

基于描述逻辑的特征模型^{*})

范少锋 张乃孝 赵新宇

(北京大学信息科学系 北京 100871)

摘要 由于特征模型缺乏严格语义基础,难以对模型进行一致性推理,无法保证模型的一致性,进而影响最终软件产品的质量。本文讨论了如何使用描述逻辑对特征模型进行形式化,给出了将特征模型转化为描述逻辑 ALCQI 知识库的算法,通过验证转化后的知识库的一致性得到所对应特征模型的一致性。并利用描述逻辑的推理机 RACER 对转换得到的知识库进行推理,自动完成特征模型的一致性检查,为特征模型的一致性检验提供一种方法,实践证明这种方法具有可靠性和高效性。

关键词 描述逻辑,特征模型,一致性,RACER

Feature Model Based on Description Logics

FAN Shao-Feng ZHANG Nai-Xiao ZHAO Xin-Yu

(Department of Information Science, Peking University, Beijing 100871)

Abstract It is rather difficult to perform consistency reasoning on feature models since they lack the formal semantics. Without guaranteed consistency of the feature models, the quality of the software products, based on the feature models, can not be guaranteed. In this paper, how to formalize feature models with Description Logics is investigated. Following the translation principles, each feature model can be formalized into an ALCQI knowledge base. Hence the consistency reasoning on the feature model turns into the consistency reasoning on the corresponding ALCQI knowledge base. Especially, the latter reasoning can be automatically performed via the description logic reasoner RACER. It has been proven that this approach can not only improve the reasoning efficiency but also enhance the reliability.

Keywords Description logic, Feature model, Consistency, RACER

1 引言

描述逻辑(Description Logic)起源于 20 世纪 70 年代的知识表示方法语义网络(semantic network)和框架(frame)系统,它是一种用于知识表示的形式化方法^[1]。在众多知识表示的形式化方法中,描述逻辑受到人们特别关注的主要原因在于:具有严格的语义基础;对概念性知识的处理,特别是对概念分层的处理非常有效;提供有效的推理机制,支持可判定的推理服务。

目前国内外许多学者对描述逻辑进行了深入研究,特别是描述逻辑的应用研究。例如:以描述逻辑理论为依据,研究语义 Web 本体语言的形式语义^[2,3];研究基于描述逻辑的主体服务匹配机制^[4];研究基于描述逻辑的数据库的实体-关系 ER 模型^[1,5];利用描述逻辑来形式化基于角色的访问控制模型^[6]等。

特征和特征模型的概念源于面向特征的领域分析方法(FODA)^[7]。领域特征模型通过记录领域具有的特征以及特征之间的关系反映整个领域的软件需求^[8],它是一种对软件需求进行组织的有效方式^[9]。由于具有结构简单、表达直观、支持复用等优点,特征模型作为捕获领域需求的重要模型,已被主流的领域工程方法所接受^[10]。现阶段已有多种领域工程方法以特征模型为中心来组织领域模型,包括 ODM^[11]、KAPTUR^[12]、FORM^[13]、FeatuRSEB^[14]和 PLA^[15]等。

软件需求是软件生命周期的起点,而对于以开发可重用软件为目标的领域工程方法来说,捕获领域需求的特征模型尤为重要。特征模型构成基于重用的领域软件开发的基础^[9]。因此,特征模型的一致性将直接影响到最终软件产品

的质量。但是由于特征模型的结构元素缺乏严格的语义基础,目前在特征建模时,设计者通常需要对一致性问题进行手工推理,导致推理效率和可靠性不高。特别是对于复杂的领域,特征模型中将包含较多的特征和约束,难以通过手工来验证模型的一致性。

针对以上问题,本文提出使用描述逻辑 ALCQI 来形式化特征模型,给出将特征模型转化为 ALCQI 知识库的方法,从而将特征模型的一致性问题转化为 ALCQI 知识库的一致性问题。利用描述逻辑有效的推理机制,通过描述逻辑推理机 RACER^[16]可自动实现知识库的一致性检查,也即自动验证特征模型的一致性,大大提高推理的效率和可靠性,为基于特征模型软件开发奠定了基础。

2 图形化的特征模型

所谓特征,是被关注概念的显著性质^[11]。一个概念的特征模型表示了该概念的内涵(intention),此模型所表示的实例的集合叫做概念的外延(extension)。特征模型包括两部分:特征图和关于特征的约束信息,例如特征之间的静态依赖关系等。对于一个特征模型,如果此模型所表示的实例的集合非空,也即存在满足其特征图和特征约束的概念实例,则称此特征模型是一致的。下面对特征模型作简要介绍,有关特征模型的更详细内容可参考文[8]。

2.1 特征图

特征图是特征模型的主要部分。一个特征图包含三类元素:一组结点、一组有向边和一组边的标记。特征图的结点和有向边构成一个树状结构,其根结点叫概念结点,表示一个概

^{*})本课题得到国家自然科学基金项目(编号 60473056)资助。范少锋 博士研究生,主要研究方向为软件自动化、形式化方法;张乃孝 教授,博士生导师,主要研究方向为软件方法学;赵新宇 博士研究生,主要研究方向为 Agent 技术、人工智能。

念,其他结点表示特征,叫做特征结点。有向边从父亲结点指向其子结点。边的标记用弧线连接源自同一个结点的多条有向边来表示。

下面将结合简化的汽车概念的特征图(见图1)简单介绍特征图的基本记法。

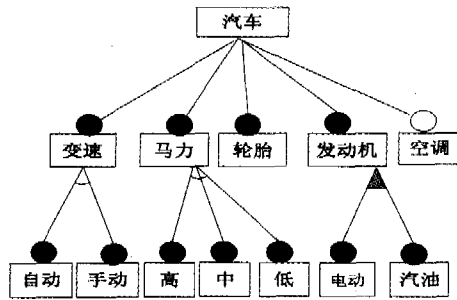


图1 简化的汽车概念对应的特征图

1)必需(mandatory)特征:由以实心点结束的简单边(为区别于带弧线标记的边,称那些不带弧线标记的边为简单边。)指向的特征结点表示,表示该特征包含于概念的实例当且仅当该特征结点的父结点包含于此实例,如图1中的“发动机”、“马力”等都是“汽车”的必需子特征。

2)可选(optional)特征:由以空心点结束的简单边指向的特征结点表示,表示该特征可能包含于某概念的实例当且仅当该特征结点的父结点包含于此实例,如图1中的“空调”是可选特征。

3)异或(alternative)特征:由弧线连接的一组边所指向的特征结点表示,表示如果一组异或特征结点的父亲结点包含于概念的实例,那么此组特征中仅有一个特征结点包含于概念的实例,如图1中的“自动”和“手动”为“变速”的异或(子)特征。

4)或(or)特征:由填充弧线连接的一组边所指向的特征结点来表示,表示如果一组或特征结点的父亲结点包含于概念的实例,那么此组特征中至少有一个特征结点包含于概念的实例,如图1中所示的“电动”和“汽油”为“发动机”的或(子)特征。

通过图1这个特征图,可以直观地描述:2(变速) * 3(马力) * 3(发动机) * 2(空调)=36种不同的汽车实例。

2.2 特征间的约束

特征图清晰地描述了概念的各个实例之间的共同点和不同点,其共同点由必需特征来体现,不同点(也即可变点)通过可选特征、异或特征、或特征来体现。但在描述实际应用领域的时候,并非任意的特征组合都具有实际意义。通过描述特征的其他信息,如添加特征间的依赖关系,来对特征图进行补充性描述,限制特征的组合,可以使特征模型更能准确地反映领域需求。本文主要考察三种类型的约束:

1)前提关系:记为 $F_1 \text{ require } F_2$,表示特征 F_1 出现在概念实例中的前提是特征 F_2 出现在概念实例。

2)互斥关系:记为 $F_1 \text{ exclude } F_2$,表示特征 F_1 和 F_2 不能同时被包含在概念实例中。

3)数量限制关系:表示特征之间的数量关系,例如一辆汽车对应4个轮胎。

对于图1表示的汽车概念,如果在此基础上再添加约束“空调 require 高马力”,则在此约束限制下,汽车实例将由原来的36种减少为 $2 * 3 + 2 * 3 * 3 = 24$ 种。

特征模型独立于实现,在较高的抽象层次上为用户提供了一种对应用领域的显式表示。其结构简单、直观,但也存在

如下缺点:当包含较多的特征时,特征图的结构将过于繁琐,失去其直观的优点;缺乏严格的语义基础,一般情况下建模者需要手工对特征模型的一致性问题进行推理,导致推理效率和可靠性不高,特别当特征模型包含较多的特征和约束时,模型的一致性很难通过手工进行检验。而描述逻辑具有严格的语义基础,提供很有效的推理机制,并支持可判定的推理服务。因此,下面我们提出使用描述逻辑来形式化特征模型。

3 描述逻辑 ALCQI

描述逻辑的基本构件包括概念(concept)和关系(role)。原子概念和原子关系是描述逻辑的基本元素,概念和关系可以通过构造符(constructor)在原子概念和关系的基础上构造出来。为了形式化的特征模型,本文选取具有较强表达能力的描述逻辑语言 ALCQI。下面简要介绍 ALCQI 的语法、语义和知识库推理相关的概念。

3.1 ALCQI 的语法

ALCQI 的语法形式如下:

- $C, D \rightarrow T$ | (全概念)
- A | (原子概念)
- $\neg C$ | (概念取非)
- $C \sqcap D$ | (概念合取)
- $C \sqcup D$ | (概念析取)
- $\exists R. C$ | (存在限制)
- $\forall R. C$ | (任意限制)
- $\geq n R. C$ | (最小数量限制)
- $\leq n R. C$ | (最大数量限制)
- $R \rightarrow P$ | (原子关系)
- P^- | (关系的逆)

其中, C, D 表示概念, R 表示关系。这里 n 表示自然数。为了提高可读性,我们用 \perp 表示 $\neg T \geq n R. C$, 用 $= n R. C$ 表示 $\geq n R. C \sqcap \leq n R. C$, 用 $\geq n R$ 表示 $\geq n R. T$ 。

3.2 ALCQI 的语义

描述逻辑具有清晰严格的语义,在描述逻辑中,概念解释成论域 Δ 的一个子集,关系解释成 $\Delta \times \Delta$ 上的二元关系。形式上, $I = \{\Delta, \cdot^I\}$ 是一个解释,其中 Δ 是一个非空的个体集合,叫做这个解释的论域, \cdot^I 叫做解释函数。描述逻辑 ALCQI 的语义如表1所示。

表1 描述逻辑 ALCQI 的语义

语法	语义
T	Δ^I
A	$A^I \subseteq \Delta^I$
$\neg C$	$\Delta^I \setminus C^I$
$C \sqcap D$	$C^I \cap D^I$
$C \sqcup D$	$C^I \cup D^I$
$\forall R. C$	$\{a \in \Delta^I \mid \forall b. (a, b) \in R^I \rightarrow b \in C^I\}$
$\exists R. C$	$\{a \in \Delta^I \mid \exists b. (a, b) \in R^I \wedge b \in C^I\}$
$\geq n R. C$	$\{o \in \Delta^I \mid \{o' \mid (o, o') \in R^I \wedge o' \in C^I\} \geq n\}$
$\leq n R. C$	$\{o \in \Delta^I \mid \{o' \mid (o, o') \in R^I \wedge o' \in C^I\} \leq n\}$
P^-	$\{(o, o') \in \Delta^I \times \Delta^I \mid (o', o) \in P^I\}$

其中, $|S|$ 表示集合 S 的势。

3.3 知识库

描述逻辑的知识库 KB 包含两部分: TBox 和 ABox。其中 TBox 是术语公理(terminology axiom)的集合,用来表示领域中的一般性知识,包括领域中的概念以及概念之间的关系。TBox 具有两种形式: $C \sqsubseteq D$ 和 $C \equiv D$, 其中 C, D 均为概念,前

者表示概念包含,后者表示概念等价。概念 C 包含于概念 D , 当且仅当对 TBox 的所有解释 I 都有 $C^I \subseteq D^I$; 概念 C 等价于概念 D 当且仅当对 TBox 的所有解释 I 都有 $C^I \equiv D^I$ 。显然, $C \equiv D$ 当且仅当 $C \sqsubseteq D$ 且 $D \sqsubseteq C$ 。ABox 是断言公理(assertion axiom)的集合,它是关于个体的断言,用来指明个体所属的概念以及个体之间的关系。

描述逻辑中的两种最基本的推理形式为可满足性(satisfiability)和内含(subsumption),其他形式的推理都能够归为这两种形式。一个概念 C 在知识库 KB 中是可满足的,当且仅当存在一个 KB 的解释 I ,使得 $C^I \neq \emptyset$,此时,称 I 是 C 的一个模型。一个概念 C 内含于另一个概念 D ,当且仅当对 KB 的所有解释 I 都有 $C^I \subseteq D^I$ 。对于知识库 KB,如果存在解释 I 使得 TBox 中的概念都可满足、ABox 的断言都成立,则称此知识库 KB 是一致的,此时称 I 是 KB 的一个模型。

下面将介绍如何将特征模型转化为 ALCQI 的知识库,从而利用 ALCQI 的推理机制对特征模型的一致性等问题进行自动推理。

4 基于描述逻辑的特征模型

为了利用描述逻辑 ALCQI 的推理机制对特征模型的一致性进行推理,首先需要将特征模型转化为 ALCQI 的知识库,将特征模型的一致性问题转化为相应知识库的一致性问题。

4.1 特征模型与 ALCQI

给定包含特征图 FD 和相关的特征约束信息的特征模型 FM ,与 FM 对应的 ALCQI 的知识库 $K = \varphi(FM)$ 按下列规则得到:

- ①特征图 FD 中的任意结点 V 对应为 ALCQI 概念 C ;
- ②假设结点 V 已经转化为 ALCQI 概念 C ,其子结点 V_i ($i \geq 1$) 分别转化为 ALCQI 概念 C_i ,对于 V 和其子结点 V_i 之间的各有向边分别对应 ALCQI 关系 R_i ,表示 C 和 C_i 之间的关系,而 C_i 和 C 之间的二元关系用 R_i^- 表示;

③给出特征图的结点和有向边的转化规则后,初步得到了 ALCQI 知识库所包含的概念和关系。下面讨论如何将特征图的基本结构(其他结构都可以转化为这几种基本结构。另外不失一般性,为了描述简洁,这里仅讨论含两个子特征的结构,对于含多个子特征的结构可做类似讨论。)(见图 2)转化为 ALCQI 知识库的公理:

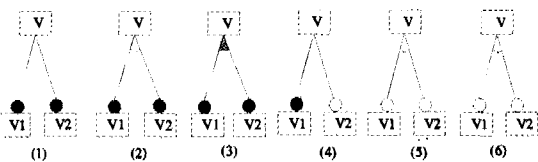


图 2 特征图的基本结构

1)结构(1)记为 $all(V_1, V_2)$,表示 V_1, V_2 都是 V 的必需子特征,对应于此结构向知识库中引入术语公理:

$$C \sqsubseteq \forall R_1. C_1 \sqcap \forall R_2. C_2;$$

2)结构(2)记为 $one-of(V_1, V_2)$,表示 V_1, V_2 是 V 的异或子特征,对应于此结构向知识库引入术语公理:

$$C \sqsubseteq C_1 \sqcup C_2,$$

$$C_1 \sqsubseteq C \text{ 和 } C_2 \sqsubseteq C \sqcap \neg C_1$$

其中,后面两个公理保证 C_1 和 C_2 两个概念是不相交的。

3)结构(3)记为 $more-of(V_1, V_2)$,表示 V_1, V_2 是 V 的

或子特征,该结构等价于

$$one-of(V_1, V_2, all(V_1, V_2))$$

因此可利用结构(1)、(2)的转化规则向知识库中引入如下术语公理:

$$C \sqsubseteq C_1 \sqcup C_2 \sqcup (\forall R_1. C_1 \sqcap \forall R_2. C_2),$$

$$C_1 \sqsubseteq C, C_2 \sqsubseteq C \sqcap \neg C_1$$

$$(\forall R_1. C_1 \sqcap \forall R_2. C_2) \sqsubseteq C \sqcap \neg (C_1 \sqcup C_2)$$

4)结构(4)记为 $all(V_1, V_2?)$,其中 $V_2?$ 表示 V_2 是一个可选子特征,而 V 包含必需子特征 V_1 和可选子特征 V_2 ,该结构等价于

$$one-of(V_1, all(V_1, V_2?)),$$

可以利用结构(1)、(2)的转化规则仿照步骤 3)向知识库中引入术语公理。

5)结构(5)记为 $one-of(V_1?, V_2?)$,表示 V 包含两个异或可选子特征 V_1 和 V_2 ,该结构等价于

$$one-of(Null, V_1, V_2)$$

其中,Null 为新引入的 ALCQI 原子概念,用于表示可选特征都不出现在概念实例中的情况。利用结构(1)、(2)的转化规则仿照步骤 3)向知识库中引入术语公理:

$$C \sqsubseteq C_1 \sqcup C_2 \sqcup Null,$$

$$C_1 \sqsubseteq C, C_2 \sqsubseteq C \sqcap \neg C_1$$

$$Null \sqsubseteq C \sqcap \neg (C_1 \sqcup C_2)$$

说明:这里引入的原子概念 Null 不同于 ALCQI 中的空概念 \perp ,因为 $\perp^I = \emptyset$,而 $Null^I \neq \emptyset$ 。

6)结构(6)记为 $all(V_1?, V_2?)$,表示 V 包含两个可选子特征 V_1 和 V_2 ,该结构等价于:

$$one-of(Null, V_1, V_2, all(V_1, V_2?))$$

可仿照步骤 5)进行转化。

④对于特征模型中特征间的约束信息,依次做如下转化:

- 1)约束 V_1 require V_2 ,表示如果 V_1 出现在概念实例中,则 V_2 必然出现在概念实例中。为了描述此约束,可先按下算法(表 2)得到从 V_1 和 V_2 的最近公共祖先结点(V_1 和 V_2 的公共祖先结点必然存在,显然根结点必然是一个公共祖先结点。所谓最近,是指在 V_1 和 V_2 上溯过程中到达的第一个公共祖先结点。)到达 V_1 经过的有效的有向边的信息 $E_{11} \dots E_{1i}$ 。

表 2 一种算法

//定义 temp 为中间结点变量
// parent(temp)表示 temp 的父结点
//Common V 表示 V_1 和 V_2 最近公共祖先结点
temp= V_1
while(temp! =CommonV)
{ if (temp 与其兄弟结点为 all 结构)
then 记录 temp 父结点与其有向边所对应的原子关系 E_{1i}
else
temp=parent(temp)
}

对于 V_2 可类似得到有效的有向边的信息 $E_{21} \dots E_{2i}$ 。然后对此约束可引入如下术语公理:

$$(\forall R_{11} \dots (\forall R_{1i}. C_1) \sqsubseteq (\forall R_{21} \dots (\forall R_{2i}. C_2))$$

其中, R_{11}, \dots, R_{1i} 是分别与有向边 $E_{11} \dots E_{1i}$ 对应的原子关系, R_{21}, \dots, R_{2i} 是分别与有向边 $E_{21} \dots E_{2i}$ 对应的原子关系。

2)约束 V_1 exclude V_2 ,表示如果 V_1 出现在概念实例中,

则 V_2 必然不出现在概念实例中,反之亦然。对此结构可按 *require* 约束中的算法引入以下术语公理:

$$(\forall R_{11} \dots (\forall R_{1i} \dots C_1)) \sqcap (\forall R_{21} \dots (\forall R_{2j} \dots C_2)) \sqsubset \perp$$

其中, C 为 V_1 和 V_2 的最近的公共祖先结点对应的 ALCQI 概念,此公理表示同时包含特征 V_1 和 V_2 的概念实例所对应的 ALCQI 概念内含于空概念,说明既包含 V_1 又包含 V_2 的概念实例不存在。

3) 对于特征间的数量限制关系,通过使用 ALCQI 数量限制构造符,转化 ALCQI 术语公理。

上述原则涵盖了第 2 节中讨论的特征模型的所有元素,实际上给出了将特征模型转化为 ALCQI 知识库的算法。

对于转化函数 φ 的正确性,与文[5]类似,可通过建立两个映射,即从特征模型 FM 的概念实例到对应知识库 $\varphi(FM)$ 的模型的映射和从 $\varphi(FM)$ 的模型到 FM 的概念实例的映射,来证明上述转化的正确性。但由于此处给出的转化规则比较直观,因此本文省略转化的正确性证明的部分,更多细节可参考文[5]。

说明:由于特征模型不涉及个体实例,所以特征模型转化得到的 ALCQI 知识库不含 ABox,因此本文的讨论限于知识库的 TBox。

4.2 例子

为了理解如何将一个特征模型转化为 ALCQI 的知识库,下面举一个课程概念的简单例子。

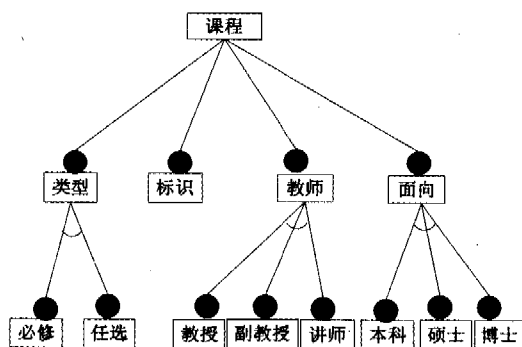


图3 简单的课程概念对应的特征图

图3是课程概念的特征图,其中课程标识、课程类型、任课教师和修读对象是课程的必需特征。而必修和任选是类型的异或子特征;教授、副教授、讲师是教师的异或子特征;修读对象分为本科生、硕士生、博士生三类。

另外,假设存在以下几点特征的约束:

- 1)“必修”*exclude*“讲师”(为了保证课程的质量,要求所有必修课需要具有副教授以上职称的教师讲授);
- 2)“博士”*require*“教授”(为了保证课程内容具有一定的深度,要求面向博士生的课程必须由具有教授职称的教师讲授);
- 3)每门课程只能由一位教师讲授(为了避免课程的重复开设);
- 4)一位教师至多讲授一门课程(为了保证课程任务分配的公平性)。

例如,在条件1)的约束限制下“必修”和“讲师”的组合是非法的,遵循条件2)“博士”和“讲师”的组合也是非法的。

为了提高可读性,下文中我们采用以下约定:ALCQI 概念的名字都以“C”开头,ALCQI 关系的名字都以小写字母开头。根据4.1节的转化规则,可以将上述特征模型转化为

ALCQI 知识库,如图4所示。

5 使用 RACER 推理

将特征模型 FM 转化为 ALCQI 知识库 $\varphi(FM)$, FM 的每个实例对应了 $\varphi(FM)$ 的模型。若特征模型 FM 一致,则 FM 的概念实例的集合非空,也即必然存在满足特征模型 FM 的概念实例,对应地, $\varphi(FM)$ 的模型必然存在,也即知识库 $\varphi(FM)$ 是一致的。反之亦然。因此有以下定理成立。

表3 RACER 脚本

(in-box course)
(signature :
:atomic-concepts
(CCourse CKind CId CTeacher COrient CRequired CLimited CArbitrary
CProf CAssoProf CTA CUnGrad CGrad CPhd)
:roles(
(kind :domain CCourse;range CKind)
(id :domain CCourse;range CId)
(taughtby;inverse taught;domain CCourse ;range CTeacher)
(orient :domain CCourse;range COrient)
(required :domain CKind;range CRequired)
(arbitrary;domain CKind;range CArbitrary)
(prof :domain CTeacher;range CProf)
(assoprof ;domain CTeacher;range CAssoProf)
(ta :domain CTeacher;range CTA)
(ungrad :domain COrient;range CUnGrad)
(grad :domain COrient;range CGrad)
(phd :domain COrient;range CPhd)
))
(equivalent CCourse (and (all kind CKind) (and (all id CId) (and
(all taughtby CTeacher) (all orient COrient))))
(implies CKind (or CRequired CArbitrary))
(implies CRequired CKind)
(implies CArbitrary and (CKind (not CRequired)))
.....
(implies and (all kind CRequired)(all taughtby CTA)) * bottom *)
(implies (all orient CPhd)(all taughtby CProf))
(implies CCourse (exactly 1 taughtby CTeacher))
(implies CTeacher (at-most 1 taught CCourse))

定理 给定一个特征模型 FM , $\varphi(FM)$ 是 FM 通过转化得到的 ALCQI 知识库,则特征模型 FM 是一致的,当且仅当 $\varphi(FM)$ 是一致的。

此时,特征模型的一致性问题就转化为相应知识库的一致性问题。

本文使用描述逻辑推理机 RACER(RACER 是实现了描述逻辑 ALCQHIR+ 的知识表示系统,它基于唯一假设,支持对 TBox 和 ABox 的各种推理,并可以处理多个 TBoxes 和 ABoxes。)(Renamed ABox and Concept Expression Reasoner) 自动实现知识库的一致性检查;基于 RACER 的语法,定义相应的 RACER 脚本文件;使用 RACER 图形化界面(RACER Interactive Client Environment, RICE)作为 RACER 的客户端,将脚本文件导入到 RACER 知识库中;在 RACER 中对知识库进行各种自动推理,这里我们关注的是知识库(也即 TBox)的一致性推理。

针对4.2节的课程概念的例子,根据已经定义的知识库,我们给出表3所示的 RACER 脚本文件。

CCourse $\sqsubseteq \forall$ kind, CKind $\sqcap \forall$ id, CId $\sqcap \forall$ taughtby, CTeacher $\sqcap \forall$ orient, COrient	//“课程”的各必需子特征
CKing \sqsubseteq CRequired \sqcup CArbitrary	//“类型”的异或子特征
CRequired \sqsubseteq CKind CArbitrary \sqsubseteq CKind $\sqcap \rightarrow$ CRequired	
.....	//“教师”、“面向”的异或子特征表示与“类型”类似
$(\forall$ kind, CRequired) \sqcap $(\forall$ taughtby, CTA) $\sqsubseteq \perp$	//“必修” exclude “讲师”
\forall orient, CPhd $\sqsubseteq \forall$ taughtby, CProf	//“博士” require “教授”
CCourse \sqsubseteq $(=1$ taughtby, CTeacher)	//每门课程只能由一位教师讲授
CTeacher \sqsubseteq $(\leq 1$ taughtby, CCourse)	//一位教师至多讲授一门课程

图 4 课程概念的特征图转换后的知识库片断

使用 RICE 作为客户端,将上述脚本文件导入 RACER 知识库中,然后输入一致性检查的命令:

```
(check-tbox-coherence course (tbox (current-tbox)))
```

结果返回 Nil,表示当前知识库中所有概念都是可满足的,也即当前知识库是一致的。相应地,当前的特征模型是一致的。

如果在原有的特征模型上再添加新的约束“要求每位教授至少讲授 2 门课程”,也即向原 TBox 中添加公理:

```
(implies CProf (at-least 2 taught CCourse))
```

更新 RACER 脚本后,再进行一致性检查,结果如图 5 所示,表示当前知识库不一致,从而检查出目前的特征模型不一致。特别地,RACER 还指出问题所在:概念 CProf 不满足。此时,返回检查特征模型中对应于概念 CPro 的特征的相关约束,发现新加入的约束“要求每位教授至少讲授 2 门课程”与“一位教师至多讲授一门课程”间存在冲突。通过修改约束,可保持特征模型的一致性。

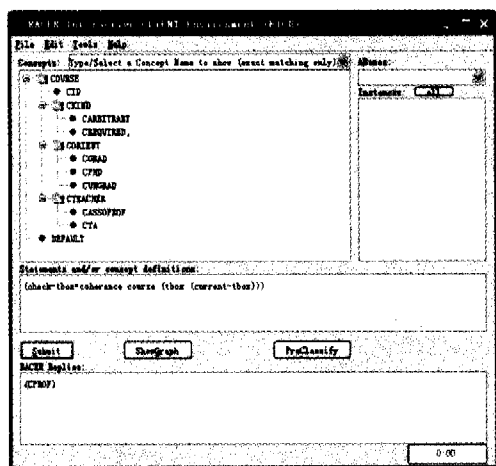


图 5 使用 RACER 进行一致性推理

显然,使用推理机 RACER 来对 ALCQI 知识库进行自动推理,大大提高了推理效率。同时,基于逻辑的严格推理保证了推理的可靠性。这也是本文研究基于描述逻辑的特征模型的动机和背景。

结论 特征模型由于具有结构简单、表达直观、支持复用等优点,已经成为捕获领域需求的重要模型。目前,多种领域工程方法都采用了面向特征的领域需求组织方式。但由于特征模型缺乏形式化语义基础,很难严格地保证模型的一致性,因此如何检验特征模型的一致性就成为一个值得研究的重要问题。Griss 提出了使用预定义的 UML 建模元素来实现特征模型的方法^[14],但由于 UML 本身就缺乏严格的形式化语义,因此该方法并不能根本解决特征模型的一致性问题的。

Deursen 定义了用于描述特征的文本式语言 FDL^[16],讨论了约束的可满足性规则,但该方法只适用于简单的可满足性推理,且尚不支持自动推理,因此推理效率低下。

本文提出基于描述逻辑的特征模型,给出了将特征模型转化为 ALCQI 知识库的算法,从而将特征模型的一致性检查转化为 ALCQI 知识库的一致性推理,并使用 RACER 自动完成该推理过程,从而验证了对应的特征模型的一致性。为特征模型的一致性检查提供了一种高效、可靠的方法,对基于特征模型的软件开发实践具有重要意义。

参 考 文 献

- 1 Baader F, Calvanese D, McGuinness D, et al. eds. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge: Cambridge University Press, 2003
- 2 Horrocks I. DAML+OIL: A description logic for the semantic Web. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2002, 25(1): 4~9
- 3 Baader F, Horrocks I, Sattler U. Description logics as ontology languages for the semantic Web. In: Hutter D, Stephan W. eds. Mechanizing Mathematical Reasoning: Essays in Honor of Jörg Siekmann on the Occasion of His 60th Birthday. LNAI 2605, Springer-Verlag, 2005. 228~248
- 4 史忠植,蒋运承,张海俊,等. 基于描述逻辑的主体服务匹配. 计算机学报, 2004, 27(5): 625~635
- 5 Calvanese D, Lenzerini M, Nardi D. Unifying class-based representation formalisms. Journal of Artificial Intelligence Research, 1999, 11(2): 199~240
- 6 Chen Zhao, Nuermaiti H, Liu Shengping, et al. Representation and Reasoning on RBAC: A Description Logic Approach. In: Proceedings of the International Colloquium on Theoretical Aspects of Computing (ICTAC 2005), LNCS 3722, Springer-Verlag, October 2005. 394~406
- 7 Kang KC, Cohen SG, Hess JA, et al. Feature-Oriented Domain Analysis (FODA) feasibility study: [Technique Report]. CMU/SEI-90-TR-21. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1990
- 8 张伟,梅宏. 一种面向特征的领域模型及其建模过程. 软件学报, 2003, 14(8): 1345~1356
- 9 Czarnecki K, Eisenecker U. Generative Programming: Methods, Tools, And Applications. Addison-Wesley, 2000
- 10 范少锋,张乃孝. 生成式程序设计研究概述. 计算机科学, 2005, 32(3): 12~16
- 11 Simos M, Creps D, Klinger C, et al. Organization Domain Modeling (ODM) Guidebook, Version 2. 0. Informal: [Technical Report]. STARS, 1996
- 12 Bailin S. Domain Analysis with KAPTUR. Tutorial of TRIAda' 93, Vol. I, ACM, NewYork, NY, September 1993
- 13 Kang KC, Kim S, Lee J, et al. FORM: A feature-oriented reuse method with domain-specific architecture. Annals of Software Engineering, 1998, 5: 143~168
- 14 Griss ML, Favaro J, d' Alessandro M. Integrating feature modeling with the RSEB. In: Devanbu P, Poulin J. eds. Proceedings of the 15th International Conference on Software Reuse. Victoria: IEEE Computer Society, 1998. 76~85
- 15 Chastek G, Donohoe P, Kang KC, et al. Product line analysis: a practical introduction: [Technical Report]. CMU/SEI-2001-TR-001. Pittsburgh: Carnegie Mellon University, Software Engineering Institute, 2001. 1~42
- 16 Haarslev V, Möller R. RACER Users Guide and Reference Manual, 2004
- 17 van Deursen A, Klint P. Domain-Specific Language Design Requires Feature Descriptions. Journal of Computing and Information Technology, 2002