

基于 Racer 和 nRQL 的本体查询与推理^①

Ontology Query and Reasoning Based on Racer & nRQL

李信本 陈仲委 (浙江万里学院计算机与信息学院 浙江宁波 315100)

摘要:语义网技术的兴起促进了本体技术的发展,本体作为语义网的基石,在知识表示与知识推理方面发挥着重要作用。本体描述语言与描述逻辑相结合,为本体推理的合理性和有效性提供了保证。在介绍描述逻辑及其系统的有关概念与查询推理机制的基础上,重点讨论了推理机 Racer 及其查询语言 nRQL 的功能,开发了一个基于 Racer 和 nRQL 的智能信息查询程序并应用于大学本体的信息查询。

关键词:本体 描述逻辑 推理机 Racer nRQL

1 引言

尽管万维网(WWW)已成为人们获取信息和服务的重要渠道之一,但 WWW 是面向人类阅读的,不便于计算机自动理解和处理。为使网络中的信息具有形式语义并能够被计算机理解,方便人机交互与协作,语义网(Semantic Web)技术的研究已成为当前的热点。本体(Ontology)作为语义网技术的基石,在语义网的研究中具有非常重要的作用,特别是,开发适当的本体语言及相应的本体查询语言是语义网研究的两个重要方面^[1]。已经提出了几种本体语言,其中 W3C 组织推出的 OWL 语言(Web Ontology Language)是一个典型代表,它以描述逻辑和框架逻辑为形式基础,以描述逻辑为语义推理基础,其中的 OWL DL 既提供了丰富的表达能力,也可以使描述逻辑语义推理的判定性有一定保证。随着这些本体语言的推出,出现了一些本体查询语言和系统, Racer 就是一个支持 AL-CQHI+(D-)的描述逻辑本体推理系统,而 nRQL 就是 Racer 使用的本体查询语言。本文首先介绍基于描述逻辑的本体查询与推理的有关概念,探讨了 Racer 和 nRQL 的查询推理机制及其应用。

化,也叫概念表示语言或术语逻辑。它是一阶逻辑的一个可判定子集,具有合适定义的语义,并且具有很强的表达能力。一个描述逻辑系统由描述语言、知识库以及推理机三部分组成,如图 1 所示。

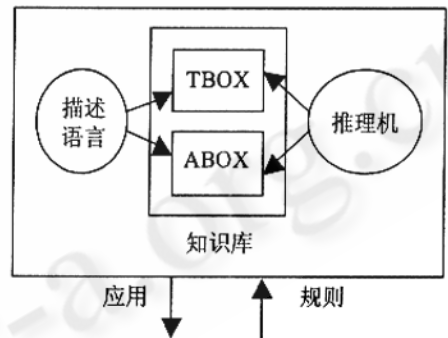


图 1 描述逻辑的知识表示系统结构

知识库是一个二元组 $\langle TBox, ABox \rangle$, 其中 TBOX 是一个关于包含断言的有限集合(也称为术语公理集合),主要用于声明概念之间的包含关系或引入概念的名称。表示概念 C 被概念 D 所包含可形式表示为 $C \sqsubseteq D$, 而引入概念 A 的过程可表示为 $A \sqsubseteq C$ 或 $A \sqcap C$ (其中 $A \sqcap C$ 又可形式表示为 $A \sqsubseteq C$ 同时 $C \sqsubseteq A$)。ABOX 是实例断言的有限集合,用于指明个体的属性或者个体之间的关系,一般形式为 $C(a)$ 或 $R(a, b)$, 其中 $C(a)$ 表示个体 a 属于概念 C (或者说个体 a 是概念 C 的一个实例), 而 $R(a, b)$ 表示两个个体 a 和 b 之间

2 Racer 与描述逻辑

2.1 描述逻辑

描述逻辑是一种基于对象的知识表示的形式 1

^① 基金资助:浙江省教育厅资助项目(20050208)

具有R关系(或称为角色)。

描述逻辑语言的两个基本元素是概念和关系,概念表示个体的集合,关系则表示领域中个体之间所具有的相互关系。描述语言通常还包含若干构造算子,通过这些构造算子,人们就可从简单的原子概念和关系构造出复杂的概念和关系。最基本的描述逻辑语言称为ALC,其中包含有交、并、非,存在量词和全称量词等构造算子。在ALC的基础上再添加不同的构造算子,就构成不同表达能力的描述逻辑语言,例如Racer的描述逻辑基础是ALCQHIR+,它是在基本描述逻辑ALC的基础上增加了数量限制(Q)、角色层次(H)、反向角色(I)和角色传递闭包(R+)等扩展而成。有关这些不同描述逻辑系统的详细语法和语义详见^[2]。

假如将知识库看成一个存放相关领域信息的系统,人们从系统中不仅可查询明显存储的术语和断言,还可以借助推理机推出隐含的信息。对于知识库中的TBOX,系统能回答的查询主要有:TBOX中概念的一致性(即一个概念所描述的对象集合是否为空集合)和概念的包含关系(即由两个概念所描述的对象集合之间是否有子集关系);对于知识库中的ABOX,可回答的查询更多,包括ABOX是否一致、个体a是否为某个概念C的实例、查找概念C的所有实例等等。根据描述语言的逻辑语义,不同种类的查询被定义为推理问题,回答一个查询也被称为提供推理服务。因此,描述逻辑系统中的查询推理的基本问题主要包括概念的可满足性、概念的包含关系、实体检测、一致性检测等,其中概念的可满足性问题是基本的推理问题,其它推理任务基本都可转化为概念的可满足性问题^[3]。

2.2 基于描述逻辑的本体推理机 Racer

Racer是由V. Haarslev等人编写的基于描述逻辑的推理机的一种。首先,Racer是一个语义网推理系统和信息仓库,既可用于查询OWL本体文档的外延信息,也可用于推理OWL本体的内涵信息,如本体的一致性检查和本体子类之间的包含关系等。其次,Racer是一个基于描述逻辑的知识表示系统,它的描述逻辑基础是SHIQ,按照ALC的命名规则,SHIQ可以称为ALCQHIR+,它是在基本描述逻辑ALC的基础上增加了数量限制(Q)、角色层次(H)、反向角色(I)和角色传递闭包(R+)等算子后扩展而成,既提供了丰富的表达能力,也可以使描述逻辑语义推理的判定

性有一定保证。第三,Racer是描述逻辑和特定关系代数的混合体,它提供的查询推理语言nRQL不仅可以查询ABOX中断言的关系,还可以查询推理空间和时态的关系^[4]。

Racer的主要查询推理功能如下:

第一,基本的对于本体的满足性测试,本体的一致性通过满足性测试来实现。本体作为建模和语义描述的工具,如果产生不一致的情况,就会使得本体应用的领域产生形式和语义的矛盾,使得在本体上的具体工作产生歧义。

第二,对于包含性的测试也转化为对满足性的测试。通过对包含性的测试,使得TBOX中的概念划分了层次,形成了概念层次,便于计算概念描述的父(或者子)描述,以及计算其祖先和后代的描述。

第三,实例检测可以为某些个体确定相应的概念描述。并且可以对属于某些概念描述的个体进行分类。

第四,一些辅助的推理功能也是具体应用的重要工具。比如在知识库中对概念名字和个体名字的检索,与角色相关的个体对的检索,还有角色层次的检索,即角色的父子层次等。

作为一个有丰富表达力的描述逻辑系统的推理机,Racer由于增加了反转角色和数量限制,导致其满足性测试算法的复杂性大大提高。然而,Racer服务器做了大量的优化处理,使得运行效率仍然很高,如对ABOX的一致性优化,对数量限制的优化,对大量的ABOX的检索优化等。Racer系统的另一个优势是支持多个TBOX和多个ABOX,使得一个Racer服务器可同时为多个客户提供推理服务。

3 基于 Racer 和 nRQL 的查询与推理

3.1 基于 Racer 和 nRQL 的查询模式

典型的Racer查询可被看作为由Racer服务器和Racer客户组成“客户/服务器”模型。客户端准备并提交查询请求给Racer服务器;服务器在对知识库进行推理计算后,将查询结果反馈给客户端;客户/服务器间的交互通过nRQL查询语言的语句来完成。Racer系统当前版本的服务器称为RacerPro,用于存放知识库和提供推理服务。而Racer客户可以是系统提供的交互式图形用户客户端程序RacerPorter,也可以是任何基于网络通信协议的客户应用程序。

Racer 服务器和客户程序可以驻留在同一台计算机上运行,也可以分布在网络上的不同节点,从而实现分布式的推理服务。此外,Racer 系统还提供知识库编辑程序 RacerEditor,用户用 RacerPorter 和 RacerEditor 就可进行交互式的本体输入、本体编辑、本体查询推理等活动。

nRQL 是 Racer 系统的新版查询语言,既可用于查询描述逻辑的知识库,也可用于查询 RDF/OWL 文档。NRQL 查询语句类似于数据库中 SQL 语句,可以作为交互式命令使用,也可以将其以字符串形式嵌入到客户程序中,由客户程序提交查询请求和接收处理查询结果。

nRQL 查询的一般格式为:

```
(retrieve (query-head)(query-body))
```

其中 retrieve 是查询关键字,query-head 是查询头,query-body 是查询体。查询头是由变量或 ABOX 中的个体名组成的列表(可以为空),用于指定查询结果的输出格式。查询体是由原子查询项按照 nRQL 语法构成的查询表达式,可以是简单的原子查询,也可以是复杂的逻辑查询。

根据查询语法的不同复杂性,nRQL 查询可分为一元概念原子查询、二元角色原子查询、复杂查询等。概念原子查询用于检索一个概念或 OWL 文档类的所有个体,例如查询:

```
(retrieve (? x)(? x FullProfessor))
```

将返回当前 ABOX 中概念类 FullProfessor 的所有实例。本例中的查询头是(? x),查询体为(? x FullProfessor)。

角色原子查询是二元查询,主要用于检索当前 ABOX 中满足某特定关系的个体对。假如要检索所有学生的选课情况,则 nRQL 语句为:

```
(retrieve (? student ? course)
```

```
(? student ? course takesOf))
```

其中 takesOf 是一个二元关系,其结果将返回一个以(学生名 课程名)为元素的列表。本例中的查询头是(? student ? course),查询体为(? student ? course takesOf)。

复杂查询是指查询体为一个复杂的查询表达式。前述的概念原子、角色原子等都是简单的查询项,这些简单查询项可通过 AND、OR、NOT 等布尔运算符以及

NEG、INV、PROJECT - TO 等运算符递归地形成更复杂的查询项。复杂查询的例子见图 2 的 Q5,更详细的说明见^[4]。

```

D:\JRacer18>type LxbQueryZwu.txt

Q1: 查询计算机学院的所有教授?"
(retrieve (?x) (?x |http://www.zwu.edu.cn/JRacer18/zwu.owl#FullProfessor|))

Q2: 计算机学院副教授职称的教师吗?
retrieve () (?x |http://www.zwu.edu.cn/JRacer18/zwu.owl#AssociateProfessor|))

Q3: 教师WanZhong是一名副教授吗?
(retrieve () (?x |http://www.zwu.edu.cn/JRacer18/zwu.owl#WanZhong| |http://www.zwu.edu.cn/JRacer18/zwu.owl#AssociateProfessor|))

Q4: 查询教师JinMin教的所有课程?
(RETRIEVE (?X) (AND (?X |http://www.zwu.edu.cn/JRacer18/zwu.owl#Course|) (|http://www.computer.zwu.edu.cn/JinMin| ?X |http://www.zwu.edu.cn/JRacer18/zwu.owl#teacherOf |)))

Q5: 查询教师WanZhong教的 interface1课程的所有学生?
(RETRIEVE (?X) (AND (?X |http://www.zwu.edu.cn/JRacer18/zwu.owl#Student|) (?X |http://www.computer.zwu.edu.cn/interface1| |http://www.zwu.edu.cn/JRacer18/zwu.owl#takesCourse|)))

D:\JRacer18>
  
```

图 2 nRQL 查询语句

应注意的是,在查询描述逻辑的知识库时,nRQL 查询语句的格式如上所述,但在查询 OWL 知识库和 RDF 数据时,必须将概念类名或角色名等放在“管道”符号“|...|”中间,而且在类名或角色名前必须加上正确的名空间,如图 2 的 Q1 中的“|http://www.zwu.edu.cn/JRacer18/ZWU.owl#FullProfessor|”。

3.2 Racer 和 nRQL 的应用

作为一个语义网推理系统和信息仓库,Racer 和 nRQL 不仅可用于查询 OWL 本体文档的外延信息,也可用于推理 OWL 本体的内涵信息。下面结合我们构造的大学本体来说明 Racer 和 nRQL 在本题查询推理方面的应用。大学本体由 16 个本体类(即描述逻辑中概念)、25 个角色和一些属性组成。这些本体类包括:大学、学院、系、人员、教师、教授、副教授、讲师、学生、本科生、研究生、课程、本科课程、研究生课程、发表论文、出版教材。每个本体类有一些属性,如教师本体有姓名、电话、email 地址、学位及取得学校等。一些

本体之间有子类关系,子类继承了父类的属性,如教授、副教授、讲师都是教师类的子类,这些子类就继承了教师类的一些属性如姓名、电话、Email 地址等,也继承了教师类的一些角色如讲授课程 (teacherOf)、为部门工作 (workFor) 等等。角色主要包括: headOf, subOrganizationOf, teacherOf, takesOf, subClassOf, memberOf, bachelorDegreeFrom, workFor, advisorOf, masterDegreeFrom, doctoralDegreeFrom, teachingAssistantOf 等。有些角色有传递特性如 subOrganizationOf, 因为 computer&information 学院是 ZWU 大学的一个部门,而 computer 系又是 computer&information 学院的一个部门,则 computer 系当然也是 ZWU 大学的一个部门。另一些角色有反向角色,如角色 takesOf 表示学生选修课程,其反向角色 takenBy 表示课程由学生修读。总之,前述的那些本体类之间经由某一(些)角色相连,从而构成了整个大学本体的语义网络,其语义信息既有利于人理解也便于计算机处理。

启动 RacerPro 和 RacerPorter 后,就可借助 RacerEditor 工具进行大学本体的编辑输入,也可以在 RacerPorter 中通过交互式命令完成图 2 所示的查询以及其它查询和推理。尽管交互命令方式在知识库编辑输入、知识库管理维护、系统工作方式设置等方面提供了很大方便,但不利于在应用系统中使用。由于 Racer 系统提供了 TCP/IP 协议,可以把 Racer 作为网络推理服务器使用,我们在 Racer 系统提供的 JRacer 接口包基础上,用 JAVA 程序设计语言开发了一个可远程访问 RacerPro 服务器的 nRQL 客户查询程序 LxbQueryZWU,该程序的基本框架如图 3 所示,其中第二行用 Racer 服务器的 IP 地址和端口号生成 RacerServer 的一个实例,第三行打开该实例与 Racer 服务器的通信端口,并在第四行以后开始发送各种对知识库进行管理的命令或 nRQL 查询命令,从而完成相应的查询推理请求。例如第四行就是请求加载知识库到服务器的命令,而第五行就是查询命令 Q1。

Racer 服务器在回答一个 nRQL 查询(即为客户提供推理服务)时,经常要进行复杂的推理计算才能得出完整的答案,例如,当要查询大学的所有部门时,由于 subOrganizationOf 角色的传递性,推理机不仅要给出大学的各个学院等外延信息,还应给出各学院的

系、乃至各系的研究小组等内涵信息,因篇幅原因不再详细讨论。

```

.....
String filename="\JRacer18/zwu.owl";
RacerServer racer1=new RacerServer(ip,port);
racer1.openConnection();
System.out.println(racer1.send("(owl-read-file
"+filename+""));

System.out.println("Q1: 查询计算机系的所有教授?");
System.out.println(racer1.send("(retrieve (?x) (?x "+
"http://www.zwu.edu.cn/JRacer18/zwu.owl#" +
"FullProfessor)"));

.....
System.out.println("Q5: 查询教师WanZhong教的interface1
课程的所有学生?");
System.out.println(racer1.send("(RETRIEVE (?X) "+
"(AND (?X |http://www.zwu.edu.cn/ JRacer18"+
"/zwu.owl#USStudent) (?X |http://"+
"www.computer.zwu.edu.cn/interface1|http://"+
"www.zwu.edu.cn/JRacer18/zwu.owl#takesCourse)"));

racer1.closeConnection();
.....

```

图 3 nRQL 查询的 JAVA 程序框架

4 结束语

随着互连网上信息的急剧增长,基于知识的信息检索(也称为语义检索或智能检索)成为目前的研究热点。语义网、本体、描述逻辑等概念的提出推动了知识检索的研究。本文在介绍描述逻辑及其系统的有关概念与查询推理机制的基础上,重点讨论了语义推理机 Racer 及其查询语言 nRQL 的基本功能,开发了一个基于 Racer 和 nRQL 的远程本体查询程序并说明了其应用。

参考文献

- 1 Z. Zhang J. Miler, Ontology Query Languages for The Semantic Web.
- 2 F. Baader, etc. editors. The Description Logic Handbook: Theory, Implementation and Application. Cambridge University Press. 2003. 472 - 494.
- 3 史忠植等,语义 WEB 的逻辑基础. 中国科学(E 辑. 信息科学),2004,34(10),1123 - 1138.
- 4 Racer Systems GmbH & Co. KG. RacerPro Reference manual v1.9. 200.