TCRA:一种结合本体概念的数据推理算法

卢桂芳 汪璟玢
(福州大学数学与计算机科学学院 福州 350108)

摘 要 OWL DL (DL即Description Logic)使用类运算式来表示概念。任意复杂的概念可用集合的运算来表示。当前OWL DL的集合运算中只能使用交集、并集和补集,而对于有数量约束的集合运算式,OWL无法进行表示。在OWL DL的基础上对集合基数限制进行扩展,实现数量约束的描述逻辑。在集合基数扩展的基础上提出一种树型分类推理算法(Tree Classify Reasoning Algorithm),并通过人力资源筛选的应用证明此方法的表达灵活性和分类有效性。

关键词 本体,OWL,描述逻辑,基数限制,推理
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.12.051

TCRA:A Data Reasoning Algorithm Combined with Ontology Concept

LU Gui-fang WANG Jing-bin
(College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China)

Abstract A class description is used to represent the concept in OWL DL (Description Logic). Collection operation in theory can lead to arbitrarily complex concept. The current collection operation of OWL DL can be achieved with intersection, union and complement, and OWL can not represent set expression with a number of constraints. This paper extended the cardinality constraint on the basis of OWL DL set limit, which can achieve the description logic with number of set constraints. And we proposed a tree classify reasoning algorithm based on collection cardinality. The application of human resources screening proves that this method is flexible in expression and very effective in individual classification.

Keywords Ontology, OWL, Description logic, Cardinality constraint, Reasoning

1 引言

目前,随着Web的广泛普及,Web中知识的可重用性和可复用性尤为重要。本体通过对特定领域知识的形式化规范说明和层次化描述,解决了语义Web信息的共享与交换。当前,越来越多的本体采用基于描述逻辑(Description Logic)的本体语言,如OWL DL进行表达[1]。这样,本体既能对领域知识进行合理的表达,形成领域本体,又能利用DL的推理服务支持有效地推理。描述逻辑通过定义应用域相关的概念并运用这些概念描述该应用域中个体的特性与个体之间的关系。Web中具有大量模糊信息,经典描述逻辑只能表示精确知识。为了解决本体模型基础上语义Web中的不精确性和不确定性的知识表示的限制,已有大量的研究工作致力于OWL和描述逻辑的模糊扩展。王海龙等人[2]提出了F-SHOZQ的自定义模糊数据类型扩展F-SHOIQ(G),用以支持含有自定义模糊数据类型及自定义模糊数据类型谓词的模糊数据信息;程经纬等人[3]提出针对f-SH这样一族极富表达能力的模糊DL知识库模糊合取查询蕴涵判定算法。在国外也有一些研究工作,Stoilos建立了模糊扩展语义f-OWL,提出了OWL基于模糊包容和模糊名词另外两个模糊扩展[4]以及推理模糊描述逻辑F-SHOIQ和f-SROIQ的算法[5]。Bobillo[6]提出了Delorean,其支持标准语言OWL和OWL2模糊扩展,以及从模糊粗糙本体语言到经典本体语言SROIQ(D)的转换。但是,上述模糊描述逻辑都是以模糊集合作为集合论的基础,并不能处理描述逻辑中带基数限制的不精确集合运算,在数量约束方面也有一些研究,如李言辉等人[7]提出了一种支持数量约束的扩展模糊描述逻辑FALCN。然而上述工作仅在描述逻辑的层面上,在OWL本体建模中,并没有针对OWL本体语言对数量约束的集合运算进行扩展。OWL DL中复杂的类可由基本的类、属性、个体通过基本的类操作创建,即并、交和补集运算,这些类操作可以表达集合关系明确的类。在许多领域上,基于不精确集的数据挖掘、决策分析的应用变得越来越普遍,如在人员筛选的运用中,往往列出一系列的条件,而要求筛选的人员只需满足所有条件中的几个以上即可。用OWL DL来描述由某几个特征属性的析取组成的概念,仅通过并、交和补集运算是无法精确表达

到稿日期:2013-06-28 返修日期:2013-08-20 本文受福州大学科技发展基金项目(2013-XQ-32),空间数据挖掘与信息共享教育部重点实验室开放研究基金项目(201006),2011年福建省科技拥军基金项目(JG2011005),福建省自然科学基金项目(2012J01168)资助。
卢桂芳(1990—),女,硕士生,主要研究领域为海量数据管理、智能技术,E-mail:562964694@qq.com;汪璟玢(1973—),女,硕士,副教授,主要研究领域为海量数据管理、网络数据库和智能技术,E-mail:wjbcc@263.net(通信作者)。

的。本文在本体建模的基础上，通过增强本体语言的描述逻辑，将基数限制集合(CardinaltySets)引入其中，从而使得本体语言能够清晰地表达复杂不精确集合概念知识；并在扩展集合基数限制的基础上进行推理研究，实现对模糊概念的个体分类。

本文第2节对OWL、描述逻辑与推理机等相关知识进行介绍；第3节介绍TCRA算法的数据结构和算法的实现过程；第4节在标准数据集和人力资源数据集上对本文方法进行运用，并对结果进行分析；最后进行总结。

2 相关基础知识

2.1 OWL DL

OWL是一个语义标记语言，用于万维网上发布和共享本体。OWL主要由个体(Individual)、属性(Property)和类(Class)3部分组成。个体是类的实例，属性是个体之间的二元关系，属性有两类，一类是对象属性，它用于将个体与个体关联起来，另一类是数据类型属性，它把个体与数据值关联起来。OWL的逻辑基础是描述逻辑，OWL在保证更加强大的语义表达能力的同时，还要保证描述逻辑的可判定推理[8]。

描述逻辑通过定义领域中相关概念表示该领域的知识，并使用这些概念指出该领域内对象和个体的性质[9]。概念(Concept)、角色(Role)和个体常量(Individual Constant)是构建描述逻辑的基础。概念是用来描述对象的集合，如“学生”。角色定义了二元关系，如“朋友”。个体常量是对个体的简单命名，如“张三”。从OWL数据建模的观点看，概念对应OWL的类，角色对应OWL的属性。描述逻辑最显著的特征是其内置的推理机制，利用这一机制，可以由知识库中显式存储的知识推导出隐含的信息。ALC(Attributive Concept Description Language with Complements)是最基本的一种描述逻辑语言，包含交(⋂)、并(⋃)、非(¬)，以及量词约束：存在量词(∃)和全称量词(∀)[8]。本体描述语言OWL DL和描述逻辑语言在语法上存在一定的对应关系，如表1所列。

表1 OWL DL的构造算子和描述逻辑语法的对应

OWL构造算子	描述逻辑语法
unionOf	C⋃D
intersectionOf	C⋂D
complementOf	¬A
someValueFrom	∃R.C
allVaulesForm	∀R.C
maxCardinality	≥nR.C
minCardinality	≤nR.C
subclassOf	C⊆D
equivalentClass	C=D
disjoinWith	C⊆¬D
sameIndividualAs	{x}={y}
differentFrom	{x}=¬{y}
subPropertyOf	P1⊆P2
equivalentProperty	P1=P2

交集、并集和补集是描述逻辑中使用的更高级的类构造器。对于n个原子概念$C[f_1,f_2,f_3,\cdots,f_n]$，以及每个原子概念的实例集$R[I_1,I_2,I_3,\cdots,I_n\}$，其中$I_i(i=1\sim n)$为原子概念$i$的实例集合，$C_n^k(k=1\sim n)$为$n$原子概念中满足$k$个原子概念的实例集合。根据这3类运算式的类构造方式，我们有如下定义。

定义1 给定一个概念C，其由n个条件概念的交集运算式构成，则概念C包含的实例，是集合中所有概念都包含的个体。即$C_n^n=I_1\cap I_2\cap I_3\cap\cdots\cap I_n$。

定义2 给定一个概念C，其其由n个条件概念的并集运算式构成，则概念C包含的实例，至少存在于集合中其中一个概念包含的个体。即$C_n^1=I_1\cup I_2\cup I_3\cup\cdots\cup I_n$。

定义3 给定一个概念C，其由一个条件概念I的补集运算式构成，则概念C包含的实例，包含那些不是补集概念的个体。即$C=\neg I$。

由基本集合构造的类需要明确在哪些集合范围内或范围外，如果要描述n个概念中至少满足k个概念的个体集合，则需要对集合运算进行扩展。

OWL DL兼顾了语言的表达能力与计算的完备性和可判定性，因此本文针对OWL DL进行基于集合运算的基数限制扩展。

2.2 推理机

描述逻辑作为语义Web的逻辑基础，使其具备了可推理的性质。描述逻辑的推理要依靠描述逻辑推理机来实现。近年来，对经典描述逻辑推理机的研究已经有较大的进展。FaCT/FaCT++[10]是一种描述逻辑分类器，能用于模态逻辑(Modal Logic)满意度测试。RacerPro[11]是一个商业推理机，有更强大的功能，支持系统资源感知计算。Pellet[12]是另一个实用的推理机，它更能体现语义Web许多重要的特色，如个体推理、查询能力、语义支持等。Jena是由HP实验室开发的一组开源模块推理引擎，用于支持语义网的应用，支持的主要推理方式有：OWL推理、RDFS推理等。上述所有的推理机都工作在本体的分类结构上，依据它们可推出新的知识。

3 树型分类推理算法TCRA

类提供了一种根据相似的特征将资源分类的抽象机制。每一个OWL类都与一组被称为类外延(class extension)的个体相关联。类外延中的个体被称作类的实例(instance)。许多情况下，使用本体是为了用它进行关于个体的推理，根据OWL定义的类，对RDF数据集中的个体进行分类，如果某个个体的三元组信息符合这个类的约束限制条件，则可推理得到此个体是这个类的实例。对不精确集合的概念进行个体分类，现有的推理机不支持用户定制的基数限制集合运算的推理，而这在数据挖掘的运用上是非常重要的。为了对基数限制集合运算式进行描述，本文提出一种集合基数限制标签，在扩展了基数限制标签的条件下，完成不确定性类的描述，并通过树型分类推理算法(TCRA)完成对个体的分类。

3.1 集合基数限制标签

根据基数限制的集合运算式的性质，有如下定义：

定义4 给定一个概念C，由n个条件概念中k个概念的交集运算式构成，则概念C包含的实例至少存在于其中k个概念都包含的个体。即：

$$S_1^nC_n^k=(I_1\cap S_2^nC_{n-1}^{k-1})\cup(I_2\cap S_3^nC_{n-2}^{k-1})\cup(I_3\cap S_4^nC_{n-3}^{k-1})\cup\cdots\cup(I_{n-k+1}\cap S_{n-k+2}^nC_{k-1}^{k-1})=\bigcup_{i=1}^{n-k+1}(I_i\cap S_{i+1}^nC_{n-i}^{k-1}) \tag{1}$$

其中，$S_{i+1}^nC_{n-i}^0=1(i=1,\cdots,n)$，$I_i\cap 1=I_i(i=1,\cdots,n)$。$S_i^nC_{n-i+1}^k(i=1\sim n,k=1\sim n)$为全集中从$i$至$n$个概念中至少存在于其中$k$个概念都包含的个体集合。

式(1)采用递归的思想，在 n 个概念中满足至少 k 个概念的集合等于满足第一个概念后满足其后 $n-1$ 个概念中 $k-1$ 个概念的实例集合并上第一个概念不满足、满足第二个概念后满足之后 $n-2$ 个概念的实例集合一直并到前 $n-k+1$ 个概念不满足、满足最后 k 个概念的实例集合。

根据以上规则，要在全部实例中筛选出符合 C_n^* 要求的实例有两个方法，方法一是从概念出发，找到每个条件的实例集合，再使用集合的运算得到满足条件的 k 个以上(包括 k)子集。方法二是从实例出发，判断当前实例是否满足 k 个以上(包括 k)概念，如果满足则将此实例输出到结果集。第二种方法省去了中间结果的存储和集合运算，灵活性更高，在这里我们使用第二种方法，具体实现方法将在第 3.2.3 节详细介绍。

在 OWL DL 的语言描述中，交集的标签是 intersectionOf，并集的标签是 unionOf，补集的标签是 complementOf，以及 Cardinalty 的值限制标签。但对于集合的基数限制，OWL 没有进行定义。基数限制集合 C_n^* 的数量约束为 k，本文以 kCardinaltySetsOf 作为 k 集合的基数限制标签。其中 intersectionOf 可以表示为 nCardinaltySetsOf，unionOf 可以表示为 1CardinaltySetsOf，C_n^2 可以表示为 2CardinaltySetsOf。

3.2 个体分类推理过程

针对基数限制集合的运算思想，本文提出了一种基于多叉树结构的分类推理算法 TCRA，即将限制条件转换成一棵推理树，对每个实例对象在树中遍历，从叶子节点开始匹配，如果能达到根节点，则满足这个规则，此个体对象是此概念的实例。在本节中个体分类推理的方法以人力资源筛选的运用为例，以便于对本文方法的理解。

3.2.1 人力资源本体

基于本体的知识库可以按照如下步骤构建[8]：(1)依据学科内容，形成本学科的知识框架，建立相关本体的框架描述；(2)依据本学科的体系框架，列举出所有概念，形成一个词汇表，并对该概念进行解释；(3)提取领域中重要的术语、概念；(4)定义领域中概念及概念之间的关系；(5)给出知识库的标识，建立知识库结构；(6)定义各类本体实例，形成各类知识库。

图1 人力资源本体定义

人力资源知识库中存放的是人力资源领域的知识，由5部分构成：基本信息、教育信息、掌握技能、工作信息、获奖事件。通过对简历文本的分析，创建人力资源本体，本体中包括了相关的实体概念，将简历中的教育经历、工作经历和获奖经历作为事件概念，设计概念之间的关系。创建简历本体中的类和属性(包括对象属性和数据属性)，定义类与类之间的关系、属性与属性之间的关系、类和属性的关系。

本文针对某软件公司用人单位关心的信息，利用 protégé 工具创建了人力资源本体，如图1所示。

3.2.2 匹配规则转化

以软件开发工程师的概念为例，限制条件为(RQ1)：(1)要求应聘者需要熟练或精通至少一门语言(即 C，C++，java 等)；(2)熟练或精通至少一个开发工具；(3)熟练或精通至少一种数据库；(4)获得本科以上学历；(5)毕业于“211”以上学校，具有开发经验，如果是非“211”以上学校需要有至少两年的开发经验。(6)以上条件至少满足4个。RQ1 的定义运用了条件限制和集合运算。其中条件限制包括以下几个：

equivalentClass：等价类，即两个类具有相同的实例。

intersectionOf：交集组合，即必须满足集合下的所有条件限制才是这个类的实例。

unionOf：并集组合，即只要满足集合下的一个条件限制就是这个类的实例。

someValuesFrom：实例中该属性所作用的值中至少有一个是该类型的。

allValuesForm：实例中该属性所作用的值中都是该类型的。

minCardinalty：最小值限制，实例中该属性所作用的值必须大于等于这个最小值限制。

maxCardinalty：最大值限制，实例中该属性所作用的值必须小于等于这个最大值限制。

Cardinalty：值限制，实例中该属性所作用的值必须等于这个值限制。

集合运算包括 C_n^1(并集)，C_n^n(交集)，C_n^*(基数限制 k 集合)。

RQ1 定义中的条件(1)(2)(3)的集合运算为 C_2^1。条件(4)本科以上的学历有3个：学士、硕士、博士，因此集合运算为 C_3^1。条件(5)有三级嵌套，毕业于“211”以上学校的类型有3个：“211”、“985”、海外，因此集合运算为是 C_3^1，毕业于“211”以上学校，具有开发经验是 C_2^2，最外层是二选一，为 C_2^1。条件(6)为5个条件至少满足4个，为 C_5^4。

将上述的类运算式转化成一棵条件树，树的根节点即为软件开发工程师，中间的节点都是集合运算即 C_n^1(并集)，C_n^n(交集)，C_n^*(k 子集)，叶子节点都是一个条件限制，可以是 someValuesFrom，allValuesForm，minCardinalty，maxCardinalty 限制等。equivalentClass 可忽略。根据 RQ1 限制条件的层级关系，将其转化成树模型的层级，转化后的树模型如图2所示。

其中，R 代表根节点，数字 1－15 代表每一个限制条件：

1. 熟练 min 1 语言；2. 精通 min 1 语言；3. 熟练于 min 1 开发工具；4. 精通于 min 1 开发工具；5. 熟练 min 1 数据库；6. 精通 min 1 数据库；7. 毕业学历是 somevalue 学士；8. 毕业学历是 somevalue 硕士；9. 毕业学历是 somevalue 博士；10. 毕业于 somevalue “211”；11. 毕业于 somevalue “985”；12. 毕业于

somevalue 海外；13. 具有开发经验 somevalue int；14. 毕业于 somevalue 非“211”和“985”；15. 具有开发经验 min 2。

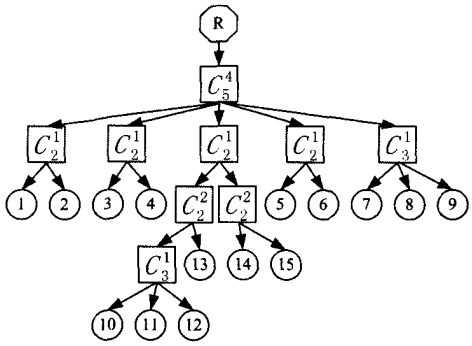

图 2 根据 RQ1 得到的分类推理模型

3.2.3 个体分类算法

当判断一个个体是否属于软件开发工程师的推理查询条件(RQ1)的实例时，提取出一个个体的所有三元组，并根据 RDFS 规则中的 5,7,9,11 规则对这个实例的三元组进行扩充，再与树模型进行匹配。从其中一个叶子节点开始，如果满足当前节点，则查看当前节点的父节点；如果父节点是 C_n^1，则父节点匹配，进入父节点，如果父节点是 $C_n^k(2\leqslant k\leqslant n)$，则需要继续判断其他的兄弟节点是否满足 C_{n-1}^{k-1}，直到满足才能进入父节点；如果不满足当前条件，若父节点是 C_n^n，则此实例不满足，退出模型，如果父节点是 $C_n^k(1\leqslant k\leqslant n-1)$，则需要继续判断其他的兄弟节点是否满足 C_{n-1}^k，直到满足才能进入父节点。如果当前节点是枝节点，进入兄弟节点判断时，要顺着树往下找直至找到兄弟节点下的叶子节点，再从叶节点开始判断。顺着树往上走，如果能走到根节点，则此个体是软件开发工程师的一个实例。

判断过程的伪代码如下：

输入：筛选类型，RDF 数据集

输出：属于该类型的实例集合

过程：

Step 1 从本体中找到该类型的定义条件，提取该类型的类运算式，生成树模型 tree。

Step 2 得到所有的用户对象 userlist，结果集 result=∅。

Step 3 i=0。

Step 4 user = userlist. get(i)。

Step 5 提取 user 的所有三元组。

Step 6 根据本体文件对三元组进行扩充，对 p 为 type 的所有 o，得到所有 o 的所有父辈，生成 s p o；对 p 不为 type 的所有 p，找到 p 的所有父辈，以及 p 对应 o 的所有父辈，进行组合，生成对应三元组。

Step 7 对用户 s 的所有三元组生成用户信息 individualMap 进入树匹配，取第一个叶子节点为当前节点。

Step 8 匹配当前节点，如果当前节点是根节点，将 user 加入结果集 result，转到 Step 4；如果是叶子节点，转到 Step 9；如果是枝节点，转到 Step 11。

Step 9 判断叶子节点是否匹配，如果匹配，查看父亲节点类型，如果是 C_n^1，将父节点作为当前节点，转到 Step 8，如果是 $C_n^k(2\leqslant k\leqslant n)$，若后面的兄弟节点数≥k，将后面的第一个兄弟节点作为当前节点，同时将父亲节点的类型记录为 C_{n-1}^{k-1}，转到 Step 8；若后面的兄弟节点数<k，此实例不满足，退出模型，转到 Step 12。

Step 10 如果不匹配，查看父亲节点类型，如果是 C_n^n，此实例不满足，退出模型，转到 Step 12；如果是 $C_n^k(1\leqslant k\leqslant n-1)$，若后面的兄弟节点数≥k，将后面的第一个兄弟节点作为当前节点，同时将父亲节点的类型记录为 C_{n-1}^k，转到 Step 8；若后面的兄弟节点数<k，此实例不满足，退出模型，转到 Step 12。

Step 11 查看父亲节点类型，如果是根节点，则将根节点作为当前节点，转到 Step 8；如果是 C_n^1，将父节点作为当前节点，转到 Step 8；如果是 $C_n^k(2\leqslant k\leqslant n)$，若后面的兄弟节点数≥k，按深度优先策略找到后面的第一个兄弟节点下的一个叶子节点作为当前节点，转到 Step 8；若后面的兄弟节点数<k，此实例不满足，退出模型，转到 Step 12。

Step 12 i++；如果 i<userlist. size，转到 Step 4。

Step 13 输入 result，退出。

4 实验结果

实验所使用的硬件环境为 Pentium(R) Dual-Core CPU E5300 2.60GHz，内存 4GB，磁盘 500 GB。软件环境为操作系统 Windows7 64 位，采用 java 作为编程语言，开发环境为 eclipse。

4.1 TCRA 算法在标准数据集上的实验对比

LUBM[13]数据集是语义 Web 领域通用的基准测试集。本文的推理模型也在 LUBM 数据集上进行了实验。在 Q12 的查询语言中有?x rdf:type ub:Chair，Chair 是一个复杂类，由类运算式 intersectionOf 构建。

数据集中的个体都没有明确定义哪些是 Chair 的实例，因此查询的?x rdf:typeub:Chair 需要用到推理。在 LUBM10 结合 TCM 的方法对?x rdf:typeub:Chair 进行查询，查询结果如表 2 所列。

表 2 LUBM 中查询?x rdf:typeub:Chair 的时间

数据集	TCRA 响应时间(s)	FaCT++响应时间(s)
LUBM1	9.534	126.899
LUBM5	43.665	4892.693
LUBM10	87.097	内存溢出
LUBM20	176.323	内存溢出

4.2 TCRA 算法在人力资源数据集上的实验对比

为验证本方法的效率和有效性，本文对某用人单位提供的电子简历文档进行实验，对电子文档进行处理后根据创建的领域本体生成 RDF 数据集，以这些数据作为实验数据。采用分类的准确率和查询的响应时间作为评价的指标。

从 RDF 数据集中选取 1000 个、10000 个、100000 个实例，用 TCRA 算法判断 RQ1 后，对实例的筛选结果与人工筛选结果进行对比，计算分类准确率和分类查全率，结果如表 3 所列。

$$\text{分类准确率}(p)=\frac{\text{正确分类的实例数}}{\text{分类模型实际分类的实例数}}\times 100\%$$

$$\text{分类查全率}(r)=\frac{\text{分类正确的实例数}}{\text{人工分类的实例数}}\times 100\%$$

表 3 RQ1 基本本体的分类准确率、查全率

测试集	准确率	查全率
1000 个实例	100%	100%
10000 个实例	100%	100%
100000 个实例	100%	100%

为了比较本文推理模型的效率，选择一个能用 OWL 表示的类型，即只存在 C_n^1 和 C_n^n，将采用 TCRA 的推理时间和官

(下转第 268 页)

[15] Wang Q, Pang J, Liu G, et al. Color Maximal-Dissimilarity Pattern for Pedestrian Detection[C]// Proceedings of the 2012 21st International Conference on Pattern Cognition. 2012: 1952-1955

[16] Wang Q, Pang J, Qin L, et al. Justifying the importance of color cues in object detection: a case study on pedestrian[M]// The Era of Interactive Media. Springer, 2013: 387-397

[17] Van De Sande K E, Gevers T, Snoek C G. Evaluating color descriptors for object and scene recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9): 1582-1596

[18] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 971-987

[19] Wu J. A fast dual method for HIK SVM learning[M]// Computer Vision-ECCV 2010. Springer, 2010, 552-565

[20] Lampert C H, Blaschko M B, Hofmann T. Efficient subwindow search: A branch and bound framework for object localization [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(12): 2129-2142

(上接第 237 页)

方下载的 FaCT++、Pellet、RacerPro 的推理引擎时间对 RQ2(将 RQ1 的最后一个条件去掉)的分类进行对比。对每个查询测试了 10 次,计算平均响应时间。经过测试,FaCT++在分类的推理运用上比 RacerPro 和 Pellet 效率更高,对比时间如表 4 所列。在数据量较大时,本文实验与 FaCT++进行对比,对比时间如表 5 所列。

表 4 RQ2 各推理机分类推理时间对比(单位 s)

RDF 三元组数(个) 推理机	5000	10000	20000	50000
FaCT++	0.086	0.237	0.529	1.109
Pellet	21.951	86.542	134	261.276
RacerPro	3.626	6.62	9.366	15.098
TICM	3.442	4.821	5.475	6.576

表 5 RQ2FaCT++和本文分类推理的时间对比(单位 s)

RDF 三元组数(个) 推理机	100000	200000	500000	1000000	2000000	3000000
FaCT++	4.583	37.053	311.67	1256.357	内存溢出	内存溢出
TICM	8.069	11.56	29.224	32.449	76.404	410.556

通过实验对比发现,在数据量较小时,本文的效率不如 FaCT++,但随着数据量的增大,本文的效率会比 FaCT++高。本文的方法适用于用复杂集合运算定义的类的分类推理,从实例出发,只对需要推理的规则进行推理,不产生中间结果;而其他推理引擎在推理的过程中会对其他规则也进行推理,在推理的过程中产生大量的中间结果,之后再对结果进行过滤,因而当数据量增大时,时间效率会越来越差。但本文的方法只适用于个体分类推理,如要完成其他类型的推理,可与其他推理引擎嵌套使用。

4.3 TCRA 算法分析

采用查询处理时间来评价本方法中分类推理模型的效率。假设数据集中实例个数为 M,RDF 三元组数为 P,每个三元组按照 RDFS 的 5,7,9,11 规则进行扩充,设时间为 Q,查询句子中的所有 C_n^x 中 n 叠加得 N,最坏的情况下要遍历树上所有的节点才能判断此实例是否属于这个类,时间复杂度为 $O(PQ+MN)$。最好的情况下,只要匹配第一个节点就可以判断此实例不属于这个类,时间复杂度为 $O(PQ+M)$。

结束语 本文在描述逻辑的基础上提出基数限制集合的表示方式,并在 OWL DL 语言中扩展了集合基数限制标签 kCardinaltySetsOf,实现了类运算中除交集、并集、补集以外的基数 k 集合表示法,并根据基数 k 集合的运算思想将分类运算转化成树型分类推理算法(TCRA),通过个体在 TCRA 中的遍历匹配,判断个体的类型。本文以 RDF 数据格式存储简历中的实体信息,实现人才知识库的存储和分类。实验结果表明,本文的方法在准确率和查询效率上达到了良好的效果,能帮助用人单位更快、更准地找到所需要的信息,提高了人才分类的效率。本文的 TCRA 本质上是一种两类个体分类器,对于两类分类,它的时间复杂度是线性的,但是如果进行多类个体分类则必须构造多个 TCRA,类别数目越多则分类时间越多,当查询条件更改时需要重新建立推理模型。

参 考 文 献

[1] Tsarkov D. Horrocks I. Optirnised Classification for Taxonomic Knowledge Bases[C]// Proceedings of the 2005 International Description Logic Workshop(DL 2005). Manchester, UK, 2005

[2] 王海龙,马宗民,严丽,等. 支持模糊数据类型表示的模糊描述逻辑 F-SHOIQ(G)[J]. 计算机学报,2009,2(8):1511-1524

[3] 程经纬,马宗民,严丽,等. 模糊描述逻辑知识库查询蕴涵的判定方法[J]. 计算机学报,2012,35(4):767-785

[4] Stoilos G, Stamou G, Pan J. Fuzzy Extensions of OWL: Logical Properties and Reduction to Fuzzy Description Logics[J]. International Journal of Approximate Reasoning, 2010, 51(6): 656-679

[5] Stoilos G, Stamou G. Reasoning with fuzzy extensions of OWL and OWL 2[J]. Knowledge and Information Systems, 2013, 40(1): 205-242

[6] Bobillo F, Delgado M, Gómez-Romero J. DeLorean: A reasoner for fuzzy OWL 2[J]. Expert Systems with Applications, 2012, 39: 258-272

[7] 李言辉,豫宝文,陆建江,等. 支持数量约束的扩展模糊描述逻辑复杂性研究[J]. 软件学报,2006,5(17):968-975

[8] 邓志鸿,唐世渭,张铭,等. Ontology 研究综述[J]. 北京大学学报,2002,35(5):730-738

[9] 石莲,孙吉贵. 描述逻辑综述[J]. 计算机科学,2006,33(1):194-197

[10] Tsarkov D, HorrocksI. FaCT++ Description Logic Reasoner: System Description[C]// Proceedings of the International Joint Conference on Automated Reasoning (IJCAR'06). 2006: 292-297

[11] Haarslev V, Moller R. Racer: A Core Inference Engine for the Semantic Web Ontology Lanuage(OWL)[C]// Proceedings of the International Workshop on Evaluation of Ontology-based Tools. 2003: 27-36

[12] Sirin E, Parsia B, Grau B, et al. Pellet: A practical OWl-DL reasoned[J]. Journal of Web Semantles, 2007, 5(2): 51-53

[13] Guo Y, Pan Z, Heflin J. LUBM: A benchmark for OWL knowledge base systems[J]. Journal of Web Semantics, 2005, 3(2/3): 158-182