第 26 卷 第 6 期
2003 年 6 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 26 No. 6
June 2003

一种带缺省推理的描述逻辑

董明楷 蒋运承 史忠植
(中国科学院计算技术研究所 北京 100080)

摘 要 该文提出了一种新的带缺省推理的描述逻辑,它以描述逻辑为主要框架,对单调逻辑和非单调逻辑进行了整合,但又避免了一般缺省逻辑的困难.基于带缺省推理的描述逻辑,构建了一种同时具有 Tbox,Abox 和缺省规则的知识库系统,研究了带缺省推理的描述逻辑的可满足性、缺省可满足性、概念包含、缺省包含以及实例检测等推理问题,提出了一种用来检测可满足性和缺省可满足性的 Tableau-D 算法,并得到了缺省可满足性和缺省包含的转换定理.

关键词 描述逻辑;缺省逻辑;缺省可满足性;缺省包含
中图法分类号 TP18

A Description Logic with Default Reasoning

DONG Ming-Kai JIANG Yun-Cheng SHI Zhong-Zhi
(*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)

Abstract As a formal tool for knowledge representation and reasoning, description logic provides the decidable reasoning service. This paper proposes a new description logic with default reasoning, which integrates monotonic reasoning and non-monotonic reasoning while avoiding the difficulty of Reiter's default logic. Based on the description logic with default reasoning, we construct a knowledge base system that incorporates Tbox, Abox and default rules. Then we discuss several reasoning problems including concept satisfiability, default satisfiability, concept subsumption, default subsumption, and instance checking. Furthermore, we present a new algorithm called Tableau-D that is used to check the satisfiability and default satisfiability. Finally a transformation theorem of default satisfiability and default subsumption is also attained.

Keywords description logic; default logic; default satisfiability; default subsumption

1 引 言

描述逻辑(description logic)[1]是基于对象的知识表示的形式化,它吸收了 KL-ONE[2] 的主要思想,是一阶谓词逻辑的一个可判定子集.它与一阶谓词逻辑不同的是,描述逻辑系统能提供可判定的推理服务.除了知识表示以外,描述逻辑还用在其它许多领域,它被认为是以对象为中心的表示语言的最为重要的归一形式.描述逻辑的重要特征是它具有很强的表达能力和可判定性,它保证推理算法总能停止,并返回正确的结果.可满足性问题是描述逻辑推理的核心问题,Schmidt- Schauß 和 Smolka[3] 首先建立了基于描述逻辑 ALC 的 Tableau 算法,该算

收稿日期:2002-04-08;修改稿收到日期:2003-01-23.本课题得到国家“八六三”高技术研究发展计划项目(2001AA113121)资助.**董明楷**,男,1973 年生,博士研究生,主要研究领域为智能主体、知识表示与推理、网络智能信息处理.E-mail: dongmk@ics.ict.ac.cn.**蒋运承**,男,1974 年生,博士研究生,主要研究领域为智能主体和多主体系统、智能互联网.**史忠植**,男,1941 年生,研究员,博士生导师,主要研究方向为人工智能、机器学习和智能主体等.

法能在多项式时间内判定描述逻辑 ALC 概念的可满足性问题. 目前, Tableau 算法已用于各种描述逻辑中(如 ALCN, ALCQ 等).

非单调推理则是知识表示和常识推理的另外一个分支. 大多数非单调形式系统, 如 Reiter 的缺省逻辑(default logic)[4]、McCarthy 的限定推理(circumscription)[5]都使用一阶谓词逻辑作为它们的语言. 在一般形式下, 其形式化通常是不可判定的. 基于这个原因, 用于可判定的一些子情形的过程基本上都限制在命题逻辑, 从而使得命题逻辑和完全的一阶逻辑之间有了更大差距. 由于大多数概念描述语言可以看作是一阶逻辑的可判定的子类, 但它仍然比命题逻辑的表达力强. 它可以作为非单调推理形式系统的有意义的检验情形. 我们将看到, 这不仅仅在算法上是可用的, 而且在语义上也是可用的.

已有的对缺省逻辑的研究中, 一般把所有的规则都看作是缺省规则, 有的可能是没有前提的, 有的可能是没有检验条件的, 等等. 即把一般的单调推理中的规则与缺省规则等同看待, 而有的至多添加一些优先级或者构造有序的缺省理论[6]. 这种做法试图利用非单调推理框架来包容单调推理的主要框架. 这种做法并不能取得令人满意的结果, 往往还导致了许多更为复杂的问题. 毕竟单调推理是最为普遍的推理, 非单调推理应该是对单调推理的补充和修改, 以维护知识库的一致性, 并给出了一种信念修改的途径. 所以单调推理和非单调推理是知识库系统的两个必要组成部分, 以单调推理为主, 非单调推理为辅. 因此从知识表示和推理的角度来看, 应当从整体出发对单调推理和非单调推理进行整合, 分析两种推理的区别与联系, 扬长避短, 互为补充.

本文的工作立足于一般的描述逻辑 ALC, 在此基础上进行缺省扩展, 它继承了描述逻辑的特点和优点, 同时又具有非单调推理的能力. 我们构建的知识库同时包括 Tbox, Abox 和缺省规则三个部分, Tbox 和 Abox 部分即构成一般描述逻辑框架的知识库, 而缺省规则是为缺省推理所用. 本文在描述逻辑的基础上进行了扩充, 给出了概念的缺省可满足性和缺省包含的定义, 得到了缺省可满足性与缺省包含的转换定理, 给出了能检测缺省可满足性的 Tableau-D 算法, 并深入研究了带缺省的描述逻辑中的基本推理问题.

2 描述逻辑基础

描述逻辑是一种基于对象的知识表示的形式化, 也叫概念表示语言或术语逻辑. 它建立在概念(concept)和关系(role)之上, 其中概念解释为对象的集合, 关系解释为对象之间的二元关系. 描述逻辑是一阶逻辑的一个可判定的子集, 具有合适定义的语义, 并且具有很强的表达能力. 一个描述逻辑系统包含四个基本组成部分: 表示概念和关系的构造集; Tbox 包含断言; Abox 实例断言; Tbox 和 Abox 上的推理机制. 一个描述逻辑系统的表示能力和推理能力取决于对以上几个要素的选择以及不同的假设.

一个知识库 $\mathcal{K}=\langle \mathcal{T},\mathcal{A}\rangle$ 由两个部分组成: Tbox $\mathcal{T}$ 和 Abox $\mathcal{A}$. 其中, Tbox 是一个关于包含断言的有限集合, 具有如下形式: $C\sqsubseteq D$, 这里 C 和 D 为概念. 通常我们用 $C\equiv D$ 作为 $C\sqsubseteq D$ 和 $D\sqsubseteq C$ 的缩写. Abox 是实例断言的有限集合, 形为 $C(a)$, 其中 C 是一个概念, a 是一个个体的名字; 或者形为 $P(a,b)$, 其中 P 为一个原始关系, a,b 为两个个体的名字.

一般地, 描述逻辑依据提供的构造算子, 在简单的概念和关系上构造出复杂的概念和关系. 通常描述逻辑至少包含以下算子: 交($\sqcap$), 并($\sqcup$), 非($\neg$), 存在量词($\exists$)和全称量词($\forall$). 这种最基本的描述逻辑称之为 ALC. 在 ALC 的基础上再添加不同的构造算子, 则构成不同表达能力的描述逻辑. ALC 的语法和语义如表 1 所示.

表 1 ALC 的语法和语义

构造算子	语法	语义
原子概念	A	$A^I\subseteq\Delta^I$
原子关系	P	$P^I\subseteq\Delta^I\times\Delta^I$
顶部	$\top$	Δ^I
底部	$\bot$	$\varnothing$
交	$C\sqcap D$	$C^I\cap D^I$
并	$C\sqcup D$	$C^I\cup D^I$
非	$\neg C$	Δ^I-C^I
存在量词	$\exists R.C$	$\{x\mid \exists y,(x,y)\in R^I\wedge y\in C^I\}$
全称量词	$\forall R.C$	$\{x\mid \forall y,(x,y)\in R^I\Rightarrow y\in C^I\}$

ALC 语义将概念解释为一定领域的子集, 关系是该领域上的二元关系. 形式上, 一个解释 $I=(\Delta^I,\cdot^I)$ 由解释的领域 Δ^I 和解释函数 $\cdot^I$ 所构成, 其中解释函数把每个原子概念 A 映射到 Δ^I 的子集, 而把每个原子关系映射到 $\Delta^I\times\Delta^I$ 的子集.

一个解释 I 是包含断言 $C\sqsubseteq D$ 的模型, 当且仅当 $C^I\subseteq D^I$; 解释 I 是实例断言 $C(a)$ 的模型, 当且仅当 $a\in C^I$; I 是 $P(a,b)$ 的模型, 当且仅当 $(a,b)\in P^I$; 解释 I 是知识库 $\mathcal{K}$ 的模型, 当且仅当 I 是 $\mathcal{K}$ 中每个包含断言和实例断言的模型; 若 $\mathcal{K}$ 有模型, 则称 $\mathcal{K}$ 是可满足的; 若断言 σ 对于 $\mathcal{K}$ 的每个模型是满足的, 则称 $\mathcal{K}$ 逻辑蕴含 σ, 记为 $\mathcal{K}\models\sigma$. 对概念 C, 若 $\mathcal{K}$ 有一

个模型 I 使得 $C^I \neq \varnothing$,则称 C 是可满足的. 知识库 $\mathcal{K}$ 中的概念 C 的可满足性可以逻辑表示为 $\mathcal{K} \not\models C \sqsubseteq \bot$.

3 描述逻辑的缺省扩展

描述逻辑作为知识表示的形式化基础,具有很强的知识表示和推理能力,但是一般的描述逻辑在处理非单调的、不完备的知识时,却无能为力. 我们考虑在一般描述逻辑的基础上添加缺省规则,即扩展为带有缺省推理的描述逻辑,研究这种扩展后的描述逻辑的知识表示问题、可满足性问题以及推理问题. 一般的缺省逻辑的优点是能处理例外的情况,能进行非单调的表示和推理,但其总体的知识表示能力却有很多的限制,特别是在进行大规模的知识表示时,由于例外毕竟是少数,这样就导致了表示和推理的复杂化. 为此,我们将描述逻辑和缺省逻辑进行整合,扬长避短,形成带缺省推理的描述逻辑.

定义 1. 一个缺省规则是形如 $\frac{C:E_1, E_2, \cdots, E_n}{D}$ 这样的表达式,其中 $C, E_1, \cdots, E_n, D$ 为概念名. C 称为前提条件,$E_1, E_2, \cdots, E_n$ 称为检验条件,D 称为缺省的结论.

缺省规则的具体含义为:对于任意的 x,若 $x:C$,且对所有的 $1 \leq i \leq n$,推导不出 $x:\neg E_i$,则我们可以使用该缺省规则,得到结论 $x:D$.

一般地,我们采用例外作为检验条件的方法来构造缺省规则,具体做法是在缺省规则的检验条件中列举出每个例外的否定,而结论部分并不出现在检验条件中. 这里前提条件、检验条件和结论是相对独立的,这样构造出来的缺省规则既不是正规的,也不是半正规的. 乍一看这种缺省规则似乎很复杂,但对概念的可满足性检测是很有利的,下面将作具体阐述.

现在我们举例说明缺省规则的构造方法. 例如对于"哺乳动物一般情况下是胎生的,除非它是鸭嘴兽",可以构造出如下的缺省规则:

$$\frac{Mammal:\neg Platypus}{Viviparous} \tag{D1}$$

同样地,对于经典的缺省逻辑的例子"一般情况下鸟是会飞的,但企鹅和鸵鸟不会飞",其缺省规则可以表示为

$$\frac{Bird:\neg Penguin, \neg Ostrich}{Fly} \tag{D2}$$

在一般的描述逻辑框架下,一个知识库通常包括 Tbox 和 Abox 两个部分. 当我们对描述逻辑进行了缺省扩展后,增加了缺省规则,下面给出带缺省规则的知识库的定义.

定义 2. 一个带有缺省规则的知识库是一个三元组 $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}, \mathcal{D} \rangle$,其中 $\mathcal{T}$ 为包含断言集合 Tbox,$\mathcal{A}$ 为实例断言集合 Abox,$\mathcal{D}$ 为缺省规则集.

我们在知识库中添加缺省规则后,将一些例外的断言公式加入到 Tbox 和 Abox 中. 其中 Tbox 中包含的例外的断言是在概念和关系层的,而 Abox 中包含的例外的断言是在个体实例层的,是闭公式. 例如,对于"一般情况下鸟是会飞的,但企鹅和鸵鸟不会飞"这样的事实,除了在缺省集中加入缺省规则 (D2) 以外,还需要在 Tbox 中添加两个事实,包含"企鹅和鸵鸟不会飞". 因为这是已知的事实,可以作为公理(事实). 将例外情形加入到 Tbox 或者 Abox 中,是对知识库的补充,这样既能保证知识库的一致性,又能保证知识库的完整性. 这两个例外在 Tbox 中可以表示如下:

$$Penguin \sqsubseteq \neg Fly \tag{T1}$$

$$Ostrich \sqsubseteq \neg Fly \tag{T2}$$

由于缺省的本质意义就是在人们的常识推理中经常会出现一些一般情况下成立的事实或者特征,形式化表示即为"若 C 成立,则一般情况下有 D 成立",或者表述为"若 x 属于 C,则一般情况下有 x 属于 D". 既然是一般情形,就说明有特殊情形或者例外,而一旦特殊情形出现,则这个一般性的结论就不再成立. 所以这种一般情形的推断的优先级应该比特殊情形推断的优先级低,否则先有一般性后有特殊性将会导致矛盾的出现. 因而我们将特殊(例外)情形列举到 Tbox 或者 Abox 中,推理的优先级高,而缺省规则的使用优先级则比较低. 推理时,由于例外在先,只要出现了例外情形,就会先在 Tbox 或者 Abox 中得出,从而阻挡其相应缺省规则的使用.

4 带缺省的描述逻辑的推理

4.1 可满足性问题

定义 3. 概念 C 在知识库 $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ 中是可满足的当且仅当 C 不包含于 $\bot$. 否则称概念 C 是不可满足的.

定义 4. 概念 C 在带缺省规则的知识库 $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}, \mathcal{D} \rangle$ 中是缺省可满足的当且仅当 C 在知识库 $\langle \mathcal{T}, \mathcal{A} \rangle$ 中是可满足的,且在加入缺省规则集 $\mathcal{D}$ 后仍有 C 不包含于 $\bot$. 若 C 在知识库 $\langle \mathcal{T}, \mathcal{A} \rangle$ 中是可满足的,而当加入缺省规则集 $\mathcal{D}$ 后有 C 包含于 $\bot$,此时称

C 为缺省不可满足的.

缺省可满足性是在带缺省规则的知识库中可满足性的一种补充定义，是为后面的缺省推理做准备. 从上述定义可以看出，在知识库 $\langle \mathcal{T}, \mathcal{A} \rangle$ 中可满足的概念在知识库 $\langle \mathcal{T}, \mathcal{A}, \mathcal{D} \rangle$ 中未必是缺省可满足的，它被细分为两种情形：缺省可满足和缺省不可满足. 因此，在需要的情况下，我们还将在知识库 $\langle \mathcal{T}, \mathcal{A}, \mathcal{D} \rangle$ 中继续讨论其缺省可满足性问题，以方便进一步的缺省推理任务. 例如，概念 Bird $\sqcap \neg$Fly 在知识库 $\langle \mathcal{T}, \mathcal{A} \rangle$ 中是可满足的，但在包含缺省规则(D2)的知识库 $\langle \mathcal{T}, \mathcal{A}, \mathcal{D} \rangle$ 中则是缺省不可满足的.

对于知识库的一致性问题，它可以化为概念的可满足性问题，这里就不再详细讨论.

4.2　Tableau-D 算法

可满足性问题是描述逻辑推理中的核心问题，因为其它许多问题(如包含检测、一致性问题等)都可化为可满足性问题. Schmidt-Schauß 和 Smolka[3] 首先建立的基于描述逻辑 ALC 的 Tableau 算法能在多项式时间内判断描述逻辑 ALC 概念的可满足性问题. 为了解决带缺省的描述逻辑中的推理问题，我们一般都把这些推理问题转换为可满足性问题和缺省可满足性问题. 这里我们对 Tableau 算法进行了扩充，建立了 Tableau-D 算法. 下面具体设计概念的可满足性和缺省可满足性的检测算法. 这个算法将对由变量、简单概念和关系组成的约束进行操作.

假设存在变量符号字母表，用 x, y, z 表示，一个约束是具有以下两种形式之一的句法对象：$x:C$，xPy. 其中 C 是一个简单的概念，P 是一个原子关系.

令 I 为一个解释，一个 I-赋值是映射每个变量到 Δ^I 的一个元素的函数 α. 我们称 α 满足 $x:C$，若 $\alpha(x) \in C^I$. 若 $\alpha(x), \alpha(y) \in P^I$，称 α 满足 xPy. 如果有一个解释 I 和 I-赋值 α 使得 α 满足约束 c，则称约束 c 是可满足的. 一个约束系统 S 是一个有穷非空的约束集. 如果一个 I-赋值 α 满足约束系统 S 中的每个约束，则称 α 满足 S. 如果存在一个解释 I 和 I-赋值 α 使得 α 满足约束系统 S，则称约束系统 S 是可满足的.

为了检测概念 C 的可满足性和缺省可满足性，Tableau-D 算法由约束系统 $S = \{x:C\}$ 开始，具体的演算由以下五条规则给出：

$\sqcap$规则：若 $\{x:C_1 \sqcap C_2\} \in S$，且 $\{x:C_1, x:C_2\} \not\subseteq S$，则 $S \leftarrow \{x:C_1, x:C_2\} \cup S$；

$\sqcup$规则：若 $\{x:C_1 \sqcup C_2\} \in S$，且 $\{x:C_1, x:C_2\} \cap S = \varnothing$，则 $S \leftarrow \{x:D\} \cup S$，其中 $D = C_1$ 或 $D = C_2$；

$\exists$规则：若$(x:\exists R.C) \in S$，且没有 y 使得$(x\,R\,y) \in S$ 且$(y:C) \in S$，则 $S \leftarrow \{y:C, x\,R\,y\} \cup S$；

$\forall$规则：若$(x:\forall R.C) \in S$，$(x\,R\,y) \in S$，且$(y:C) \notin S$，则 $S \leftarrow \{y:C\} \cup S$；

缺省规则：若上述规则中没有规则可用，则考虑从缺省规则集中选择缺省规则. 若缺省规则集中有规则 $\dfrac{C:E_1, E_2, \cdots, E_n}{D}$，且$(x:C) \in S$，$(x:\neg E_i) \notin S$，其中 $1 \le i \le n$，则 $S \leftarrow \{x:D\} \cup S$.

在上述 5 条规则中，前 4 条规则称为单调规则，缺省规则是非单调规则. 这里我们定义单调规则的使用优先级高于缺省规则的使用优先级. 只有当没有单调规则可用时，才考虑使用缺省规则，而使用缺省规则进行推理的结果也带有不确定成分. 基于概念的可满足性定义和缺省可满足性定义，我们构建的 Tableau-D 算法包括两个层次的推理，一是单调推理，另一种是缺省推理.

Tableau-D 算法的单调推理部分用来检测概念的可满足性，它限定只能使用 4 条单调的规则. 对于这部分，已有的 Tableau 算法中所使用的定理同样有效，即有下述定理.

定理 1.　一个概念 C 是可满足的，当且仅当约束系统 $\{x:C\}$ 是可满足的.

定理 2.　一个完备的约束系统是可满足的，当且仅当它不包含冲突.

为了检测一个简单概念 C 的满足性问题，就必须从约束系统 $S = \{x:C\}$ 开始，然后不断添加约束到 S 中，直到产生矛盾或满足 C 的一个解释能从结果系统中获得，从而生成一个完备的约束系统，它们是有穷多个的. 若所有这些系统中都包含有冲突，则 C 是不可满足的，否则为可满足的. 如果没有传播规则运用于一个约束系统，则该约束系统是完备的. 一个冲突是形如 $\{x:\perp\}$ 或者 $\{x:A, x:\neg A\}$ 这样的约束系统. 注意，在$\sqcup$规则中，有两个分支：$\{x:C_1\} \cup S$ 和 $\{x:C_2\} \cup S$. 只有当两个分支都有冲突时，约束系统才是有冲突的，否则就是没有冲突的.

对于缺省推理部分，我们在单调推理的基础上进行. 只有当完备的约束系统是可满足的时候，才考虑使用缺省推理. 若约束系统中没有任何单调规则可用时，则我们将从缺省规则集中选择缺省规则，按缺省规则的条件进行推理，以产生新的结果. 显然，对可以使用缺省规则的约束系统，依据缺省可满足性的定义，我们可以得到下述结果.

定理 3.　一个概念 C 是缺省可满足的，当且仅当带缺省的约束系统 $\{x:C\}$ 是缺省可满足的.

定理 4.　一个带缺省规则的约束系统是缺省可满足的，当且仅当它不包含冲突.

此时，我们只需要在约束系统中检查是否有缺省规则可用. 若有缺省规则可用，且不包含冲突，则约束系统是缺省可满足的. 若使用缺省规则后包含了冲突，则约束系统是缺省不可满足的. 据此我们相应地判断概念是否为缺省可满足的.

在使用缺省规则时，可能牵涉到缺省规则的先后顺序问题. 通常缺省规则的使用从小入手，即假设有两个缺省规则可用，其中一个规则 d_1 的前提条件所指的概念包含于另一个规则 d_2 所指的概念，则就先使用 d_1 这个规则，再使用另一个规则. 这样做的主要目的是使推理中尽可能使用更贴近的概念和特征，特殊情形在先，一般情形在后，让推理更为精确具体. 例如，当人们看到一匹马时，他的第一反应是“这是一匹马”，而不说“这是一只动物”，也就是使用更贴切更具体的概念和特征.

4.3　概念包含问题

在带缺省推理的描述逻辑中，概念的包含问题同样满足下述定理.

定理 5.　概念 C 包含于概念 D，当且仅当 $C\sqcap\neg D$ 是不可满足的.

缺省规则一般用于属性、特征方面的描述，即描述某个概念或者个体是否具有某个属性、特征，或者满足某个性质. 例如描述“一般情况下鸟会飞”，即鸟具有“飞翔”这个能力或者属性. 又如“人会说话，除非他是哑巴或者婴儿”，“说话”是一般人都具有的能力. 但是，由于在一般的描述逻辑框架中只有概念和关系，我们不妨可以把上述说法换为概念的包含关系. 例如，“一般情况下鸟是属于能飞行的事物”，“一般情况下人是能说话的”，“哺乳动物一般情况下是属于胎生的”等. 由此，缺省规则也可以解释为概念之间的一种缺省的包含关系，只不过它不是集合论中的严格包含关系，存在有例外. 下面给出缺省包含的定义，它是一种近似的包含关系.

定义 5.　在知识库 $\langle\mathcal{T},\mathcal{A},\mathcal{D}\rangle$ 中，对于概念 C 和 D，如果 $C\not\sqsubseteq D$，但存在有穷非空的缺省规则集 $\mathcal{D}'\subseteq\mathcal{D}$，使得 $\mathcal{D}'$ 中的所有缺省规则都能在从 $\{x:C\}$ 开始的约束系统 S 中使用，且能导出 $(x:D)\in S$，则称概念 D 缺省包含概念 C，记为 $C\subsetneq D$.

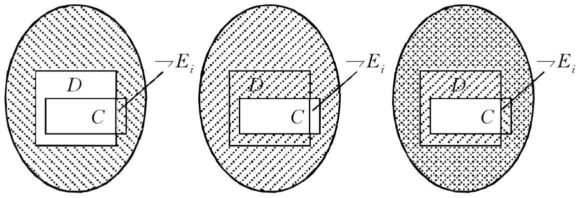
缺省包含的语义可以直观地解释为“一般情况下属于概念 C 的对象也属于概念 D”，或者解释为“一般情况下属于概念 C 的对象具有属性 D”. 实际上这本身就是对缺省规则的体现和扩展，使缺省规则有更为具体的语义. 尽管这里定义的缺省包含只是一种近似的包含关系，并不是严格的包含关系，但这恰好是为了弥补严格包含关系的不足，引入了近似的、非单调的推理. 这里把传统的 Reiter 缺省理论中的缺省推理转换成为更接近于描述逻辑的缺省包含的推理.

在描述逻辑的概念包含问题中，若 $C\sqsubseteq D$，则有 $\neg D\sqsubseteq\neg C$，这是显而易见的. 同样地，由于在我们定义的缺省规则中，结论并不出现在检验条件中，因此这些缺省规则满足类似的性质. 可以把一个缺省规则等价地转换为其逆否形式，即把前提和结论都换为其否定形式，然后将结论和前提相互交换. 于是我们得到下述引理.

引理 1.　缺省规则 $\dfrac{C:E_1,E_2,\cdots,E_n}{D}$ 可以转化为其等价形式 $\dfrac{\neg D:E_1,E_2,\cdots,E_n}{\neg C}$.

证明.　由缺省规则 $\dfrac{C:E_1,E_2,\cdots,E_n}{D}$，所以若 $x:C$，且 $x:\neg E_i$ 不成立，其中 $1\le i\le n$，则缺省情况下有 $x:D$. 如图 1 所示，其中圆形代表全体论域，正方形代表概念 D，长方形代表概念 C，$\neg E_i$ 即为例外情形. $\neg D$ 即图 1(a) 中的阴影区域，$\neg C$ 为图 1(b) 中的阴影区域，显然 $\neg E_i\sqsubseteq\neg D$，但 $\neg E_i\not\sqsubseteq\neg C$，即除 $\neg E_i$ 以外，$\neg D$ 的其余部分全部包含于 $\neg C$，如图 1(c) 所示. 因而我们可以构建出缺省规则，把例外的否定放在其检验条件中，即成为 $\dfrac{\neg D:E_1,E_2,\cdots,E_n}{\neg C}$.

证毕.



(a) $\neg D$　　(b) $\neg C$　　(c) $\dfrac{\neg D:E_1,E_2,\cdots,E_n}{\neg C}$

图 1　缺省规则的等价形式示意图

引理 2.　在知识库 $\langle\mathcal{T},\mathcal{A},\mathcal{D}\rangle$ 中，若从 $\{x:\neg D\}$ 开始的约束系统 S 能导出 $(x:\neg C)\in S$，则同样地从 $\{x:C\}$ 开始的约束系统 S' 能导出 $(x:D)\in S'$.

证明.　设从 $\{x:\neg D\}$ 开始的约束系统 S 能推导出 $(x:\neg C)\in S$，则必定存在一个有穷的缺省规则集 $\mathcal{D}'\subseteq\mathcal{D}$，使得 $\mathcal{D}'$ 中的所有缺省规则都能在 S 中使用，且在知识库中能导出 $(x:\neg C)\in S$. 即存在一个概念包含和缺省规则的序列 $P_1,P_2,\cdots,P_k$，使得从

$\{x:\neg D\}$开始的约束系统 S 由这个序列可以导出 $(x:\neg C)\in S$. 由引理 1 及 $A\sqsubseteq B$ 有 $\neg B\sqsubseteq\neg A$,我们可以将这个序列中所有形如 $\frac{\neg B:E_1,E_2,\cdots,E_n}{\neg A}$ 的缺省规则等价地改写为 $\frac{A:E_1,E_2,\cdots,E_n}{B}$,把形如 $\neg B\sqsubseteq\neg A$ 的公理等价地改写为 $A\sqsubseteq B$,从而得到序列 $P_1',P_2',\cdots,P_k'$,再将该序列倒置,得到 $P_k',\cdots,P_1'$,这个序列即为从$\{x:C\}$开始的约束系统 S' 导出$(x:D)\in S'$的序列. 证毕.

为了能在带缺省的描述逻辑中方便地进行缺省推理,我们得到了关于缺省包含和缺省可满足性的一个很好的结果,表述为下述定理.

定理 6. 概念 D 缺省包含概念 C 当且仅当 $C\sqcap\neg D$ 在知识库$\langle\mathcal{T},\mathcal{A},\mathcal{D}\rangle$中是缺省不可满足的.

证明. (1) 必要性. 设概念 D 缺省包含概念 C,对于 $x:C\sqcap\neg D$,则 $x:C$ 且 $x:\neg D$,由缺省包含的定义,有 $C\not\sqsubseteq D$,且存在有穷非空的缺省规则集 $\mathcal{D}'\subseteq\mathcal{D}$,使得 $\mathcal{D}'$中的所有缺省规则都能在从$\{x:C\}$开始的约束系统 S 中使用,且能导出$(x:D)\in S$,从而$(x:D)\in S$ 且$(x:\neg D)\in S$,得出冲突. 所以 $C\sqcap\neg D$ 是缺省不可满足的.

(2) 充分性. 设 $C\sqcap\neg D$ 在知识库$\langle\mathcal{T},\mathcal{A},\mathcal{D}\rangle$中是缺省不可满足的. 由缺省不可满足的定义,则 $C\sqcap\neg D$ 在知识库$\langle\mathcal{T},\mathcal{A}\rangle$中是可满足的,即有 $C\not\sqsubseteq D$,且从$\{x:C\sqcap\neg D\}$开始的约束系统 S 包含冲突. 对于冲突的出现,只可能有以下 3 种情形:

① 存在有穷非空的缺省规则集 $\mathcal{D}'\subseteq\mathcal{D}$,使得由$\{x:C\}$开始的约束系统 S 使用 $\mathcal{D}'$中的规则可以导出$(x:D)\in S$,与 $x:\neg D$ 相冲突. 依缺省包含的定义,故概念 D 缺省包含概念 C.

② 存在有穷非空的缺省规则集 $\mathcal{D}'\subseteq\mathcal{D}$,使得由$\{x:\neg D\}$开始的约束系统 S 使用 $\mathcal{D}'$中的规则可以导出$(x:\neg C)\in S$,从而与 $x:C$ 相冲突. 依据引理 2,所以由$\{x:C\}$开始的约束系统 S 使用 $\mathcal{D}'$中的规则可以导出$(x:D)\in S$,从而概念 D 缺省包含概念 C.

③ 存在概念 A 以及缺省规则集 $\mathcal{D}$ 的两个子集 $\mathcal{D}_1$ 和 $\mathcal{D}_2$,使得由$\{x:C\}$开始的约束系统 S 使用 $\mathcal{D}_1$ 中的规则可以导出$(x:A)\in S$,而由$\{x:\neg D\}$开始的约束系统 S' 使用 $\mathcal{D}_2$ 中的规则可以导出$(x:\neg A)\in S'$,即由 $x:C\sqcap\neg D$ 能得到 $x:A$ 和 $x:\neg A$ 这样的冲突. 其中 $\mathcal{D}_1$ 和 $\mathcal{D}_2$ 至少有一个不空,因为若它们同时为空,则必有 $C\sqsubseteq A$ 且 $\neg D\sqsubseteq\neg A$,即 $A\sqsubseteq D$,从而得出 $C\sqsubseteq D$,与 $C\not\sqsubseteq D$ 矛盾. 但不管 $\mathcal{D}_1$ 和 $\mathcal{D}_2$ 是否为空,都说明由$\{x:C\}$开始的约束系统 S 可以导出$(x:A)\in S$,由$\{x:\neg D\}$开始的约束系统 S' 可以导出$(x:\neg A)\in S'$. 对后者,利用引理 2,得到由$\{x:A\}$开始的约束系统 S'' 可以导出$(x:D)\in S''$. 再与前者相结合,即可以断言由$\{x:C\}$开始的约束系统 S 可以导出$(x:D)\in S$. 因此再依据缺省包含的定义,我们有概念 D 缺省包含概念 C.

这里的概念 A 只是导出冲突的一种形式,对应地,由$\{x:C\}$导出$(x:\neg A)\in S$ 且由$\{x:\neg D\}$导出$(x:A)\in S$ 的情形与上述证明是相应的,不再详细证明.

综合上述三种情形,我们得到由 $C\sqcap\neg D$ 的缺省不可满足性能导出概念 D 缺省包含概念 C.

证毕.

有了定理 6 作保证,则缺省包含问题就可以转换为概念的缺省可满足性问题,这对于缺省推理来说是一个重要的结果. 尽管这里只是使用概念间缺省包含问题,但由于我们在定义概念时可以使用广义上的概念,它可以代表具有某些属性或者特征的个体的集合,因而其适用范围是很广的. 下面的例子就是将缺省包含问题转换为缺省可满足性问题,然后利用 Tableau-D 算法来进行推理.

例 1. 设带缺省规则的知识库 $\mathcal{K}=\langle\mathcal{T},\mathcal{A},\mathcal{D}\rangle$如下:

$$\mathcal{T}=\{Sparrow\sqsubseteq Bird,Penguin\sqsubseteq Bird,$$
$$Penguin\sqsubseteq\neg Fly,Ostrich\sqsubseteq\neg Fly\}$$
$$\mathcal{A}=\varnothing,$$
$$\mathcal{D}=\left\{\frac{Bird:\neg Penguin,\neg Ostrich}{Fly}\right\}.$$

现在要解答的问题是"麻雀会飞吗?",即判断概念"麻雀"和"会飞的"之间是否有包含关系或者缺省包含关系. 先将 Tbox 和 Abox 转换为约束形式,有

$$S=\{x:(\neg Sparrow\sqcup Bird),x:(\neg Penguin\sqcup Bird),$$
$$x:(\neg Penguin\sqcup\neg Fly),x:(\neg Ostrich\sqcup\neg Fly)\},$$

然后将问题转化为概念的可满足性和缺省可满足性问题. 将$\{x:(Sparrow\sqcap\neg Fly)\}$加入到约束系统 S 中,开始使用 Tableau-D 算法进行推理,具体如下:

(1) $x:(Sparrow\sqcap\neg Fly)$	
(2) $x:Sparrow$	$\sqcap$规则
(3) $x:\neg Fly$	$\sqcap$规则
(4) $x:(\neg Sparrow\sqcup Bird)$	Tbox
(5) $x:\neg Sparrow$	$\sqcup$规则
(6) 〈$\sqcup$规则的一个分支冲突,即(2)和(5)冲突〉	
(7) $x:Bird$	$\sqcup$规则

(8) x:Fly 缺省规则

(9) 〈(3)和(8)冲突〉

从而 $Sparrow \sqcap \neg Fly$ 是缺省不可满足的,所以有 $Sparrow \sqsubsetneq Fly$. 即 $Sparrow$ 缺省包含于 Fly,问题的解答应该是"麻雀一般情况下是会飞的".

由于判断概念 D 和 C 之间的缺省包含关系是通过检测 $C \sqcap \neg D$ 的缺省可满足性来实现的,因此若我们把缺省规则的结论部分也加入到检验条件中,则由于 $\neg D$ 的出现,就会阻挡缺省规则的使用,得不出结论 D,当然也就得不出矛盾. 一般缺省理论由于采用的是正规的或者半正规的缺省规则,就不能采用类似一阶逻辑中的"反证"方法,而是使用缺省证明的方式[4]. 这就使得一般的缺省推理变得更为复杂,难以实现. 而我们提出的将缺省的结论不放入检验条件中这种缺省规则,目的就是要避免一般缺省推理所存在的这种问题,使我们的缺省推理适合于描述逻辑的推理方式.

4.4 实例检测

基于上述的缺省可满足性和缺省包含问题,我们可以类似地应用于实例检测中. 一般地,实例检测问题可以转化为知识库的一致性检测问题. 同样地,由于我们的缺省规则中结论部分并没有出现在缺省检验条件中,因而也可以采用这种方法. 检测方法就是将对个体的断言的否定加入到知识库中,再检测知识库是否是一致的. 若是一致的,则将个体断言加入到知识库中,进而再检测其可满足性和缺省可满足性.

5 相关工作比较

Baader 将 Reiter 的缺省逻辑整合到描述逻辑中,构造出术语缺省理论(Terminological default theory)[7]. 术语缺省理论是一个二元组 $\langle \mathcal{A}, \mathcal{D} \rangle$,其中 $\mathcal{A}$ 是一个实例断言集合 Abox,$\mathcal{D}$ 为一个有穷的缺省规则集合,其前提、检验和结论都是概念项. 显然,由于 Abox 可以看作是闭公式集合,且概念可以看作是带一个自由变元的公式,术语缺省理论包含在 Reiter 的开缺省理论中.

在术语缺省理论基础上,Baader 在文献[8]中还引入了带优先级的缺省理论,它主要用来处理有前提条件的缺省规则之间的优先级问题. 在这种缺省理论中,他引入了在缺省规则集 $\mathcal{D}$ 上的严格偏序关系"$<$",并定义了一种带优先级的扩张——P-扩张. P-扩张是 Reiter 扩张中考虑了优先级的那些扩张,它的个数一般比 Reiter 扩张的个数要少.

在 Baader 的术语缺省理论中,缺省规则只能运用于 Abox,即只包含闭公式的个体断言,不能运用于 Tbox 上,因而推理是很有限的. 由于在缺省理论中,对缺省规则进行 Skolem 化通常会破坏概念公式的构造特性,所以 Baader 只考虑了 Abox 的情形. 但是,实际上这是一般缺省理论都面临的问题,如果仍然是在 Reiter 缺省理论的框架下进行研究,采用计算扩张的方法来进行推理,就不能避免这个问题. 尽管 Baader 的工作是基于描述逻辑的,但它还是处于 Reiter 的缺省逻辑的框架之下,因此这并不能避免缺省逻辑所面临的问题,也不便于扩展.

Haarslev[9] 的工作则是在 Baader 术语缺省理论的基础上进行了应用扩展,用来处理关于空间信息的缺省推理. 为了解决搜索引擎处理相关文档存在的问题,Lambrix[10] 在知识库中同时引入分类信息和缺省信息,但其工作主要还是 Baader 的术语缺省逻辑的应用和扩展,没有提出更新的结果.

还有一项工作是 Sebastiani 和 Straccia[11] 提出的另一种术语缺省逻辑 TDL^-,它吸收了带例外的多继承网络 MINEs 的主要思想,使用特殊化原理对例外情况进行优先处理,即子类优先的原理. 它继承了 Reiter 的缺省逻辑,但又克服了其缺点,同样采用计算扩张的方法来进行缺省推理. 其基本思想类似于 Baader 在文献[8]中的工作.

总体来看,上述几项工作与其说是在描述逻辑上进行缺省的扩展,不如说是采用描述逻辑的思想对缺省逻辑进行扩展,因为他们更倾向于把缺省逻辑作为主要框架,采用计算扩张的方法进行缺省推理,而只是把描述逻辑作为知识表示的工具,并没有把它作为推理工具.

我们的扩展方法是对缺省规则进行重新的处理,缺省规则的结论不出现在检验条件中,在已有的描述逻辑基础上添加了缺省规则,其主要思想还是采用描述逻辑的框架,将 Tableau 算法进行了扩充. 因而我们的工作并没有采用计算扩张的方法进行缺省推理,避免了一般缺省逻辑所面临的问题. 在引入了缺省可满足性和缺省包含这两个定义后,我们得到了缺省可满足性与缺省包含的转换定理这一重要结果,从而改变了传统的缺省推理方式,使得缺省推理转变为缺省可满足性问题. 这样既包容了一般描述逻辑中的推理问题,又能进行缺省的推理,这是一种新的、有效的整合方式.

6 结 论

本文在一般描述逻辑的基础上进行了缺省的扩展,使其既具备一般知识表示和推理的能力,又具有非单调推理的能力.我们构建的带缺省规则的知识库包括 Tbox,Abox 和缺省规则集三个部分,缺省规则中将例外的否定作为检验条件,结论部分并不出现在检验条件中,同时把例外事实加入到 Tbox 和 Abox 中.这种带缺省推理的描述逻辑仍然以描述逻辑作为主要框架,从而避免了缺省理论在推理上存在的困难.文中给出了缺省可满足性和缺省包含等新的定义,研究了带缺省推理的描述逻辑的可满足性、缺省可满足性、概念包含、缺省包含以及实例检测等推理问题,提出了一种用来检测可满足性和缺省可满足性的新算法——Tableau-D 算法.这种整合方式是对描述逻辑在非单调推理上的有效的扩充,提出了一种新的缺省推理方式.进一步的工作将考虑带缺省推理的描述逻辑中的信念修改、Tableau-D 算法的优化等问题,我们将另文给出.

参 考 文 献

1 Baader F, *et al*. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge: Cambridge University Press, 2002

2 Brachman R J, Schmolze J G. An overview of the KL-ONE knowledge representation system. Cognitive Science, 1985, 9(2): 171～216

3 Schmidt-Schauß M, Smolka G. Attributive concept descriptions with complements. Artificial Intelligence, 1991, 48(1): 1～26

4 Reiter R. A logic for default reasoning. Artificial Intelligence, 1980, 13(1): 81～132

5 McCarthy J. Circumscription——A form of non-monotonic reasoning. Artificial Intelligence, 1980, 13(1): 27～39

6 Brewka G. Nonmonotonic Reasoning: Logical Foundations of Commonsense. Cambridge: Cambridge University Press, 1991

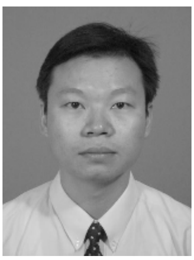
7 Baader F, Hullunder B. Embedding defaults into terminological representation systems. Journal of Automated Reasoning, 1995, 14(1): 149～180

8 Baader F, Hollunder B. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. Journal of Automated Reasoning, 1995, 15(1): 41～68

9 Haarslev V, Möller R, Turhan A Y, Wessel M. On terminological default reasoning about spatial information: Extended abstract. In: Proceedings of the International Workshop on Description Logics (DL'99), Linkoping, Sweden, 1999. 155～159

10 Lambrix P, Shahmehri N, Wahllöf N. A default extension to description logics for use in an intelligent search engine. In: Proceedings of the 31st Hawaii International Conference on System Sciences, Volume V - Modeling Technologies and Intelligent Systems Track, Hawaii, 1998. 28～35

11 Sebastiani F, Straccia U. Default reasoning in a terminological logic. Computers and Artificial Intelligence, 1995, 14(3): 225～251



DONG Ming-Kai, born in 1973, Ph. D. candidate. His research interests include intelligent agent, knowledge representation and reasoning, Web intelligent information processing.

JIANG Yun-Cheng, born in 1974, Ph. D. candidate. His research interests include intelligent agent, multi-agent system, intelligent Web.

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