近年来得到了研究者的关注。F. Baader 等^[13]提出了一个基于描述逻辑的动作形式系统。在该系统中,描述逻辑中的 TBox 和 ABox 被分别用来刻画领域知识和当前状态;每个原子动作被刻画为形如(*pre*,*occ*,*post*)的三元组形式,其参数分别表示动作的前提条件、执行过程中的不确定成分以及执行后产生的影响;复杂动作仅仅由原子动作的有限序列构成。在此基础上,Baader 等将动作的可执行性问题和投影问题转换为 ABox 的一致性问

题。Y. Gu 等^[14]以描述逻辑为参照提出了情景演算的一种改造形式:一方面将情景演算中刻画初始情景和动作的公式都限制为 C° 逻辑中的公式,另一方面将描述逻辑中的 TBox 和 RBox 显式地引入情景演算中。最终得到的动作系统既有较强的描述能力,又保证了相关推理问题的可判定性。在此基础上,Y. Gu 等研究了动作的可执行性问题和投影问题。

与上述两项工作相比,我们工作的主要特点是:首先将动作嵌入到描述逻辑中,构造出具有二维特征的动态描述逻辑系统,然后在这类动态描述逻辑的基础上构造出动作理论。如果将 F. Baader 等和 Y. Gu 等的工作看作是以描述逻辑为参照对情景演算进行了限制,则我们的工作可以看作是以描述逻辑为参照对基于命题动态逻辑的动作理论进行了扩展。

针对描述逻辑只能表示和推理静态知识的局限,Wolter^[15]将描述逻辑 ALC 与命题动态逻辑 PDL 结合,提出了命题动态描述逻辑 PDLC。PDLC 的特点是在描述逻辑中引入了动态算子 $\langle\pi\rangle$ 和 $[\pi]$,用于构造形如 $\langle\pi\rangle C$ 和 $[\pi]C$ 的概念以及形如 $\langle\pi\rangle\varphi$ 和 $[\pi]\varphi$ 的公式,其中 π 是根据命题动态逻辑的语法规则生成的任一动作。应用 PDLC,一方面可以刻画出具有动态内涵的概念,另一方面也可以对原子动作的前提条件和执行后产生的影响进行刻画。例如公式 $\langle\alpha\rangle\text{true}\rightarrow\varphi$ 表示在执行动作 α 之前必须满足公式 φ ,公式 $[\alpha]\varphi$ 表示在执行了动作 α 之后公式 φ 一定成立等。然而,为了支持对动作的各种推理,PDLC 中必须引入其他机制来处理动作理论中的框架问题和分枝问题。

结束语 基于动态描述逻辑 DDL 的动作理论具有以下特点。首先,采用描述逻辑语言刻画领域知识、世界状态、动作的前提条件以及执行动作后产生的影响等,描述能力明显强于采用命题语言的动作理论。其次,它提供了具有可判定性特征的推理服务,在推理能力和推理性能上优于基于一阶谓词或高阶逻辑的动作理论。最后,与其它基于描述逻辑的动作理论相比,它提供了更为丰富的推理服务。

下一步的工作是将基于 DDL 的动作理论与智能主体和语义 Web 服务等结合起来,为语义 Web 服务的功能自动化开发相应的支撑平台。

参考文献

- [1] Reiter R. Knowledge in action; logical foundations for describing and implementing dynamical systems [M]. Cambridge, MA: MIT Press, 2001
- [2] Thielscher M. From situation calculus to fluent calculus: state update axioms as a solution to the inferential frame problem [J]. Artificial intelligence, 1999, 111(1/2): 277-299
- [3] Fikes R. STRIPS: a retrospective [J]. Artificial intelligence, 1993, 59(1-2): 227-232
- [4] Giacomo G, Lenzerini M. PDL-based framework for reasoning about actions [C] // Proceedings of the 4th Congress of the Italian Association for Artificial Intelligence. LNAI 992. Berlin: Springer, 1995: 103-114
- [5] Calvanese D, Giacomo G, Vardi M. Reasoning about actions and planning in LTL action theories [C] // Proceedings of the 8th International Conference on Principles and Knowledge Representation and Reasoning. Menlo Park: AAAI Press, 2002: 593-602
- [6] Shi Zhong-zhi, Dong Ming-kai, Jiang Yun-cheng, et al. A logical foundation for the semantic Web [J]. Science in China, Ser. F: Information Sciences, 2005, 48(2): 161-178
- [7] 常亮, 史忠植, 邱莉榕, 等. 动态描述逻辑的 Tableau 判定算法 [J]. 计算机学报, 2008, 31(6): 896-909
- [8] 史忠植, 常亮. 基于动态描述逻辑的语义 Web 服务推理 [J]. 计算机学报, 2008, 31(9): 1599-1611
- [9] Niu Wen-jia, Shi Zhong-zhi, Cao Peng, et al. Semantic filtering for DDL-based service composition [C] // Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence. LNCS 5351. Berlin: Springer, 2008: 778-787
- [10] 梅婧, 林作铨. 从 ALC 到 SHOQ(D): 描述逻辑及其 Tableau 算法 [J]. 计算机科学, 2005, 32(3): 1-11
- [11] Chang Liang, Lin Fen, Shi Zhong-zhi. A Dynamic Description Logic for Representation and Reasoning About Actions [C] // Proceedings of the 2nd International Conference on Knowledge Science, Engineering and Management. LNCS 4798. Berlin: Springer, 2007: 115-127
- [12] Liu Hong-kai, Lutz C, Milicic M, et al. Updating description logic ABoxes [C] // Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning. Menlo Park: AAAI Press, 2006: 46-56
- [13] Baader F, Lutz C, Milicic M, et al. Integrating description logics and action formalisms: first results [C] // Proceedings of the 12th National Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2005: 572-577
- [14] Gu Yi-lan, Soutchanski M. Decidable reasoning in a modified situation calculus [C] // Proceedings of the 20th International Joint Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2007: 1891-1897
- [15] Wolter F, Zakharyashev M. Temporalizing description logics [M]. Gabbay D, Rijke M. eds. Frontiers of Combining Systems II, Studies Press/Wiley, 2000: 379-401