

面向角色关系和公理的数据挖掘方法研究

周 峰, 宋 衡, 童 林

(扬州大学 信息工程学院, 江苏 扬州 225127)

摘 要: 由于缺乏有效的公理获取与验证系统, 无法实现自动推理, 严重阻碍了语义 Web 的发展. 本文针对政府角色及角色关系, 提出一种半自动政府角色关系和公理的获取与验证方法. 首先分析了政府角色关系公理的完备性, 进而给出了政府角色关系和公理中可能存在的错误, 并对错误进行分类; 最后设计了一个政府角色关系和公理的知识模型实现系统, 实现该知识模型系统的自动验证. 实验表明, 本文提出的方法能有效的进行政府角色关系公理的获取与正确性验证.

关键词: 语义 Web; 角色; 公理; 获取; 验证

中图分类号: TP39

文献标识码: A

文章编号: 1000-7180(2012)11-0099-04

Research on Acquisition and Verification of Government Role Relations and Their Axioms

ZHOU Feng, SONG Heng, TONG Lin

(School of Information Technology, Yangzhou University, Yangzhou 225127, China)

Abstract: In the absence of effective system to access and verify axiom, automated reasoning can't be realized, it also severely hinders the development of the Semantic Web. According to government roles and role relationships, this article proposed a semi-automatic government roles relationships and a method of obtaining and verifying axioms. Firstly it analyzed the completeness of government roles relations axiom, then gives errors that may exist in the government roles relations and axioms, and classification; finally it design a system to realized the knowledge model of the role of a government relations and axioms, realized the automatic verification of the knowledge model system. The experiments show that the proposed method can effectively carry out the access of government roles relations axiom and correctness verification.

Key words: semantic web; government characters; axiom; acquisition; verification

1 引言

形式化本体奠定了语义 Web 的知识表示基础, 提供了 Internet 上的语义支撑^[1-3]. 语义 Web 的检索、推理及分布式语义融合等任务依赖自动化推理, 因此, 领域公理知识的获取与验证将直接影响语义 Web 功能的实现^[4-5]. 知识获取就是人们把一些人类理解的解决问题的知识用一种合理的方式提取出来, 转移到计算机中, 用程序来让计算机理解和识别, 是一种高级的人工智能理念; 知识验证是通过人工或者自动方式判定知识的完备性及正确性^[6-7].

角色是“社会角色”的简称, 指个人在社会关系位置上的行为模式. 它规定一个人活动的特定范围和与人的地位相适应的权利义务与行为规范, 是对一个处于特定地位的人的行为期待^[8]. 政府角色是指在政府这个特殊场合中的“角色”. 角色关系是指所定义的角色之间存在的关系. 政府角色关系是指在政府这个特定场合中, 所定义的政府角色相互之间存在的关系.

本文以政府角色为例, 研究一种半自动政府角色关系和公理的获取与验证方法. 首先分析了政府角色关系公理的完备性, 进而给出了政府角色关系

和公理中可能存在的错误,并对错误进行分类;最后设计了一个政府角色关系和公理的知识模型实现系统,实现该知识模型系统的自动验证.实验表明,本文提出的方法能有效的进行政府角色关系公理的获取与正确性验证.

2 政府角色关系公理的完备性

所谓政府角色关系公理的完备性,是指在政府这样的一个特定场合中,所有角色关系之间的公理都写出来了,没有遗漏的公理.为了保证政府角色关系公理的完备性,即保证所有关系之间的公理都写出来了,要采用一定的方法来实现这个目标.下面将要介绍保证公理完备性的方法.

政府角色关系公理分为六大类:(1)内部公理.(2)外部公理.(3)单个角色公理.(4)两个角色形成的公理.(5)单个角色与单个角色关系形成的公理.(6)级别高于(级别低于)关系.

内部公理,是指公理中包含两个主体,如果这两个主体间存在一种关系,那么可以推出它们又存在另一种关系,这样的推导就是一个内部公理.

外部公理,是指公理中包含三个主体,由两个二维关系相交推导出另一个二维关系,其中相交的两个二维关系之间必须存在一个公共的主体名.每个二维关系由一个关系名谓词,一个场合的个体,两个主体名的个体组成.这样的推导就是一个外部公理.

也就是说,两个关系名相交产生外部公理的时候,我们要考虑它们有几种组合方式,不同的组合方式产生不同的外部公理,只有把所有的组合方式都写出来,才能得到完备的外部公理.

单个角色公理,是指单个主体满足一种角色定位的同时,他也满足另一个角色定位.这样的推导就是一个单个角色公理.

两个角色形成的公理,是指两个单独角色之间形成的一种角色关系.公理中包含两个主体,由两个一维关系推导出一个二维关系.这样的推导就是一个单个角色公理.

单个角色与单个角色关系形成的公理,是指两个角色之间的一种角色关系在确定其中一个角色定位之后得出的另一种角色关系.这样的推导就是一个单个角色与单个角色关系形成的公理.

级别高于(级别低于)关系公理,是指它是一种辅助关系,是为了能够简化角色关系而设立的一种特殊公理形式,它可以避免烦琐的表示所带来的不便.它是由两个主体名在同一个场合中推导形成一

个级别高低关系的公理.

3 政府角色关系和公理中可能存在的错误

3.1 公理语法错误

当写出一条公理后,这条公理可能并不符合公理的书写规范.当公理与公理的书写规范有差别,我们就说这条公理存在语法错误.

在这个研究中,公理有六种出现方式,但因语法错误种类很多造成篇幅较大,我们只写出这六种公理的规范定义,语法错误就详细写两种公理:内部公理和外部公理.这六种公理的书写规范定义为:

形式一:关系名 1(<场合>, <主体名 1>, <主体名 2>)->关系名 1(<场合>, <主体名 1>, <主体名 2>)

下面举例说明公理的语法错误.

公理的输入存在 27 种语法错误类型,错误列表举例如下:

语法错误类型 1:输入公理中,字符‘,’左侧的字符不是‘<’,或者逗号右侧的字符不是‘>’时,公理就存在语法错误.

在编写验证算法程序时,把可能出现的语法错误都考虑进去.程序运行时,只要正在接受验证的公理存在其中的任意一个语法错误,则屏幕上提示该公理存在语法错误,并显示该公理是形式一还是形式二,随后报告该语法错误的类型码.

3.2 公理中存在的冗余性错误

冗余就是重复的意思.所谓公理的冗余性,也就是说所获取的公理中存在相同的公理.

即如果在获取的公理中存在这样的两条公理: $P1 \cdot P2 \rightarrow P$, $P1 \cdot P2 \rightarrow P$,则这两条公理就存在冗余性.

把所有的公理从 txt 文本中读出来,写入一个二维的字符数组中,数组的每一行就是一条公理,数组实际的行数就是公理的条数.然后每次从数组中读出两行(也即取出两条公理)进行检查,看取出的两条公理的关系名,群体名,主体名是否都对应相同,如果都相同的话,那么这两条公理就存在冗余性,并报告用户.

3.3 公理中存在的包含性错误

如果在获取的公理中存在这样的两条公理:两条公理的推导符号‘ $\rightarrow$ ’的右侧相同,一条公理的左侧包含在另一条公理的左侧中,即: $P1 \cdot P2 \rightarrow P$, $P1 \rightarrow P$,那么这两条公理就存在包容性.

把所有的公理从 txt 文本中读出来,写入一个

二维的字符数组中,每一行就是一条公理,数组实际的行数就是公理的条数.然后每次从所有的公理中取出两条公理进行检查,看两条公理的右侧的主体名是否相同,一条公理左侧的主体名是否包含在另一条公理左侧的主体名中,如果都相同的话,那么这两条公理就存在包含性.

3.4 公理中存在的潜在不一致性

如果存在这样的两条公理,两条公理的推导符号的左侧完全相同,而右侧的关系名不相同,即: $P1 \cdot P2 \rightarrow P$, $P1 \cdot P2 \rightarrow P'$,那么这两条公理就存在潜在不一致性.

把所有的公理从 txt 文本中读出来,写入一个二维的字符数组中,每一行就是一条公理,数组实际的行数就是公理的条数.然后每次从所有的公理中取出两条公理进行检查,看这两条公理的左侧的关系名,左侧的主体名,右侧的主体名是否都相同,如果相同的话,再看右侧的关系名是否不相同,如果不相同的话,那么这两条公理就存在潜在不一致性.

4 政府角色关系知识模型的实现

4.1 实现系统的设计

在政府角色关系知识模型的实现过程中,用参数来代替主体名进行分析.例如把主体名 1 定义为参数 x,主体名 2 定义为参数 y,主体名 3 定义为参数 z 等,在程序中将 x,y,z 参数定义为字符数组.

公理:上级下级关系(<场合>,<主体名 1>,<主体名 2>)->同事关系(<场合>,<主体名 1>,<主体名 2>)

对于上面的一条公理,可以把主体名替换为参数后,表示如下:

公理:上级下级关系(<场合>,<x>,<y>)->同事关系(<场合>,<x>,<y>)

在设计的验证程序中,角色关系通过逻辑公式来表示,逻辑公式的谓词就是政府关系名,个体词就是公理的主体名.把政府关系用逻辑公式表示出来,然后把所得的逻辑公式用字符数组表示,方便程序进行处理.

在设计的验证程序中,公理是用字符数组来表示的,即公理中的每个字符都是字符数组的一个字符.所有的公理采用二维字符数组来表示,二维数组的行数就是公理的条数,每一行字符就是一条公理.后面将讲到程序中所设计的参数提取函数,这个函数可以将公理中的所有参数提取出来.

在本研究中,政府角色关系公理首先是存于

word 文档中的,为了方便,当用程序进行语法错误检验,冗余性、潜在不一致性、包含性检验的时候,需要先把 word 文档转化为 txt 文件.程序运行时,通过屏幕提示用户输入 txt 文件的名字.输入名字结束后,程序在屏幕上显示出 txt 文件中所有的公理.显示结束后,程序提示用户是否进行公理的语法错误分析,输入‘Y’进行,输入‘N’就不进行;紧接着程序提示是否进行公理的冗余性、潜在不一致性、包含性的检验,输入‘Y’进行冗余性、潜在不一致性、包含性的检查并输出检查结果,输入‘N’则不进行这三性的检验.程序结束.

4.2 公理的冗余性、不一致性及包含性检查的实现

公理的冗余性检查是在函数 check_up(char s1[],char s2[])中实现的.函数的功能是先调用函数 judge_intersect(char s[])分别判断这两条公理 s1 和 s2 是形式一还是形式二,然后再根据它们的形式来调用参数提取函数(形式一则调用 extract_a 函数,形式二则调用 extract_b 函数)来提取出公理中的各种参数.最后根据这些参数来判断 s1 和 s2 是不是有冗余性.

如果两条公理的形式相同(形式不同的公理不存在冗余性)的话,那么判断它们所有的主体名是否对应相同,所有的群体名是否都相同,关系名是否都对应相同.如果这些参数都相同的话,那么这两条公理就存在冗余性,否则就没有冗余性.

接下来,我们把程序进行编译,无错误后,就可以运行了.运行时,屏幕首先提示用户输入公理所在的文件名(包括后缀名).如图 1 所示.

现在从文件中调入公理:
Input filename:(包括文件的后缀名)
—

图 1 程序提示用户输入公理所在文件名的结果

上面提到,把公理所在的 txt 文件命名为“role.txt”,因此我们在屏幕提示的下方输入“role.txt”,如图 2 所示.

现在从文件中调入公理:
Input filename:(包括文件的后缀名)
role.txt

图 2 用户在程序中输入公理所在的文件名的结果

程序接受用户的输入,把 txt 文档中的所有公理输出到屏幕,并显示文件中公理的个数,部分显示的公理如图 3 所示.

```

现在从文件中调入公理:
Input filename:(包括文件的后缀名)
role.txt
第一条公理为:
上级下级(<s1>,<a1>,<a2>)-上级下级(<s1>,<a2>,<a3>)->上级下级(<s1>,<a1>,<a3>)
第二条公理为:
上级下级(<s1>,<a1>,<a2>)-上级下级(<s1>,<a1>,<a3>)->同事(<s1>,<a2>,<a3>)
第三条公理为:
上级下级(<s1>,<a1>,<a2>)-上级下级(<s1>,<a1>,<a3>)->级别高于(<s1>,<a2>,<a3>)->上级下级(<s1>,<a2>,<a3>)
.....

```

图 3 程序在屏幕上输出所有公理的部分结果

屏幕把文档中的所有公理输出来之后,出现提示用户是否进行公理语法分析的界面,如图 4 所示。

```

是否进行公理的语法分析?
Y:进行语法分析
N:不进行语法分析
-

```

图 4 提示用户是否进行语法分析的界面

输入‘Y’,程序进行公理的语法错误分析,如果有语法错误,屏幕提示有错误,并指出错误的类型,方便用户改正。如图 5 所示。

```

是否进行公理的语法分析?
Y:进行语法分析
N:不进行语法分析
Y
现在进行公理的语法错误分析!
现在进行第 1 条公理的语法分析:
分析结果:
公理没有语法错误!
现在进行第 2 条公理的语法分析:
分析结果:
形式二,语法错误 28:公理中,'('和')'的数目不相等!
.....

```

图 5 进行语法错误分析的部分结果

语法分析结束后,屏幕提示用户是否进行公理的冗余性,潜在不一致性和包含性的检查,如图 6 所示。

```

是否进行公理的冗余性,潜在不一致性和包含性检查?
Y:进行语法分析
N:不进行语法分析

```

图 6 提示用户进行公理的冗余性,潜在不一致性和包含性检查的界面

当需要进行公理的冗余性,潜在不一致性和包含性检查的时候,我们输 Y,程序进行公理的冗余性,潜在不一致性和包含性检查。

5 结束语

对政府角色和政府关系进行研究,然后手工写

出政府关系公理,最后用程序实现对公理的语法错误、潜在不一致性、冗余性、包含性的检查。由于政府角色的定义具有不一定的特点,因此不同的人定义出来的政府角色可能是不一样的,又由于后面的政府关系的定义和政府关系公理的推导是在政府角色的基础上实现的,所以获取的关系和公理带有“不确定性”。然后,设计验证程序对公理的输入很方便,通过本程序可以快速地进行公理的语法错误、冗余性、潜在不一致性和包含性的检查。

参考文献:

- [1] Junwu Zhu, Jiandong Wang, Bin Li. A formal method for integrating distributed ontologies and reducing the redundant relations[J]. Kybernetes, 2009, 38(10): 1870-1879.
- [2] Maria Vargas-Vera, Enrico Motta, John Domingue, et al. Ontology driven semi-automatic and automatic support for semantic markup knowledge engineering and knowledge management: ontologies and the semantic web[C] // Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. USA: Hawaii, 2002(2473): 213-221.
- [3] Junwu Zhu, Jiandong Wang, Li Bin, Jiang Yi. On semantic-based cooperation among web services[C] // CSCWD 2006. China: Nanjing, 700-703
- [4] 杜小勇, 李曼, 王大治. 语义 Web 与本体研究综述[J]. 计算机应用. 2004(10): 14-16
- [5] 田春虎. 国内语义 Web 研究综述[J]. 情报学报, 2005(2): 243-249
- [6] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, et al. Pellet: a practical OWL-DL reasoner[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2007, 5(2): 41-50.
- [7] 徐天伟. 基于本体的领域知识库构建及推理研究[J]. 云南师范大学学报: 自然科学版, 2006(4): 7-10.
- [8] Soma R, Prasanna V. Parallel inferencing for OWL knowledge bases[C] // Proceedings of the 37th International Conference on Parallel Processing. Washington, DC: IEEE Computer Society, 2008: 75-82.

作者简介:



周 峰 男, (1967—), 实验师. 研究方向为数据挖掘等。