一种描述逻辑 SHIF 的 ABox 一致性判定算法*

彭 立, 杨恒伏
(湖南第一师范学院 信息科学与工程系, 长沙 410205)

摘 要: 为了判定 SHIF 的 ABox 一致性,提出了一种 Tableau 算法。该算法先通过预处理将 ABox 转换成标准形式,然后按照特定的完整策略将一套 Tableau 规则应用于 ABox,直到将它扩展成完整的 ABox 为止。ABox 与 TBox 一致,当且仅当算法能产生一个无冲突的完整的 ABox。算法所采用的阻塞机制可以避免 Tableau 规则的无限次执行。为了提高算法的效率,该机制允许一个新个体被在其之前创建的任意新个体直接阻塞,而不仅仅局限于其祖先。通过对算法的可终止性、合理性和完备性进行证明,算法的正确性得以确认。

关键词: 描述逻辑 SHIF; ABox 一致性判定; Tableau 算法; 阻塞机制; 正确性
中图分类号: TP301.2 **文献标志码**: A **文章编号**: 1001-3695(2013)02-0423-06
doi:10.3969/j.issn.1001-3695.2013.02.029

ABox consistency decision algorithm for description logic SHIF

PENG Li, YANG Heng-fu
(*Dept. of Information Science & Engineering, Hunan First Normal University, Changsha* 410205, *China*)

Abstract: To decide ABox consistency for SHIF, this paper presented a Tableau algorithm. The algorithm first converted ABox into a standard form by pre-disposal, and then applied a set of Tableau rules to ABox according to specific completion strategies until it extended to a complete ABox. ABox was consistent with regard to TBox if and only if the algorithm could yield a clash-free and complete ABox. It adopted a blocking mechanism by the algorithm could avoid infinite execution of Tableau rules. To raise the efficiency of the algorithm, the mechanism allowed a new individual to be directly blocked by any individual created before it, which was not restricted to its ancestor. Through proving the termination, soundness and completeness of the algorithm, confirmed its correctness.

Key words: description logic SHIF; ABox consistency decision; Tableau algorithm; blocking mechanism; correctness

描述逻辑(description logics, DLs)是一类用于知识表示和推理的逻辑,它广泛地应用于信息系统、数据库、软件工程和语义 Web 等领域[1,2]。经典的 DL 语言 ALC 可满足一般知识建模的需要,但在领域知识较为复杂的情况下,仍暴露出表达力不足的缺陷。为 ALC 增加传递角色[3]、角色层次[4]、反向角色[5]和功能限制这些新特征后,便形成了一种表达力较强的 DL 语言 $ALCHIF_{R+}$,简写为 SHIF[6],它能解决较为复杂的知识建模问题。DL 系统最基本的功能是保存经某种 DL 语言建模后的应用域知识。DL 系统的知识库包括 TBox 和 ABox 两部分。领域中概念和关系的描述以术语公理的形式存放于 TBox;领域中个体的描述以断言公理的形式存放于 ABox。此外,为了能在实际应用中充分发挥作用,DL 系统还应提供与 TBox 和 ABox 相关的推理服务。ABox 一致性判定即判断 ABox 是否与 TBox 一致,它是最基本的 DL 推理服务。要判定 ABox 的一致性,可采用两种方法:a)预完整方法[7,8],其思想是将 ABox 转换成预完整的 ABox,从而将 ABox 一致性判定归约为概念可满足性判定,该方法较难设计,目前尚未看到适用于 SHI[5]的预完整方法,更不用说 SHIF;b)设计专门的 Tableau 算法,通过 Tableau 算法判定 ABox 的一致性,这也是目前解决 ABox 一致性判定问题普遍采用的方法。

1 描述逻辑 SHIF

本章对 SHIF 的语法和语义进行了定义。事先规定:N_C 为概念名集合,N_R 为角色名集合,N_T 为传递角色名集合,N_P 为非传递角色名集合,O 为个体集合,$N_C \cap N_R = \varnothing$,$N_C \cap O = \varnothing$,$NR \cap O = \varnothing$,$N_R = N_T \cup N_p$,$N_T \cap N_P = \varnothing$。

定义 1 角色集: $= N_R \cup \{R^- \mid R \in NR\}$($R^-$ 为 R 的反向角色)。概念集为满足以下条件的最小集合:a)N_c 中的每一个元素、$\top$ 和 $\bot$ 都是概念;b)若 C 和 D 是概念,R 和 S 是角色且 S 为简单角色,则 $C \cap D$、$C \cup D$、$\neg C$、$\exists R.C$、$\forall R.C$、$\leqslant 1S$ 和 $\geqslant 2S$ 都是概念。

定义 2 形式为 $R \subseteq S$ 的公理是角色包含公理,其中 R 和 S 都是角色。形式为 $C \subseteq D$ 的公理是一般概念包含公理(general concept inclusion, GCI),其中 C 和 D 都是概念。TBox 是这两种术语公理的有限集合。

定义 3 让 R_I 为 TBox 中所有角色包含公理的集合,H: = $(R_I \cup \{\text{inv}(R) \subseteq \text{inv}(S) \mid R \subseteq S \in R_I\}, \subseteq^*)$ 被称为角色层次。其中 inv 函数被定义为:若 $R \in N_R$,则 $\text{inv}(R) := R^-$;若 $R = S^-$ 且 $S \in N_R$,则 $\text{inv}(R) := S$;$\subseteq^*$ 为 $R_I \cup \{\text{inv}(R) \subseteq \text{inv}(S) \mid R \subseteq S \in R_I$ 上 $\subseteq$ 的自反传递闭包。

收稿日期: 2012-07-08; **修回日期**: 2012-08-20 **基金项目**: 国家自然科学基金资助项目(61073191);湖南第一师范学院校级课题项目(XYS10N09)

作者简介: 彭立(1974-),男,湖南长沙人,讲师,硕士,主要研究方向为描述逻辑、语义 Web(goodbetter@163.com);杨恒伏(1973-),男,副教授,博士,主要研究方向为数字水印、语义 Web.

定义4 让 H 为角色层次。若 $R\subseteq^* S\in H$,则 S 为 R 的父角色,R 为 S 的子角色。角色 R 为传递角色,当且仅当 $R\in N_T$ 或 $\mathrm{inv}(R)\in N_T$。若角色 R 和其任意子角色都不是传递角色,则 R 为简单角色。

定义5 形式为 $a:C$ 的公理是概念断言公理,而形式为 $a\ R\ b$ 的公理是角色断言公理,形式为 $a\neq b$ 的公理是不等公理,其中 C 为概念,R 为角色,$a,b\in O$。ABox 是这三种断言公理的有限集合。

下面给出了 SHIF 的模型论语义。

定义6 一个解释 $I:=(\Delta^I,\cdot^I)$ 包含了一个非空集合 Δ^I(解释域)和一个解释函数 $\cdot^I$。解释函数将每一个概念映射为 Δ^I 的一个子集,将每一个角色映射为 $\Delta^I\times\Delta^I$ 的一个子集,将每一个个体映射为 Δ^I 中的一个元素。让 C、D 为概念,R 为角色,$P\in N_R$,$Q\in N_T$,#表示集合的势,解释函数应满足以下约束:

$\top^I=\Delta^I\quad\bot^I=\varnothing\quad(\neg C)^I=\Delta^I\backslash C^I$

$(C\cap D)^I=C^I\cap D^I\quad(C\cup D)^I=C^I\cup D^I$

$(\exists R.C)^I=\{x\in\Delta^I\mid$存在 $y\in\Delta^I$ 使得$(x,y)\in R^I$ 且 $y\in C^I\}$

$(\forall R.C)^I=\{x\in\Delta^I\mid$对于任意 $y\in\Delta^I$,若$(x,y)\in R^I$,则 $y\in C^I\}$

$(\leqslant 1R)^I=\{x\in\Delta^I\mid\#\{y\mid(x,y)\in R^I\}\leqslant 1\}$

$(\geqslant 2R)^I=\{x\in\Delta^I\mid\#\{y\mid(x,y)\in R^I\}\geqslant 2\}$

$(x,y)\in P^I$ 当且仅当$(y,x)\in(P^-)^I$

若$(x,y)\in Q^I$ 且$(y,z)\in Q^I$,则$(x,z)\in Q^I$

让 T 为 TBox,A 为 ABox,H 为角色层次,I 为解释。I 满足 H,当且仅当 I 满足所有的 $R\subseteq^* S\in H$,也就是使 $R^I\subseteq S^I$。I 满足 T,当且仅当:a)I 满足 H;b)I 满足所有的 $C\subseteq D\in T$(C 和 D 为概念),也就是使 $C^I\subseteq D^I$,此时 I 被称做 T 的模型。I 满足 A,当且仅当:a)I 满足所有的 $a:C\in A$,也就是使 $a^I\in C^I$;b)I 满足所有的 $a\ R\ b\in A$,也就是使$(a^I,b^I)\in R^I$;c)I 满足所有的 $a\neq b\in A$,也就是使 $a^I\neq b^I$,此时 I 被称做 A 的模型。A 与 H 一致,当且仅当存在 I 能同时满足 A 和 H;A 与 T 一致,当且仅当存在 I 能同时满足 T 和 A。

2 ABox 一致性判定算法

ABox 一致性判定是判断 ABox 是否与 TBox 一致。在文献[6]提出的 SHIF 概念可满足性判定算法的基础上,本文为 SHIF 设计了一种 ABox 一致性判定 Tableau 算法。该算法先通过预处理将 ABox 转换成标准形式,然后按照特定的完整策略将一套 Tableau 规则应用于 ABox,直到它扩展成完整的 ABox 为止。如果算法能产生一个无冲突的完整的 ABox,则 ABox 与 TBox 一致;否则不一致。

2.1 预处理

先给出与预处理有关的定义。

定义7[9] 形式为 $\forall x.x:C$ 的公理是通用概念公理,其中 C 为概念,$x\notin O\cup N_C\cup N_R$。解释 I 满足 $\forall x.x:C$,当且仅当 $C^I=\Delta^I$。

定义8 对于一个概念,若¬ 运算子只出现在它所包含的概念名的前面,则该概念具有否定标准形式。所有的概念都可以通过以下规则转换成否定标准形式:

$\neg\top=\bot\quad\neg\bot=\top\quad\neg(\neg C)=C$

$\neg(C\cap D)=\neg C\cup\neg D\quad\neg(C\cup D)=\neg C\cap\neg D$

$\neg(\exists R.C)=\forall R.\neg C\quad\neg(\forall R.C)=\exists R.\neg C$

$\neg(\leqslant 1R)=\geqslant 2R\quad\neg(\geqslant 2R)=\leqslant 1R$

给定 TBox T 和 ABox A,预处理按以下规则将 A 转换成标准形式:a)为每个 $C\subseteq D\in T$(C 和 D 为概念),在 A 中添加相应的通用概念公理 $\forall x.x:(\neg C\cup D)$;b)将 A 中出现的每一个概念转换成否定标准形式。

2.2 阻塞机制

预处理结束后,算法通过执行 Tableau 规则来对 ABox 进行扩展。为了防止 Tableau 规则被无限次执行,从而确保算法能终止,本文采用了一种新颖的阻塞机制,该机制是对 pair-wise blocking[10] 的改进,它类似于 anywhere blocking[11,12]。

定义9 让 A 为 ABox,a 和 b 为 A 中出现的个体。a 的概念集 $L(a):=\{C\mid a:C\in A\}$,a 和 b 的角色集 $L(a,b):=\{R\mid a\ R\ b\in A\}$。

定义10 让 A 为 ABox,H 为角色层次。若 $a\ R\ b\in A$,则 a 为 b 的前驱,b 为 a 的后继,a 和 b 互为邻居。若 $b\ S^-\ a\in A$ 且 $S\subseteq^* R\in H$,则 b 为 a 的 R 前驱;若 $a\ S\ b\in A$ 且 $S\subseteq^* R\in H$,则 b 为 a 的 R 后继;b 为 a 的 R 邻居,当且仅当 b 为 a 的 R 前驱或 R 后继。祖先关系为前驱关系的传递闭包,后代关系为后继关系的传递闭包。

定义11 ABox 中出现的个体分为老个体和新个体两类。老个体是算法执行之前 ABox 中已经存在的个体,新个体是算法在执行过程中创建的个体。

定义12 规定用"$<$"来表示 ABox A 中新个体的创建顺序。若算法在 A 中创建一个新个体 b,则对于 A 中所有已经存在的新个体 a,有 $a<b$。

定义13 阻塞。一个个体被阻塞,当且仅当它被直接阻塞或被间接阻塞。让 a、b、a'、b' 为 ABox 中存在的个体,a'、b' 分别为 a、b 的前驱。a 被 b 直接阻塞,当且仅当:a)a 和 b 均为新个体且 $b<a$;b)a 没有祖先被阻塞;c)b 未被阻塞;d)$L(a)=L(b)$,$L(a')=L(b')$,$L(a',a)=L(b',b)$。一个个体被间接阻塞,当且仅当它的前驱被阻塞。

与 pair-wise blocking 不同的是,该阻塞机制允许一个新个体被在其之前创建的任意新个体直接阻塞,而不仅仅局限于其祖先,这无疑可以减少 Tableau 规则的应用次数,从而提高算法的效率。

2.3 Tableau 规则

本文定义 Tableau 规则。让 A 为 ABox,A' 为 A 扩展成的 ABox,H 为角色层次。

定义14 Tableau 规则。

$\cap$规则

条件:a)$a:C\cap D\in A$,a 未被间接阻塞;b)$\{a:C,\ a:D\}\not\subseteq A$

动作:$A'=A\cup\{a:C,\ a:D\}$

$\cup$规则

条件:a)$a:C\cup D\in A$,a 未被间接阻塞;b)$\{a:C,\ a:D\}\cap A=\varnothing$

动作:$A'=A\cup\{a:C\}$或$A'=A\cup\{a:D\}$

$\exists$规则

条件:a)$a:\exists R.C\in A$,a 未被阻塞;b)a 没有 R 邻居 b 使得 $b:C\in A$

动作:$A'=A\cup\{a\ R\ b,\ b:C\}$($b$ 为 A 中不存在的个体)

$\forall$规则

条件:a)$a:\forall R.C\in A$,a 未被间接阻塞;b)a 有一个 R 邻居 b 使得 $b:C\notin A$

动作:$A'=A\cup\{b:C\}$

$\forall_T$规则

条件:a)$a:\forall R.C\in A$,a 未被间接阻塞;b)存在一个传递

角色 S 使得 $S\subseteq^{*}R\in H$;c)a 有一个 S 邻居 b 使得 $b:\forall S.C\notin A$

动作:$A'=A\cup\{b:\forall S.C\}$

$\forall_x$ 规则

条件:a)$\forall x.x:C\in A$;b)A 中存在一个未被间接阻塞的个体 a 使得 $a:C\notin A$;c)A 中不存在另一个体 b 使得 $a\doteq b\in A$

动作:$A'=A\cup\{a:C\}$

$\geqslant 2R$ 规则

条件:a)$a:\geqslant 2R\in A$,a 未被阻塞;b)a 的 R 邻居数 <2

动作:$A'=A\cup\{a\ R\ b_1,\ a\ R\ b_2\}\cup\{b_1\neq b_2,\ b_2\neq b_1\}$($b_1$ 和 b_2 为 A 中不存在的个体)

$\leqslant 1R$ 规则

条件:a)$a:\leqslant 1R\in A$,a 未被间接阻塞;b)a 有两个 R 邻居 b_1 和 b_2(其中 b_1 是新个体且为 a 的后继)使得 $b_1\neq b_2\notin A$

动作:a)若 b_2 为 b_1 的祖先,则将 A 中每一条包含了 a 和 b_1 的角色断言公理 $a\ S\ b_1$ 转换成 $b_1\ \mathrm{inv}(S)\ a$;b)$A'=A[b_1/b_2]\cup\{b_1\doteq b_2\}$($A[b_1/b_2]$表示在 A 中用 b_2 替代 b_1)

$\leqslant o1R$ 规则

条件:a)$a:\leqslant 1R\in A$,a 未被间接阻塞;b)a 有两个 R 邻居 b_1 和 b_2(b_1 和 b_2 均为老个体)使得 $b_1\neq b_2\notin A$

动作:$A'=A[b_1/b_2]\cup\{b_1\doteq b_2\}$($A[b_1/b_2]$表示在 A 中用 b_2 替代 b_1)

$\cap$和$\cup$规则能对个体 a 的概念集 $L(a)$进行扩展。$\exists$规则可为 a 产生新的 R 邻居。$\forall$和$\forall_T$规则能将 a 的概念集 $L(a)$中的某个概念或相关概念传播到其 R 邻居。$\forall x$ 规则用于对通用概念公理进行处理。$\geqslant 2R$ 规则能为 a 产生两个不等的 R 后继。

$\leqslant 1R$ 规则能对 a 的两个 R 邻居进行合并,即用其中的一个替代另一个,并要求两者中至少有一个是新个体。它不允许老个体被新个体替代,从而确保了新个体不会成为老个体的祖先。此外,该规则的动作 1 避免了"a 的 R 后继 b_1 被 R 前驱 b_2 替代后,b_2 同时为 a 的前驱和后继"这种情况的出现,从而确保了所有新个体都只有一个前驱。$\leqslant_o 1R$ 规则也能对 a 的两个 R 邻居进行合并,但要求两者都是老个体。之所以让$\leqslant 1R$ 和$\leqslant_o 1R$ 规则在 ABox 中添加 $b_1\doteq b_2$ 形式的公理,是为了证明定理 1。

当 a 被阻塞时,$\exists$和$\geqslant 2R$ 规则不能再应用于 a,从而确保算法能终止。当 a 被间接阻塞时,$\cap$、$\cup$、$\forall$、$\forall_T$、$\forall_x$、$\leqslant 1R$ 和$\leqslant_o 1R$规则不能再应用于 a,从而避免了不必要的规则应用,算法也得以优化。在这些规则中,$\cap$、$\exists$、$\forall$、$\forall_T$、$\forall_x$ 和$\geqslant 2R$ 规则是确定规则,$\cup$、$\leqslant 1R$ 和$\leqslant_o 1R$ 规则是非确定规则,因为前者产生的结果是确定的,而后者产生的结果是不确定的。因为非确定规则的存在,算法所产生的完整的 ABox 也是不确定的,也就是说该算法是非确定性算法。

2.4　完整策略

完整策略规定了 Tableau 规则的执行次序,从而确保算法能产生完整的 ABox。

定义 15　当 ABox 中出现了冲突或无 Tableau 规则可应用时,ABox 就成为了完整的 ABox,此时算法停止执行。

定义 16　ABox A 中出现了冲突,是指:

a)$a:\bot\in A$,或

b)$\{a:C,\ a:\neg C\}\in A$(其中 $C\in N_c$),或

c)$a:\leqslant 1R\in A$,且 a 有两个 R 邻居 b_1 和 b_2 使得 $b_1\neq b_2\in A$。

算法按以下策略将 ABox 扩展成完整的 ABox。

策略 1　按以下步骤对 ABox 中的个体进行处理:

a)将 ABox 中所有的老个体都标记为"未处理"。

b)按任意顺序将所有"未处理"的老个体处理一遍。

c)按"$<$"顺序将所有"未处理"的新个体处理一遍。

d)重复步骤 b)c),当所有的个体都被标记为"已处理"时,算法停止执行。

策略 2　在对某个个体进行处理时,按以下步骤将 Tableau 规则应用于该个体:

a)反复应用$\cap$、$\cup$、和$\forall_x$ 规则,直到它们无法应用或 ABox 中出现冲突为止。若 ABox 中出现了冲突,则算法停止执行。

b)反复应用$\exists$和$\geqslant 2R$ 规则,并将规则产生的新个体标记为"未处理",直到它们无法应用为止。

c)反复应用$\forall$、$\forall_T$、$\leqslant 1R$ 和$\leqslant_o 1R$ 规则,并将规则所影响的其他个体标记为"未处理",直到它们无法应用或 ABox 中出现冲突为止。若 ABox 中出现了冲突,则算法停止执行。

d)将该个体标记为"已处理",并开始对下一个个体进行处理。

2.5　算法正确性的证明

给定 TBox 和 ABox,算法能产生一个不确定的完整的 ABox。若能证明"算法是可以终止的,而且 ABox 与 TBox 一致,当且仅当算法能产生一个无冲突的完整的 ABox",则说明可通过该算法来判定 ABox 的一致性。

引理 1　让 A 为 ABox,A' 为 A 的标准形式,T 为 TBox,H 为角色层次。A 与 T 一致,当且仅当 A'与 H 一致。

证明　将 A 转换成 A',也就是为每一个 $C\subseteq D\in T$(C 和 D 为概念)在 A 中添加相应的公理 $\forall x.x:(\neg C\cup D)$,并且将 A 中出现的所有概念都转换成否定标准形式。显然,一个解释能满足 $C\subseteq D$ 当且仅当它满足 $\forall x.x:(\neg C\cup D)$;此外,对于任意概念和其否定标准形式,两者的解释必然相同。因此,若存在一解释能同时满足 A 和 T,则它能同时满足 A'和 H;反之亦然。引理得证。

定义 17　对于 ABox A 中存在的任意个体 a,若 A 中存在另一个体 b 使得 $a\doteq b\in A$,则 a 被称为被替代的个体;否则被称为未被替代的个体。

定理 1　合理性。若算法能产生一个无冲突的完整的 ABox,则 ABox 与 TBox 一致。

证明　让 T 为 TBox,A 为 ABox,H 为角色层次,A'为 A 的标准形式,A''为 A'扩展成的无冲突的完整的 ABox。从 A''中去除那些包含了被间接阻塞个体的公理,可得到一新的 ABox A^{*},显然 A^{*}也是无冲突的完整的 ABox。本文先为 A^{*}和 H 定义一个解释 I。考虑到 SHIF 不具有有限模型属性,因此当 A^{*}中存在阻塞时,I 应被定义成无限模型,为此引入了路径[13]的概念。一条路径 p 是 A^{*}中个体的一个序列,其形式为 $p=[(a_0,a_0'),\cdots,(a_n,a_n')]$。对于路径 p,定义 $\mathrm{tail}(p):=a_n$ 且 $\mathrm{tail}'(p):=a_n'$。用$[p\mid(a_{n+1},a_{n+1}')]$来表示路径$[(a_0,a_0'),\cdots,(a_n,a_n'),(a_{n+1},a_{n+1}')]$。$A^{*}$的路径集 $\mathrm{path}(A^{*})$被递归地定义如下:

a)对于 A^{*}中未被替代的老个体 a_i,$[(a_i,a_i)]\in\mathrm{path}(A^{*})$。

b)对路径 $p\in\mathrm{path}(A^{*})$和 A^{*}中未被替代的新个体 a、b:

(a)如果 a 是 $\mathrm{tail}(p)$的后继,且它没有被阻塞,则$[p\mid(a,a)]\in\mathrm{path}(A^{*})$,或者

(b)如果 a 是 $\mathrm{tail}(p)$的后继,且 a 被 b 直接阻塞,则$[p\mid(b,a)]\in\mathrm{path}(A^{*})$。

根据以上递归定义,可得出 $\mathrm{path}(A^{*})$的一些属性:

a)老个体不能对路径进行扩展,只能出现在路径起始处。

b)对于任意 $p \in \text{path}(A^*)$,tail(p)没有被阻塞。

c)对于任意 $p \in \text{path}(A^*)$,tail(p) = tail'(p)当且仅当tail'(p)没有被阻塞。

d)对于任意 $p \in \text{path}(A^*)$,$L(\text{tail}(p)) = L(\text{tail}'(p))$。

给定一条路径$[(a_0,a_0'),\cdots,(a_n,a_n')]$,若 $n=0$ 或当 $n>0$ 时,对于 $0 \leqslant i \leqslant n-1$ 有 $a_i = a_i'$,则该路径被称为简单路径。

下面为 A^* 和 H 定义一个解释 I:

a)$\Delta^I = \text{path}(A^*)$。

b)对于 A^* 中的任意个体 a,若 a 是未被替代的个体,则 $a^I = p(p \in \Delta^I$为简单路径且 tail'(p) = $a)$;若 a 是被替代的个体,则 $a^I = b^I(a \doteq b \in A^*)$。

c)对于 A^* 中的任意概念名 C,$C^I = \{p \in \Delta^I \mid C \in L(\text{tail}'(p))\}$。

d)对于 A^* 和 H 中的任意角色 R,让 $\varepsilon(R) = \{(p,[p'|(a,a')]) \in \Delta^I \times \Delta^I \mid a'$是 tail($p$)的 R 后继$\} \cup \{([q|(a,a')],q) \in \Delta^I \times \Delta^I \mid \text{tail}(q)$是的 a' 的 R 前驱$\} \cup \{([(a^i,a^i)],[(a^j,a^j)]) \in \Delta^I \times \Delta^I \mid a^i$ 和 a^j 为老个体且 a^j 是 a^i 的 R 邻居$\}$。

若 R 为传递角色,则 $R^I = \varepsilon(R)^+$,否则 $R^I = \varepsilon(R) \cup \bigcup_{P \subseteq {}^* R \in H, P=R} P^I(\varepsilon(R)^+$表示 $\varepsilon(R)$的传递闭包)。

接下来证明 I 满足 A^* 中的每一条断言公理。

对于任意 $a \neq b \in A^*$,因为 tail'(a^I) = a,tail'(b^I) = b,而且 Tableau 规则确保了 $a \neq b$,所以 $a^I \neq b^I$。对于任意 $a\ R\ b \in A^*$,因为 a 未被阻塞,所以 tail(a^I) = tail'(a^I) = a。b 可能是老个体,也可能是新个体。若 b 为老个体,则 a 必然也为老个体,而且 $b^I = [(b,b)]$,$a^I = [(a,a)]$;若 b 为新个体,则 $b^I = [a^I|(c,b)]$。因为 b 是 a 的 R 后继,所以无论是哪种情况,都有$(a^I,b^I) \in R^I$。

现在考虑 $a:C$ 形式的公理。只需证明这样一个命题:对于 $p \in \Delta^I$ 和概念 C,若 $C \in L(\text{tail}(p))$,则 $p \in C^I$。若该命题成立,则对于任意 $a:C \in A^*$,因为 $C \in L(a) = L(\text{tail}'(a^I)) = L(\text{tail}(a^I))$,所以 $a^I \in C^I$。下面通过对概念 C 的结构进行归纳来证明 $p \in C^I$。

a)若 $C \in N_c$,因为 $C \in L(\text{tail}(p)) = L(\text{tail}'(p))$,所以 $p \in C^I$。若 $C = \top$,显然 $p \in \Delta^I = \top^I$。$C = \bot$ 这种情况不可能出现,因为 A^* 是无冲突的。

b)若 $C = \neg D$,则 $D \in N_c$,因为 A^* 中所有的概念都具有否定标准形式。A^* 是无冲突的,所以 $D \notin L(\text{tail}(p)) = L(\text{tail}'(p))$,$p \notin D^I$,$p \in (\neg D)^I$。

c)若 $C = C_1 \cap C_2$,因为 A^* 是完整的,所以 $C_1 \in L(\text{tail}(p))$ 且 $C_2 \in L(\text{tail}(p))$。根据归纳假设,有 $p \in C_1^I$ 且 $p \in C_2^I$,所以 $p \in (C_1 \cap C_2)^I$。

d)若 $C = C_1 \cup C_2$,因为 A^* 是完整的,所以 $C_1 \in L(\text{tail}(p))$ 或 $C_2 \in L(\text{tail}(p))$。根据归纳假设,有 $p \in C_1^I$ 或 $p \in C_2^I$,所以 $p \in (C_1 \cup C_2)^I$。

e)若 $C = \forall R.D$,则只要证明:对于任意 $q \in \Delta^I$,若$(p,q) \in R^I$,有 $q \in D^I$。要使$(p,q) \in R^I$,只有以下两种可能:

(a)$(p,q) \in \varepsilon(R)$。p 和 q 的关系存在三种可能:

①$q = [p|(\text{tail}(q),\text{tail}'(q))]$,tail'($q$)是 tail($p$)的 R 后继。因为 $\forall R.D \in L(\text{tail}(p))$ 且 A^* 是完整的,所以 $D \in L(\text{tail}'(q)) = L(\text{tail}(q))$。

②$p = [q|(\text{tail}(p),\text{tail}'(p))]$,tail($q$)是 tail'($p$)的 R 前驱。因为 $\forall R.D \in L(\text{tail}(p)) = L(\text{tail}'(p))$ 且 A^* 是完整的,所以 $D \in L(\text{tail}(q))$。

③$p = [(\text{tail}(p),\text{tail}(p))]$,$q = [(\text{tail}(q),\text{tail}(q))]$,tail($p$)和 tail($q$)均为老个体,且 tail($q$)是 tail($p$)的 R 邻居。因为 $\forall R.D \in L(\text{tail}(p))$ 且 A^* 是完整的,所以 $D \in L(\text{tail}(q))$。

因此,若$(p,q) \in \varepsilon(R)$,总有 $D \in L(\text{tail}(q))$。根据归纳假设,有 $q \in D^I$。

(b)$\{(p,p_1),(p_1,p_2),\cdots,(p_n,q)\} \subseteq \varepsilon(S)$,$S$ 为传递角色,且 $S \subseteq {}^* R \in H$。p 和 p_1 的关系同样存在三种可能,类似地可证明 $\forall S.D \in L(\text{tail}(p_1))$,同理可得 $\forall S.D \in L(\text{tail}(p_i))$ $(2 \leqslant i \leqslant n)$。既然 $\forall S.D \in L(\text{tail}(p_n))$ 且$(p_n,q) \in \varepsilon(S)$,那么根据上述证明可知 $q \in D^I$。

f)若 $C = \exists R.D$,则只要证明:存在 $q \in \Delta^I$ 使得$(p,q) \in R^I$ 且 $q \in D^I$。让 $b = \text{tail}(p)$,$b' = \text{tail}'(p)$。因为 A^* 是完整的且 b 未被阻塞,所以 b 必然有一 R 邻居 c 使得 $D \in L(c)$。b 和 c 的关系有两种可能:

(a)c 是 b 的 R 后继。若 c 是老个体,则存在 $q \in \Delta^I$ 使得 $q = [(c,c)]$,而且 b 必然也是老个体,$p = [(b,b)]$;若 c 为新个体,则存在 $q \in \Delta^I$ 满足 $q = [p|(d,c)]$。

(b)c 是 b 的 R 前驱。

①若 b 为老个体,则 $p = [(b,b)]$,而且存在 $q \in \Delta^I$ 使得 $q = [(c,c)]$。

②若 b 为新个体且 $b = b'$,则存在 $q \in \Delta^I$ 使得 tail(q) = c 且 $p = [q|(b,b)]$。

③若 b 为新个体且 $b \neq b'$,则存在 $q \in \Delta^I$ 使得 tail(q) $\neq c$ 且 $p = [q|(b,b')]$。因为 b 直接阻塞了 b',根据阻塞的条件可知 $L(c,b) = L(\text{tail}(q),b')$ 且 $L(c) = L(\text{tail}(q))$,显然 tail(q)是 b' 的 R 前驱且 $D \in L(\text{tail}(q))$。

因此不管是什么情况,始终存在 $q \in \Delta^I$ 使得$(p,q) \in R^I$ 且 $D \in L(\text{tail}(q))$。根据归纳假设,有 $q \in D^I$。

g)若 $C = \geqslant 2R$,则只要证明:存在 $q_1,q_2 \in \Delta^I (q_1 \neq q_2)$ 使得$(p,q_1) \in R^I$ 且$(p,q_2) \in R^I$。让 $b = \text{tail}(p)$,$b' = \text{tail}'(p)$。因为 A^* 是完整的且 b 未被阻塞,所以 b 至少有两个 R 邻居 c_1 和 c_2。关于 b 有两种可能:

(a)b 为老个体。若 $c_i(i=1$ 或 $2)$为老个体,则存在 $q_i \in \Delta^I$ 使得 $q_i = [c_i,c_i]$且$(p,q_i) \in R^I$;若 c_i 为新个体,则存在 $q_i \in \Delta^I$ 使得 $q_i = [p|(d_i,c_i)]$且$(p,q_i) \in R^I$。不难看出,若 $c_1 \neq c_2$,则 $q_1 \neq q_2$。

(b)b 为新个体。

①若 c_1 和 c_2 都是 b 的 R 后继,则对于每一个 c_i,都存在 $q_i \in \Delta^I$使得 $q_i = [p|(d_i,c_i)]$且$(p,q_i) \in R^I$。不难看出,若 $c_1 \neq c_2$,则 $q_1 \neq q_2$。

②若 c_1 和 c_2 中有一个为 b 的 R 前驱,不妨假设它为 c_1,则 c_2 必为 b 的 R 后继。对于 c_1,根据"$C = \exists R.D$"的证明可知,存在 $q_1 \in \Delta^I$ 使得$(p,q_1) \in R^I$,而且"$q_1 = [(c_1,c_1)]$或 tail(q_1) = c_1,$p = [q_1|(b,b)]$或 tail(q_1) $\neq c_1$,$p = [q_1|(b,b')]$";对于 c_2,必然存在 $q_2 \in \Delta^I$ 使得 $q_2 = [p|(d,c_2)]$且$(p,q_2) \in R^I$。显然 $q_1 \neq q_2$。

h)若 $C = \leqslant 1R$,则只要证明:最多只存在一个 $q \in \Delta^I$ 使得$(p,q) \in R^I$。需要注意的是:在形式为 $\leqslant 1R$ 的概念中,R 必须是简单角色,因此 $R^I = \varepsilon(R)$。假设存在 $q_1,q_2 \in \Delta^I (q_1 \neq q_2)$ 使得$(p,q_1) \in R^I$ 且$(p,q_2) \in R^I$,现证明这是不可能的。让 $b = \text{tail}(p)$,$b' = \text{tail}'(p)$。关于 b 有两种可能:

(a)b 为老个体。$q_i(i=1$ 或 $2)$的形式只可能是$[c_i,c_i]$或

$[p|(d_i,c_i)]$。若 $q_1=[(c_1,c_1)]$,$q_2=[(c_2,c_2)]$,因为 $q_1\neq q_2$,所以 $c_1\neq c_2$。若 $q_1=[(c_1,c_1)]$,$q_2=[p|(d_2,c_2)]$ 或 $q_2=[(c_2,c_2)]$,$q_1=[p|(d_1,c_1)]$,显然 $c_1\neq c_2$,因为老个体只能出现在路径的开始处。若 $q_1=[p|(d_1,c_1)]$,$q_2=[p|(d_2,c_2)]$,也可得出 $c_1\neq c_2$,因为一旦 $c_1=c_2$,若 c_1 未被阻塞,则 $d_1=c_1=c_2=d_2$,这意味着 $q_1=q_2$;若 c_1 被阻塞,则 $d_1=d_2$,这也意味着 $q_1=q_2$。因此不管是哪种情况,都说明 b 有两个 R 邻居 c_1 和 c_2,这意味着 A^* 是不完整的,从而产生矛盾。

(b)b 为新个体。考虑两种可能:

①$b=b'$。若 $p\neq[q_i|(b,b)]$($i=1$ 或 2),则 q_i 的形式只能是 $[p|(d_i,c_i)]$,由上面的证明可知这是不可能的。因此,必然有且只有一个 $q_i\in\{q_1,q_2\}$ 使得 $p=[q_i|(b,b)]$,且 $\text{tail}(q_i)$ 为 b 的 R 前驱,不妨假定它为 q_1。q_2 的形式只能是 $[p|(d,c)]$,且 c 为 b 的 R 后继。显然 $\text{tail}(q_1)\neq c$,同样会产生矛盾。

②$b\neq b'$。同样可知必然有一个且只有一个 $q_i\in\{q_1,q_2\}$ 使得 $p=[q_i|(b,b')]$,而且 $\text{tail}(q_i)$ 是 b' 的 R 前驱,不妨假定它为 q_1。因为 b' 被 b 阻塞,所以 b 必有一前驱 c_1 使得 $L(c_1,b)=L(\text{Tail}(q_1),b')$,$c_1$ 显然是 b 的 R 前驱。q_2 的形式只能为 $[p|(d,c_2)]$,而且 c_2 为 b 的 R 后继。显然 $c_1\neq c_2$,所以也会产生矛盾。

对于任意 $\forall x.\,x:C\in A^*$,$\forall_x$ 规则可以确保:对于 A^* 中所有未被替代的个体 a,有 $a:C\in A^*$。对于任意 $p\in\Delta^I$,A^* 中必然存在一个未被替代的个体 b 使得 $\text{tail}'(p)=b$。因为 $C\in L(b)=L(\text{tail}'(p))=L(\text{tail}(p))$,根据之前的证明可知 $p\in C^I$,这意味着 $C^I=\Delta^I$。

至此已证明了 I 满足 A^*,现证明 I 满足 A'。Tableau 规则不会删除公理,只可能将公理中包含的某些个体用其他个体来替代,因此对于 A' 中的任意公理 Axm,在 A^* 中都有一条对应的公理 Axm'。对于 Axm 中包含的任意个体 a,若它成为了被替代的个体,则在 Axm' 中它必然被另一个体 a' 替代;否则它必然存在于 Axm' 中。因为 $a^I=a'^I$,而且 I 满足 Axm',所以 I 也满足 Axm。

现在证明 I 满足 H。对于任意 $R\sqsubseteq^* S\in H$,若 $(p,q)\in R^I$,则存在两种可能:

a)$(p,q)\in\varepsilon(R)$。因为 $R\sqsubseteq^* S$,所以 $(p,q)\in\varepsilon(S)\subseteq S^I$。

b)$(p,p_1),(p_1,p_2),\cdots,(p_n,q)\}\subseteq\varepsilon(Q)$,$Q$ 为传递角色,且 $Q\sqsubseteq^* R\in H$。显然 $\{(p,p_1),(p_1,p_2),\cdots,(p_n,q)\}\subseteq\varepsilon(R)\subseteq\varepsilon(S)$,$(p,q)\in Q^I$。因此无论 S 是不是传递角色,都有 $(p,q)\in S^I$。

因为 I 满足 A' 和 H,根据引理 1 可知定理 1 成立。

定义 18　让 C 为具有否定标准形式的概念,A 为 ABox,H 为角色层次。$\text{sub}(C,H)$ 是满足以下条件的最小概念集:a)$C\in\text{sub}(C,H)$;b)若 $D\in\text{sub}(C,H)$,则 D 的所有子概念也属于 $\text{sub}(C,H)$;c)若 $\forall R.\,D\in\text{sub}(C,H)$,$S\sqsubseteq^* R\in H$,且 S 为传递角色,则 $\forall S.\,D\in\text{sub}(C,H)$。集合 $\text{sub}(A,H):=\bigcup_{a:C\in A}\text{sub}(C,H)$。

定理 2　可终止性。给定 TBox 和 ABox,算法可以终止。

证明　让 T 为 TBox,A 为 ABox,n_T 为 T 的大小,n_A 为 A 的大小。算法可分为两个阶段:

a)预处理阶段。将 A 转换成标准形式,也就是将 T 中所有的 GCIs 转换成 A 中相应的广义概念公理,而且将出现在 A 中的所有概念转换成否定标准形式。转换操作的次数显然受 n_T+n_A 的约束,因此 a)阶段是可以终止的。

b)Tableau 规则执行阶段。按完整策略执行 Tableau 规则,直到产生一个完整的 ABox 为止。该阶段是否能终止,取决于 Tableau 规则的执行次数是否有限。算法每执行一次 Tableau 规则,都将产生新的公理($\leqslant 1R$ 和 $\leqslant_o 1R$ 规则能产生形式为 $C\doteq D$的公理)。因此,若能证明算法所产生的任意一个完整的 ABox 中只包含有限个公理,则意味着 b)阶段是可以终止的。

让 A' 为 A 的标准形式,则 A' 的大小 $n_{A'}=O(n_A+n_T)$。让 H 为角色层次,则 H 的大小 $n_H=O(n_T^2)$。对于任意概念 C(假定其大小为 n_C),文献[14]已经证明 $\#\text{sub}(C,H)=O(n_C\times n_H)$(#表示集合的势),所以 $\#\text{sub}(A',H)=O(n_A'\times n_H)$。让 $R_{A',H}$ 为 A' 和 H 中所有角色及其反向角色的集合,则 $\#R_{A',H}=O(n_{A'}+n_H)$。让 $m=\#\text{sub}(A',H)$,$k=\#R_{A',H}$,$n=n_A+n_T$,则 $m=O(n^3)$,$k=O(n^2)$,$n_{A'}=O(n)$。

让 A_r 为 A' 扩展而成的任意 ABox。可以认为 A_r 中的任意老个体及其所有的新个体后代构成了一棵树:该老个体是树的根节点,其新个体后代为根节点的子孙节点;若个体 a 是个体 b 的前驱,则 a 为 b 的父节点。对于树中任意一对父子节点 a 和 b,$L(a)$ 和 $L(b)$ 都是 $\text{sub}(A',H)$ 的子集,$L(a,b)$ 是 $R_{A',H}$ 的子集,显然 $(L(a),L(a,b),L(b))$ 的取值最多只有 2^{2m+k} 种可能。这意味着当树中任意路径的长度达到 $2^{2m+k}+1$ 时,该路径中必然有两对父子节点 (a',a) 和 (b',b)(其中 b 是 a 的祖先)满足 $(L(a'),L(a',a),L(a))=(L(b'),L(b',b),L(b))$。假定 a 未被阻塞,那么 a 所有的祖先都不可能被阻塞,根据阻塞的定义可知 a 必定被 b 直接阻塞,从而产生矛盾,因此 a 必然被阻塞,路径的长度不可能再增加,这说明树的深度 $\leqslant 2^{2m+k}+2$。对于树中的任意节点 a,A_r 中形式为 $a:\exists R.\,C$ 或 $a:\geqslant 2R$ 的公理最多有 m 条,而且对于其中任意一条公理,$\exists$ 或 $\geqslant 2R$ 规则最多能对它应用一次,因此 a 的后继最多有 $2m$ 个,a 的出度 $\leqslant 2m$,树中节点的数量 $\leqslant(2m)^{2+2^{2m+k}}$。$A_r$ 中最多有 $n_{A'}$ 个老个体,所以树的数量 $\leqslant n_{A'}$,A_r 中个体的数量 $\leqslant n_{A'}\times(2m)^{2+2^{2m+k}}=O(n^{2^{n^3}})$。

显然,A_r 中形式为 $a:C$ 的公理的数量 $\leqslant m\times$ 个体数 $=O(n^{2^{n^3}})$;形式为 $a\ R\ b$ 的公理的数量 $\leqslant k\times$ 个体数$^2=O(n^{2^{n^3}})$;形式为 $a\neq b$ 的公理的数量 $\leqslant$ 个体数$^2=O(n^{2^{n^3}})$;形式为 $a\doteq b$ 的公理的数量 $\leqslant$ 个体数$^2=O(n^{2^{n^3}})$。既然 A_r 中包含的公理为有限多个,那么算法所产生的任意一个完整的 ABox 中包含的公理必为有限多个,这说明第二个阶段是可以终止的。至此,定理 2 得证。

引理 2　让 A 和 A' 为 ABox,H 为角色层次。

a)若 A 与 H 一致,则对 A 应用一条确定规则所得到的 A' 与 H 一致。

b)若 A 与 H 一致,而且有一条非确定规则可应用于 A,则该规则能以某种方式执行,从而产生一个与 H 一致的 A'。

证明　为了证明 a),分别对每一条确定规则进行分析。假定 $I:=(\Delta^I,\cdot^I)$ 满足 A 和 H。

若 $\sqcap$ 规则被应用于 $a:C\sqcap D\in A$,则 $A'=A\cup\{a:C,\ a:D\}$。因为 $a^I\in(C\sqcap D)^I$,所以 $a^I\in C^I$ 且 $a^I\in D^I$。显然 I 满足 $a:C$ 和 $a:D$,因而也满足 A' 和 H。

若 $\forall$ 规则被应用于 $a:\forall R.\,C\in A$,则 a 有一个 R 邻居 b 使得 $A'=A\cup\{b:C\}$。根据定理 1 的证明,不难发现 $(a^I,b^I)\in R^I$。因为 $a^I\in(\forall R.\,C)^I$,所以 $b^I\in C^I$。显然 I 满足 $b:C$,因而也满足 A' 和 H。

若 $\forall_T$ 规则被应用于 $a:\forall R.\,C$,则存在一传递角色 S 使得 $S\sqsubseteq^* R\in H$,且 a 有一 S 邻居 b 使得 $A'=A\cup\{b:\forall S.\,C\}$。根据

定理1的证明,不难发现$(a^I,b^I)\in S^I$。假设$b^I\notin(\forall S.C)^I$,则必然存在$x\in\Delta^I$使得$(b^I,x)\in S^I$且$x\notin C^I$。因为S为传递角色,所以$(a^I,x)\in S^I\subseteq R^I$。因为$a^I\in(\forall R.C)^I$,所以$x\in C^I$。这说明假设不成立,$b^I\in(\forall S.C)^I$。显然$I$满足$b:\forall S.C$,因而也满足$A'$和$H$。

若$\forall_x$规则被应用于$\forall x.x:C\in A$,则A中存在一个个体a使得$A'=A\cup\{a:C\}$。因为$C^I=\Delta^I$,所以$a^I\in C^I$。显然I满足$a:C$,因而也满足A'和H。

若$\exists$规则被应用于$a:\exists R.C\in A$,则该规则会产生一个新个体b,使得$A'=A\cup\{a\ R\ b,\ b:C\}$。因为$a^I\in(\exists R.C)^I$,所以必然存在$x\in\Delta^I$使得$(a^I,x)\in R^I$且$x\in C^I$。定义一个新的解释函数$I'$,并规定:对于$A$中所有的个体$a$都有$a^{I'}=a^I$,对于新个体$b$有$b^{I'}=x$,对于$A$和$H$中所有的角色$R$都有$R^{I'}=R^I$。显然$I'=(\Delta^I,\cdot^{I'})$满足$a\ R\ b$和$b:C$,因而也满足$A'$和$H$。

若$\geqslant$规则被应用于$a:\geqslant 2R\in A$,则该规则会产生两个新个体b_1和b_2,使得$A'=A\cup\{a\ R\ b_1,\ a\ R\ b_2,\ b_1\neq b_2\}$。因为$a^I\in(\geqslant 2R)^I$,所以必然存在两个不同的$x_i\in\Delta^I(i=1$或$2)$使得$(a^I,x_i)\in R^I$。定义一个新的解释函数$.I'$,并规定:对于$A$中所有的个体$a$都有$a^{I'}=a^I$,对于新个体$b_i$有$b_i^{I'}=x_i$,对于$A$和$H$中所有的角色$R$都有$R^{I'}=R^I$。显然$I':=(\Delta^I,\cdot^{I'})$满足$a\ R\ b_1$,$a\ R\ b_2$和$b_1\neq b_2$,因而也满足$A'$和$H$。

为了证明b),分别对每一条非确定规则进行分析。假定$I:=(\Delta^I,\cdot^I)$满足A和H。

若$\cup$规则可应用于$a:C\cup D\in A$,则该规则能产生的结果为:$A'=A\cup\{a:C\}$或$A'=A\cup\{a:D\}$。因为$a^I\in(C\cup D)^I$,所以$a^I\in C^I$或$a^I\in D^I$。若$a^I\in C^I$,则让$A'=A\cup\{a:C\}$,否则让$A'=\{a:D\}$。显然I满足A'和H。

若$\leqslant 1R$规则可应用于$a:\leqslant 1R\in A$,则a的R邻居数>1,而且必然有一新个体为a的R后继。因为$a^I\in(\leqslant 1R)^I$,所以最多有一个$x\in\Delta^I$使得$(a^I,x)=R^I$,这意味着a所有R邻居的解释都相同。从a的R后继中任选一个新个体b_1,并从a的R邻居中任选另一个体b_2,因为$b_1^I=b_2^I$且I满足A,所以$b_1\neq b_2\notin A$。$A'=A[b_1/b_2]$显然是该规则能产生的一种结果。因为I满足A和H且$b_1^I=b_2^I$,所以I也满足A'和H。

若$\leqslant_o 1R$规则可应用于$a:\leqslant 1R\in A$,则a至少有两个邻居为老个体。因为$a^I\in(\leqslant 1R)^I$,所以最多有一个$x\in\Delta^I$使得$(a^I,x)=R^I$,这意味着a所有R邻居的解释都相同。从a的R邻居中任选两个老个体b_1和b_2,因为$b_1^I=b_2^I$且I满足A,所以$b_1\neq b_2\notin A$。$A'=A[b_1/b_2]$显然是该规则能产生的一种结果。因为I满足A和H且$b_1^I=b_2^I$,所以I也满足A'和H。至此,引理2得证。

定理3 完备性。若ABox与TBox一致,则算法能产生一个无冲突的完整的ABox。

证明 让A为ABox,T为TBox,H为角色层次。根据定理2和引理1、2可知,算法必定能产生一个与H一致的完整的ABox A'。既然A'与H一致,那么A'中显然不可能存在冲突。定理3得证。

定理4 Tableau算法是判定SHIF的ABox一致性的非确定性算法。

证明 根据定理1~3可知该定理成立。

3 结束语

本文给出了一种SHIF的ABox一致性判定Tableau算法,并证明了该算法是正确的。该算法先通过预处理将ABox转换成标准形式,然后按照特定的完整策略将一套Tableau规则应用于ABox,直到它扩展成完整的ABox为止。为了让算法在ABox只有无限模型的情况下仍能正确终止,本文采用了一种新颖的阻塞机制,该机制能避免多余的规则应用,从而提高算法的效率。在证明算法的正确性时,本文并未像许多文献一样通过构建Tableau来进行证明,而是采用了一种更为直接、简洁方法。因为算法所采用的Tableau规则中包含了非确定规则,所以该算法是一种非确定性算法。如果为$\cup$、$\leqslant 1R$和$\leqslant_o 1R$规则去除非确定性,完全可以将该算法改造成确定性算法,这也是下一阶段的工作。

参考文献:

[1] BAADER F, McGUINNESS D L, NARDI D, *et al*. The description logic handbook: theory, implementation, and applications[M]. 2nd ed. Cambridge: Cambridge University Press, 2007.

[2] Van HARMELEN F, LIFSCHITZ V, PORTER B. Handbook of knowledge representation[M]. San Diego: Elsevier Science, 2008.

[3] SATTLER U. A concept language extended with different kinds of transitive roles[EB/OL]. [2012-06-25]. http://lat.inf.tu-dresden.de/research/papers/1996/Sattler-KI-96.ps.gz.

[4] HORROCKS I, GOUGH G. Description logics with transitive roles [EB/OL]. [2012-06-25]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.164.1297&rep=rep1&type=pdf.

[5] HORROCKS I, SATTLER U. A description logic with transitive and inverse roles and role hierarchies[J]. Logic and Computation, 1999, 9(3): 385-410.

[6] HORROCKS I, SATTLER U, TOBIES S. A description logic with transitive and converse roles, role hierarchies and qualifying number restrictions[EB/OL]. [2012-06-25]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.45.9605&rep=rep1&type=pdf.

[7] TESSARIS S, GOUGH G. ABox reasoning with transitive roles and axioms[C]//Proc of International Workshop on Description Logics. 1999: 31-44.

[8] TESSARIS S, HORROCKS I. Abox satisfiability reduced to terminological reasoning in expressive description logics[C]//Proc of the 9th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning. London: Springer-Verlag, 2002: 435-449.

[9] HAARSLEV V, MÖLLER R. Expressive ABox reasoning with number restrictions, role hierarchies, and transitively closed roles[EB/OL]. [2012-6-25]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.62.1108&rep=rep1&type=pdf.

[10] HORROCKS I, SATTLER U, TOBIES S. Practical reasoning for very expressive description logics[J]. Logic Journal of the IGPL, 2000, 8(3): 239-264.

[11] MOTIK B, SHEARER R, HORROCKS I. Hypertableau reasoning for description logics[J]. Artificial Intelligence Research, 2009, 36(1): 165-228.

[12] GLIMM B, HORROCKS I, MOTIK B. Optimized description logic reasoning via core blocking[C]//Proc of the 5th International Conference on Automated Reasoning. Berlin: Springer-Verlag, 2010: 457-471.

[13] HORROCKS I, SATTLER U. A Tableau decision procedure for SHOIQ[J]. Automated Reasoning, 2007, 39(3): 249-276.

[14] TOBIES S. Complexity results and practical algorithms for logics in knowledge representation[EB/OL]. [2012-06-25]. http://lat.inf.tu-dresden.de/research/phd/Tobies-PhD-2001.pdf.