软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(11):2777−2795 [doi: 10.13328/j.cnki.jos.004920]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于保守扩充理论的模块化本体重用*

李 璞[1,2], 蒋运承[1], 王 驹[3]

[1](华南师范大学 计算机学院,广东 广州 510631)

[2](郑州轻工业学院 软件学院,河南 郑州 450000)

[3](广西师范大学 计算机科学与信息工程学院,广西 桂林 541004)

通讯作者: 李璞, E-mail: superlipu@163.com, http://www.scnu.edu.cn

摘 要: 分析了本体重用的研究现状和目前重用方法只适用于单个独立本体的不足,以$\mathcal{E}$-Connections 语言构建的模块化本体库为研究对象,基于保守扩充理论提出了本体模块知识完整性概念,并证明了知识完整性的相关性质.在此基础上,给出了一种针对模块化本体库的保守扩充重用算法 ERMMO(extracting reused modules from modular ontologies),讨论了该算法的两种子算法 $EMMO_{IK}$ 和 EMO_{IK} 的特点及适用条件.分析并验证了 ERMMO 算法的可行性和正确性.ERMMO 算法是当前保守扩充重用算法的一般化扩充,对模块化思想应用于本体重用问题有所启示.

关键词: 本体重用;保守扩充;模块化;$\mathcal{E}$-Connections 理论

中图法分类号: TP182

中文引用格式: 李璞,蒋运承,王驹.基于保守扩充理论的模块化本体重用.软件学报,2016,27(11):2777−2795. http://www.jos.org.cn/1000-9825/4920.htm

英文引用格式: Li P, Jiang YC, Wang J. Modular ontology reuse based on conservative extension theory. Ruan Jian Xue Bao/ Journal of Software, 2016,27(11):2777−2795 (in Chinese). http://www.jos.org.cn/1000-9825/4920.htm

Modular Ontology Reuse Based on Conservative Extension Theory

LI Pu[1,2], JIANG Yun-Cheng[1], WANG Ju[3]

[1](School of Computer Science, South China Normal University, Guangzhou 510631, China)

[2](Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450000, China)

[3](College of Computer Science and Information Engineering, Guangxi Normal University, Guilin 541004, China)

Abstract: In this paper, the current research progresses of ontology reuse is reviewed and the issue that current ontology reuse algorithms can merely be applied to a single independent ontology is addressed. Focusing on the modular ontologies with $\mathcal{E}$-Connections language, the IK_{MO} (integrity of knowledge about the module in an ontology) is presented based on the theory of conservative extension. The related properties of IK_{MO} are proved. Further, an algorithm for the ontology reuse with the conservative extension ERMMO (extracting reused modules from modular ontologies) is provided. The features and conditions of two sub-algorithms of ERMMO are discussed. Lastly, the feasibility and soundness of ERMMO are analyzed and verified. ERMMO is a generalization of the current reuse algorithms based on conservative extension theory, and can be served as the guidelines for reuse of modular ontologies.

Key words: ontology reuse; conservative extension; modularization; $\mathcal{E}$-Connections theory

本体模块化构建和本体重用的研究工作借鉴了当前软件工程中的成功经验[1].本体模块化是解决本体异

* 基金项目: 国家自然科学基金(61272066); 教育部新世纪优秀人才支持计划(NCET-12-0644); 广州市科技计划(2014J4100031, 201604010098); 广西可信软件重点实验室(桂林电子科技大学)研究课题(KX201419)

Foundation item: National Natural Science Foundation of China (61272066); Program for New Century Excellent Talents in University of Ministry of Education of China (NCET-12-0644); Program for Science and Technology of Guangzhou (2014J4100031, 201604010098); Project for Guangxi Key Laboratory of Trusted Software (Guilin University of Electronic Technology) (KX201419)

收稿时间: 2015-04-29; 修改时间: 2015-08-24; 采用时间: 2015-09-15

构问题的一个重要方法[2].当前,大部分对本体的研究和构建工作都是针对特定的项目需求或研究目的而展开的.在构建过程中,本体中的知识往往涉及到多个领域,从而可能将多个领域中的概念和关系引入到待开发的本体中,形成一个规模很大的混合本体.但这种大而全的本体不利于维护、推理和重用,不能很好地发挥本体自身的共享优势.为了解决上述问题,人们将软件工程中的模块化思想应用到本体构建当中,提出了模块化本体(也称为模块化本体库)的概念[1].ε-Connections[3,4]理论是本体模块化构建的一种方法,其主要思想是通过连接关系(linkproperty)将多个互不相交的逻辑解释域相连,是由传统描述逻辑的并集再加上与连接关系相关的构造因子构成的[3],能够使用 Tableaux 算法对本体库进行推理.

本体重用是软件工程思想在本体工程中的又一种体现,其目的是为了满足本体共享的需要[5].根据当前本体构建的需求,应用本体重用技术,可将已有本体中的部分或全部知识引入到当前正在构建的本体当中.重用技术避免了开发人员自身的知识局限性和重复开发带来的额外开销,减少了本体中知识推理的错误和资源浪费,是目前本体研究的热点之一[5].

针对本体重用,近些年人们做了大量的研究,大致上可以分为结构化和逻辑化两种思想.结构化方法并没有考虑知识间的语义关系,不能保证重用知识的安全性[5].为此,人们从知识间语义的逻辑关系出发,提出了一些新的思路.其中最有代表性的就是保守扩充(conservative extension)理论[6,7],该理论是为了保证安全地重用其他本体中的知识而提出的,使得在引入新的知识后,本体中不会产生不可预期的语义[6].随后,保守扩充理论又被引入到 XML 等更为广泛的领域[8,9].

然而,对本体模块化构建和本体重用的研究还是相对独立的.目前,本体重用的理论和方法都是以一个独立本体为重用对象,没有与本体的模块化构建相结合,从而无法很好地体现出软件工程理论中模块化和共享的特点和优势.因此,从一个模块化的本体库中抽取重用模块是一个亟待解决的问题.

基于上述问题,本文根据保守扩充理论,以应用ε-Connections 方法构建的模块化本体库为研究对象,对如何从一个模块化本体库中抽取重用模块进行了研究,提出了本体模块知识完整性的概念,并对相关的性质进行了证明.在此基础上,给出了一种更加一般化的本体重用算法——ERMMO(extracting reused modules from modular ontologies).该算法根据重用符号集 S,将存在于本体库多个模块中的相关知识抽取出来,并按照保守扩充理论,对这些知识进行筛选和重组,从而得到一个关于 S 的重用模块.此外,还对 ERMMO 算法的两种子算法($EMMO_{IK}$ 和 EMO_{IK})的特点及适用条件进行了分析.最后对算法的可行性和正确性进行了验证.

1 模块化本体库和本体重用

1.1 模块化本体库

目前,对本体模块化的研究虽然已经有了一定的进展,提出了几种模块化思想和语言,但由于各中方法具有不同的特点和优势,所以并没有形成一种统一规范的理论框架.表 1 列举了目前几种模块化语言和各自的特点.对于各种语言的具体思想和理论,可参见相关文献.

Table 1 Comparison of modular ontology languages

表 1 本体模块化语言的比较

模块化语言	特点	不足
分布式描述逻辑 DDLs (distributed description logics)[10]	实现了不同模块间概念的直接引用,要求本体模块互不相交	仅允许一种类型的域关系,不支持模块间角色的通信,表达能力十分有限
C-OWL[11]	定义了描述本体间关系的桥规则和描述这些桥规则的语法,具有较强的语义表达能力	没有推理支持
ε-Connections[3,4]	是一种框架逻辑,通过连接关系相互关联,每个连接的逻辑解释域互不相交	不允许模块间的概念进行直接引用
P-DL (package-based description logics)[12]	既允许模块间概念直接引用,又提供模块间的关系联接,放开了对模块间的严格不相交性	仍处于在理论研究阶段,没有成熟的工具实现
分布式动态描述逻辑 D3L (distributed dynamic description logic)[13]	是 DDL 的扩展,引入了“动态”的概念,可以为语义 Web 提供更为合理的逻辑基础	仍处于在理论研究阶段,没有成熟的工具实现

从表 1 不难看出,目前的本体模块化研究还不十分成熟.为了满足研究的需要,我们选择ε-Connections 作为模块化语言.首先,ε-Connections 有一个相对成熟的开发平台——Swoop[14],且支持表达能力较强的描述逻辑的 Tableaux 推理;此外,虽然ε-Connections 要求本体模块互不相交,且不允许模块间的概念进行直接引用,但是模块间的概念可以通过连接关系进行引用[4].这种限制更加符合模块化本体强内聚、松耦合的特点.

ε-Connections 的最大特点就是定义了连接关系.该连接是定义在其定义域所在的本体模块中,并由定义域模块指向值域模块的二元关系.定义域模块中的概念可以通过连接关系引用值域模块中的概念,如图 1 所示.有关ε-Connections 的相关理论和内容可以参见文献[3,4].

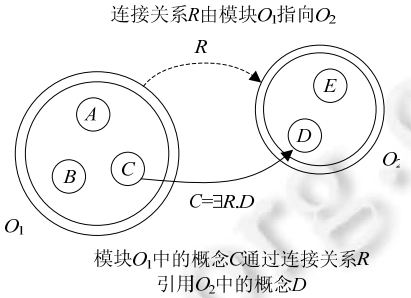

Fig.1 Structure of ε-Connections

图 1 ε-Connections 的结构

由图 1 可以看出,应用ε-Connections 构建的模块化本体库中的各个本体模块都具有相对的独立性,每个模块可以看作是一个规模较小的领域本体.但是它与当前普遍应用的独立本体又有本质不同,各个模块并非绝对独立于其他模块,而是通过少数连接关系相互关联.因此从重用的角度来看,当前应用于独立本体的重用算法显然不适用于这样的模块化本体库.这也正是本文研究的出发点.

在之前的工作中,已经对ε-Connections 构建模块化本体库的方法以及推理规则进行了研究.在实验中,针对具体的旅游领域,应用ε-Connections 理论构建了一个模块化本体库,并对构建和推理的复杂度与传统方法进行了比较和分析.具体研究内容可参见文献[15].

1.2 本体重用

最早的重用思想是比较简单的直接导入,即 OWL 提供的 import 功能[5].其思路是将一个本体 A 完全导入到另一个本体 B 中.但这种方法存在很多缺陷:一方面会造成冗余信息的增加,因为 B 可能只需要 A 中的部分而不是全部的知识;另一方面,这种简单的导入可能会产生一些新的语义,导致重用后本体中的知识推理结果与原来本体不一致,从而造成推理错误[6].为弥补上面的两个缺陷,人们提出了一些更加科学的重用方法.目前,对本体重用的研究主要分为两种思路:

(1) 结构化方法

a) Noy 等人在 2003 年提出一种本体重用的算法[16],该算法只是从知识的结构上考虑与符号集 S 的关系,实际上不一定所有包含重用符号的知识都与重用有关.所以,这种方法虽然简单易行,但不能保证重用模块中只包含相关的知识(即可能含有冗余知识).

b) Seidenberg 等人在 2006 年提出了一种应用于医学本体 GALEN 的重用算法[17].由于该算法是针对具体本体设计的,所以并不是一种通用的方法.此外,该方法只考虑 TBox 中的概念和关系,并没有对 ABox 中的断言进行判断.

(2) 基于逻辑的方法

a) Grau 等人在 2008 年提出了一种基于保守扩充理论的重用算法[18−20],该算法虽然可以保证重用模块的封闭性(即重用模块中包含了所有需要重用的知识)及安全性(即重用后的本体不会由于导入了重

用模块而产生新的语义),解决了结构化方法只考虑不同知识语法间的关系而不考虑知识内部语义间关系的缺点,但是它针对的重用对象只是一个独立本体(即从一个单个本体中抽取重用模块).

b) Kontchakov,Lutz 和 Shen 等人分别在 2009 年、2010 年和 2014 年针对保守扩充理论的不可分离性(inseparability)和复杂性(complexity)进行了研究,并提出了一些新的重用模块抽取算法[21−24].这些算法其实是对上面保守扩充算法的一种改进,在保证重用模块安全性的基础上能够得到最小的重用模块(即模块中包含的相关概念和公式最少).但这几种算法只是基于复杂度较低的描述逻辑所构建的本体,并不支持表达能力更强的描述逻辑.

1.3 保守扩充理论基础

保守扩充理论最早由 Ghilardi 等人在 2006 年提出[6],由 Grau 等人于 2008 年应用于从单个本体中抽取重用模块的算法中[20].限于篇幅,这里只介绍该算法涉及的相关概念和整体思路,具体的算法可以参见文献[20].

本体重用的目的是,将已有本体的部分或是全部知识引入到待开发的本体中[5].保守扩充思想就是要在引入新知识的同时,保证引入后得到的新本体并没有因为引入重用模块而使得被重用知识产生不可预期的新结论和不一致性[6].下面给出一个简单的例子.

例 1:设有本体 $O_1=\{A_1\sqsubseteq B_1\sqcap C_1,A_2\sqsubseteq B_1\sqcap C_2,B_1\equiv D\sqcap(\exists R_1.E)\}$,$O_2=\{C_2\sqsubseteq C_1,A_1\equiv\exists R_2.F,A_2\equiv\forall R_3.G\}$.$O_1$ 为了重用 O_2 中的概念 A_1 和 A_2,将 O_2 导入,得到 $O=O_1\cup O_2$.于是,在 O 中有下面的推理:

因为 $A_1\sqsubseteq B_1\sqcap C_1,A_2\sqsubseteq B_1\sqcap C_2,C_2\sqsubseteq C_1$,所以有 $O\models A_2\sqsubseteq A_1$.

但在初始情况下,$O_1\not\models A_2\sqsubseteq A_1$,且 $O_2\not\models A_2\sqsubseteq A_1$.

从上面例子中的推理不难看出,$A_2\sqsubseteq A_1$ 并不是开发人员所预期到的结论,而是在当 O_1 重用 O_2 后自动产生的.这种不可预期的结论会给原有本体带来新的语义,甚至会产生矛盾.保守扩充理论的提出很好地解决了这类问题.为了更好地理解后续内容,首先给出文献[20]中的几个重要概念(定义 1~定义 5).

定义 1(保守扩充(deductive conservative extension))[20]. 给定本体 O_1 和 O_2.令 $Sig(\alpha)$表示公式α中所包含的非逻辑符号集合(包括概念名、关系名、个体名等),S 是一个重用符号集合(由本体中需重用的概念名、关系名、个体名等构成).假定有 $O_2\subseteq O_1$,若所有满足 $Sig(\alpha)\subseteq S$ 的公理α都有:$O_1\models\alpha$,当且仅当 $O_2\models\alpha$,则称 O_1 是 O_2 关于 S 的保守扩充(deductive S-conservative extension).若 O_1 是 O_2 关于 S 的保守扩充,且 $S=Sig(O_2)$,则称 O_1 是 O_2 的保守扩充.

定义 2(本体安全性(safety for an ontology))[20]. 给定本体 O_1 和 O_2,若 $O_1\cup O_2$ 是 O_2 的保守扩充,则称 O_1 对 O_2 是安全的.

通过上面的定义可知,当 O_1 对 O_2 安全时,O_1 重用 O_2 后不会产生新的语义,从而保证了重用后得到的新本体 $O_1\cup O_2$ 和所重用模块(这里指 O_2)语义上的一致性.

但是,由于需要重用的模块往往并不是整个本体,而是本体的一部分,如果在满足安全性的条件下,每次都将被重用的本体整个导入,那么会使新本体中包含很多不必要的冗余知识,使得本体的规模无谓地扩大,导致推理复杂度的增加.因此在重用时,更希望只导入被重用本体中和重用符号集 S 相关的部分知识,也就是一个重用模块.

定义 3(重用模块(reuse module for an ontology))[20]. 给定本体 O_1,O_2 和 O_2',且 $O_2'\subseteq O_2$,若 $O_1\cup O_2$ 是 $O_1\cup O_2'$ 关于 $S=Sig(O_1)$的保守扩充,则称 O_2' 是 O_2 中关于 O_1 的重用模块.

该定义实际上表达的含义是,若 O_2' 是 O_2 中关于 O_1 的重用模块,那么当 O_1 需要重用 O_2 中已有的知识时,O_1 导入 O_2 和导入 O_2' 的效果是等价的.

至此,问题似乎已经解决,但不管是待开发的本体 O_1 还是被重用的本体 O_2 都在不断地更新.上面的定义都是针对静态本体而言的,如果本体发生变化,尤其是添加了新的知识,那么先前对安全性和重用模块的定义就会被破坏,必须重新按照定义进行判断和抽取,这样会降低重用模块的共享性.为了适应本体动态更新的特点,需要将上述定义 2 和定义 3 一般化.

定义 4(针对符号集的安全性(safety for a signature))[20]. 给定本体 O_1 和符号集 S,若对于每一个满足

$Sig(O_1)\cap Sig(O_2)\subseteq S$ 的本体 O_2,都有 O_1 对 O_2 是安全的($O_1\cup O_2$ 是 O_2 的保守扩充),则称 O_1 对 S 是安全的.

定义 5(针对符号集的重用模块(module for a signature))[20]. 给定本体 O_2 和 O_2' 及符号集 S,且 $O_2'\subseteq O_2$,若对于每一个满足 $Sig(O_1)\cap Sig(O_2)\subseteq S$ 的本体 O_1,都有 O_2' 是 O_2 中关于 O_1 的重用模块,则称 O_2' 是 O_2 中针对 S 的重用模块.

注:按照定义 5,从一个本体中抽取的关于 S 的重用模块并不唯一.因为该模块除了包含与 S 相关的必要知识α之外,还可能包含很多与 S 无关的知识.为了保证确定性以便于进行接下来的研究工作,后续提到的重用模块只包含和 S 中符号有关的必要知识,即最小重用模块.关于α的必要性和最小重用模块的介绍参见文献[20].

根据以上定义,图 2 给出了 Grau 等人的算法思路,其中,$\boldsymbol{O}(\cdot)$为一个安全类.关于安全类的概念及具体算法参见文献[20].

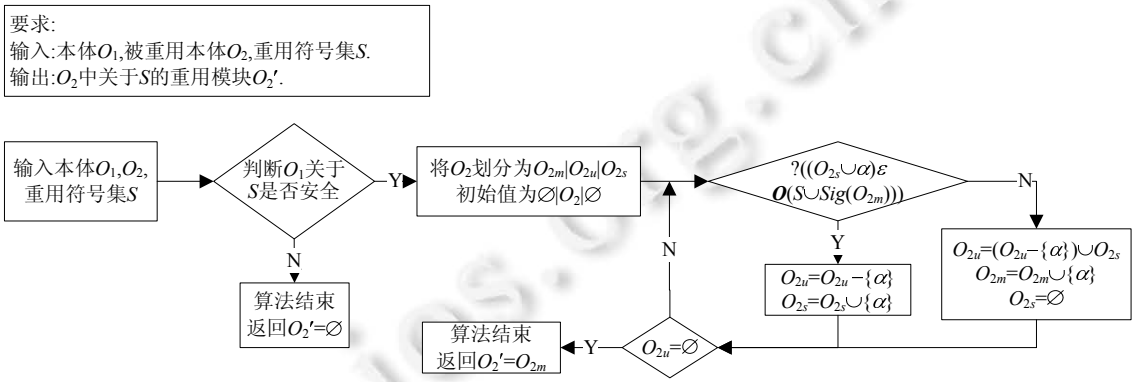

Fig.2 Conservative extension algorithm of extracting the reuse module from a single ontology

图 2 从单个本体中抽取重用模块的保守扩充算法

2 ERMMO 算法

2.1 本体模块的知识完整性

虽然图 2 中的算法保证了重用模块的封闭性及安全性,但是该算法只适用于从单个的独立本体中抽取关于重用符号集 S 的重用模块.ERMMO 算法是对上述算法的一种一般化扩充,旨在解决该算法无法从一个模块化本体库中(即一个或多个相互联系的本体模块中)抽取重用模块的问题.换言之,上述算法是 ERMMO 算法在模块化本体库中只包含一个本体模块时的特殊情况.

由第 1.1 节模块化本体库的特点可以看出,它与独立本体最大的不同就是独立本体与外界其他本体没有任何联系,而模块化本体库中的各模块之间存在连接关系,相互之间并不是绝对独立的.因此,如果按照上面的算法将本体模块视为独立的本体进行抽取,则会造成知识的缺失,破坏了保守扩充的封闭性.这也正是 ERMMO 算法所要解决的核心问题.

下面通过一个简单的例子对上述问题进行说明.

例 2:有一个模块化本体库 $O=\{O_1,O_2,O_3\}$,其中,O_1 通过连接关系 R 与 O_2 相连,O_1 中的概念 C 通过关系 R 引用了 O_2 中的概念 D(如图 1 所示).且有 $O_1=\{C\equiv\exists R.D,C\sqsubseteq\neg A\sqcap B\}$,$O_2=\{D\sqsubseteq E\sqcap F,F\equiv G\sqcup(\exists R_1.H),I\sqsubseteq D\}$,$O_3=\{J\equiv\neg K\}$.现在,开发人员在开发另一个本体 O^*时,要重用 O 中的概念 C 和 D(即重用符号集 $S=\{C,D\}$).如果按照上面的重用算法抽取(即,将本体模块 O_1 和 O_2 视为相互独立的本体),由于不考虑连接关系 R,则概念 D 在 O_1 中并不存在,从而有 $C\equiv\exists R.D\equiv\exists R.\bot$.得到的重用模块 $O'=\{C\equiv\exists R.\bot,C\sqsubseteq\neg A\sqcap B,D\sqsubseteq E\sqcap F,F\equiv G\sqcup(\exists R_1.H)\}$.虽然得到的 $O^*\cup O'$按照定义 1 并没有产生新的结论α,使得 $O^*\cup O_1'\models\alpha$,且 $O_1'\not\models\alpha$(其中,$Sig(\alpha)\subseteq S$),但重用模块 O'中并没有包含 $C\equiv\exists R.D$,而被 $C\equiv\exists R.\bot$替代.这显然造成了语义的错误和缺失,使得到的重用模块并没有包含所有与重用符号集 S 相关的

知识.

为了解决上述问题,本文以多个关联模块组成的模块化本体库为研究对象,根据第 1.3 节的保守扩充理论,提出了 ERMMO 算法.在给出该算法前,首先需要定义一些新的概念作为理论基础.

定义 6(关联知识(related knowledge in an ontology,简称 RK_O)). 给定本体 O(其中,O 可以是独立本体,也可以是模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$).对于任意 $\alpha,\alpha^*\in O$,若在 O 中存在一个序列$(\alpha_0,\alpha_1,\ldots,\alpha_n)$,使得 $\alpha_0=\alpha,\alpha_n=\alpha^*$,且 $Sig(\alpha_{i-1})\cap Sig(\alpha_i)\neq\varnothing$,其中,$i\in\{1,2,\ldots,n\}$,则称$\alpha$和$\alpha^*$互为关联知识,将此二元关系记为$\alpha\sim\alpha^*$.

定理 1. 二元关系“~”是一种等价关系.

由等价关系的定义可知,若“~”是等价关系,则必满足自反性、对称性和传递性.下面就分别从这 3 方面加以证明.

证明:

(1) 自反性.根据定义 6 可知,对于任意$\alpha\in O$,显然,O 中存在(α,α)使得 $Sig(\alpha)\cap Sig(\alpha)=Sig(\alpha)\neq\varnothing$,即$\alpha\sim\alpha$,因此,“~”是自反的.

(2) 对称性.根据定义 6 可知,对于任意$\alpha,\alpha^*\in O$,若有$\alpha\sim\alpha^*$,O 中必存在序列$(\alpha_0,\alpha_1,\ldots,\alpha_n)$,使得 $\alpha_0=\alpha,\alpha_n=\alpha^*$,且 $Sig(\alpha_{i-1})\cap Sig(\alpha_i)\neq\varnothing$,其中,$i\in\{1,2,\ldots,n\}$.由于 $Sig(\alpha_{i-1})\cap Sig(\alpha_i)=Sig(\alpha_i)\cap Sig(\alpha_{i-1})$,则序列$(\alpha_n,\alpha_{n-1},\ldots,\alpha_1)$满足定义 6 的条件,即$\alpha^*\sim\alpha$.因此,“~”是对称的.

(3) 传递性.根据定义 6 可知,对于任意$\alpha,\alpha',\alpha''\in O$,若有$\alpha\sim\alpha'$且$\alpha'\sim\alpha''$,则 O 中必定存在序列 $1(\alpha_0,\alpha_1,\ldots,\alpha_n)$,满足$\alpha_0=\alpha,\alpha_n=\alpha'$且 $Sig(\alpha_{i-1})\cap Sig(\alpha_i)\neq\varnothing$(其中,$i\in\{1,2,\ldots,n\}$)和序列 $2(\alpha'_0,\alpha'_1,\ldots,\alpha'_m)$满足 $\alpha'_0=\alpha',\alpha'_m=\alpha''$ 且 $Sig(\alpha'_{j-1})\cap Sig(\alpha'_j)\neq\varnothing$ (其中,$j\in\{1,2,\ldots,m\}$).由于 $\alpha_n=\alpha'_0=\alpha'$, 所以可将两个序列合并得到$(\beta_0,\beta_1,\ldots,\beta_{n+m})=(\alpha_0,\alpha_1,\ldots,\alpha_{n-1},\alpha_n,\alpha'_1,\ldots,\alpha'_m)$,使得$\beta_0=\alpha_0=\alpha,\beta_n=\alpha_n=\alpha'$, $\beta_{n+m}=\alpha'_m=\alpha''$ 且 $Sig(\beta_{i-1})\cap Sig(\beta_i)\neq\varnothing$,其中,$i\in\{1,2,\ldots,n+m\}$.根据定义 6 得$\alpha\sim\alpha''$,因此,“~”是传递的.

综上可知,“~”是一种等价关系. □

定义 7(本体模块知识完整性(integrity of knowledge about the module in an ontology,简称 IK_{MO})). 给定本体 O(其中,O 可以是独立本体,也可以是模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$),O 的子模块 O'和符号集 S.称 O 的子模块 O'对于 S 具有知识完整性,当以下条件成立:对$\forall\alpha\in O$,若$\exists\alpha^*\in O$,有 $Sig(\alpha^*)\cap S\neq\varnothing$且$\alpha^*\sim\alpha$成立,则$\alpha\in O'$.

由定义 7 的条件以及定理 1 的结论可知,对于 O 中所有包含 S 中符号的知识,其关联知识都在子模块 O'中.

下面继续通过例 2 对上述定义进行说明.

给定 $S=\{C,D\}$和本体库 $O=\{O_1,O_2,O_3\}$,对于 O_1 中的知识$\alpha=(C\equiv\exists R.D)$,有 $Sig(\alpha)=\{C,D,R\}$,且 $Sig(\alpha)\cap S=\{C,D\}\neq\varnothing$.又由定理 1 可知,$\alpha\sim\alpha$,所以$\alpha\in O'$.同理可得$(C\sqsubseteq\neg A\sqcap B)\in O'$,$(D\sqsubseteq E\sqcap F)\in O'$,$(I\sqsubseteq D)\in O'$.

对于 O_2 中的知识$\alpha'=(F\equiv G\sqcup(\exists R_1.H))$,$\exists\alpha^*=(D\sqsubseteq E\sqcap F)\in O$,且 $Sig(\alpha^*)\cap S=\{D\}\neq\varnothing$.又 O 中存在序列(α^*,α')满足 $Sig(\alpha^*)\cap Sig(\alpha')=\{F\}\neq\varnothing$,根据定义 6 有$\alpha^*\sim\alpha'$,所以$\alpha'\in O'$.

定义 7 表明:从一个模块化本体库中可以根据重用符号集 S,从各子模块中抽取出一个包含所有相关知识的子模块.

这里需要说明一点,满足定义 7 的子模块并不唯一.因为定义 7 中只说明对于满足条件的知识α,有$\alpha\in O'$,而对于不满足条件的知识是否属于 O'并没有进行限定.如上面的例子中,O_3 中的知识 $J\equiv\neg K$ 不满足定义 7 的条件,但 $O''=O_1\cup O_2\cup O_3$ 仍然符合定义 7 的要求,即对于 S 具有知识完整性.只是 O''中包含了诸如 $J\equiv\neg K$ 这样与 S 无关的冗余知识.

为了尽量减小子模块的规模,并保证它的唯一性,需要对定义 7 进行加强.

定义 8(最小知识完整性子模块(minimal module in an ontology for integrity of knowledge,简称 MMO_{IK})). 给定本体 O(其中,O 可以是独立本体,也可以是模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$)、O 的子模块 O'和符号集 S,称 O'是 O 中对于 S 的最小知识完整性子模块,当且仅当 O'满足以下条件:

(1) O'对于 S 知识完整性;

(2) 对$\forall\alpha\in O'$,必$\exists\alpha^*\in O'$,使得 $Sig(\alpha^*)\cap S\neq\varnothing$且$\alpha^*\sim\alpha$成立.

定义8通过条件(2)的过滤,将O中与S无关的冗余知识从定义7的O'中删除,从而保证O'中包含且只包含O中所有与S相关的知识.

直观上可以看出:定义8中描述的子模块与定义5中描述的重用模块并不相同,前者除了包含重用模块之外,还含有其他一些与重用符号集S相关的知识.接下来对定义5给出的重用模块和定义8中给出的最小知识完整性子模块进行形式化的比较,并对二者之间的关系进行讨论.

引理1. 给定本体O(其中,O可以是独立本体,也可以是模块化本体库$O=\{O_1,O_2,\ldots,O_m\}$)和符号集S,若O'是O中对于S的最小知识完整性子模块,O''是O中对于S的重用模块,那么O'与O''不一定相同.

证明:可以使用反证法.假设对任意满足条件的O'和O'',都有$O'=O''$.仍以例2为例,由定义8可得,O中对于$S=\{C,D\}$的最小知识完整性子模块$O'=O_1\cup O_2$.对于S的重用模块O'',由定义5可得,$O''=O_1\cup(O_2-\{I\sqsubseteq D\})=\{C\equiv\exists R.D, C\sqsubseteq\neg A\sqcap B, D\sqsubseteq E\sqcap F, F\equiv G\sqcup(\exists R_1.H)\}$,因为$\alpha=I\sqsubseteq D$不会在$S=\{C,D\}$的情况下对重用后的本体产生新的语义或是知识的缺失.此时,$O'\neq O''$,与假设矛盾.因此,引理1的结论成立. □

从引理1不难看出,虽然O'中包含且只包含O中与S相关的所有知识,但是这些知识对于符号集S的重用需求而言并不一定全是必要的.因此,可以对引理1进一步加强.

定理2. 给定本体O(其中,O既可以是独立本体,也可以是模块化本体库$O=\{O_1,O_2,\ldots,O_m\}$)和符号集S,若O'是本体O中对于S的最小知识完整性子模块,O''是O中对于S的重用模块,则有$O''\subseteq O'$.

为了证明定理2,首先证明另一个结论.

引理2. 若O'是O中对于S的重用模块,则O是O'关于S的保守扩充.

证明:可以使用反证法.假设O不是O'关于S的保守扩充.根据定义1,必$\exists\alpha^*$,其中$Sig(\alpha^*)\subseteq S$,使得$O\models\alpha^*$且$O'\not\models\alpha^*$.

不失一般性,不妨设存在一个O^*对于S是安全的,且$Sig(O^*)\cap Sig(O)\subseteq S$.由定义4有,$O^*\cup O$是$O$的保守扩充.即对于所有满足$Sig(\alpha)\subseteq S$的$\alpha$,都有:$O^*\cup O\models\alpha$,当且仅当$O\models\alpha$.又由假设条件$O\models\alpha^*$,所以,

$$O^*\cup O\models\alpha^* \tag{1}$$

因为O'是O中对于S的重用模块,由定义5可知,$O^*\cup O$是$O^*\cup O'$的保守扩充.根据定义1,由公式(1)可得:

$$O^*\cup O'\models\alpha^* \tag{2}$$

又$O'\subseteq O$,所以$Sig(O')\subseteq Sig(O)$,则$Sig(O^*)\cap Sig(O')\subseteq Sig(O^*)\cap Sig(O)\subseteq S$.因为$O^*$对于$S$是安全的,由定义4可知,$O^*\cup O'$是$O'$的保守扩充.根据定义1,由式(2)可得$O'\models\alpha^*$.此结论与假设$O'\not\models\alpha^*$矛盾,因此,引理2结论成立. □

下面对定理2进行证明.

证明(定理2):若要证明定理中的结论,只需证明:$\forall\alpha^*\in O''$,必有$\alpha^*\in O'$.

下面对O''中α^*所具备的特性进行讨论,进而分析α^*与O'的包含关系.

由于O''是O中对于S的重用模块,由引理2可知,对于所有满足$Sig(\alpha)\subseteq S$的α,都有:$O\models\alpha$,当且仅当$O''\models\alpha$.即O''中包含了所有和S的重用相关的必要知识.

不妨设$\Delta=\{\alpha|Sig(\alpha)\subseteq S, \text{且 } O\models\alpha\}$.则由引理2,对于任意给定的$\alpha\in\Delta$,有$O''\models\alpha$.这表明,在$O''$中必存在这样一个有限子集$\Delta^*=\{\alpha_1^*,\alpha_2^*,\ldots,\alpha_n^*\}$,使得$\Delta^*\models\alpha$,且对于任意$\alpha_i^*\in\Delta^*$,$\Delta^*-\{\alpha_i^*\}\not\models\alpha$,其中,$i\in\{1,2,\ldots,n\}$.

Δ^*具备以下两条性质:

(1) $Sig(\alpha)\subseteq(\cup Sig(\alpha_i^*)=Sig(\Delta^*)$,其中,$i\in\{1,2,\ldots,n\}$.

(2) 对于任意$\alpha_i^*,\alpha_j^*\in\Delta^*$,有$\alpha\sim\alpha_i^*\sim\alpha_j^*$.

由条件(1)和$Sig(\alpha)\subseteq S$可知,$(Sig(\Delta^*)\cap S)\supseteq Sig(\alpha)\neq\varnothing$.这说明$\Delta^*$中至少存在一个$\alpha_i^*$,其中,$i\in\{1,2,\ldots,n\}$,使得$Sig(\alpha_i^*)\cap Sig(\alpha)\neq\varnothing$,即$Sig(\alpha_i^*)\cap S\neq\varnothing$.再由条件(2)可知,$\forall\alpha^*\in\Delta^*$,必$\exists\alpha^{*\prime}\in\Delta^*$,使得$Sig(\alpha^{*\prime})\cap S\neq\varnothing$,且$\alpha^{*\prime}\sim\alpha^*$成立.又$\Delta^*\subseteq O''\subseteq O$,由定义8可得,$\alpha^*\in O'$.

由于假设中α在Δ中的任意性,以及定义5下面注明的重用模块O''的最小性,可以将此结论从$\alpha^*\in\Delta^*$扩充至$\alpha^*\in O''$,即$\forall\alpha^*\in O''$,必有$\alpha^*\in O'$.

综上可得定理中的结论,有 $O''\subseteq O'$. □

注:这里无法用类似的证明得到 $O'\subseteq O''$,因为上述证明中,Δ^*具备的两条性质均是必要的而不是充分的.这一点同样可以从引理 1 证明中的反例看出.

最后,讨论一下定义 8 给出的最小知识完整性子模块的唯一性.

定理 3(最小知识完整性子模块的唯一性). 若 O'和 O''都是本体 O 中对于 S 的最小知识完整性子模块,则必有 $O'=O''$.

这里首先要说明,一个集合中满足某种条件的最小子集并不一定唯一.一个类似的简单例子就是有向图中给定两点间的最短路径可能并不唯一.此外,一个本体中关于符号集的最小重用模块也并不唯一(对此的讨论可参见文献[20]).这也正是定理 3 的意义所在.

证明:同样使用反证法.假设存在两个 O 中对于 S 的最小知识完整性子模块 O'和 O''且 $O'\neq O''$,那么应有下面两种情况之一成立:(1) $\exists\alpha\in O'$但$\alpha\notin O''$;(2) $\exists\alpha\in O''$但$\alpha\notin O'$.不妨对第 1 种情况进行证明.

因为$\alpha\in O'$且 O'是本体 O 中对于 S 的最小知识完整性子模块,根据定义 8 条件(2)得:$\exists\alpha^*\in O'$,使得 $Sig(\alpha^*)\cap S\neq\varnothing$且$\alpha^*\sim\alpha$成立.又因为 O''是本体 O 中对于 S 的最小知识完整性子模块,由定义 7 的条件可知$\alpha\in O''$.与假设中的条件(1)$\alpha\notin O''$矛盾.同理可证假设中的条件(2)也不成立.所以假设错误.

综上可知,定理 3 中结论成立. □

由定理 3 可知,本体 O 中对于 S 的最小知识完整性子模块是唯一的.

至此,通过上面的定义和定理,对一个本体(库)关于某个给定符号集 S 的知识完整性进行了分析.

显然,定义 8 对于从一个独立本体中抽取重用模块的算法(如图 2 所示)来说意义不大.因为对于一个独立本体 O 而言,它本身就对所包含的知识具有知识完整性,只是不一定是最小的.所以在对独立本体进行重用模块抽取时,不必担心知识的缺失.如果在抽取前,先要根据 S 从中得到一个最小知识完整性模块 O',然后从 O'中抽取重用模块,未免有些画蛇添足.但是对于模块化本体库而言,定义 8 十分重要.因为与 S 有关的知识往往不会完全存在于一个本体模块中,而是通过连接关系,由多个模块中的知识相互引用所组成的.若将所有模块合并成一个独立本体,再从中抽取,显然会将大量无关的知识引入当前本体,使得合并后的本体规模过大,从而造成算法的效率大为降低.

直观上讲,随着大数据时代的到来,本体(库)中知识的数量往往会不断增长,而针对知识重用所给出的相关重用符号集 S 一般则相对固定.显然,本体(库)中每次新增的知识并不一定都与 S 相关.若每次更新后都漫无目的地将本体(库)进行合并,然后按照图 2 算法对其中的每一条知识进行安全性判断,则该算法的复杂度会随着本体(库)复杂度及规模的增大而显著增加.然而,根据本节的定义及相关定理可知,在获得最小知识完整性子模块后,对于那些不属于该模块的知识,不必对其安全性进行进一步判断.这样,对于大规模的本体(库)而言,由于最小知识完整性子模块排除了大量与 S 无关的冗余知识,所以其规模更小,复杂度更低.这样,应用 ERMMO 算法对该模块进行重用,能够避免很多不必要的安全性检测,从而能够提高重用算法的执行效率.这一结论也可从后面的实验数据及相应的分析中得到验证.

因此,ERMMO 算法的核心思想就是实现从一个模块化本体库中根据重用符号集 S,得到一个关于 S 的最小知识完整性本体,然后再从中抽取重用模块.

2.2 ERMMO算法的实现

从上面的分析不难发现,ERMMO 算法的实现可以分为两个阶段:第 1 阶段是根据重用符号集 S 从一个或多个本体模块中获得关于 S 的最小知识完整性子模块——$EMMO_{IK}$(extracting the minimal module in an ontology for integrity of knowledge);第 2 阶段是应用图 2 中的重用算法,从第 1 阶段得到的子模块中抽取出关于 S 的重用模块.

2.2.1 第 1 阶段:$EMMO_{IK}$ 算法

根据第 2.1 节定义 8 中最小知识完整性子模块所满足的条件,下面给出 $EMMO_{IK}$ 算法.

首先,针对 O 中的某一个本体模块 $O_M\in\{O_1,O_2,\ldots,O_m\}$,对于给定的符号集 S,给出从 O_M 中获得关于 S 的最

小知识完整性子模块 O'_M 的算法,见算法 1.

算法 1. *Extract*(O_M,S_M).

输入:给定一个基于ε-Connections 构建的模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$中的某个子模块 O_M,一个重用符号集 S_M,且 S_M=Sig(O_M)$\cap S$.

输出:O_M 中关于 S_M 的最小知识完整性子模块(MMO$_{IK}$): O'_M.

具体步骤:初始 $O^*_M=\varnothing, O'_M=\varnothing, O'^*_M=\varnothing$, $O^*=\varnothing$.

(1) $O^*_M=O_M$ //为了防止在递归的时候 O_M 被其他本体模块调用,这里需要使用 O_M 的副本 O^*_M

(2) **while** (Sig(O^*_M)$\cap S_M\neq\varnothing$) **do** //将 O_M 中所有与 S_M 相关的知识抽取出来,放入 O'_M

(3) select $\alpha\in O^*_M$

(4) **if** ($Sig(\alpha)\cap S_M\neq\varnothing$ **and** $\alpha\notin O'_M$)

(5) $O'_M=O'_M\cup\{\alpha\}$

(6) $S_M=S_M\cup Sig(\alpha)$

(7) $O^*_M=O^*_M-\{\alpha\}$

(8) **end if**

(9) **end while**

(10) $O'^*_M=O'_M$ //为了对 O'_M 中的每一个α进行检测,同时保留 O'_M ,需要使用 O'_M 的副本 O'^*_M

(11) **while** ($O'^*_M\neq\varnothing$) **do** //对于 O'_M 中包含连接关系的α,将所连接模块中的相关知识放入 O'_M

(12) select $\alpha\in O'^*_M$

(13) **if** ($Sig(\alpha)$ has "**LinkProperty**")

(14) O^*=the module in O, which is linked with O_M by "**LinkProperty**"

(15) S^*=the Signature in $Sig(\alpha)$ linked from O^*

(16) $O'_M=O'_M\cup$***Extract***(O^*,S^*)

(17) **end if**

(18) $O'^*_M=O'^*_M-\{\alpha\}$

(19) **end while**

(20) **return** O'_M

可以看出,算法 1 中使用了递归方法.针对一个模块中包含连接关系的关联知识循环调用该算法,将相关本体模块中的关联知识也放入所生成的模块中.这种采用深度优先策略的算法,可以从 S 中的某一个重用符号出发,沿着连接关系将所有关联知识从不同本体模块中抽取出来,直到相关知识中不再有连接关系出现.

接下来,对于 $O=\{O_1,O_2,\ldots,O_m\}$中的每一个模块循环调用算法 1,最终获得 O 中关于 S 的最小知识完整性子模块 O',见算法 2.

算法 2. EMMO$_{IK}$ 算法:*EMMO*$_{IK}$(O,S).

输入:给定一个基于ε-Connections 构建的模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$,一个重用符号集 S,且 $S\subseteq Sig(O)$.

输入:O 中关于 S 的最小知识完整性子模块(MMO$_{IK}$):O'.

具体步骤:初始 $O'=\varnothing$.

(1) **for** (i=1 to m)

(2) $S_i=S\cap Sig(O_i)$

(3) **if** ($S_i\neq\varnothing$) **do**

(4) $O'=O'\cup$***Extract***(O_i,S_i)

(5) **end if**

(6) **end for**

(7) **return** O'

通过算法 2,对 O 中的每一个本体模块循环调用算法 1,从而获得了 O 中关于 S 的最小知识完整性子模块 O'.这里说明一点:由于ε-Connections 理论要求各个模块的逻辑解释不相交(见表 1),所以可以按照算法 2 中的步骤(1)和步骤(2)得到集合 S 的一个划分,即不会遗漏 S 中的符号.

下面对算法 1 和算法 2 的正确性进行分析.

由定义 8 可知,关于 S 的最小知识完整性子模块中包含且只包含所有与 S 相关的知识.算法 1 中,步骤(2)~步骤(9)的第 1 个 while 循环就是将某个模块中所有与 S 相关的知识抽取出来,步骤(11)~步骤(19)的第 2 个 while 循环又通过对连接关系的检测,采用深度优先策略,将其他关联模块中与 S 相关的知识也抽取出来.接着,通过算法 2 中的 for 循环对 O 中每个模块循环调用算法 1,将每次循环的结果进行合并,最终得到 O'.显然,通过算法 1 和算法 2 得到的 O'符合定义 8 中的条件,即为关于 S 的最小知识完整性子模块.

2.2.2 $EMMO_{IK}$ 算法的变形:EMO_{IK} 算法

下面讨论 $EMMO_{IK}$ 算法的适用情况.

当本体库中的本体模块较少、每个模块中的概念和公式很多时,模块中包含的与 S 无关的冗余知识相对较多.此时应用 $EMMO_{IK}$ 算法,可以约减掉很多冗余知识,能够明显提高第 2 阶段抽取重用模块的效率.但由于该算法是一种递归算法,算法的复杂度相对较高.当本体库中的模块很多、每一个模块中的概念和公式较少时,各模块中包含的与 S 无关的冗余知识也相对较少.如果在这种情况下仍然应用该算法对每一项知识进行检测,虽然能够约减冗余知识,但数量不会很多,这样显然会降低算法的效率.因此针对这种情况,为了进一步提高算法的效率,可以对 $EMMO_{IK}$ 算法进行改进,下面给出改进后的 EMO_{IK}(extracting the module in an ontology for integrity of knowledge)算法.

EMO_{IK} 算法同样也要从模块化本体库 O 中得到一个对重用符号集 S 具有知识完整性的子模块 O',只是 O' 不一定是最小的 MMO_{IK}.该算法的思路是:在本体模块较多而各模块的规模较小时,不再针对 S 中的符号从各模块中选择关联知识放入 O',而是将包含相关知识的模块直接并入 O'.这样做虽然会使 O' 中含有一些冗余知识,但由于并入的本体模块规模较小,所以模块中包含的冗余知识也较少.这样在第 2 阶段重用模块时,对少量冗余知识执行算法的开销不会很大.

此外,EMO_{IK} 也并不是简单地将所有的模块都合并在一起,而是只合并通过连接关系与含有 S 相关知识的模块相连的模块.因此,与 S 无关的模块仍然会被约减.

对于 EMO_{IK} 算法,同样需要设计两种子算法.

首先,对于 S,将包含关联知识的本体模块 O_M 并入 O'.同时,因为关联知识中可能包含连接关系,所以将与 O_M 存在连接关系的其他本体模块也并入 O',见算法 3:

算法 3. *Combine*(O_M).

输入:给定一个ε-Connections 模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$中的某个子模块 O_M.

输出:O 中所有通过连接关系与相连 O_M 相连的模块的并集(含 O_M 本身): O'_M .

具体步骤:初始 $O^*=\varnothing$, $O'_M=\varnothing$,k=0.

(1) $O'_M = O'_M \cup O_M$

(2) **if** (O_M has "**LinkProperty**") //将所有与 O_M 通过连接关系相连的模块合并到 O'中

(3) k =the number of "**LinkProperty**" in O_M

(4) **for** (i=1 to k)

(5) O^*=the module in O, which is linked with O_M by "**LinkProperty**"

(6) **if** ($O^* \not\subset O'_M$) //对于两个模块相互连接,形成一个环的情况,防止产生死循环

(7) $O'_M = O'_M \cup O^*$

(8) ***Combine***(O^*)

(9) **end if**

(10) **end for**

(11) **end if**

(12) **return** O'_M

然后,对于 $O=\{O_1,O_2,\ldots,O_m\}$中的每一个模块循环调用算法 3,最终获得 O 中关于 S 的知识完整性子模块 O',见算法 4:

算法 4. EMO_{IK} 算法:$EMO_{IK}(O,S)$.

输入:给定一个基于ε-Connections 构建的模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$,一个重用符号集 S,且 $S\subseteq Sig(O)$.

输出:O 中关于 S 的知识完整性子模块(MO_{IK}):O'.

具体步骤:初始 $O'=\varnothing$.

(1) **for** (i=1 to m)

(2) **if** ($Sig(O_i)\cap S\neq\varnothing$ **and** $O_i\not\subset O'$)

(3) $O'=O'\cup$***Combine***(O_i)

(4) **end if**

(5) **end for**

(6) **return** O'

同样,对上面两种算法的正确性进行分析.

(1) 与算法 1 类似,算法 3 也采用了递归的方法将某个相关模块以及和它相连的其他模块求并.算法 3 中的步骤(5)~步骤(9)与算法 1 中的步骤(14)~步骤(16)类似,但算法 3 中对所有的连接关系全部进行了递归.此外,算法 3 是直接将模块求并,忽略了算法 1 中步骤(2)~步骤(9)对模块中每个α进行的筛选.显然,算法 3 中的结果包含算法 1 的结果.

(2) 算法 4 和算法 2 使用同样的循环策略分别调用算法 3 和算法 1,因此算法 4 的结果包含算法 2 的结果.

(3) 综上,由前面对算法 1 和算法 2 的正确性分析可知,通过算法 3 和算法 4 得到的 O'符合定义 7 中的条件,即:对于 S 具有知识完整性,当然也包含定义 8 中关于 S 的最小知识完整性子模块.

根据上面的分析,结合定理 2,有下面两条推论(对这两个推论的证明可参见之前对于 4 个算法正确性的分析和定理 2 的证明,此处省略).

推论 1. 给定一个模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$和重用符号集 S,O'是 EMO_{IK} 算法输出的结果,O''是 EMMO_{IK} 算法输出的关于 S 的最小知识完整性子模块,则有 $O''\subseteq O'$.

推论 2. 给定一个模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$和重用符号集 S,O'是 EMO_{IK} 算法输出的结果,O''是 O 中对于 S 的重用模块,则有 $O''\subseteq O'$.

2.2.3 第 2 阶段:整合后的 ERMMO 算法

由定理 2 和推论 2 可知,无论应用 EMMO_{IK} 算法或 EMO_{IK} 算法,都可以获得一个包含 O 中关于 S 的重用模块的子模块.因此,可以将上面算法中获得的子模块视为被重用的目标本体,作为图 2 中算法的输入,从中抽取一个关于 S 的重用模块.将 EMMO_{IK} 算法或 EMO_{IK} 算法和图 2 中算法进行整合,可以实现从模块化本体库 O 中根据重用符号集 S 抽取重用模块的 ERMMO 算法.

首先,对 EMMO_{IK} 算法和 EMO_{IK} 算法的选择进行讨论.需要根据本体库的特点选择其中一种算法来执行,从而尽可能地提高整体 ERMMO 算法的效率.这里只给出一个简单的判别规则,至于一些更加复杂和完善的方法,会在后续的研究工作中进一步探索.

由于选择 EMMO_{IK} 算法和 EMO_{IK} 算法的根本出发点是比较模块化本体库中本体模块数和各模块中包含知识数量的多少,因此判断标准也是根据这两个参数的相对值给出的.

定义 9(模块化本体库的平均概念数(average number of concepts in modular ontologies,简称 ANC_{MO})). 有模块化本体库 $O=\{O_1,O_2,\ldots,O_m\}$($m\geqslant 2$),令$\sum_C$为 O 中所有概念和公式的数量总和,$\sum_O$为 O 中所有本体模块的数量总和,且$\sum_O\geqslant 2$,则称 $ANC_{MO}=\sum_C/\sum_O$为 O 的平均概念数.

引入参数θ,并将$ANC_{MO}=\theta$作为一个阈值,从而有下面的选择算法:

(1) 当$ANC_{MO}\leqslant\theta$时,执行 EMO_{IK} 算法;

(2) 当$ANC_{MO}>\theta$时,执行 $EMMO_{IK}$ 算法.

需要说明的是,定义 8 中之所以限定 O 中的模块数$\sum_O\geqslant 2$,是因为当$\sum_O=1$ 时,本体库 O 就变成了一个独立本体,这时采用 $EMMO_{IK}$ 算法,会降低整个抽取算法的效率;而采用 EMO_{IK} 算法,显然得到的结果仍然是独立本体 O 本身.因此,对于独立本体不再执行上面两种算法,也不必使用定义 9 中的 ANC_{MO} 进行判断,而是按照图 2 的算法直接进行抽取.

从前面第 2.2.1 和第 2.2.2 节对 $EMMO_{IK}$ 算法和 EMO_{IK} 算法正确性的分析,显然可以得到 ERMMO 算法的正确性.由图 3 中给出的算法流程可知,图 2 中的算法是当被重用本体库中的本体模块数$\sum_O=1$(即被重用对象为一个独立本体)时的一种特殊形式.所以,ERMMO 算法是 Grau 等人提出算法的一般化扩充,是一种更加通用的算法.

整合后的 ERMMO 算法如图 3 所示.

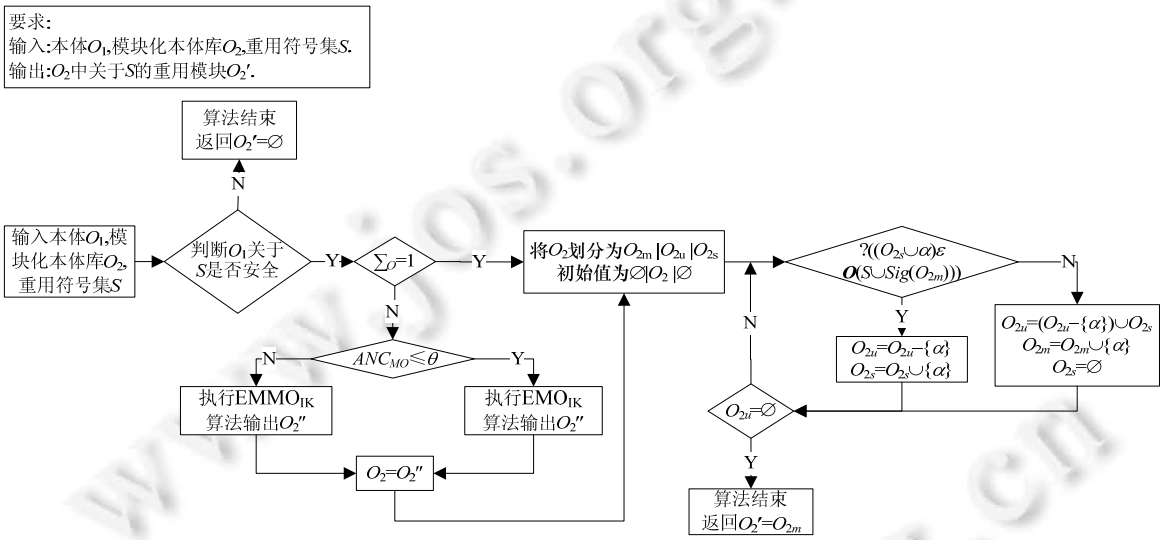

Fig.3 ERMMO algorithm

图 3 ERMMO 算法

3 ERMMO 算法实验分析

为了对 ERMMO 算法的正确性和可行性做进一步的验证,本文以旅游领域为研究对象,基于ε-Connections 理论,使用 Swoop 平台构建了一个旅游领域的模块化本体库,如图 4 所示.Swoop 是用 Java 开发的一个开放源码的本体编辑器.它可以支持ε-Connections 中的连接关系以及基于该理论的 pellet 推理机[25].但是,由于 Swoop 对于汉字编码的处理不稳定,所以在构建时均采用英文编码(具体的构建和推理过程可参见文献[15]中的相关研究结果).

将上面构建并推理完成的模块化本体库作为重用对象,根据给定的重用符号集 S,对 ERMMO 算法进行了实现.在此过程中,分别使用 $EMMO_{IK}$ 算法和 EMO_{IK} 算法抽取出对于 S 具有知识完整性的子模块,然后将此模块导入 Protégé[26].Protégé 是目前构建本体时应用最广泛的编辑工具之一,它是用 Java 开发的一个开放源码的本体编辑器.该编辑器支持对独立本体进行 Tableaux 推理的 pellet 推理机,并可以实现对本体的查询.之所以将 $EMMO_{IK}$ 算法和 EMO_{IK} 算法输出的子模块导入 Protégé,是由于 Grau 等人的算法需要在该平台上面实现[27,28],这样做更便于对实验数据进行对比和分析.

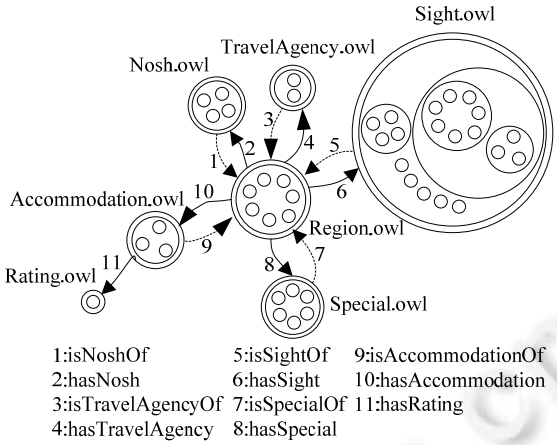

Fig.4 Modular framework of ontologies for tourism domain

图 4 旅游领域本体库的模块化框架

这里需要说明一点,从前面给出的 $EMMO_{IK}$ 算法和 EMO_{IK} 算法可知,若对图 4 给出的本体库执行这两种算法,由于多个模块之间都存在互逆的连接关系,所以进行抽取及合并的时候,很容易通过连接关系使所有的模块都参与运算,导致最终获得的输出结果和独立本体一样.实际上,模块中的大部分知识同这些互逆的连接关系是无关的.所以对于大多数重用符号集而言,如果仍然保持上面的互逆关系,那么 ERMMO 算法尤其当采用子算法 EMO_{IK} 时,很难体现出自身的优势,这样就失去了对比的意义.为此,在不改变本体库中各模块所包含知识数量的前提下,可将本体库中互逆的连接关系进行约减,只保留一个单向的关系,即,只保留图 4 中实线标注的连接关系.由于对于 ERMMO 算法和 Grau 等人的算法同时采用简化后的本体(库)作为实验对象,所以并不影响结果对比中实验数据的公平性.具体实验结果见表 2 和表 3.

虽然该模块化本体库并没有包含所有旅游领域的知识,但根据表 2 和表 3 的实验结果,仍然可以对 ERMMO 算法的特点进行分析.

(1) 从被重用的目标本体来看,ERMMO 算法抽取的对象并不是各模块简单合并后的独立本体,而是一个对重用符号集 S 具备知识完整性的子模块.这个子模块无论从规模上还是复杂程度上都比之前的独立本体要小,如图 5 所示.从图 5 中可以看出,使用 $EMMO_{IK}$ 算法获得的子模块规模小于使用 EMO_{IK} 算法获得的子模块.这也从实验的角度验证了第 2.2.2 节推论 1 中的结论.

(2) 当被重用的目标本体为独立本体时,这种大而全的本体中大部分知识是与重用符号集 S 无关的冗余知识.图 2 的算法中,对于这种大量的冗余知识也需进行计算,这显然会影响重用算法的效率.而对于 ERMMO 算法,由于通过其子算法 $EMMO_{IK}$ 和 EMO_{IK} 获得的子模块中所有或者大部分的概念和公理都和重用符号集 S 相关,因此在第 2 阶段的抽取过程中,能够提高重用算法的执行效率.对被重用的目标本体中包含冗余知识的比较如图 6 所示.对重用知识在被重用的目标本体中所占比例的比较如图 7 所示.

(3) 从得到的重用模块来看,由定理 2 和推论 2 可知,虽然被抽取的目标本体规模变小,但由 ERMMO 算法抽取的重用模块与从独立本体中抽取的重用模块仍然相同.这个结论可从表 2 中两种算法的“重用模块概念数”一列得到验证.

Table 2 Comparison of experimental results between the ERMMO algorithm and Grau's (1)

表 2 ERMMO 算法与 Grau 等人的算法实验结果比较(1)

算法	ERMMO 算法				Grau 等人的算法	
重用符号集 S	总概念数	被重用目标本体概念数		重用模块概念数	被重用目标本体概念数	重用模块概念数
		算法 2:$EMMO_{IK}$	算法 4:EMO_{IK}			
RestHouse	207	14	22	9	207	9
RestHouse Mountain_Water	207	63	138	42	207	42
Accommodation Museum Mountain_Water HistoricalPlace	207	103	138	97	207	97
Accommodation City Sight DomesticTravelAgency	207	141	207	122	207	122

Table 3 Comparison of experimental results between the ERMMO algorithm and Grau's (2)

表 3 ERMMO 算法与 Grau 等人的算法实验结果比较(2)

算法	ERMMO 算法						Grau 等人的算法	
重用符号集 S	被重用目标本体复杂度		获得被重用目标本体耗时(ms)		重用模块抽取时间(ms)		被重用目标本体复杂度	重用模块抽取时间(ms)
	$EMMO_{IK}$ 算法	EMO_{IK} 算法	$EMMO_{IK}$ 算法	EMO_{IK} 算法	$EMMO_{IK}$ 算法	EMO_{IK} 算法		
RestHouse	$\mathcal{ALCO}$	$\mathcal{ALCO}$	87	54	31	69	$\mathcal{SHOI}$	92
RestHouse Mountain_Water	$\mathcal{SOI}$	$\mathcal{SOI}$	331	217	102	198	$\mathcal{SHOI}$	326
Accommodation Museum Mountain_Water HistoricalPlace	$\mathcal{SOI}$	$\mathcal{SOI}$	549	221	286	335	$\mathcal{SHOI}$	570
Accommodation City Sight DomesticTravelAgency	$\mathcal{SOI}$	$\mathcal{SHOI}$	596	506	352	635	$\mathcal{SHOI}$	622

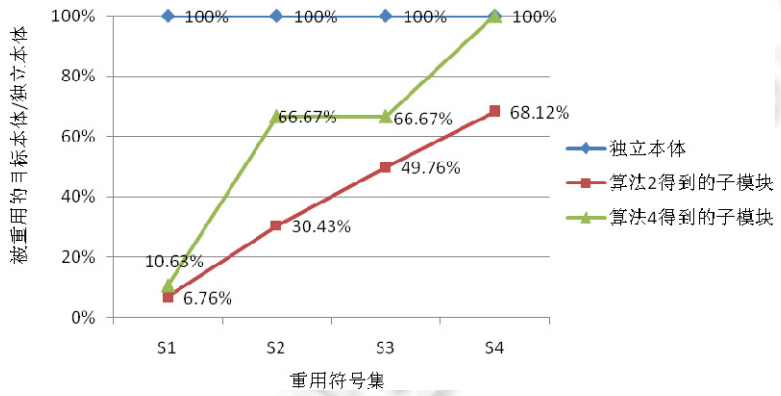

Fig.5 Comparison of the size between target ontology and independent ontology

图 5 被重用目标本体与独立本体规模的比较

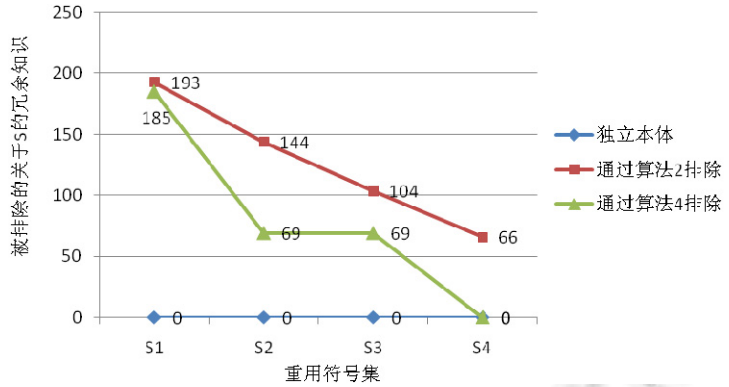

Fig.6 Redundant knowledge irrelevant to S filtered by target ontology

图 6 被重用目标本体中排除的与 S 无关的冗余知识

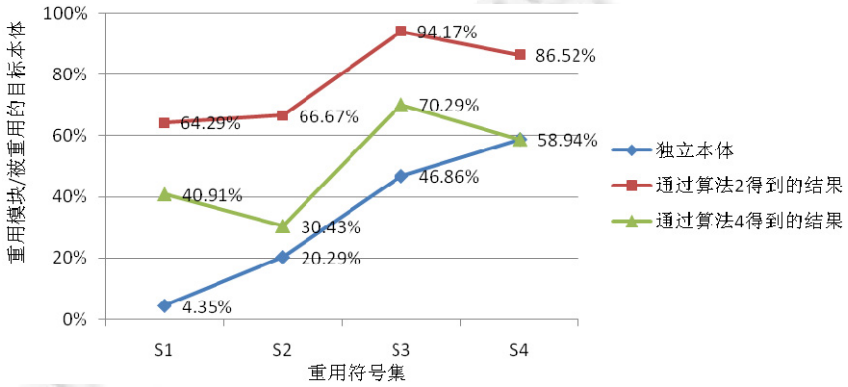

Fig.7 Percentage of the reuse module in target ontology

图 7 重用模块在目标本体中所占比例

下面再从 ERMMO 算法的运行时间方面对该算法的效率进行分析.

(1) 从 $EMMO_{IK}$ 算法和 EMO_{IK} 算法获得的被重用目标本体来看,由于 $EMMO_{IK}$ 算法过滤掉了所有与 S 无关的冗余知识,因此通过该算法获得的被重用目标本体的复杂度不会超过 EMO_{IK} 算法.但是由于 $EMMO_{IK}$ 算法需要对各个模块中的每条知识是否与给定的 S 相关进行判断,所以在算法的复杂度上要高于 EMO_{IK} 算法.因此在获得被重用目标本体的第 1 阶段,$EMMO_{IK}$ 算法的执行时间要高于 EMO_{IK} 算法,如图 8 所示.

(2) 在重用模块抽取时间方面,由于 $EMMO_{IK}$ 算法获得的被重用目标本体规模最小,所以在应用图 2 算法进行重用模块抽取时,基于 $EMMO_{IK}$ 算法的 ERMMO 算法执行时间最短.而由于 Grau 等人算法的被重用目标本体是独立本体,其中包含的知识是模块化本体库中知识的总和,所以抽取重用模块时算法的执行时间最长,如图 9 所示.

(3) 在算法的整体运行时间方面,由于 ERMMO 算法分为两个阶段,所以总体的执行时间为 t="获得被重用目标本体耗时"+"重用模块抽取时间".而 Grau 等人的算法由于是从独立本体中直接抽取重用模块,所以算法执行的时间 t'="重用模块抽取时间".由从表 3 的数据可以看出,t 并没有比 t'更短,大部分情况下甚至更长.具体比较结果如图 10 所示.但由于模块化本体库自身结构的特点及其对于本体工程的意义,所以 ERMMO 算法在时间上的代价也是不可避免的.如何进一步对文中算法进行优化,从而提高运算效率,将会在今后的研究工作中继续探索.

(4) 由图 10 的比较结果可见,虽然 ERMMO 算法的运行时间与 Grau 等人的算法相比并没有优势,但是 Grau

等人提出的算法在进行重用模块抽取时,对于一个独立本体,算法要遍历其中的每一个概念或公理,也就是对每一条知识都要判断该概念或公理是否属于安全类 $\boldsymbol{O}(S)$.通常情况下,对于给定的重用符号集 S,独立本体中大部分概念和公理都是与 S 无关的冗余知识.此时按照图 2 的流程,对某个概念或公理是否属于安全类 $\boldsymbol{O}(S)$进行判断时,算法很多时间都浪费在了对冗余知识的判断上.实际上这种判断是多余的,因为这些冗余知识本身就与 S 无关,自然属于安全类 $\boldsymbol{O}(S)$.因此,Grau 等人的算法在重用模块抽取的过程中做了许多无用功.该算法的效率会随着本体中所包含知识的增多而降低.

相比之下,ERMMO 算法在第 1 阶段得到的关于 S 的知识完整性子模块中,已将许多与 S 无关的冗余知识排除,其中所包含的概念和公理的数量比单个的独立本体要少很多.因此在执行图 3 的算法时,可以避免许多对冗余知识是否属于安全类 $\boldsymbol{O}(S)$的判断,从而节省了算法第 2 阶段抽取重用模块时的运行时间.

(5) 此外,从表 2 和表 3 不难看出,无论是文中的 ERMMO 算法(包括两种子算法)还是 Grau 等人的算法,针对不同的重用符号集 S,每个算法输出的模块中所包含的概念数和执行时间都有较大的差别,且没有明显的规律.造成这种结果的原因不仅和所构建本体自身的复杂度和知识之间的关联性有关,也与符号集 S 的选取有关.因此对于上述几种算法,很难针对所有的实验情况得到统一的结论.

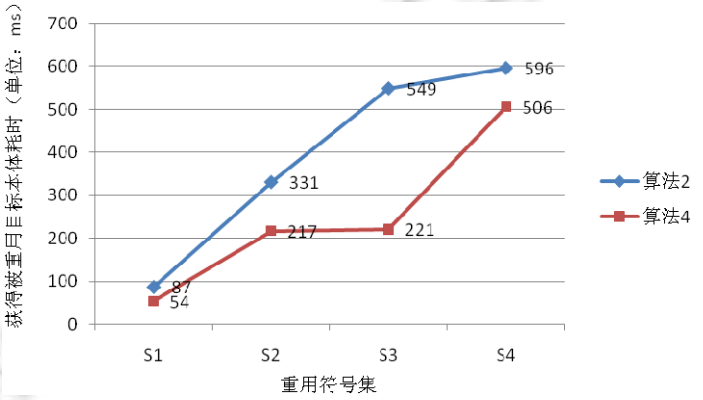

Fig.8 Runtime of two sub-algorithms of ERMMO algorithm (getting the target ontology)

图 8 ERMMO 算法的两种子算法的执行时间(获得目标本体)

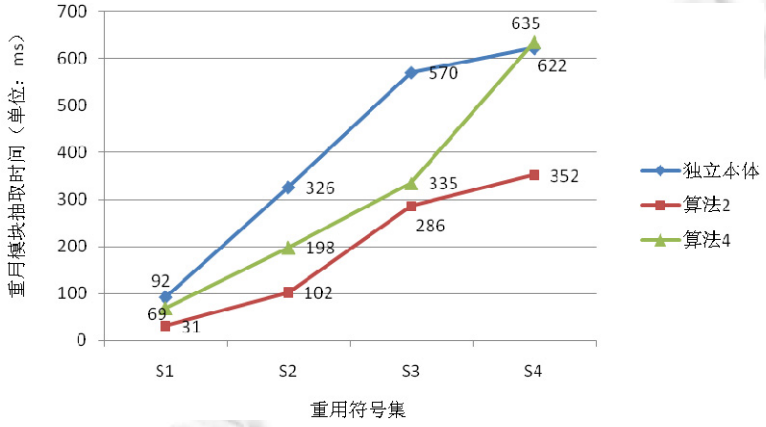

Fig.9 Runtime of extracting the reuse module

图 9 抽取重用模块的执行时间

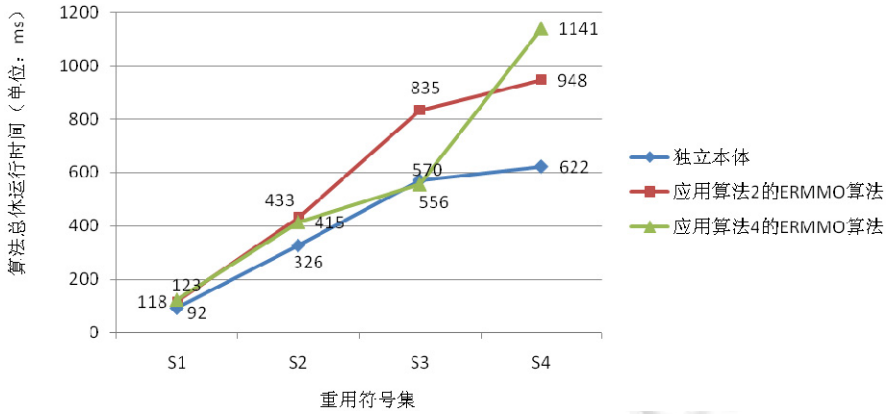

Fig.10 Overall runtime of three algorithms

图 10 3 种算法整体运行时间

这里需要说明的是,由于构造的本体规模较小,所以使用 $EMMO_{IK}$ 和 EMO_{IK} 两种子算法的 ERMMO 算法在总体时间上相差不大.由于现有的本体(如 National Cancer Institute's(NCI) thesaurus[29],Gene Ontology (GO)[30], Systematized Nomenclature of Medicine, Clinical Terms(SNOMED,CT)[31]等)都是针对某个具体领域开发的独立本体,所以虽然它们所包含的知识更加全面,但限于领域的专业性及知识的规模,对这些已有本体实现模块化重组十分困难.除了已构建的如图 4 所示的模块化本体库外,很难找到其他合适的本体库作为实验对象.因此,文中的实验没有体现出定义 9 中阈值θ的作用.但是随着本体工程的发展,对模块化本体的研究也会越来越深入,将来会出现更多规模更大、结构更加科学的模块化本体库.届时,可以有更多的本体库作为实验对象,从而能够对 $EMMO_{IK}$ 和 EMO_{IK} 两种子算法的比较和选择做进一步的研究.

4 后续工作及展望

本体的模块化构建和重用是当前本体工程和人工智能的研究热点,也是未来本体发展的一个趋势.本文正是以模块化本体库为研究对象,根据保守扩充理论,提出了本体模块知识完整性概念,并证明了相关的性质.在此基础上,提出了一种更加通用的 ERMMO 算法,验证了该算法的可行性和正确性,解决了 Grau 等人的重用算法只适用于独立本体的问题.接下来,在进一步完善所构建的旅游领域本体库的同时,将主要针对 ERMMO 算法中子算法 $EMMO_{IK}$ 和 EMO_{IK} 的选择进行研究,希望找到一种更加完善的选择判定方法.同时,进一步研究如何应用其他模块化语言来构建本体库,并以此为重用对象来扩展 ERMMO 算法.

致谢 在此,向对本文提出宝贵意见的评审专家表示衷心的感谢,向对本文的工作给予建议的老师和同学表示感谢.

References:

[1] Nicola AD, Missikoff M, Navigli R. A software engineering approach to ontology building. Information Systems, 2009,34(2): 258−275. [doi: 10.1016/j.is.2008.07.002]

[2] LIN ST. Research on the construction of modular ontology [Ph.D. Thesis]. Beijing: Beijing University of Posts and Telecommunications, 2006 (in Chinese with English abstract).

[3] Kutz O, Lutz C, Wolter F, Zakharyaschev M. $\mathcal{E}$-Connections of abstract description systems. Artificial Intelligence, 2004,156(1): 1−73. [doi: 10.1016/j.artint.2004.02.002]

[4] Kutz O, Lutz C, Wolter F, Zakharyaschev M. $\mathcal{E}$-Connections of description logics. In: Calvanese D, Giacomo GD, Franconi E, eds. Proc. of the 2003 Int'l Workshop on Description Logics (DL 2003). Rome: CEUR Workshop, 2003. 81−90.

[5] Grau BC, Horrocks I, Kazakov Y, Sattler U. Ontology reuse: Better safe than sorry. In: Calvanese D, Franconi E, Haarslev V, Lembo D, Motik B, Turhan AY, Tessaris S, eds. Proc. of the 2007 Int'l Workshop on Description Logics (DL 2007). Brixen-Bressanone: CEUR Workshop, 2007. 250−261.

[6] Ghilardi S, Lutz C, Wolter F. Did I damage my ontology? A case for conservative extensions in description logics. In: Doherty P, Mylopoulos J, Welty C, eds. Proc. of the 10th Int'l Conf. on Principles of Knowledge Representation and Reasoning (KR 2006). Menlo Park: AAAI Press, 2006. 187−197.

[7] Lutz C, Walther D, Wolter F. Conservative extensions in expressive description logics. In: Veloso M, ed. Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2007). Menlo Park: AAAI Press, 2007. 453−459.

[8] Chabin J, Ferrari MH, Musicante MA, Réty P. Conservative type extensions for XML data. In: Hameurlain A, Küng J, Wagnereds R, eds. Trans. on Large-Scale Data-and Knowledge-Centered Systems IX. LNCS 7980, Heidelberg: Springer-Verlag, 2013. 65−94. [doi: 10.1007/978-3-642-40069-8_4]

[9] Vatev S. Conservative extensions of abstract structures. In: Löwe B, Normann D, Soskov I, Soskova A, eds. Proc. of the Models of Computation in Context. LNCS 6735, Heidelberg: Springer-Verlag, 2011. 300−309. [doi: 10.1007/978-3-642-21875-0_32]

[10] Borgida A, Serafini L. Distributed description logics: Assimilating information from peer sources. Journal of Data Semantics, 2003, 2800:153−184. [doi: 10.1007/978-3-540-39733-5_7]

[11] Bouquet P, Giunchiglia F, Harmelen FV, Serafini L, Stuckenschmidt H. C-OWL: Contextualizing ontologies. Web Semantics Science Services & Agents on the World Wide Web, 2004,1(4):325−343. [doi: 10.1016/j.websem.2004.07.001]

[12] Bao J, Caragea D, Honavar VG. Package-Based description logics—Preliminary results. In: Cruz I, Decker S, Allemang D, Preist C, Schwabe D, Mika P, Uschold M, Aroyo L, eds. Proc. of the Semantic Web-ISWC 2006. LNCS 4273, Heidelberg: Springer-Verlag, 2006. 967−969. [doi: 10.1007/11926078_72]

[13] Jiang YC, Shi ZZ, Tang Y, Wang J. A distributed dynamic description logic. Journal of Computer Research and Development, 2006,43(9):1603−1608 (in Chinese with English abstract). [doi: 10.1360/crad20060917]

[14] Kalyanpur A, Parsia B, Sirin E, Grau BC, Hendler JA. Swoop: A Web ontology editing browser. Web Semantics Science Services & Agents on the World Wide Web, 2006,4(2):144−153. [doi: 10.1016/j.websem.2005.10.001]

[15] Li P, Shi YX, Jiang YC. Tourism domain ontology construction based on $\mathcal{E}$-Connections. Computer Engineering, 2010,36(22): 274−276 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-3428.2010.22.098]

[16] Noy NF, Musen MA. The PROMPT suite: Interactive tools for ontology mapping and merging. Int'l Journal of Human-Computer, 2003,59(6):983−1024. [doi: 10.1016/j.ijhcs.2003.08.002]

[17] Seidenberg J, Rector A. Web ontology segmentation: Analysis, classification and use. In: Carr L, Roure DD, Iyengar A, Goble C, Dahlin M, eds. Proc. of the 15th Int'l Conf. on World Wide Web. New York: ACM Press, 2006. 13−22. [doi: 10.1145/1135777.1135785]

[18] Grau BC, Horrocks I, Kazakov Y, Sattler U. A logical framework for modularity of ontologies. In: Veloso M, ed. Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2007). Menlo Park: AAAI Press, 2007. 298−304.

[19] Grau BC, Horrocks I, Kazakov Y, Sattler U. Just the right amount: Extracting modules from ontologies. In: Williamson C, Zurko ME, Patel-Schneider P, Shenoy P, eds. Proc. of the 16th Int'l Conf. on World Wide Web. New York: ACM Press, 2007. 717−726. [doi: 10.1145/1242572.1242669]

[20] Grau BC, Horrocks I, Kazakov Y, Sattler U. Modular reuse of ontologies: Theory and practice. Journal of AI Research, 2008,31(1): 273−318.

[21] Kontchakov R, Pulina L, Sattler U, Schneider T, Selmer P, Wolter F, Zakharyaschev M. Minimal module extraction from DL-lite ontologies using QBF solvers. In: Boutilier C, ed. Proc. of the 21st Int'l Joint Conf. on Artificial Intelligence (IJCAI 2009). Menlo Park: AAAI Press, 2009. 836−841.

[22] Kontchakov R, Wolter F, Zakharyaschev M. Logic-Based ontology comparison and module extraction, with an application to DL-lite. Artificial Intelligence, 2010,174(15):1093−1141. [doi: 10.1016/j.artint.2010.06.003]

[23] Lutz C, Wolter F. Deciding inseparability and conservative extensions in the description logic $\mathcal{E}$L. Journal of Symbolic Computation, 2010,45(2):194−228. [doi: 10.1016/j.jsc.2008.10.007]

[24] Shen YM, Wang J. Complexity of conservative extensions and inseparability in the description logic $\mathcal{EL}$. In: Zhao DY, Du JF, Wang HF, Wang P, Ji DH, Jeff ZP, eds. Proc. of the Semantic Web and Web Science. CCIS 480, Heidelberg: Springer-Verlag, 2014. 78−86. [doi: 10.1007/978-3-662-45495-4_7]

[25] Grau BC, Parsia B, Sirin E. Combining OWL ontologies using $\mathcal{E}$-Connections. Web Semantics Science Services & Agents on the World Wide Web, 2006,4(1):40−59. [doi: 10.1016/j.websem.2005.09.010]

[26] Protégé. http://protege.stanford.edu/

[27] Jiménez-Ruiz E, Grau BC, Sattler U, Schneider T, Berlanga R. Safe and economic re-use of ontologies: A logic-based methodology and tool support. In: Bechhofer S, Hauswirth M, Hoffmann J, Koubarakis M, eds. Proc. of the Semantic Web: Research and Applications. LNCS 5021, Heidelberg: Springer-Verlag, 2008. 185−199. [doi: 10.1007/978-3-540-68234-9_16]

[28] Jiménez-Ruiz E, Grau BC, Sattler U, Schneider T, Berlanga R. ProSE: A Protégé plugin for reusing ontologies, safe and economique. 2008. http://krono.act.uji.es/people/Ernesto/safety-ontology-reuse/proSE-current-version/ProSE-Manual.pdf

[29] Sioutos N, deCoronado S, Haber MW, Hartel FW, Shaiu W, Wright LW. NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. Journal of Biomedical Informatics, 2006,40(1):30−43. [doi: 10.1016/j.jbi.2006.02.013]

[30] The Gene Ontology Consortium. The Gene ontology: Enhancements for 2011. Nucleic Acids Research, 2012,40(D1):559−564. [doi: 10.1093/nar/gkr1028]

[31] Spackman KA, Campbell KE, Côté RA. SNOMED RT: A reference terminology for health care. In: Proc. of the AMIA Annual Fall Symp. 1997. 640−644.

附中文参考文献:

[2] 林松涛.模块化本体建设研究[博士学位论文].北京:北京邮电大学,2006.

[13] 蒋运承,史忠植,汤庸,王驹.一种分布式动态描述逻辑.计算机研究与发展,2006,43(9):1603−1608. [doi: 10.1360/crad20060917]

[15] 李璞,施雅贤,蒋运承.基于$\mathcal{E}$-Connections 的旅游领域本体构建.计算机工程,2010,36(22):274−276. [doi: 10.3969/j.issn.1000-3428.2010.22.098]

李璞(1983－),男,河南开封人,博士生,助教,主要研究领域为 Web 语义分析,本体工程.

王驹(1950－),男,博士,研究员,博士生导师,CCF 专业会员,主要研究领域为数理逻辑,人工智能,计算机理论.

蒋运承(1973－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为大数据语义分析,语义搜索,数据科学.