

动态描述逻辑的 Tableau 判定算法

常 亮^{1),2)} 史忠植¹⁾ 邱莉榕¹⁾ 林 芬^{1),2)}

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⁾(中国科学院研究生院 北京 100039)

摘 要 动态描述逻辑在描述逻辑的基础上引入了动态维,用于描述和推理动态领域的知识,但目前缺少有效的判定算法作为支撑.文中以描述逻辑 ALCO 的动态扩展为例,构建出动态描述逻辑 D-ALCO.以 D-ALCO 的构建过程为基础,将 ALCO 的 Tableau 算法、命题动态逻辑的 Tableau 算法以及对可能模型途径的处理有机地结合起来,给出了 D-ALCO 的 Tableau 判定算法,证明了算法的可终止性、可靠性和完备性.应用该算法,可以在采用开世界假设的情况下对 D-ALCO 中公式的可满足性进行判定.对于 D-ALCQO、D-ALCQIO 等具有更强描述能力的动态描述逻辑,可以对该算法扩展后得到相应的 Tableau 判定算法.

关键词 动态描述逻辑;动作理论;可满足性问题;Tableau 算法;可判定性

中图法分类号 TP18

A Tableau Decision Algorithm for Dynamic Description Logic

CHANG Liang^{1),2)} SHI Zhong-Zhi¹⁾ QIU Li-Rong¹⁾ LIN Fen^{1),2)}

¹⁾(Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100190)

²⁾(Graduate University of Chinese Academy of Sciences, Beijing 100039)

Abstract By introducing a dynamic dimension into the description logic, two kinds of dynamic description logics have been proposed in the literatures for representing and reasoning about knowledge of dynamic application domains. But both of them lack efficient decision algorithms. This paper presents a Tableau decision algorithm for the dynamic description logic D-ALCO. D-ALCO is a combination of the description logic ALCO, the dynamic logic, and an action theory based on the possible models approach. In D-ALCO, based on domain ontologies expressed in ALCO, atomic actions are described by specifying their preconditions and effects; With the help of standard action constructors of the dynamic logic, complex actions can be described also; Both atomic actions and complex actions are then used as modal operators in the construction of formulas. The Tableau decision algorithm for D-ALCO forms an elaborated combination of the Tableau algorithm for ALCO, the decision algorithm for propositional dynamic logic, and the embodiment of the possible models approach. The authors prove that the algorithm is terminating, sound and complete for deciding the satisfiability of D-ALCO formulas with the Open World Assumption. The algorithm is flexible and can be extended for more expressive dynamic description logics such as the D-ALCQO and the D-ALCQIO.

Keywords dynamic description logic; action theory; satisfiability; Tableau algorithm; decidability

收稿日期:2006-11-11;最终修改稿收到日期:2008-04-21. 本课题得到国家自然科学基金(90604017,60775035)、国家“八六三”高技术研究发展计划项目基金(2007AA01Z132)和国家“九七三”重点基础研究发展规划项目基金(2007CB311004)资助. 常 亮,男,1980 年生,博士研究生,主要研究方向为描述逻辑、语义 Web、智能主体. E-mail: changli@ics.ict.ac.cn. 史忠植,男,1941 年生,研究员,博士生导师,主要研究领域为人工智能、机器学习、多主体系统. 邱莉榕,女,1978 年生,博士,主要研究方向为语义 Web 服务、多主体系统. 林 芬,女,1982 年生,博士研究生,主要研究方向为多主体系统、Web 服务.

1 引言

作为一类用于知识表示的形式化工具,描述逻辑在信息系统、软件工程、自然语言处理等领域得到了成功应用^[1]. 在作为下一代 Web 技术的语义 Web 中,描述逻辑更是扮演着关键角色,成为了 W3C 推荐的 Web 本体语言 OWL 的逻辑基础^[2]. 描述逻辑的主要特点在于具有较强的描述能力,同时保证了相关推理问题的可判定性,具有有效的推理算法作为支撑.

描述逻辑只能表示和推理静态领域的知识. 针对这一局限, Wolter 等^[3]将描述逻辑 ALC 与命题动态逻辑 PDL 结合起来,构建了命题动态描述逻辑 PDLC 并证明了该逻辑具有可判定性. PDLC 的主要特点是将任一动作 π 作为模态词,用以构造形如 $\langle \pi \rangle C$ 的概念以及形如 $\langle \pi \rangle \varphi$ 的公式,从而可以刻画具有动态内涵的知识. 但 PDLC 的判定算法一直是未解决的问题.

史忠植等^[4]提出另外一种动态描述逻辑,将描述逻辑与动态逻辑以及动作理论有机地结合起来. 在该逻辑中,任一原子动作 α 都被刻画为二元组 (P_α, E_α) 的形式,分别表示执行该动作之前必须满足的前提条件以及执行动作后的效果,其中的 P_α 、 E_α 是由描述逻辑中的公式组成的集合;从原子动作以及公式出发,可以应用动态逻辑中的动作构造符构造出复杂动作;此外,任一原子动作或复杂动作又可以作为模态词,用于对概念和公式的构造. 在这种方式中,动作成为了逻辑系统的主要成员之一,与概念具有同等重要的地位,可以作为一类知识进行刻画和推理. 文献[4]在描述逻辑 ALC 的基础上构建了相应的动态描述逻辑,给出了判定算法. 但该判定算法遵循了 STRIPS 系统的思路,采用删除表和添加表的方式来处理动作,仅适用于闭世界假设 (closed world assumption) 的情况,在推理过程中需要关于世界状态的完整知识.

描述逻辑的一个典型特征,是在采用开世界假设 (open world assumption)^[1] 的情况下有效地进行推理. 开世界假设是指:如果知识库中不能推导出某个公式,则认为不知道这个公式是否成立. 与之对应的是闭世界假设:如果知识库中不能推导出某个公式,则认为该公式不成立. 由于语义 Web 的开放性和分布性,往往需要采用开世界假设,在信息不完全的情况下进行推理.

在文献[4]的基础上,本文采用可能模型途径 (possible models approach)^[5] 来定义原子动作的语义;对于刻画为二元组 (P_α, E_α) 的任一原子动作 α ,将执行动作 α 后所能产生的影响严格地限制为 E_α 中描述的内容. 本文首先以描述逻辑 ALCO 的动态扩展为例,构建出动态描述逻辑 D-ALCO,给出其语法和语义定义. 接下来,提出 D-ALCO 的 Tableau 判定算法,证明算法的可终止性、可靠性以及完备性;该算法继承了描述逻辑 Tableau 算法的基本特征,在采用开世界假设的情况下对 D-ALCO 中公式的可满足性进行判定. 最后,针对 D-ALCQO、D-ALCQIO 等具有更强描述能力的动态描述逻辑,逐步对该算法扩展后得到相应的 Tableau 判定算法.

2 动态描述逻辑 D-ALCO

D-ALCO 的基本符号包括:①由概念名组成的集合 N_C ;②由角色名组成的集合 N_R ;③由个体名组成的集合 N_I ;④由个体变元组成的集合 N_{IV} ;⑤由原子动作名组成的集合 N_A ;⑥概念构造符 $\{ \}$ 、 \neg 、 \sqcup 和 \forall ;⑦公式构造符 \neg 、 \vee 和 $\langle \rangle$;⑧动作构造符 \cup 、 $;$ 、 $*$ 和 $?$;⑨其它符号,包括定义号 \equiv 、圆括号 $()$ 以及逗号. 从这些符号出发可以构造出角色、概念、公式以及动作.

定义 1. R 是 D-ALCO 中的角色当且仅当 $R \in N_R$.

定义 2. D-ALCO 中的概念由如下产生式生成:

$$C, D ::= C_i \mid \{u\} \mid \neg C \mid C \sqcup D \mid \forall R.C,$$

其中, $C_i \in N_C$, $u \in N_I \cup N_{IV}$, R 为角色.

将形如 $\{u\}$ 、 $\neg C$ 、 $C \sqcup D$ 以及 $\forall R.C$ 的概念分别称为枚举、否定、析取以及值限定概念.

此外,也可以引入形如 $C \sqcap D$ 、 $\exists R.C$ 、 \top 以及 \perp 的概念,分别作为 $\neg(\neg C \sqcup \neg D)$ 、 $\neg(\forall R. \neg C)$ 、 $C \sqcup \neg C$ 以及 $\neg \top$ 的缩写.

令 C_i 为概念名, D 为概念,则称 $C_i \equiv D$ 为概念定义式.

对于由概念定义式组成的有限集合 \mathcal{T} ,如果每个概念名最多在 \mathcal{T} 中某个概念定义式的左边出现一次,则称 \mathcal{T} 为 TBox.

相对于某个 TBox \mathcal{T} ,如果概念名 C 在 \mathcal{T} 中某个概念定义式的左边出现,则称 C 为被定义的概念名,否则称 C 为简单概念名.

定义 3. D-ALCO 中的公式由如下产生式

生成:

$$\varphi, \psi ::= C(u) \mid R(u, v) \mid \neg \varphi \mid \varphi \vee \psi \mid \langle \pi \rangle \varphi,$$

其中, $u, v \in N_I \cup N_{IV}$, C 为概念, R 为角色, π 为动作.

将形如 $C(u)$ 、 $R(u, v)$ 、 $\neg \varphi$ 、 $\varphi \vee \psi$ 以及 $\langle \pi \rangle \varphi$ 的公式分别称为概念断言、角色断言、否定式、析取式以及动作存在性断言.

此外, 也可以引入形如 $[\pi] \varphi$ 、 $\varphi \wedge \psi$ 、 $\varphi \rightarrow \psi$ 、 true 以及 false 的公式, 分别作为 $\neg \langle \pi \rangle \neg \varphi$ 、 $\neg (\neg \varphi \vee \neg \psi)$ 、 $\neg \varphi \vee \psi$ 、 $\varphi \vee \neg \varphi$ 以及 $\neg \text{true}$ 的缩写.

给定某个 TBox \mathcal{T} , 将形如 $R(u, v)$ 、 $\neg R(u, v)$ 以及 $C(u)$ 和 $\neg C(u)$ (其中的 C 是相对于 \mathcal{T} 的简单概念名) 的公式都称为简单公式.

对于任一简单公式 φ , 如果 φ 形如 $\neg R(u, v)$ 或 $\neg C(u)$, 则用 φ^\neg 表示相应的简单公式 $R(u, v)$ 或 $C(u)$; 否则, 如果 φ 形如 $R(u, v)$ 或 $C(u)$, C 为简单概念名, 则用 φ^\neg 相应地表示 $\neg R(u, v)$ 或 $\neg C(u)$.

定义 4. 给定某个 TBox \mathcal{T} , 将 $\alpha(v_1, \dots, v_n) \equiv (P, E)$ 称为一个原子动作定义式; 其中

- (1) $\alpha \in N_A$ 为所定义的原子动作名;
- (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;
- (3) P 是由公式组成的有限集合, 表示动作执行前必须满足的前提条件;
- (4) E 是由简单公式组成的有限集合, 表示执行该动作后将会发生的影响;
- (5) P 与 E 之间满足如下约束: 对于任一 $\varphi \in E$ 都有 $\varphi^\neg \in P$.

对于由动作定义式组成的任一有限集合 \mathcal{A}_c , 如果每个原子动作名最多在 \mathcal{A}_c 中某个动作定义式的左边出现一次, 则称 \mathcal{A}_c 为 AActBox.

定义 5. 令 \mathcal{T} 为 TBox, \mathcal{A}_c 为 AActBox, $\alpha(v_1, \dots, v_n) \equiv (P, E)$ 为 \mathcal{A}_c 中的某个动作定义式, u_1, \dots, u_n 为任意 n 个个体名或个体变元, 则将 $\alpha(u_1, \dots, u_n)$ 称为一个以 \mathcal{T} 和 \mathcal{A}_c 为参照的原子动作, 简称为原子动作; 并且称 $P[u_1/v_1, \dots, u_n/v_n]$ 为该原子动作的前提条件, 称 $E[u_1/v_1, \dots, u_n/v_n]$ 为该原子动作所能产生的影响. 其中的 $P[u_1/v_1, \dots, u_n/v_n]$ 和 $E[u_1/v_1, \dots, u_n/v_n]$ 是将 v_1, \dots, v_n 在 P 和 E 中的出现分别用 u_1, \dots, u_n 进行替换后得到的两个公式集.

令原子动作 $\alpha(v_1, \dots, v_n)$ 的前提条件为 P , 所能产生的影响为 E , 则在不产生歧义的情况下, 也可以用二元组 (P, E) 表示原子动作 $\alpha(v_1, \dots, v_n)$.

每个原子动作定义式都刻画了一类原子动作.

例如, 购买图书的动作 $\text{buyBook}(u, v)$ 可以刻画为 $\text{buyBook}(u, v) \equiv (\{ \text{customer}(u), \text{book}(v), \text{instore}(v), \neg \text{owns}(u, v) \}, \{ \neg \text{instore}(v), \text{owns}(u, v) \})$.

在代入个体名 Tom 和 KingLear 后, 可以相应地得到一个原子动作 $\text{buyBook}(\text{Tom}, \text{KingLear})$; 该原子动作的直观含义为: 当条件 $\text{customer}(\text{Tom})$ 、 $\text{book}(\text{KingLear})$ 、 $\text{instore}(\text{KingLear})$ 以及 $\neg \text{owns}(\text{Tom}, \text{KingLear})$ 成立时, 该原子动作可以被执行; 在执行之后, 对状态的影响是使得 $\text{instore}(\text{KingLear})$ 变为 $\neg \text{instore}(\text{KingLear})$ 以及使得 $\neg \text{owns}(\text{Tom}, \text{KingLear})$ 变为 $\text{owns}(\text{Tom}, \text{KingLear})$.

定义 6. 令 \mathcal{T} 为 TBox, \mathcal{A}_c 为 AActBox. D-ALCO 中的动作由如下产生式生成:

$$\pi, \pi' ::= \alpha(v_1, \dots, v_n) \mid \varphi? \mid \pi \cup \pi' \mid \pi; \pi' \mid \pi^*,$$

其中, $\alpha(v_1, \dots, v_n)$ 为原子动作, φ 为公式.

将形如 $\varphi?$ 、 $\pi \cup \pi'$ 、 $\pi; \pi'$ 以及 π^* 的动作分别称为测试、选择、顺序以及迭代动作.

此外, 可以引入 $\text{if } \varphi \text{ then } \pi \text{ else } \pi' \text{ endif}$ 作为 $(\varphi?; \pi) \cup ((\neg \varphi)?; \pi')$ 的缩写, 引入 $\text{while } \varphi \text{ do } \pi \text{ endwhile}$ 作为 $(\varphi?; \pi)^*$; $(\neg \varphi)?$ 的缩写.

下面给出 D-ALCO 的语义定义.

对于描述逻辑 (例如 ALCO) 来说, 其语义解释一般形如 $I = (\Delta, \cdot^I)$. 其中的 Δ 是由个体组成的解释域; 解释函数 \cdot^I 将每个概念名 C_i 解释为 Δ 的某个子集 C_i^I , 将每个角色名 R_i 解释为 Δ 上的某个二元关系 R_i^I , 将每个个体名 p_i 解释为 Δ 中的某个元素 p_i^I .

动态描述逻辑在描述逻辑的基础上引入了动态维, 使得语义模型从整体上体现为由多个可能世界构成的可能世界空间. 在每个可能世界下都分别对概念名、角色名以及个体名进行解释. 动作被解释为关于这些可能世界的二元关系.

定义 7. D-ALCO 模型是一个三元组 $M = (\Delta, W, I)$. 其中, Δ 是由个体组成的非空集合, 作为该模型的论域; W 是由可能世界组成的集合; I 对 W 中的每个可能世界 w 赋予一个解释 $I(w) = (\Delta, \cdot^{I(w)})$, 其中的解释函数 $\cdot^{I(w)}$ 满足以下条件:

- (1) 将 N_C 中的每个概念名 C_i 解释为 Δ 的某个子集 $C_i^{I(w)}$;
- (2) 将 N_R 中的每个角色名 R_i 解释为 Δ 上的某个二元关系 $R_i^{I(w)}$;
- (3) 将 N_I 中的每个个体名 p_i 解释为 Δ 中的某个个体 $p_i^{I(w)}$, 并且满足: 对于 W 中的任一可能世界 w' 都有 $p_i^{I(w)} = p_i^{I(w')}$; 由于 p_i 的解释与可能世界无关, 因此也将 $p_i^{I(w)}$ 简记为 p_i^I .

需要指出的是,上述定义中采用了恒定解释域假设(constant domain assumption)^[6],模型中的所有可能世界都采用同一个解释域.相应地, N_I 中的个体名都作为刚性命名符(rigid designator)^[6]来处理,个体名的解释不随可能世界的变化而改变.

定义 8. 对应于任一 D-ALCO 模型 $M=(\Delta, W, I)$,将函数 $\gamma: N_{IV} \rightarrow \Delta$ 称为基于 M 的一个指派.

任一指派 γ 给出了 N_{IV} 中各个个体变元的一种赋值,分别将这些个体变元指派为 Δ 中的个体.

定义 9. 令 \mathcal{T} 为 TBox, \mathcal{A}_c 为 AActBox, $M=(\Delta, W, I)$ 为 D-ALCO 模型;对 D-ALCO 中角色、概念、公式以及动作的语义归纳定义如下.

首先,相对于 W 中任一可能世界 w 以及任一基于 M 的指派 γ ,将任一角色 R 解释为 Δ 上的某个二元关系 $R^{I(w),\gamma} \subseteq \Delta \times \Delta$,将任一概念 C 解释为 Δ 的某个子集 $C^{I(w),\gamma} \subseteq \Delta$;归纳定义如下:

- (1) $R_i^{I(w),\gamma} = R_i^{I(w)}$, 其中 $R_i \in N_R$;
- (2) $C_i^{I(w),\gamma} = C_i^{I(w)}$, 其中 $C_i \in N_C$;
- (3) $\{u\}^{I(w),\gamma} = \{u^{I,\gamma}\}$, 其中,对 $u^{I,\gamma}$ 定义为
 - ① 如果 $u \in N_I$, 则 $u^{I,\gamma} := u$;
 - ② 如果 $u \in N_{IV}$, 则 $u^{I,\gamma} := \gamma(u)$;
- (4) $(\neg C)^{I(w),\gamma} = \Delta \setminus C^{I(w),\gamma}$, 其中的“ \setminus ”为集合差运算;

(5) $(C \sqcup D)^{I(w),\gamma} = C^{I(w),\gamma} \cup D^{I(w),\gamma}$, 其中的“ \cup ”为集合并运算;

(6) $(\forall R.C)^{I(w),\gamma} = \{x \mid \text{对于任一 } y \in \Delta: \text{如果 } (x,y) \in R^{I(w),\gamma}, \text{则必然有 } y \in C^{I(w),\gamma}\}$;

其次,相对于 W 中任一可能世界 w 和任一基于 M 的指派 γ ,用 $(M,w,\gamma) \models \varphi$ 表示公式 φ 在模型 M 中的可能世界 w 下相对于指派 γ 成立;根据 φ 的结构归纳定义如下:

- (7) $(M,w,\gamma) \models C(u)$ iff $u^{I,\gamma} \in C^{I(w),\gamma}$;
- (8) $(M,w,\gamma) \models R(u,v)$ iff $(u^{I,\gamma}, v^{I,\gamma}) \in R^{I(w),\gamma}$;

(9) $(M,w,\gamma) \models \neg \varphi$ iff $(M,w,\gamma) \not\models \varphi$ (即 φ 在 M 中的可能世界 w 下相对于指派 γ 不成立);

(10) $(M,w,\gamma) \models \varphi \vee \psi$ iff $(M,w,\gamma) \models \varphi$ 或者 $(M,w,\gamma) \models \psi$;

(11) $(M,w,\gamma) \models \langle \pi \rangle \varphi$ iff 存在 $w' \in W$ 使得 $(w,w') \in \pi^I$ 并且 $(M,w',\gamma) \models \varphi$;

最后,对于任一动作 π ,将其解释为 W 上的某个二元关系 $\pi^I \subseteq W \times W$;归纳定义如下:

(12) 对于原子动作 $\alpha(v_1, \dots, v_n)$,如果根据 \mathcal{A}_c 可以确定其前提条件为 P ,所产生的影响为 E ;则

$\alpha(v_1, \dots, v_n)^I = (P, E)^I = \{(w, w') \mid \text{存在某个基于 } M \text{ 的指派 } \gamma \text{ 使得: ① 对于任一公式 } \varphi_i \in P \text{ 都有 } (M, w, \gamma) \models \varphi_i, \text{② 对于任一简单概念名 } C \text{ 都有 } C^{I(w'),\gamma} = (C^{I(w),\gamma} \cup \{u^{I,\gamma} \mid C(u) \in E\}) \setminus \{u^{I,\gamma} \mid \neg C(u) \in E\} \text{ 以及 ③ 对于任一角色名 } R \in N_R \text{ 都有 } R^{I(w'),\gamma} = (R^{I(w),\gamma} \cup \{(u^{I,\gamma}, v^{I,\gamma}) \mid R(u,v) \in E\}) \setminus \{(u^{I,\gamma}, v^{I,\gamma}) \mid \neg R(u,v) \in E\}\}$;

(13) $(\varphi?)^I = \{(w, w) \mid (M, w, \gamma) \models \varphi\}$;

(14) $(\pi \cup \pi')^I = \pi^I \cup \pi'^I$;

(15) $(\pi; \pi')^I = \{(w, w') \mid \text{存在某个可能世界 } w'' \in W \text{ 使得 } (w, w'') \in \pi^I \text{ 以及 } (w'', w') \in \pi'^I\}$;

(16) $(\pi^*)^I = \pi^I$ 的自反传递闭包.

上述定义中对原子动作的解释采用了可能模型途径^[5],将原子动作产生的影响解释为简单概念名和角色名在外延上的变化.

定义 10. 令 $M=(\Delta, W, I)$ 为 D-ALCO 模型, γ 为基于 M 的指派,则

(1) M 相对于 γ 来说是某个概念定义式 $C \equiv D$ 的模型,记为 $M, \gamma \models C \equiv D$,当且仅当对于 W 中的任一可能世界 w 都有 $C^{I(w),\gamma} = D^{I(w),\gamma}$;

(2) M 相对于 γ 来说是某个 TBox \mathcal{T} 的模型,记为 $M, \gamma \models \mathcal{T}$,当且仅当对于 \mathcal{T} 中的任一概念定义式 $C \equiv D$ 都有 $M, \gamma \models C \equiv D$.

定义 11. 令 $\mathcal{T}, \mathcal{A}_c$ 分别为 TBox 和 AActBox, φ 为公式;称公式 φ 相对于 \mathcal{T} 和 \mathcal{A}_c 是可满足的,当且仅当存在某个模型 $M=(\Delta, W, I)$ 、某个基于 M 的指派 γ 以及 W 中的某个可能世界 w 使得 $M, \gamma \models \mathcal{T}$ 和 $(M, w, \gamma) \models \varphi$.

公式的可满足性问题是动态描述逻辑中最基本的判定问题. 本文将在下一节给出 D-ALCO 的 Tableau 判定算法,用于判定 D-ALCO 中任一公式的可满足性.

需要指出的是,给定某个 TBox \mathcal{T} 和某个 AActBox \mathcal{A}_c ,由于组成 \mathcal{T} 的概念定义式中可能会出现 \mathcal{A}_c 中定义的原子动作,而在 \mathcal{A}_c 中定义原子动作时又使用了 \mathcal{T} 中定义的非初始概念名,因而可能会出现循环定义的情况. 此外,即使不涉及 \mathcal{A}_c 时, \mathcal{T} 中也可能出现描述逻辑中遇到的循环定义^[1]. 循环定义会使得知识库丧失可定义性,即根据初始概念名的解释不能确定非初始概念名的相应解释. 含有循环定义时的判定算法是描述逻辑长期以来的研究难点,目前最好的研究结果尚不能处理描述逻辑 ALCO^[7-8], 因此,本文不考虑含有循环定义的情况.

3 D-ALCO 的 Tableau 判定算法

令 $\mathcal{T}, \mathcal{A}_c$ 分别为 TBox 和 AActBox. 下面先引入一系列符号和术语.

首先, 对于任一 D-ALCO 公式 φ , 用 $nf_{\mathcal{T}, \mathcal{A}_c}(\varphi)$ 表示根据如下过程构造得到的公式: (1) 对于 φ 中出现的每个被定义的概念名, 用 \mathcal{T} 中相应概念定义式右边的概念进行替换; (2) 对于 φ 中出现的每个原子动作, 根据 \mathcal{A}_c 中的原子动作定义式将其改写为由前提条件以及所产生的影响组成的二元组形式; (3) 反复进行上述过程; 直到 φ 中不含有被定义的概念名, 并且每个原子动作都已经改写为二元组形式.

其次, 在 D-ALCO 中引入形如 $@_u C$ 的概念; 其中 $u \in N_I \cup N_{IV}$, C 为任一概念. 对概念 $@_u C$ 的语义定义为: 对于任一 D-ALCO 模型 $M = (\Delta, W, I)$ 、该模型中的任一可能世界 $w \in W$ 以及基于 M 的任一指派 γ : 如果 $u^{I, \gamma} \in C^{I(w), \gamma}$, 则 $(@_u C)^{I(w), \gamma} = \Delta$; 否则 $(@_u C)^{I(w), \gamma} = \emptyset$, 其中的“ \emptyset ”表示空集.

引入概念 $@_u C$ 的目的, 是为了便于表述下文 B_2 -规则中对概念 $D^{Regress(\sigma, \varepsilon)}$ 的递归构造过程, 是判定算法中用到的一种临时表示形式. 实际上, 根据语义定义, 对于任一个体名或个体变元 v , 公式 $(@_u C)(v)$ 以及 $(\neg @_u C)(v)$ 分别与公式 $C(u)$ 和 $(\neg C)(u)$ 等价. 因此, 引入形如 $@_u C$ 的概念并没有增强 D-ALCO 的描述能力.

接下来, 对于由简单公式组成的任一集合 E , 用 E^- 表示集合 $\{\varphi^- \mid \varphi \in E\}$. 在此基础上, 引入前缀以及带前缀的公式.

定义 12. 任一前缀 σ, ε 由一个顺序动作 σ 以及关于简单公式的集合 ε 构成, 并且满足如下产生式规则:

$$\sigma, \varepsilon ::= (\emptyset, \emptyset), \emptyset \mid \sigma; (P, E), (\varepsilon \setminus E^-) \cup E,$$

其中的 (P, E) 为原子动作, $\sigma; (P, E)$ 是新生的顺序动作, $(\varepsilon \setminus E^-) \cup E$ 是新生的由简单公式组成的集合.

也将 $(\emptyset, \emptyset), \emptyset$ 称为初始前缀, 表示为 σ_0, ε_0 .

令 σ, ε 为前缀, φ 为公式, 则称 $\sigma, \varepsilon: \varphi$ 为一个带前缀的公式.

引入前缀的目的, 是为了在构建公式的语义模型时通过前缀区分出不同的可能世界, 并且通过前缀体现出这些可能世界之间存在的由原子动作引起的可达关系. 例如, 令前缀 $\sigma', \varepsilon' = \sigma; (P, E), (\varepsilon \setminus E^-) \cup E$; 如果前缀 σ, ε 和 σ', ε' 分别对应于某个模型 $M = (\Delta, W, I)$ 中的可能世界 w 以及 w' , 则 w 与 w' 之间满足

关系 $(w, w') \in (P, E)^I$.

下面分别引入分枝、分枝的扩展、饱和的分枝、可忽略的分枝以及存在冲突的分枝.

定义 13. 任一分枝 \mathcal{B} 是由以下 4 类元素组成的有限集合:

(1) 带前缀的公式;

(2) 形如 $X \equiv \langle \pi^* \rangle \varphi$ 的可能性标记, 其中的 $\langle \pi^* \rangle \varphi$ 是由迭代动作 π^* 和公式 φ 构成的动作存在性断言, X 为字符串;

(3) 形如 $u = v$ 的关于个体名和个体变元的等式, 其中 $u, v \in N_I \cup N_{IV}$;

(4) 形如 $u \neq v$ 的关于个体名和个体变元的不等式, 其中 $u, v \in N_I \cup N_{IV}$.

对于任一分枝 \mathcal{B} , 用 \mathcal{B}_{ind} 表示由 \mathcal{B} 中关于个体名和个体变元的所有等式以及不等式组成的集合. 在此基础上, 用 $\mathcal{B}_{\text{ind}}^*$ 表示满足以下条件的最小集合: ① $\mathcal{B}_{\text{ind}} \subseteq \mathcal{B}_{\text{ind}}^*$; ② 对于出现在 \mathcal{B}_{ind} 中的任一个体名或个体变元 u 都有 $u = u \in \mathcal{B}_{\text{ind}}^*$; ③ 如果 $u = v \in \mathcal{B}_{\text{ind}}^*$, 则有 $v = u \in \mathcal{B}_{\text{ind}}^*$; ④ 如果 $u = v \in \mathcal{B}_{\text{ind}}^*$ 并且 $v = r \in \mathcal{B}_{\text{ind}}^*$, 则有 $u = r \in \mathcal{B}_{\text{ind}}^*$; ⑤ 如果 $u \neq v \in \mathcal{B}_{\text{ind}}^*$, 则有 $v \neq u \in \mathcal{B}_{\text{ind}}^*$; ⑥ 如果 $u = v \in \mathcal{B}_{\text{ind}}^*$ 并且 $v \neq r \in \mathcal{B}_{\text{ind}}^*$, 则有 $u \neq r \in \mathcal{B}_{\text{ind}}^*$. 即 $\mathcal{B}_{\text{ind}}^*$ 是 \mathcal{B}_{ind} 关于等价关系“=”和对称关系“ \neq ”的闭包.

对于任一分枝 \mathcal{B} , 如果其满足表 1~表 4 中某条扩展规则的前件, 则可以应用该扩展规则对 \mathcal{B} 进行扩展, 在 \mathcal{B} 中加入新的带前缀的公式、新的可能性标记以及关于个体名和个体变元的等式或不等式.

定义 14. 对于任一分枝 \mathcal{B} , 如果表 1 到表 4 中所有扩展规则都不能对 \mathcal{B} 进行扩展, 则称 \mathcal{B} 为饱和的.

定义 15. 对于出现在分枝 \mathcal{B} 中的可能性标记 $X \equiv \langle \pi^* \rangle \varphi$, 如果存在某两个前缀 σ, ε 和 σ', ε' 使得 $\sigma, \varepsilon: X \in \mathcal{B}, \sigma', \varepsilon': \varphi \in \mathcal{B}$ 以及 $\varepsilon = \varepsilon'$, 则称 $X \equiv \langle \pi^* \rangle \varphi$ 在 \mathcal{B} 中被实现了; 否则称其未被实现.

定义 16. 对于任一分枝 \mathcal{B} , 如果 \mathcal{B} 是饱和的并且 \mathcal{B} 中含有未被实现的可能性标记, 则称 \mathcal{B} 为可忽略的, 否则称 \mathcal{B} 为不可忽略的.

定义 17. 对于任一分枝 \mathcal{B} , 如果至少出现以下某种情况, 则称 \mathcal{B} 存在冲突, 否则称 \mathcal{B} 为无冲突的:

(1) 存在某个公式 φ 以及某两个前缀 σ, ε 和 σ', ε' , 使得 $\sigma, \varepsilon: \varphi \in \mathcal{B}, \sigma', \varepsilon': \neg \varphi \in \mathcal{B}$ 以及 $\varepsilon = \varepsilon'$;

(2) 存在某个概念 C 、某个个体名或个体变元 u 以及某两个前缀 σ, ε 和 σ', ε' , 使得 $\sigma, \varepsilon: C(u) \in \mathcal{B}, \sigma', \varepsilon': (\neg C)(u) \in \mathcal{B}$ 以及 $\varepsilon = \varepsilon'$;

(3) 存在某个个体名或个体变元 u , 使得 $u \neq u \in \mathcal{B}_{\text{ind}}^*$.

下面对扩展规则进行简要说明. 表 1 和表 2 中的扩展规则直观地体现了 D-ALCO 中对各种形式的概念以及除 $\langle \pi \rangle \varphi$ 之外的各种公式的语义定义. 如果去掉个体变元, 去掉扩展规则中出现的各个前缀,

并且去掉 $\neg @$ -规则和 $@$ -规则, 则得到描述逻辑 ALCO 中关于概念和公式的扩展规则. 需要注意的是, $\neg \forall$ -规则和 \forall -规则中仅考察初始前缀 σ_0, ε_0 的情况.

表 1 关于概念的扩展规则

规则名	规 则
$\neg \neg_C$ -规则	如果 $\sigma, \varepsilon : (\neg(\neg C))(x) \in \mathcal{B}$, 并且 $\sigma, \varepsilon : C(x) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : C(x)\}$.
$\neg \{ \}$ -规则	如果 $\sigma, \varepsilon : (\neg \{u\})(x) \in \mathcal{B}$, 并且 $x \neq u \notin \mathcal{B}_{\text{ind}}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{x \neq u\}$.
$\{ \}$ -规则	如果 $\sigma, \varepsilon : \{u\}(x) \in \mathcal{B}$, 并且 $x = u \notin \mathcal{B}_{\text{ind}}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{x = u\}$.
$\neg \sqcup$ -规则	如果 $\sigma, \varepsilon : (\neg(C_1 \sqcup C_2))(x) \in \mathcal{B}$, 并且 $\sigma, \varepsilon : (\neg C_1)(x) \notin \mathcal{B}$ 或 $\sigma, \varepsilon : (\neg C_2)(x) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : (\neg C_1)(x), \sigma, \varepsilon : (\neg C_2)(x)\}$.
\sqcup -规则	如果 $\sigma, \varepsilon : (C_1 \sqcup C_2)(x) \in \mathcal{B}$, 并且 $\sigma, \varepsilon : C_1(x) \notin \mathcal{B}$ 以及 $\sigma, \varepsilon : C_2(x) \notin \mathcal{B}$, 则对于某个 $C \in \{C_1, C_2\}$ 令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : C(x)\}$.
$\neg @$ -规则	如果 $\sigma, \varepsilon : (\neg @_u C)(x) \in \mathcal{B}$, 并且 $\sigma, \varepsilon : (\neg C)(u) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : (\neg C)(u)\}$.
$@$ -规则	如果 $\sigma, \varepsilon : (@_u C)(x) \in \mathcal{B}$, 并且 $\sigma, \varepsilon : C(u) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : C(u)\}$.
$\neg \forall$ -规则	如果 $\sigma_0, \varepsilon_0 : (\neg(\forall R.C))(x) \in \mathcal{B}$, 并且不存在个体名或个体变元 y 使得 $\sigma_0, \varepsilon_0 : R(x, y) \in \mathcal{B}$ 以及 $\sigma_0, \varepsilon_0 : (\neg C)(y) \in \mathcal{B}$, 则引入一个未在 \mathcal{B} 中出现的个体名 y , 然后令 $\mathcal{B} := \mathcal{B} \cup \{\sigma_0, \varepsilon_0 : R(x, y), \sigma_0, \varepsilon_0 : (\neg C)(y)\}$.
\forall -规则	如果 $\sigma_0, \varepsilon_0 : (\forall R.C)(x) \in \mathcal{B}$, 并且 $\sigma_0, \varepsilon_0 : R(x, y) \in \mathcal{B}$ 以及 $\sigma_0, \varepsilon_0 : C(y) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma_0, \varepsilon_0 : C(y)\}$.

表 2 除 $\langle \pi \rangle \varphi$ 之外的关于公式的扩展规则

规则名	规 则
\neg_f -规则	如果 $\sigma, \varepsilon : \neg(C(x)) \in \mathcal{B}$, 并且 $\sigma, \varepsilon : (\neg C)(x) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : (\neg C)(x)\}$.
$\neg \neg_f$ -规则	如果 $\sigma, \varepsilon : \neg(\neg \varphi) \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \varphi\}$.
$\neg \vee$ -规则	如果 $\sigma, \varepsilon : \neg(\varphi \vee \psi) \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \neg \varphi \notin \mathcal{B}$ 或 $\sigma, \varepsilon : \neg \psi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \neg \varphi, \sigma, \varepsilon : \neg \psi\}$.
\vee -规则	如果 $\sigma, \varepsilon : \varphi \vee \psi \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \varphi \notin \mathcal{B}$ 以及 $\sigma, \varepsilon : \psi \notin \mathcal{B}$, 则对于某个 $\phi \in \{\varphi, \psi\}$ 令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \phi\}$.

表 3 给出了关于公式 $\langle \pi \rangle \varphi$ 以及动作的扩展规则; 其中假设各个原子动作 $\alpha(v_1, \dots, v_n)$ 都已经根据

其前提条件以及所产生的影响改写为二元组 (P, E) 的形式.

表 3 关于公式 $\langle \pi \rangle \varphi$ 和动作的扩展规则

规则名	规 则
$\neg \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \neg \langle \pi_1; \pi_2 \rangle \varphi \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \neg \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \neg \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi\}$.
$\langle \rangle$ -规则	如果 $\sigma, \varepsilon : \langle \pi_1; \pi_2 \rangle \varphi \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi\}$.
$\neg ? \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \neg \langle \psi? \rangle \varphi \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \neg \psi \notin \mathcal{B}$ 以及 $\sigma, \varepsilon : \neg \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \neg \psi\}$, 或者令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \neg \varphi\}$.
$? \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \langle \psi? \rangle \varphi \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \psi \notin \mathcal{B}$ 或者 $\sigma, \varepsilon : \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \psi, \sigma, \varepsilon : \varphi\}$.
$\neg \cup \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \neg \langle \pi_1 \cup \pi_2 \rangle \varphi \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \neg \langle \pi_1 \rangle \varphi \notin \mathcal{B}$ 或者 $\sigma, \varepsilon : \neg \langle \pi_2 \rangle \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \neg \langle \pi_1 \rangle \varphi, \sigma, \varepsilon : \neg \langle \pi_2 \rangle \varphi\}$.
$\cup \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \langle \pi_1 \cup \pi_2 \rangle \varphi \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \langle \pi_1 \rangle \varphi \notin \mathcal{B}$ 以及 $\sigma, \varepsilon : \langle \pi_2 \rangle \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \langle \pi_1 \rangle \varphi\}$, 或者令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \langle \pi_2 \rangle \varphi\}$.
$\neg * \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \neg \langle \pi^* \rangle \varphi \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \neg \varphi \notin \mathcal{B}$ 或者 $\sigma, \varepsilon : \neg \langle \pi \rangle \langle \pi^* \rangle \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \neg \varphi, \sigma, \varepsilon : \neg \langle \pi \rangle \langle \pi^* \rangle \varphi\}$.
$* \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \langle \pi^* \rangle \varphi \in \mathcal{B}$, 并且 \mathcal{B} 中不存在可能性标记 $X \equiv \langle \pi^* \rangle \varphi$ 使得 $\sigma, \varepsilon : X \in \mathcal{B}$, 则引入一个未在 \mathcal{B} 中出现的字符串 X , 然后令 $\mathcal{B} := \mathcal{B} \cup \{X \equiv \langle \pi^* \rangle \varphi, \sigma, \varepsilon : X\}$.
X-规则	如果 \mathcal{B} 中存在可能性标记 $X \equiv \langle \pi^* \rangle \varphi$ 使得 $\sigma, \varepsilon : X \in \mathcal{B}$, 并且 $\sigma, \varepsilon : \varphi \notin \mathcal{B}$ 以及 $\sigma, \varepsilon : \langle \pi \rangle X \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \varphi\}$, 或者令 $\mathcal{B} := \mathcal{B} \cup \{\neg \varphi, \sigma, \varepsilon : \langle \pi \rangle X\}$.
$\neg \text{atom} \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \neg \langle (P, E) \rangle \varphi \in \mathcal{B}$, 并且 $\{\sigma, \varepsilon : \neg \psi \mid \psi \in P\} \cap \mathcal{B} = \emptyset$ 以及存在某个前缀 σ', ε' 使得 $\varepsilon' = (\varepsilon \setminus E \neg) \cup E$ 和 $\sigma', \varepsilon' : \neg \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma', \varepsilon' : \neg \varphi\}$, 或者对于某个 $\psi \in P$ 令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \neg \psi\}$.
$\text{atom} \langle \rangle$ -规则	如果 $\sigma, \varepsilon : \langle (P, E) \rangle \varphi \in \mathcal{B}$, 并且 $\{\sigma, \varepsilon : \psi \mid \psi \in P\} \not\subseteq \mathcal{B}$, 或者不存在某个前缀 σ', ε' 使得 $\varepsilon' = (\varepsilon \setminus E \neg) \cup E$ 和 $\sigma', \varepsilon' : \varphi \in \mathcal{B}$, 则 1. 首先, 如果 $\{\sigma, \varepsilon : \psi \mid \psi \in P\} \not\subseteq \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \varepsilon : \psi \mid \psi \in P\}$; 2. 接下来, 如果不存在某个前缀 σ', ε' 使得 $\varepsilon' = (\varepsilon \setminus E \neg) \cup E$ 以及 $\sigma', \varepsilon' : \varphi \in \mathcal{B}$, 则 2. 1. 如果不存在某个前缀 σ', ε' 使得 $\varepsilon' = (\varepsilon \setminus E \neg) \cup E$, 则引入前缀 $\sigma', \varepsilon' := \sigma; (P, E).(\varepsilon \setminus E \neg) \cup E$, 然后令 $\mathcal{B} := \mathcal{B} \cup \{\sigma', \varepsilon' : \varphi\}$; 2. 2. 否则, 对于某个满足 $\varepsilon' = (\varepsilon \setminus E \neg) \cup E$ 的前缀 σ', ε' , 令 $\mathcal{B} := \mathcal{B} \cup \{\sigma', \varepsilon' : \varphi\}$.

表 4 进一步给出了关于动作所产生效果的扩展规则,体现了本文在定义原子动作的语义时采用的可能模型途径.其中,对于 B_2 -规则来说,令 $Obj(\epsilon)$ 是由出现在 ϵ 中的所有个体名和个体变元组成的集合,则 $D^{Regress(\sigma, \epsilon)}$ 表示从概念 D 出发根据前缀 σ, ϵ 中 ϵ 的内容新构造出的概念,并且递归构造如下:

① 对于简单概念名 C_i ,有

$$C_i^{Regress(\sigma, \epsilon)} := (C_i \sqcup \bigsqcup_{C_i(u) \in \epsilon} \{u\}) \sqcap \bigsqcap_{\neg C_i(u) \in \epsilon} \neg \{u\};$$

② $\{u\}^{Regress(\sigma, \epsilon)} := \{u\};$

$$\textcircled{3} (\textcircled{a}_u C)^{Regress(\sigma, \epsilon)} := \textcircled{a}_u C^{Regress(\sigma, \epsilon)};$$

$$\textcircled{4} (\neg C)^{Regress(\sigma, \epsilon)} := \neg C^{Regress(\sigma, \epsilon)};$$

$$\textcircled{5} (C \sqcup C')^{Regress(\sigma, \epsilon)} := C^{Regress(\sigma, \epsilon)} \sqcup C'^{Regress(\sigma, \epsilon)};$$

$$\textcircled{6} (\forall R.C)^{Regress(\sigma, \epsilon)} :=$$

$$((\bigsqcup_{u \in Obj(\epsilon)} \{u\}) \sqcup \forall R.C^{Regress(\sigma, \epsilon)}) \sqcap$$

$$((\bigsqcap_{u \in Obj(\epsilon)} \neg \{u\}) \sqcup \forall R.((\bigsqcup_{v \in Obj(\epsilon)} \{v\}) \sqcup C^{Regress(\sigma, \epsilon)})) \sqcap$$

$$\bigsqcap_{u, v \in Obj(\epsilon), R(u, v) \notin \epsilon, \neg R(u, v) \notin \epsilon} ((\neg \{u\}) \sqcup \forall R.((\neg \{v\}) \sqcup$$

$$C^{Regress(\sigma, \epsilon)})) \sqcap \bigsqcap_{R(u, v) \in \epsilon} ((\neg \{u\}) \sqcup \textcircled{a}_v C^{Regress(\sigma, \epsilon)}).$$

表 4 关于动作所产生效果的扩展规则

规则名	规 则
B_1 -规则	如果 $\sigma, \epsilon: \varphi \in \mathcal{B}$, φ 为简单公式, $\varphi \notin \epsilon$, 并且 $\sigma_0, \epsilon_0: \varphi \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma_0, \epsilon_0: \varphi\}$.
B_2 -规则	如果 $\sigma, \epsilon: D(u) \in \mathcal{B}$, D 是形如 $\forall R.C$ 或 $\neg \forall R.C$ 的概念, 并且 $\sigma_0, \epsilon_0: D^{Regress(\sigma, \epsilon)}(u) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma_0, \epsilon_0: D^{Regress(\sigma, \epsilon)}(u)\}$.

该构造过程保证了如下性质:对于任一模型 $M=(\Delta, W, I)$, W 中的任意两个可能世界 w, w' 以及基于 M 的任一指派 γ :如果 $(w, w') \in \sigma^I$, 则必然有 $(D^{Regress(\sigma, \epsilon)})^I(w, \gamma) = D^I(w', \gamma)$. 相关证明可见引理 1.

最后,对 D-ALCO 的 Tableau 判定算法描述如下.

算法 1. 对于 D-ALCO 中的任一公式 φ ,按以下步骤判定 φ 相对于 \mathcal{T} 和 \mathcal{Ac} 是否可满足:

1. 构造一个分枝 $\mathcal{B} := \{\sigma_0, \epsilon_0: nf_{\mathcal{T}, \mathcal{Ac}}(\varphi)\}$;
2. 以任意顺序应用表 1 到表 4 中的扩展规则对 \mathcal{B} 进行扩展;
3. 如果通过扩展后能得到某个饱和的分枝 \mathcal{B}' , 并且 \mathcal{B}' 是无冲突的以及不可忽略的, 则返回结果“ φ 相对于 \mathcal{T} 和 \mathcal{Ac} 是可满足的”;否则返回结果“ φ 相对于 \mathcal{T} 和 \mathcal{Ac} 是不可满足的”.

4 判定算法的性质

本节证明上文给出的 Tableau 判定算法是可终止的、可靠的以及完备的.

4.1 算法的可终止性

令 $\mathcal{T}, \mathcal{Ac}$ 分别为 TBox 和 AActBox, φ 为需要判定可满足性的公式.

首先,对于任一概念、公式或动作 Y ,用 $size(Y)$ 表示如下集合中的所有元素在 Y 中累计出现的次数: $N_C \cup N_R \cup N_I \cup N_{IV} \cup \{\neg, \sqcup, \forall, <, >, \vee, \cup, ;, *, ?, @\}$.

其次,用 $cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 表示满足以下条件的最小集合:

- (1) $nf_{\mathcal{T}, \mathcal{Ac}}(\varphi) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;
- (2) 如果 $\psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 并且 ψ 不是以“ \neg ”

开头的公式, 则 $\neg \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(3) 如果 $\neg \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $\psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(4) 如果 $\phi \vee \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $\{\phi, \psi\} \subseteq cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(5) 如果 $\langle (P, E) \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $P \subseteq cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, $E \subseteq cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 以及 $\psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(6) 如果 $\langle \phi? \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $\phi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 以及 $\psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(7) 如果 $\langle \pi \cup \pi' \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $\langle \pi \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 以及 $\langle \pi' \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(8) 如果 $\langle \pi; \pi' \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $\langle \pi \rangle \langle \pi' \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(9) 如果 $\langle \pi^* \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $\langle \pi \rangle \langle \pi^* \rangle \psi \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(10) 如果 $\neg(C(x)) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $(\neg C)(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(11) 如果 $C(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 并且 C 不是以“ \neg ”开头的概念, 则 $(\neg C)(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(12) 如果 $(\neg C)(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $C(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(13) 如果 $(\textcircled{a}_u C)(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $C(u) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$;

(14) 如果 $(C \sqcup D)(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$, 则 $C(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 以及 $D(x) \in cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$.

显然, 集合 $cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 中元素的个数 $\#cl_f(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 与 $size(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 成线性关系.

此外,用 $AtomAct(nf_{\mathcal{T}, \mathcal{Ac}}(\varphi))$ 表示由出现在

$nf_{T,Ac}(\varphi)$ 中的所有原子动作组成的集合; 在此基础上, 用 $Eff(nf_{T,Ac}(\varphi))$ 表示集合 $\bigcup_{(P_i, E_i) \in AtomAct(nf_{T,Ac}(\varphi))} E_i$.

定理 1. 算法 1 是可终止的.

证明.

令 $s := size(nf_{T,Ac}(\varphi))$, $f := \#cl_f(nf_{T,Ac}(\varphi))$, $e := \#Eff(nf_{T,Ac}(\varphi))$, $count_1$ 为 $nf_{T,Ac}(\varphi)$ 中出现的不同的个体名和个体变元的个数, $count_V$ 为概念构造符“ \forall ”在 $nf_{T,Ac}(\varphi)$ 中累计出现的次数, $count_*$ 为动作构造符“ $*$ ”在 $nf_{T,Ac}(\varphi)$ 中累计出现的次数, 则算法 1 具有以下性质:

(1) 每应用一次任意一条扩展规则之后, 分枝 \mathcal{B} 中的元素个数都至少增加 1.

(2) 所有可能性标记都是通过 $\langle \rangle$ -规则引入的, 因此, 分枝 \mathcal{B} 中出现的所有可能性标记最多为 $count_*$ 个.

(3) 除初始前缀 $\sigma_0.\epsilon_0$ 之外, 所有前缀都是通过应用 $atom_{\langle \rangle}$ -规则引入的; 根据该规则, 任一前缀 $\sigma.\epsilon$ 中的 ϵ 都与其它前缀中的不同. 因此, 分枝 \mathcal{B} 中除 $\sigma_0.\epsilon_0$ 之外的前缀最多有 $2^e - 1$ 个.

(4) 对于任一非初始前缀 $\sigma.\epsilon$, 分枝 \mathcal{B} 中出现的以 $\sigma.\epsilon$ 为前缀的带前缀的公式最多为 f 个.

(5) 应用 B_2 -规则引入的形如 $\sigma_0.\epsilon_0 : D^{Regress(\sigma.\epsilon)}(u)$ 的带前缀的公式最多为 $count_V \times (2^e - 1)$ 个.

并且, 对于应用 B_2 -规则引入的任一带前缀的公式 $\sigma_0.\epsilon_0 : D^{Regress(\sigma.\epsilon)}(u)$, 根据对概念 $D^{Regress(\sigma.\epsilon)}$ 的构造过程, 必然存在某两个多项式 p_1 和 p_2 , 使得 $size(D^{Regress(\sigma.\epsilon)}(u))$ 最多为 $2^{p_1(size(D))} \times p_2(\#\epsilon)$, 其中 $\#\epsilon$ 表示集合 ϵ 中元素的个数; 该结论与文献[9]关于 ABox 更新算法复杂度的相应结果类似. 同时, 如果概念构造符“ \forall ”在概念 D 中出现的次数为 i , 则“ \forall ”在概念 $D^{Regress(\sigma.\epsilon)}$ 中出现的次数最多为 $(count_1 \times (count_1 - 1) + 2)^i$.

此外, 对于应用 B_2 -规则引入的任一带前缀的公式 $\sigma_0.\epsilon_0 : D^{Regress(\sigma.\epsilon)}(u)$, 在其基础上进一步应用表 1 中除 $\rightarrow V$ -和 V -规则之外的所有规则之后, 所引入的以 $\sigma_0.\epsilon_0$ 为前缀的带前缀的公式在数量上最多与 $2^{p_1(size(D))} \times p_2(\#\epsilon)$ 成线性关系. 由于 $size(D) \leq s$ 和 $\#\epsilon \leq e$, 因此, 必然存在某个与 p_2 相关的多项式 $p_{2,1}$, 使得在没有应用 $\rightarrow V$ -规则和 V -规则的情况下, 分枝 \mathcal{B} 中出现的以 $\sigma_0.\epsilon_0$ 为前缀的带前缀的公式最多为 $f + 2^{p_1(s)} \times p_{2,1}(e) \times count_V \times (2^e - 1)$ 个; 并且, “ \forall ”在这些带前缀的公式中出现的次数最多为 $(count_1 \times (count_1 - 1) + 2)^{count_V} \times (count_V \times (2^e -$

$1)) + count_V$, 简记为 NUM_V .

(6) 考察应用 $\rightarrow V$ -规则和 V -规则的情况. 应用 $\rightarrow V$ -规则会引入新的个体名或个体变元; 对应于每个“ \forall ”符号最多能应用一次 $\rightarrow V$ -规则, 因此, 最多会引入 NUM_V 个个体名或个体变元.

每应用 $\rightarrow V$ -规则引入一个个体名或个体变元 y 的同时, 将会引入某两个带前缀的公式 $\sigma_0.\epsilon_0 : R(x, y)$ 和 $\sigma_0.\epsilon_0 : C(y)$. 之后, 由于 y 的引入, 针对 \mathcal{B} 中每对带前缀的公式 $\sigma_0.\epsilon_0 : (\forall R.C)(x)$ 和 $\sigma_0.\epsilon_0 : R(x, y)$, 将会进一步通过应用 V -规则引入一个带前缀的公式 $\sigma_0.\epsilon_0 : C(y)$; 总共最多会引入 $NUM_V - 1$ 个这样的公式. 再接下来, 对于这些公式中的每个成员 $\sigma_0.\epsilon_0 : C(y)$, 在其基础上进一步应用表 1 中除 $\rightarrow V$ -和 V -规则之外的所有扩展规则之后, 所引入的以 $\sigma_0.\epsilon_0$ 为前缀的带前缀的公式在数量上最多与 $size(C)$ 成线性关系; 由于 $size(C)$ 小于等于 s 与 $2^{p_1(s)} \times p_{2,1}(e)$ 中的最大值, 因而必然存在某个自然数 k 使得这些带前缀的公式最多为 $k \times (s + 2^{p_1(s)} \times p_{2,1}(e))$.

因此, 由于新的个体名或个体变元的引入, 最多将增加 $(2 \times NUM_V) + NUM_V \times (NUM_V - 1) \times (k \times (s + 2^{p_1(s)} \times p_{2,1}(e)))$ 个以 $\sigma_0.\epsilon_0$ 为前缀的带前缀的公式.

(7) 由于所有个体名和个体变元的数量最多为 $count_1 + NUM_V$, 因此, 根据 $\{\}$ -规则, 分枝 \mathcal{B} 中出现的关于个体名和个体变元的等式最多为 $count_1 + NUM_V - 1$ 个. 同理, 根据 $\rightarrow \{\}$ -规则, 分枝 \mathcal{B} 中出现的关于个体名和个体变元的不等式最多为 $(count_1 + NUM_V) \times (count_1 + NUM_V - 1) / 2$ 个.

综上, 每应用一次扩展规则之后, \mathcal{B} 的元素个数至少增加 1; 同时, \mathcal{B} 中所有元素的个数是有限的. 因此, 算法是可终止的. 并且, 对上述数据化简可得, 算法中应用扩展规则的次数最多为 $p_3(s) \times 2^{p_4(s)} \times count_1^{4 \times count_V}$, 其中的 p_3, p_4 为多项式. 证毕.

4.2 算法的可靠性和完备性

首先, 引入分枝-模型映射, 将分枝 \mathcal{B} 中出现的前缀映射为某个 D-ALCO 模型中的可能世界.

定义 18. 令 $\mathcal{T}, \mathcal{Ac}$ 分别为 TBox 和 AActBox, \mathcal{B} 为分枝, Σ_p 为出现在 \mathcal{B} 中的所有前缀, $M = (\Delta, W, I)$ 为 D-ALCO 模型, γ 为基于 M 的指派. 将某个函数 $\delta : \Sigma_p \rightarrow W$ 称为相对于 γ 的从 \mathcal{B} 到 M 的分枝-模型映射, 当且仅当对于 Σ_p 中的任意两个前缀 $\sigma.\epsilon$ 和 $\sigma'.\epsilon'$: 如果存在某个原子动作 (P, E) 使得 $\sigma' = \sigma; (P, E)$ 以及 $\epsilon' = (\epsilon \setminus E^-) \cup E$, 则必然有

(1) 对于出现在 \mathcal{B} 中的任一简单概念名 C 都有

$C^{I(\delta(\sigma', \epsilon'))}, \gamma = (C^{I(\delta(\sigma, \epsilon))}, \gamma \cup \{u^{I, \gamma} \mid C(u) \in E\}) \setminus \{u^{I, \gamma} \mid \rightarrow C(u) \in E\}$ 以及

(2) 对于出现在 \mathcal{B} 中的任一角色名 R 都有 $R^{I(\delta(\sigma', \epsilon'))}, \gamma = (R^{I(\delta(\sigma, \epsilon))}, \gamma \cup \{(u^{I, \gamma}, v^{I, \gamma}) \mid R(u, v) \in E\}) \setminus \{(u^{I, \gamma}, v^{I, \gamma}) \mid \rightarrow R(u, v) \in E\}$.

根据定义 12 中关于前缀的产生式规则, 容易证明分枝-模型映射具有如下性质.

性质 1. 令 Σ_p 为出现在分枝 \mathcal{B} 中的所有前缀, γ 为基于模型 $M = (\Delta, W, I)$ 的指派, 函数 $\delta: \Sigma_p \rightarrow W$ 是相对于 γ 的从 \mathcal{B} 到 M 的分枝-模型映射当且仅当对于 Σ_p 中的任一前缀 σ, ϵ 都满足:

(1) 对于出现在 \mathcal{B} 中的任一简单概念名 C 都有 $C^{I(\delta(\sigma, \epsilon))}, \gamma = (C^{I(\delta(\sigma_0, \epsilon_0))}, \gamma \cup \{u^{I, \gamma} \mid C(u) \in \epsilon\}) \setminus \{u^{I, \gamma} \mid \rightarrow C(u) \in \epsilon\}$ 以及

(2) 对于出现在 \mathcal{B} 中的任一角色名 R 都有 $R^{I(\delta(\sigma, \epsilon))}, \gamma = (R^{I(\delta(\sigma_0, \epsilon_0))}, \gamma \cup \{(u^{I, \gamma}, v^{I, \gamma}) \mid R(u, v) \in \epsilon\}) \setminus \{(u^{I, \gamma}, v^{I, \gamma}) \mid \rightarrow R(u, v) \in \epsilon\}$.

基于该性质可进一步得出如下结论.

性质 2. 令 δ 是相对于 γ 的从 \mathcal{B} 到 M 的分枝-模型映射, 则

(1) 对于 \mathcal{B} 中出现的任意两个前缀 $\sigma, \epsilon, \sigma', \epsilon'$ 来说, $\delta(\sigma, \epsilon) = \delta(\sigma', \epsilon')$ 当且仅当 $\epsilon = \epsilon'$;

(2) 对于 \mathcal{B} 中出现的任一前缀 σ, ϵ 以及 ϵ 中的任一公式 φ , 都有 $(M, \delta(\sigma, \epsilon), \gamma) \models \varphi$.

应用分枝-模型映射, 可以证明 B_2 -规则中对概念 $D^{Regress(\sigma, \epsilon)}$ 的递归构造过程具有如下引理.

引理 1. 令 $D^{Regress(\sigma, \epsilon)}$ 为 B_2 -规则中应用递归构造过程得到的概念, 则对于任一模型 $M = (\Delta, W, I)$, 基于 M 的任一指派 γ 以及相对于 γ 的从 \mathcal{B} 到 M 的任一分枝-模型映射 δ , 都有 $(D^{Regress(\sigma, \epsilon)})^{I(\delta(\sigma_0, \epsilon_0))}, \gamma = D^{I(\delta(\sigma, \epsilon))}, \gamma$.

证明. 对概念 D 的结构进行归纳.

如果 D 形如 $\{u\}$, 结论显然成立. 如果 D 为简单概念名 C_i , 则代入递归构造过程后再根据性质 1, 可得 $C_i^{I(\delta(\sigma, \epsilon))}, \gamma = (C_i^{Regress(\sigma, \epsilon)})^{I(\delta(\sigma_0, \epsilon_0))}, \gamma$.

如果 D 形如 $@_u C, \neg C$ 或者 $C \sqcup C'$, 基于归纳假设以及概念的语义定义, 结论显然成立.

如果 D 形如 $\forall R.C$, 则首先可观察到如下结论: 对于解释域 Δ 中的任意两个个体 x, y 以及任一角色名 R 来说, $(x, y) \in R^{I(\delta(\sigma, \epsilon))}, \gamma$ 当且仅当如下情况之一成立:

(1) $x \notin \bigcup_{u \in Obj(\epsilon)} \{u^{I, \gamma}\}$ 并且 $(x, y) \in R^{I(\delta(\sigma_0, \epsilon_0))}, \gamma$, 即动作 σ 的执行没有对 x 产生影响;

(2) $x \in \bigcup_{u \in Obj(\epsilon)} \{u^{I, \gamma}\}$, $y \notin \bigcup_{v \in Obj(\epsilon)} \{v^{I, \gamma}\}$ 并且 $(x, y) \in R^{I(\delta(\sigma_0, \epsilon_0))}, \gamma$, 即 σ 的执行对 x 产生了影响, 但没有对 y 产生影响;

(3) 存在 $u, v \in Obj(\epsilon)$ 使得 $x = u^{I, \gamma}$, $y = v^{I, \gamma}$, $\rightarrow R(u, v) \notin \epsilon$, $R(u, v) \notin \epsilon$ 以及 $(x, y) \in R^{I(\delta(\sigma_0, \epsilon_0))}, \gamma$, 即 σ 的执行对 x 和 y 都产生了影响, 但在 $R^{I(\delta(\sigma_0, \epsilon_0))}, \gamma$ 的基础上既没有加入 (x, y) 也没有删去 (x, y) ;

(4) 存在 $u, v \in Obj(\epsilon)$ 使得 $x = u^{I, \gamma}$, $y = v^{I, \gamma}$ 以及 $R(u, v) \in \epsilon$, 即 σ 的执行对 x 和 y 都产生了影响, 并且在 $R^{I(\delta(\sigma_0, \epsilon_0))}, \gamma$ 的基础上加入了 (x, y) .

接下来, 令 x 为 Δ 中的任一个体; 代入递归构造过程并进行逻辑变换后容易得出: $x \in ((\forall R.C)^{Regress(\sigma, \epsilon)})^{I(\delta(\sigma_0, \epsilon_0))}, \gamma$ 当且仅当 $x \in (\forall R.C)^{I(\delta(\sigma, \epsilon))}, \gamma$. 因此可得

$$((\forall R.C)^{Regress(\sigma, \epsilon)})^{I(\delta(\sigma_0, \epsilon_0))}, \gamma = (\forall R.C)^{I(\delta(\sigma, \epsilon))}, \gamma.$$

证毕.

上述证明中, D 形如 $\forall R.C$ 时的分析参考了文献[9]中对 ABox 更新算法正确性的证明.

接下来, 引入分枝的可满足性.

定义 19. 令 \mathcal{B} 为分枝, $M = (\Delta, W, I)$ 为 D-ALCO 模型, γ 为基于 M 的指派, δ 为相对于 γ 的从 \mathcal{B} 到 M 的分枝-模型映射, 则称模型 M 、指派 γ 和映射 δ 满足分枝 \mathcal{B} , 表示为 $M, \gamma, \delta \models \mathcal{B}$, 当且仅当满足以下条件:

(1) 对于 \mathcal{B} 中任一带前缀的公式 $\sigma, \epsilon, \varphi$, 都有 $(M, \delta(\sigma, \epsilon), \gamma) \models \varphi$;

(2) 对于 \mathcal{B} 中任一关于个体名和个体变元的等式 $u = v$, 都有 $u^{I, \gamma} = v^{I, \gamma}$;

(3) 对于 \mathcal{B} 中任一关于个体名和个体变元的不等式 $u \neq v$, 都有 $u^{I, \gamma} \neq v^{I, \gamma}$.

定义 20. 对于任一分枝 \mathcal{B} , 如果存在某个模型 M 、某个基于 M 的指派 γ 以及某个相对于 γ 的从 \mathcal{B} 到 M 的分枝-模型映射 δ , 使得 $M, \gamma, \delta \models \mathcal{B}$, 则称分枝 \mathcal{B} 是可满足的.

基于上述定义, 下面依次证明算法 1 的几条性质, 分别如引理 2~引理 5 所述.

引理 2. 令 \mathcal{B} 为任一分枝. 对于在 \mathcal{B} 上应用任意一条扩展规则的过程来说, \mathcal{B} 在扩展之前是可满足的当且仅当可以在扩展后得到某个可满足的分枝.

证明. 一方面, 令 \mathcal{B}' 是在 \mathcal{B} 上应用扩展规则后得到的任一分枝, 则必然有 $\mathcal{B} \subseteq \mathcal{B}'$; 因此, 如果 \mathcal{B}' 是可满足的, 则 \mathcal{B} 也是可满足的. 另一方面, 令存在某

个模型 M 、基于 M 的指派 γ 以及相对于 γ 的从 \mathcal{B} 到 M 的分枝-模型映射 δ , 使得 $M, \gamma, \delta \models \mathcal{B}$; 对表 1~表 4 中的扩展规则逐一进行考察, 基于对概念、公式以及动作的语义定义, 容易证明结论成立. 证毕.

引理 3. 如果分枝 \mathcal{B} 是饱和的、无冲突的并且不可忽略的, 则 \mathcal{B} 一定是可满足的.

证明. 令 $N_I^{\mathcal{B}}$ 、 $N_{IV}^{\mathcal{B}}$ 、 $N_C^{\mathcal{B}}$ 、 $N_R^{\mathcal{B}}$ 以及 $\Sigma_p^{\mathcal{B}}$ 分别是由出现在 \mathcal{B} 中的所有个体名、个体变元、概念名、角色名以及前缀组成的集合. 按如下步骤构造模型 $M = (\Delta, W, I)$ 、基于 M 的指派 γ 以及函数 $\delta: \Sigma_p^{\mathcal{B}} \rightarrow W$.

(1) 构造解释域 $\Delta := \{[u]_{=\mathcal{B}} \mid u \in N_I^{\mathcal{B}} \cup N_{IV}^{\mathcal{B}}\}$, 其中, $[u]_{=\mathcal{B}}$ 是个体名或个体变元 u 在 \mathcal{B} 中关于等价关系“=”的等价类, 即 $[u]_{=\mathcal{B}} := \{v \mid u = v \in \mathcal{B}_{\text{Ind}}^*\}$.

(2) 对于 $N_I^{\mathcal{B}}$ 中的各个个体名 p , 将其解释为 $p^I := [p]_{=\mathcal{B}}$.

(3) 对于 $N_{IV}^{\mathcal{B}}$ 中的各个个体变元 v , 将其指派为 $\gamma(v) := [v]_{=\mathcal{B}}$.

(4) 对应于初始前缀 σ_0, ϵ_0 , 引入一个可能世界 w_0 . 为该可能世界构造解释 $I(w_0)$, 将 $N_C^{\mathcal{B}}$ 中的任一概念名 C 解释为 $C^{I(w_0)} := \{u^{I, \gamma} \mid \sigma_0, \epsilon_0 : C(u) \in \mathcal{B}\}$, 将 $N_R^{\mathcal{B}}$ 中的任一角色名 R 解释为 $R^{I(w_0)} := \{(u^{I, \gamma}, v^{I, \gamma}) \mid \sigma_0, \epsilon_0 : R(u, v) \in \mathcal{B}\}$. 在此基础上, 设置 $\delta(\sigma_0, \epsilon_0) := w_0$.

(5) 对应于 Σ_p 中的每个非初始前缀 σ_i, ϵ_i , 相应地引入一个可能世界 w_i . 为每个可能世界 w_i 构造解释 $I(w_i)$, 将 $N_C^{\mathcal{B}}$ 中的任一概念名 C 解释为 $C^{I(w_i)} := (C^{I(w_0)} \cup \{u^{I, \gamma} \mid C(u) \in \epsilon_i\}) \setminus \{u^{I, \gamma} \mid \neg C(u) \in \epsilon_i\}$, 将 $N_R^{\mathcal{B}}$ 中的任一角色名 R 解释为 $R^{I(w_i)} := (R^{I(w_0)} \cup \{(u^{I, \gamma}, v^{I, \gamma}) \mid R(u, v) \in \epsilon_i\}) \setminus \{(u^{I, \gamma}, v^{I, \gamma}) \mid \neg R(u, v) \in \epsilon_i\}$. 在此基础上, 设置 $\delta(\sigma_i, \epsilon_i) = w_i$.

根据性质 1, δ 是相对于 γ 的从 \mathcal{B} 到 M 的分枝-模型映射. 接下来只需要证明 $M, \gamma, \delta \models \mathcal{B}$.

首先, 对于 \mathcal{B} 中关于个体名和个体变元的任一等式 $u = v$, 必然有 $u^I = [u]_{=\mathcal{B}} = [v]_{=\mathcal{B}} = v^I$.

其次, 由于 \mathcal{B} 是无冲突的, 对应于 \mathcal{B} 中关于个体名和个体变元的任一不等式 $u \neq v$, 必然有 $u = v \notin \mathcal{B}_{\text{Ind}}^*$ (否则会出现 $u \neq u \in \mathcal{B}_{\text{Ind}}^*$, 导致矛盾), 因此有 $[u]_{=\mathcal{B}} \neq [v]_{=\mathcal{B}}$, 即 $u^I \neq v^I$.

最后, 对公式 φ 的结构进行归纳, 证明对于任一 $\sigma, \epsilon: \varphi \in \mathcal{B}$ 都有 $(M, \delta(\sigma, \epsilon), \gamma) \models \varphi$. 在归纳证明的过程中, 对应于公式 φ 的每一种形式, 判定算法中都有一条扩展规则与之对应; 基于归纳假设以及动作和公式的语义, 可以容易地得出结论. 证毕.

引理 4. 对于任一分枝 \mathcal{B} , 如果 \mathcal{B} 存在冲突, 则

\mathcal{B} 一定是不可满足的.

根据定义 17 和性质 2, 该引理显然成立.

引理 5. 如果算法 1 构造的初始分枝 $\mathcal{B} = \{\sigma_0, \epsilon_0 : nf_{T, A_c}(\varphi)\}$ 是可满足的, 则在应用扩展规则后得到的所有饱和的并且可满足的分枝中, 必然存在某个分枝是不可忽略的.

证明. 首先, 引入关于前缀的偏序关系“ \leq_p ”, 对其递归定义为 ① $\sigma, \epsilon \leq_p \sigma, \epsilon$, ② $\sigma, \epsilon \leq_p \sigma'; (P, E).(\epsilon \setminus E^-) \cup E$, ③ 如果 $\sigma, \epsilon \leq_p \sigma'. \epsilon'$ 并且 $\sigma'. \epsilon' \leq_p \sigma''. \epsilon''$, 则 $\sigma, \epsilon \leq_p \sigma''. \epsilon''$.

下面用反证法证明. 假设应用扩展规则后得到的各个饱和且可满足的分枝都是可忽略的.

需要指出的是, 各个可忽略的分枝可能是由于不同的未实现的可能性标记导致的; 但不失一般性, 我们只需考察某个可能性标记 $X \equiv \langle \pi^* \rangle \varphi$. 不妨令 \mathcal{B}' 是应用扩展规则后得到的某个饱和的分枝, \mathcal{B}' 中含有未被实现的可能性标记 $X \equiv \langle \pi^* \rangle \varphi$, 并且 \mathcal{B}' 是可满足的, 即存在某个模型 $M = (\Delta, W, I)$ 、基于 M 的指派 γ 以及相对于 γ 的从 \mathcal{B}' 到 M 的分枝-模型映射 δ , 使得 $M, \gamma, \delta \models \mathcal{B}'$.

令 $path(X, \mathcal{B}')$ 表示由出现在 \mathcal{B}' 中的满足 $\sigma, \epsilon: X \in \mathcal{B}'$ 的所有前缀组成的集合, 即 $path(X, \mathcal{B}') := \{\sigma, \epsilon \mid \sigma, \epsilon: X \in \mathcal{B}'\}$, 则 $path(X, \mathcal{B}')$ 与关系“ \leq_p ”一起构成一个全序集. 此外, 由于 $path(X, \mathcal{B}')$ 是有限集, 不妨令 $path(X, \mathcal{B}')$ 中的所有元素为 $\sigma_1, \epsilon_1, \dots, \sigma_m, \epsilon_m$, 并且对于任一 $1 \leq i < j \leq m$ 都有 $\sigma_i, \epsilon_i \leq_p \sigma_j, \epsilon_j$, 即 $path(X, \mathcal{B}')$ 中的每个前缀 σ_j, ϵ_j 都是在前一个前缀 $\sigma_{j-1}, \epsilon_{j-1}$ 的基础上, 通过应用 X-规则, 然后再经历一个 π -过程后引入的, 根据 X-规则, 对于 $path(X, \mathcal{B}')$ 中的任一前缀 σ, ϵ 都有 $\sigma, \epsilon: \neg \varphi \in \mathcal{B}'$.

令 σ_M, ϵ_M 是出现在 \mathcal{B}' 中的满足 $\sigma_m, \epsilon_m \leq_p \sigma_M, \epsilon_M$ 的最大前缀, 即对于 \mathcal{B}' 中满足 $\sigma_m, \epsilon_m \leq_p \sigma, \epsilon$ 的任一前缀 σ, ϵ 都有 $\sigma, \epsilon \leq_p \sigma_M, \epsilon_M$. 不失一般性, 令组成 σ_M, ϵ_M 的动作 σ_M 为 $\sigma_m; (P_1, E_1); \dots; (P_K, E_K)$. 由于 \mathcal{B}' 是饱和的, 根据 $atom_{\langle \rangle}$ -规则, \mathcal{B}' 中必然存在某个形如 $\sigma_M, \epsilon_M: \langle (P_{K+1}, E_{K+1}) \rangle \langle \pi' \rangle X$ 的带有前缀的公式、某个前缀 σ_N, ϵ_N 以及相应的带前缀的公式 $\sigma_N, \epsilon_N: \langle \pi' \rangle X$, 满足 $\epsilon_N = (\epsilon_M \setminus E_{K+1}^-) \cup E_{K+1}$ 以及 $\langle (P_1, E_1); \dots; (P_K, E_K); (P_{K+1}, E_{K+1}); \pi' \rangle^I = \pi^I$.

令 σ_n, ϵ_n 是 $path(X, \mathcal{B}')$ 中满足 $\sigma_n, \epsilon_n \leq_p \sigma_N, \epsilon_N$ 的最大前缀, 即对于 $path(X, \mathcal{B}')$ 中满足 $\sigma, \epsilon \leq_p \sigma_N, \epsilon_N$ 的任一前缀 σ, ϵ 都有 $\sigma, \epsilon \leq_p \sigma_n, \epsilon_n$. 由于 $\sigma_n, \epsilon_n: X \in \mathcal{B}'$ 和 $(M, \delta, \gamma) \models \mathcal{B}'$, 必然存在某个非负整数 u 使得从

$\delta(\sigma_n, \epsilon_n)$ 出发经过 u 个 π -过程后到达满足 φ 的可能世界,即存在 W 中的 $u+1$ 个可能世界 $w_n, w_{n+1}, \dots, w_{n+u}$, 使得 $\delta(\sigma_n, \epsilon_n) = w_n, (M, w_{n+u}, \gamma) \models \varphi$ 以及对于所有 $0 \leq i \leq u-1$ 都有 $(w_{n+i}, w_{n+i+1}) \in \pi'$.

首先,对于满足上述条件的某个非负整数 u , 必然有 $u > m-n$. 否则将有 $\sigma_{n+u}, \epsilon_{n+u} \in path(X, \mathcal{B}')$, 进而有 $\sigma_{n+u}, \epsilon_{n+u} : \neg\varphi \in \mathcal{B}'$, 即 $(M, \delta(\sigma_{n+u}, \epsilon_{n+u}), \gamma) \models \neg\varphi$; 同时,由于 $M, \gamma, \delta \models \mathcal{B}'$, 必然存在 W 中的 $u+1$ 个可能世界 w_n, \dots, w_{n+u} 使得 $\delta(\sigma_n, \epsilon_n) = w_n, \dots, \delta(\sigma_{n+u}, \epsilon_{n+u}) = w_{n+u}$, 因而 $(M, \delta(\sigma_{n+u}, \epsilon_{n+u}), \gamma) \models \varphi$; 矛盾.

其次,如果 $u > m-n$, 则从可能世界 $\delta(\sigma_M, \epsilon_M)$ 出发经过一个 (P_{K+1}, E_{K+1}) -过程后到达可能世界 $\delta(\sigma_N, \epsilon_N)$, 再依次经过一个 π' -过程和 $u-(m-n)-2$ 个 π -过程后到达可能世界 w_{n+u} ; 即从可能世界 $\delta(\sigma_n, \epsilon_n)$ 出发经过 $u-(m-n)-1$ 个 π -过程后可以到达可能世界 w_{n+u} . 构建新的映射 δ' 如下: ① $\delta'(\sigma_n, \epsilon_n) := \delta(\sigma_n, \epsilon_n)$; ② 对于 \mathcal{B}' 中的任一前缀 σ, ϵ , 如果 $\sigma_n, \epsilon_n \leq_p \sigma, \epsilon$ 不成立, 则令 $\delta'(\sigma, \epsilon) := \delta(\sigma, \epsilon)$. 根据扩展规则以及引理 2, 必然存在某个饱和的并且可满足的分枝 \mathcal{B}'' , 使得 $M, \gamma, \delta' \models \mathcal{B}''$, 并且, 从可能世界 $\delta'(\sigma_n, \epsilon_n)$ 出发经过 $u-(m-n)-1$ 个 π -过程后到达可能世界 w_{n+u} . 由于 $u-(m-n)-1 < u$, 我们可以重复上述过程, 直到 $u \leq m-n$, 从而再次产生矛盾.

证毕.

最后,基于上述引理容易证明算法 1 的可靠性和完备性, 即有以下定理.

定理 2. 算法 1 返回“ φ 相对于 \mathcal{T} 和 \mathcal{A}_c 是可满足的”当且仅当 φ 相对于 \mathcal{T} 和 \mathcal{A}_c 是可满足的.

5 对判定算法的扩展

动态描述逻辑是描述逻辑的一类动态扩展. 对应于各个描述逻辑系统 (例如 ALCQO、ALCIO、ALCQIO 等), 可以分别引入动态维, 得到相应的动

态描述逻辑 (例如 D-ALCQO、D-ALCIO 以及 D-ALCQIO 等). 相应地, 可以对上文给出的判定算法进行扩展, 得到这些逻辑系统的 Tableau 判定算法.

对于上文的判定算法来说, 可以将其中的扩展规则分为两部分. 表 1 和表 2 中的为第一部分; 表 3 和表 4 中的为第二部分. 在不考虑个体变元以及 $\neg @-$ 和 $@-$ 规则的情况下, 第一部分的扩展规则与描述逻辑 ALCO 中关于角色、概念以及公式的构造符一一对应. 第二部分的扩展规则用于处理动态描述逻辑中的动作; 其中, 与描述逻辑 ALCO 中的各种构造符相关的, 仅仅是 B_2 -规则中对概念 $D^{Regress(\sigma, \epsilon)}$ 的递归构造过程. 因此, 为了得到 D-ALCQO、D-ALCIO 以及 D-ALCQIO 等的判定算法, 只需要在第一部分的扩展规则中增加对 Q 和 I 的处理, 同时在概念 $D^{Regress(\sigma, \epsilon)}$ 的递归构造过程中增加相应的处理.

例如, 动态描述逻辑 D-ALCQO 在 D-ALCO 的基础上增加了定性数量限定算子, 用于构造形如 $\leq nR.C$ 的概念. 即 D-ALCQO 中的概念由如下产生式生成:

$$C, D ::= C_i | \{u\} | \neg C | C \sqcup D | \forall R.C | \leq nR.C,$$

其中, $C_i \in N_C, u \in N_I \cup N_{IV}, R \in N_R, n$ 为非负整数.

D-ALCQO 的语义定义只需在定义 9 的基础上增加对概念 $\leq nR.C$ 的语义解释, 即

$$(16) (\leq nR.C)^{I(w), \gamma} = \{x | \# \{y | (x, y) \in R^{I(w), \gamma} \text{ 并且 } y \in C^{I(w), \gamma}\} \leq n\},$$

其中的“ $\#$ ”表示集合的秩.

为了判定 D-ALCQO 中公式的可满足性, 需要对 D-ALCO 的判定算法进行两处扩充.

首先, 在表 1 的基础上增加关于定性数量限定算子的扩展规则, 用于处理形如 $\neg(\leq nR.C)$ 和 $\leq nR.C$ 的概念, 如表 5 所示. 其中, 对于任一个体名或个体变元 u , 用 $[u]_{=B}$ 表示 u 在 B 中关于等价关系 “=” 的等价类, 即 $[u]_{=B} := \{v | u=v \in B_{\text{Ind}}^*\}$.

表 5 关于定性数量限定算子(Q)的扩展规则

规则名	规 则
$\rightarrow \leq$ -规则	如果 $\sigma_0, \epsilon_0 : (\neg(\leq nR.C))(x) \in B$, 并且不存在 $n+1$ 个个体名或个体变元 y_1, \dots, y_{n+1} , 使得 ① 对于任一 $1 \leq i < j \leq n+1$ 都有 $[y_i]_{=B} \neq [y_j]_{=B}$ 以及 ② 对于任一 $1 \leq k \leq n+1$ 都有 $\sigma_0, \epsilon_0 : R(x, y_k) \in B$ 和 $\sigma_0, \epsilon_0 : C(y_k) \in B$, 则引入 $n+1$ 个未在 B 中出现的个体名 y_1, \dots, y_{n+1} , 然后令 $B := B \cup \{\sigma_0, \epsilon_0 : R(x, y_k) 1 \leq k \leq n+1\} \cup \{\sigma_0, \epsilon_0 : C(y_k) 1 \leq k \leq n+1\} \cup \{y_i \neq y_j 1 \leq i < j \leq n+1\}$.
\leq -规则	如果 $\sigma_0, \epsilon_0 : (\leq nR.C)(x) \in B$, 并且至少存在 $n+1$ 个个体名或个体变元 y_1, \dots, y_{n+1} , 使得 ① 对于任一 $1 \leq i < j \leq n+1$ 都有 $[y_i]_{=B} \neq [y_j]_{=B}$ 以及 ② 对于任一 $1 \leq k \leq n+1$ 都有 $\sigma_0, \epsilon_0 : R(x, y_k) \in B$ 和 $\sigma_0, \epsilon_0 : C(y_k) \in B$, 则选择其中的某两个个体名或个体变元 y_i, y_j , 然后令 $B := B \cup \{y_i = y_j\}$.

其次,对于 B_2 -规则调用的对概念 $D^{Regress(\sigma,\epsilon)}$ 的递归构造过程,增加如下情况:

$$\textcircled{7} (\leq nR.C)^{Regress(\sigma,\epsilon)} :=$$

$$\begin{aligned} & ((\bigcap_{u \in Obj(\epsilon)} \neg \{u\}) \sqcap \leq nR.C^{Regress(\sigma,\epsilon)}) \sqcup \\ & \bigcup_{u \in Obj(\epsilon)} (\{u\} \sqcap (\bigcup_{n_1+n_2+n_3=n} (\leq n_1R.(\bigcap_{v \in Obj(\epsilon)} \neg \{v\} \sqcap \\ & C^{Regress(\sigma,\epsilon)}) \sqcap \leq n_2R.(\bigcup_{v \in Obj(\epsilon), R(u,v) \notin \epsilon, \neg R(u,v) \notin \epsilon} \{v\} \sqcap \\ & C^{Regress(\sigma,\epsilon)}) \sqcap \bigcap_{O \sqsubseteq \{v | R(u,v) \in \epsilon\}, \#O = n_3+1} (\bigcup_{v \in O} \neg @_v C^E))))), \end{aligned}$$

其中的 n_1, n_2, n_3 为某 3 个非负整数。

可以证明,上述扩展后的判定算法是可终止的、可靠的以及完备的。具体的证明过程只需对上一节的证明进行相应的扩展,这里不再详述。

再如,动态描述逻辑 D-ALCQIO 在 D-ALCQO 的基础上进一步引入了角色逆反算子,用于构造形如 R_i^- 的角色。即 D-ALCQIO 中的角色由如下产生式生成:

$$R ::= R_i | R_i^-,$$

其中, $R_i \in N_R$ 。

对于 D-ALCQIO 的语义定义来说,一方面需要在 D-ALCQO 语义定义的基础上增加对角色 R_i^- 的语义解释,即

$$(18) (R_i^-)^{I(w),r} = \{(x,y) | (y,x) \in R_i^{I(w),r}\}.$$

另一方面需要将定义 9 中对原子动作的语义解释(12)更改如下:

$$(12') \text{ 对于原子动作 } \alpha(v_1, \dots, v_n), \text{ 如果根据 } \mathcal{Ac}$$

可以确定其前提条件为 P ,所产生的影响为 E ;则

$$\alpha(v_1, \dots, v_n)^I = (P, E)^I = \{(\omega, \omega') | \text{存在某个基于 } M \text{ 的指派 } \gamma \text{ 使得: } \textcircled{1} \text{ 对于任一公式 } \varphi_i \in P \text{ 都有 } (M, \omega, \gamma) \models \varphi_i, \textcircled{2} \text{ 对于任一简单概念名 } C \text{ 都有 } C^{I(\omega'), \gamma} = (C^{I(\omega), \gamma} \cup \{u^{I, \gamma} | C(u) \in E\}) \setminus \{u^{I, \gamma} | \neg C(u) \in E\} \text{ 以及 } \textcircled{3} \text{ 对于任一角色名 } R \in N_R \text{ 都有 } R^{I(\omega'), \gamma} = (R^{I(\omega), \gamma} \cup \{(u^{I, \gamma}, v^{I, \gamma}) | R(u, v) \in E \text{ 或者 } R^-(v, u) \in E\}) \setminus \{(u^{I, \gamma}, v^{I, \gamma}) | \neg R(u, v) \in E \text{ 或者 } \neg R^-(v, u) \in E\}\}.$$

为了判定 D-ALCQIO 中公式的可满足性,可以对 D-ALCQO 的判定算法进行两个方面的扩展。

首先,需要扩展对前缀的定义,将角色的逆反形式考虑进去。为了表述方便,引入关于角色的函数 $Inv()$,使得对于任一 $R_i \in N_R$ 都有 $Inv(R_i) = R_i^-$ 以及 $Inv(R_i^-) = R_i$ 。另外,对于由简单公式组成的任一集合 E ,用 E^* 表示满足以下条件的最小集合: $\textcircled{1} E \subseteq E^*$; $\textcircled{2}$ 如果 $R(u, v) \in E^*$,则 $Inv(R)(v, u) \in E^*$; $\textcircled{3}$ 如果 $\neg R(u, v) \in E^*$,则 $\neg Inv(R)(v, u) \in E^*$ 。在此基础上,对关于前缀的定义 12 扩充如下。

定义 12'. 任一前缀 σ, ϵ 由一个顺序动作 σ 以及关于简单公式的集合 ϵ 构成,并且满足如下产生式规则:

$$\sigma, \epsilon ::= (\emptyset, \emptyset), \emptyset | \sigma; (P, E), (\epsilon \setminus (E^*)^-) \cup E^*.$$

其次,增加关于角色逆反算子的扩展规则,用于处理形如 R^- 的角色,如表 6 所示。

表 6 关于角色逆反算子(I)的扩展规则

规则名	规 则
$\neg r^-$ -规则	如果 $\sigma, \epsilon: \neg R^-(x, y) \in \mathcal{B}$, 并且 $\sigma, \epsilon: \neg R(y, x) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \epsilon: \neg R(y, x)\}$ 。
r^- -规则	如果 $\sigma, \epsilon: R^-(x, y) \in \mathcal{B}$, 并且 $\sigma, \epsilon: R(y, x) \notin \mathcal{B}$, 则令 $\mathcal{B} := \mathcal{B} \cup \{\sigma, \epsilon: R(y, x)\}$ 。

同样,类似于上一节的证明过程,可以证明上述扩展后的判定算法对于动态描述逻辑 D-ALCQIO 来说是可终止的、可靠的以及完备的。这里不再详述。

6 相关工作比较

与 Wolter 等^[3]关于动态描述逻辑的工作相比,本文的主要特点是进一步将任一原子动作 α 刻画为二元组 (P, E) 的形式,分别表示执行该动作之前必须满足的前提条件以及执行该动作后将会产生的影响;在采用可能模型途径对动作的语义进行严格定义之后,给出了动态描述逻辑的 Tableau 判定算法。

本文采用了文献[4]提出的构建动态描述逻辑的基本思路,将动作作为逻辑系统的主要成员之一,

与概念一起作为知识进行刻画和推理。本文对文献[4]的改进在于采用了可能模型途径来定义原子动作的语义;在此基础上,所给出的判定算法继承了描述逻辑 Tableau 算法的基本特征,在采用开世界假设的情况下进行判定。

文献[10]在描述逻辑 ALCO@的基础上构建了动态描述逻辑 $D_{ALCO@}$ 并给出其 Tableau 判定算法;在此基础上,该文研究了动作的可实现性、可执行性、投影以及规划等推理问题,将这些推理问题转换为动态描述逻辑中公式的可满足性问题。本文对文献[10]的改进在于 3 个方面。首先,在逻辑系统中引入了个体变元;允许个体变元出现在概念和公式的构造过程中,进而将个体变元应用于对原子动作定义式的刻画。其次,在证明判定算法可终止性的同

时,分析了影响算法复杂度的因素,证明了判定算法的复杂度上限.最后,研究了判定算法的可扩展性;对于 D-ALCQO、D-ALCQIO 等具有更强描述能力的逻辑系统,给出了相应的 Tableau 判定算法.

动态描述逻辑、时序描述逻辑以及模态描述逻辑等都可以看作对描述逻辑的二维扩展,它们分别在描述逻辑的基础上引入了动态维、时序维以及模态维,将描述逻辑语义解释中单一的可能世界扩展为由多个可能世界构成的可能世界空间.

Wolter 等^[11]提出了一类时序描述逻辑并证明了其可判定性.该逻辑在描述逻辑中引入了时序算子 \mathcal{U} 、 \mathcal{S} ,用于生成形如 $C\mathcal{U}D$ 、 $C\mathcal{S}D$ 的概念以及形如 $\varphi\mathcal{U}\psi$ 和 $\varphi\mathcal{S}\psi$ 的公式.之后, Lutz^[12] 和 Sturm^[13] 先后设计了该时序描述逻辑的 Tableau 判定算法,分别针对解释域不变的情况以及解释域可扩展的情况.从语义模型上看,时序描述逻辑中所有可能世界之间是一个线序关系;而对于动态描述逻辑来说,每一个动作都对应于可能世界之间的一种可达关系.因此,时序描述逻辑中的 Tableau 判定算法不能解决动态描述逻辑中的推理问题.

Baader 等^[14]较先对描述逻辑的模态扩展进行了研究,在描述逻辑中引入多组模态算子 \Box_i 和 \Diamond_i ,其中 $i=1,2,\dots,n$,用于构造形如 $\Box_i C$ 、 $\Diamond_i C$ 的概念以及形如 $\Box_i \varphi$ 、 $\Diamond_i \varphi$ 的公式. Wolter^[6]进行了较全面的总结,以 ALC 为例给出了模态描述逻辑 ALC_M 并证明了该逻辑系统的可判定性.之后, Lutz^[15] 针对更为简单的模态描述逻辑 K_{ALC} 给出了 Tableau 判定算法.如果将每组模态算子 \Box_i 、 \Diamond_i 作为一个原子动作 α_i ,则 Lutz 的工作相当于给出了一种受限形式下动态描述逻辑的推理算法,即只能含有原子形式的动作和以原子动作序列的形式出现的动作,不能处理通过动作构造算子 \mathcal{U} 和 $*$ 构造而得的复杂动作.

最后,本文的 B_2 -规则中对概念 $D^{Regress(\sigma,\epsilon)}$ 的递归构造过程参考了文献[9]给出的 ABox 更新算法.文献[9]研究了描述逻辑中 ABox 的更新问题.首先,其将作用于 ABox 的更新集 U 定义为由形如 $R(p,q)$ 、 $\neg R(p,q)$ 、 $C(p)$ 或者 $\neg C(p)$ 的简单公式组成的有限集合,其中的 p,q 为个体名, R 为角色, C 为未在 TBox 中进行定义的简单概念名.其次,对于描述逻辑中的两个语义解释 $I=(\Delta, \cdot^I)$ 和 $I'=(\Delta', \cdot^{I'})$,称 I' 是更新集 U 作用于 I 的结果,记为 $I \Rightarrow_U I'$,当且仅当 ① 对于任一简单概念名 C 都有 $C^{I'}=(C^I \cup \{p^I \mid C(p) \in U\}) \setminus \{p^I \mid \neg C(p) \in U\}$ 以及 ② 对于任一角色名 R 都有 $R^{I'}=(R^I \cup \{(p^I, q^I) \mid R(p,q) \in U\}) \setminus \{(p^I, q^I) \mid \neg R(p,q) \in U\}$. 在此基础

上,用 $M(T, \mathcal{A})$ 表示由满足 $I \models T$ 和 $I \models \mathcal{A}$ 的所有语义解释 I 组成的集合,则对 ABox 的更新定义如下:称 ABox \mathcal{A}' 是更新集 U 作用于 ABox \mathcal{A} 的结果,记为 $\mathcal{A} * U = \mathcal{A}'$,当且仅当 $M(T, \mathcal{A}')$ 与集合 $\{I' \mid I \in M(T, \mathcal{A}) \text{ 并且 } I \Rightarrow_U I'\}$ 相同.基于上述定义,文献[9]给出了计算 $\mathcal{A} * U$ 的算法.在该算法中,对于任一概念 C ,将递归地构造出在更新集 U 的作用下新得到的概念 C^U . 本文将该递归构造过程与定义 12 中引入的具有特殊结构的前缀结合起来,采用倒推理的形式,从与当前前缀 $\sigma.\epsilon$ 相关的概念 C 构造出与初始前缀 $\sigma_0.\epsilon_0$ 相关的概念 $C^{Regress(\sigma,\epsilon)}$.

7 结 论

动态描述逻辑 D-ALCO 将描述逻辑 ALCO、动态逻辑以及基于可能模型途径的动作理论有机地结合了起来;既可以基于描述逻辑刻画和推理静态的领域知识,又可以在这些知识的基础上将动作也作为一类知识进行刻画和推理.本文给出了 D-ALCO 的 Tableau 判定算法并证明了算法的可终止性、可靠性和完备性.应用该算法,可以在采用开世界假设的情况下对 D-ALCO 中任一公式的可满足性进行判定.该算法具有较好的可扩展性;对应于 D-ALCQO、D-ALCQIO 等具有更强描述能力的逻辑系统,可以对其扩展后得到相应的 Tableau 判定算法.本文的工作为动态描述逻辑的实际应用提供了可靠的算法基础.进一步的工作是对算法进行优化,开发相应的动态描述逻辑推理机.

致 谢 感谢审稿专家提出的宝贵意见!

参 考 文 献

- [1] Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P et al. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge: Cambridge University Press, 2002
- [2] Horrocks I, Patel-Schneider P F, Harmelen F V. From SHIQ and RDF to OWL: The making of a Web ontology language. Journal of Web Semantics, 2003, 1(1): 7-26
- [3] Wolter F, Zakharyashev M. Dynamic description logic//Zakharyashev M, Segerberg K, Rijke M, Wansing H eds. Advances in Modal Logic, Vol. 2. Stanford: CSLI Publications, 2000: 449-463
- [4] Shi Zhong-Zhi, Dong Ming-Kai, Jiang Yun-Cheng, Zhang Hai-Jun. A logical foundation for the semantic Web. Science in China, Series F, 2005, 48(2): 161-178

- [5] Winslett M. Reasoning about action using a possible models approach//Proceedings of the 7th National Conference on Artificial Intelligence. St. Paul, Minnesota, 1988; 89-93
- [6] Wolter F, Zakharyashev M. Satisfiability problem in description logics with modal operators//Cohn A, Schubert L, Shapiro S eds. Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning. San Francisco: Morgan Kaufmann Publishers, 1998; 512-523
- [7] Jiang Yun-Cheng, Wang Ju, Deng Pei-Min, Tang Yong. Semantics and reasoning of terminological cycles in description logic FL^- . Chinese Journal of Computers, 2008, 32(2): 185-195(in Chinese)
(蒋运承, 王驹, 邓培民, 汤庸. 描述逻辑 FL^- 循环术语集的语义及推理. 计算机学报, 2008, 32(2): 185-195)
- [8] Wang Ju, Jiang Yun-Cheng, Shen Yu-ming. Satisfiability and reasoning mechanism of terminological cycles in description logic vL . Science in China, Series F, 2008 (be addressed)(in Chinese)
(王驹, 蒋运承, 申宇铭. 描述逻辑系统 vL 循环术语集的可满足性及推理机制. 中国科学, F 辑, 2008, 待发表)
- [9] Liu H, Lutz C, Milicic M, Wolter F. Updating description logic ABoxes//Doherty P, Mylopoulos J, Welty C eds. Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning. Cambridge: AAAI Press, 2006; 46-56
- [10] Chang Liang, Lin Fen, Shi Zhongzhi. A dynamic description logic for representation and reasoning about actions//Zhang Z, Siekmann J eds. Proceedings of the 2nd International Conference on Knowledge Science, Engineering and Management. Berlin: Springer-Verlag, 2007; 115-127
- [11] Wolter F, Zakharyashev M. Temporalizing description logics//Gabbay D, Rijke M eds. Frontiers of Combining Systems II. Dordrecht : Kluwer Academic Publishers, 2000; 379-401
- [12] Lutz C, Sturm H, Wolter F, Zakharyashev M. Tableau calculus for temporal description logic: The constant domain case//Gore A, Leitsch A, Nipkow T eds. Proceedings of the 1st International Joint Conference on Automated Reasoning. Berlin: Springer-Verlag, 2001; 121-136
- [13] Sturm H, Wolter F. A tableau calculus for temporal description logic: The expanding domain case. Journal of Logic and Computation, 2002, 12(5): 809-838
- [14] Baader F, Laux A. Terminological logics with modal operators//Mellish C eds. Proceedings of the 14th International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 1995; 808-814
- [15] Lutz C, Sturm H, Wolter F, Zakharyashev M. A tableau decision algorithm for modalized ALC with constant domains. Studia Logica, 2002, 72(2): 199-232



CHANG Liang, born in 1980, Ph.D. candidate. His research interests include description logics, the Semantic Web, and intelligent Agent.

SHI Zhong-Zhi, born in 1941, professor and Ph.D. supervisor. His main research interests include artificial intelligence, machine learning and multi-agent system.

QIU Li-Rong, born in 1978, Ph.D. Her research interests include semantic Web services and multi-agent system.

LIN Fen, born in 1982, Ph.D. candidate. Her research interests include multi-agent system and Web services.

Background

The description logic (DL) is playing an important role in the Semantic Web, acting as the basis of the W3C- recommended Web ontology language OWL. The main strength of DL is that it offers considerable expressive power going far beyond propositional logic, while reasoning is still decidable. A limitation of the DL is that it can only describe and reason about knowledge of static application domains.

In order to represent and process the knowledge about semantic Web services, an obvious concern is to combine in some way the static description of ontologies on the Semantic Web with the dynamic description of computations provided by Web services. In the authors' previous work, by embracing actions into the DL, a dynamic description logic named DDL was proposed. The DDL provides an effective approach to combine the static description of ontologies with the dynamic

description of services. In this paper, the authors focus on the tableau decision algorithm for dynamic description logics. By taking the dynamic description logic D-ALCO as an example, a decision algorithm applicable to the Open World Assumption is proposed.

This research is supported by the National Natural Science Foundation of China under grant No.60775035, which title is the Logical Foundation for the Semantic Web Service. It is also partially supported by the National Natural Science Foundation of China (No.90604017), the National High Technology Research and Development Program (863 Program) of China (No.2007AA01Z132) and the National Basic Research Program (973 Program) of China (No.2007CB311004).