

基于半模型的 OWL 本体理由探求方法研究

张 瑜^{1,2)} 欧阳丹彤^{1,2)} 崔仙姬³⁾ 叶育鑫^{1,2)}

¹⁾(吉林大学计算机科学与技术学院 长春 130012)

²⁾(符号计算与知识工程教育部重点实验室 长春 130012)

³⁾(大连民族大学信息与通信工程学院 辽宁 大连 116600)

摘 要 OWL 本体理由探求是语义 Web 推理的重要任务之一. 随着语义 Web 数据的急剧增长以及本体规模的不断扩大,目前的本体理由探求策略已难以满足它们对推理性能的要求. 该文以基于黑盒的探求技术为研究对象,黑盒法是基于“扩张”和“收缩”两个阶段实现理由探求任务的,“扩张”阶段的目标是获得蕴涵目标公理的理由的一个超集,“收缩”阶段对得到的理由超集进行删减至极小集合. 然而,这两个阶段的主要时间开销在于频繁地调用推理机进行变化的公理集合与目标公理之间的蕴涵关系的检测,这会严重影响理由探求的效率. 为了解决这一问题,通过观察理由探求过程中公理集合的变化情况,给出增量本体序列定义,并揭示了增量本体序列中的最大增量本体与理由之间的关系. 增量本体序列的生成过程主要涉及两方面因素:(1)后继本体对先驱本体的有效扩充(必须保证是拟序关系);(2)对本体链中的本体是否蕴涵目标公理的推理判定(必须保证当且仅当最大本体蕴涵目标公理). 在增量本体序列生成过程中,利用半模型证明了后继增量本体与目标公理之间的蕴涵关系是半可判定的,进而给出基于半模型增量推理的理由超集探求算法及其正确性证明. 半模型增量推理的增量体现在:保留上一次得到的模型作为下一次判定的初始条件之一,从而避免传统蕴涵判定中,每一次都完全重构模型的冗余计算. 最后,提出了一种与现有的收缩过程相反的理由求解方案——基于扩张的理由求解策略. 通过迭代地添加公理过程,探测该公理集下的所有理由的公共元素. 利用探测到的所有公共元素构造目标公理的理由. 随后利用该文提出的增量推理任务分别给出新的“扩-缩”理由探求方法和“双扩”理由探求方法. 实验结果表明,改进后的“扩-缩”理由探求方法在求解性能上优于原有的“扩-缩”理由探求方法;而新提出的基于增量推理的“双扩”理由探求方法在现有的求解策略中表现最优.

关键词 语义 Web 推理; OWL 本体; 理由; 黑盒技术; 蕴涵关系

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2018.02720

Semi-Models Based Justifications Detection for OWL Ontologies

ZHANG Yu^{1,2)} OUYANG Dan-Tong^{1,2)} CUI Xian-Ji³⁾ YE Yu-Xin^{1,2)}

¹⁾(College of Computer Science and Technology, Jilin University, Changchun 130012)

²⁾(Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Changchun 130012)

³⁾(College of Information and Communication Engineering, Dalian Minzu University, Dalian, Liaoning 116600)

Abstract Detecting justifications for OWL ontologies is an important task in semantic web reasoning. Because of the rapid growth of semantic web data and the continuous increase of scales of ontologies, existing semantic web reasoning strategy has been difficult to adapt to such environment. In this paper, we discussed black-box techniques of detecting justifications. Black box method for detecting justifications is based on the “expansion” stage and “contraction” stage. The “expansion” stage can achieve a superset of justification which entails the target axiom. The “contraction” stage prunes the superset to obtain a minimal set of axioms. However, the main

收稿日期:2017-05-24;在线出版日期:2018-04-18. 本课题得到国家自然科学基金项目(61672261,61502199)资助. 张 瑜,男,1982 年生,博士研究生,主要研究方向为本体推理. E-mail: zhangyujlu@163.com. 欧阳丹彤,女,1968 年生,博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为模型诊断、自动推理. 崔仙姬,女,1986 年生,博士,讲师,主要研究方向为语义 Web 与描述逻辑. 叶育鑫(通信作者),男,1981 年生,博士,副教授,主要研究方向为语义 Web 与本体推理. E-mail: yeyx@jlu.edu.cn.

cost of it is the frequent call to the ontology reasoners to determine the implication relationship between changing axioms and target axioms, thus affecting the efficiency of the detecting. In order to solve this problem, the definition of the incremental ontologies sequence is given and the relationship between the maximum of incremental ontology and the justification is revealed based on the observation of the changes of axiom sets. The generation of incremental ontologies sequence is mainly involves two important factors: (1) the effective expansion from the successor ontology to the precursor ontology (it is necessary to ensure that this expansion must be a quasi-order relation), (2) the reasoning judgment on whether the ontology in the ontology chain entails the target axiom. During the generation of the incremental ontologies sequence, a new reasoning task, incremental reasoning task, is discovered. Furthermore, semi-decision between next incremental ontology and target axiom is proven by semi-model which definition is based on the current incremental ontology and its model. It leads to the algorithm about detecting the super-set of justification, and the correction of this algorithm is proven. Semi-model incremental reasoning embodies in: reserves the model obtained in the last time as one of the initial conditions, so as to avoid the completely reconstruction in the traditional entailment judgments. At last, a new method to detect justifications which is in converse direction of shrinking process is proposed, named expansion based justification detection strategy. This strategy iteratively adds the axioms to detect the common elements of all the justifications for the set of axioms. And then use all the public elements to construct the justification of the target axiom. We exploit incremental reasoning to devise an incremental reasoning based on “expansion-shrink” framework and an incremental reasoning based on “double-expansion” framework separately. The experimental results show that the capability of incremental reasoning based on “expansion-shrink” framework is better than the one without incremental reasoning strategy, and the incremental reasoning based on “double-expansion” framework is best in all of methods mentioned in this paper.

Keywords semantic web reasoning; OWL ontology; justification; black-box techniques; entailments

1 引言

在语义 Web 的体系结构中,本体是其核心组成部分,本体是“概念模型明确的规范说明”,它能够描述应用领域的知识,从而形成对领域知识的共同理解.本体描述语言是开发本体的关键,本体语言不仅要具备对领域知识的描述能力,还要具备一定的推理能力,目前最常用的 OWL(网络本体语言)是基于描述逻辑的,描述逻辑是一阶逻辑的可判定子集,它具有较强的表达能力并提供了可判定的推理服务^[1]. OWL 可以被看作是利用 RDF 语法制定的特殊描述逻辑语言,OWL 推理问题的可判定性和复杂度亦可以按照描述逻辑语言的相应性质进行理解.在这种关系下,描述逻辑中已有的一些解决关键推理问题(尤其是可满足问题)的算法,很自然地应用到 OWL 本体的推理问题上.本体与目标公理的蕴涵关系判定是本体推理的基本任务之一,它可

以和其它本体推理基本任务相互转化.本文所讨论的 OWL 本体理由是本体蕴涵关系的“前提”(目标公理称为与其相对的“结论”).在部分小节我们将蕴涵关系的判定转换为相应的概念可满足问题讨论.探求某一目标公理的理由,实际上就是在判定本体蕴涵目标公理成立的前提下,找到包含最小数目的公理集合.理由(Justification)探求任务的前身是 Schlobach 和 Cornet 针对解释本体术语集的语义冲突调试而提出的一个新的非标准本体推理任务^[2].

目前主流的理由探求方法是基于白盒的探求方法和基于黑盒的探求法^[3-4].白盒法是依赖本体推理机的探测方法.该类算法是建立在关于描述逻辑已有的 Tableau 演算判定过程基础上的.实现白盒算法需要彻底修改推理机内部的推理规则.主要涉及到在推理过程中如何有效跟踪公理.黑盒法是独立于本体推理机的方法.该类算法将推理机当作单独的子程序,并且不必对推理机内部做任何改动.本文主要谈理由探求的黑盒算法.跟白盒算法相比,它

虽然在调用推理次数和执行效率上没有优势,但该类方法不需要针对不同语言表达能力调整程序,也不需要修改内部的推理机制.较白盒法具有更强的普适性和通用性.例如,本体 Economy 的描述逻辑语言表达能力为 $ALCH^D$,使用白盒法需要修改处理这一类构造算子的推理机内部算法,使之能够执行 $ALCH^D$ 的判定任务,如果本体变成 MadCow,由于 MadCow 表达能力为 $ALCHOIN^D$,处理本体 Economy 的算法就无法适用于本体 MadCow,因为它们的构造算子不同.而黑盒法只需调用推理机进行可满足性检测即可,因而处理这两个本体时无需任何修改.

黑盒法探求理由是“扩张”与“收缩”两个阶段实现的,通过“扩张”获得蕴涵目标公理的理由的一个超集,再通过“收缩”对超集进行精简从而获得一个极小的理由集合.然而,它的主要代价在于频繁调用本体推理机判定不断变化的公理子集与目标公理之间的蕴涵关系,从而影响理由探求的效率.针对这一问题,本文定义了增量本体序列和半模型的概念,并给出基于增量推理的“扩-缩”和“双扩”理由探求框架,提升理由探求的求解性能.主要贡献有如下 5 点:

(1) 通过观察基于黑盒的理由探求过程中公理集合的变化情况,给出增量本体序列定义.揭示增量本体序列中的最大增量本体与理由之间的关系,并给出计算增量本体序列及其最大增量本体的方法;

(2) 研究了增量本体序列生成过程中伴随的推理过程,给出一种特殊的推理任务——基于半模型的增量推理任务.在该推理任务中,定义了增量本体的半模型,并证明了利用半模型判定后继增量本体与目标公理之间的蕴涵关系是半可判定的.进而利用该性质,给出基于增量推理的理由超集探求算法,并证明了算法的正确性;

(3) 利用半模型增量推理的理由超集探求方法设计了半模型增量推理的“扩-缩”探求方法,实验证明该方法优于传统的“扩-缩”探求方法;

(4) 通过定义关键公理这一概念,发现了一种与收缩过程相反的理由探求方法——我们称为基于“扩张”的理由探求方法.该方法将某一理由的每一个元素都定义为关键公理,关键公理的定位要依靠一个迭代的扩展过程;

(5) 利用基于半模型增量推理的理由超集探求方法设计了基于半模型增量推理的“双扩”理由探求方法,实验证明该方法是目前黑盒方法中的最

优算法.

这 5 方面工作的内在联系是:贡献(1)在于从“扩张”的角度重新定义了包含理由的一个子本体;贡献(2)提出一个基于半模型的增量推理方法快速定位贡献(1)定义的那个子本体(即最大增量本体);贡献(3)在于利用贡献(2)中的方法替换了传统“扩-缩”方法中的“扩”阶段,给出新的理由探求策略,从而将贡献(1)定义的最大增量本体定位到待求的理由.贡献(3)只改动了传统“扩-缩”框架中的扩展阶段,为了进一步提升求解性能,贡献(4)提出了一种不带收缩过程的理由求解框架,并在贡献(5)中将贡献(2)的方法应用在这一框架下,尽可能地使理由探求效率得到最大提升.

2 相关工作

在本体调试中,Schlobach 和 Cornet 定义了“极小不可满足保持子术语集”MUPS(Minimal Unsatisfiability-Preserving Sub-TBox)这一概念来定位导致本体中概念不可满足的极小公理集合.而后,Kalyanpur 和 Parsia 等人在文献[3]中,将不可满足概念的解释推广到一般公理的解释,正式给出了理由的定义.Kalyanpur^[5]指出,求解不可满足概念的 MUPS 这一问题可以等价转换为探求蕴涵该不可满足概念的理由这一问题.理由探求方法包括白盒法和黑盒法两种,白盒法首先由 Schlobach 等人^[6]提出用于 ALC 的 MUPS 求解,而后由 Nyssen^[7]与 Baader 等人^[8]进一步推广到一般本体的调试.接下来,Baader 等人^[9-10]提出的白盒法理由探求针对的是一类轻量级的 EL 和 EL⁺ 本体,Horridge 等人^[11]使用黑盒法求解不可满足概念的理由,并在文献[12]中从给定的蕴涵抽取解释该蕴涵的理由.Suntisrivaraporn^[13]采用基于模块的黑盒方法探求 EL⁺ 本体中蕴涵公理的所有理由.模块化的方法也同样由 Horridge^[14]应用于 MUPS 的黑盒法求解中.Lam 等人^[15]为了定位更精确的理由,采用细粒度的方法跟踪蕴涵产生的路径从而构造出对应的理由.细粒度的思想也被应用于 Du 等人^[16]的调试与修复工作.Zhou 等人^[17]通过为 EL⁺ 构造解释依赖图的方式探求给定蕴涵的理由.对于轻量级的 DL-Lite 本体,付雪峰等人^[18-19]利用 DL-Lite 的特殊结构特征,构造肯定与否定逻辑闭包形成图结构求解蕴涵不可满足概念的理由.Ji 等人^[20]对理由探求的白盒法与黑盒法进行了实验上的比较.欧阳丹彤等人^[21]则从求解效率这

一角度对基于黑盒法的五种推理机进行性能上的比较. 此外, Jannach 等人^[22]将并行技术用于本体诊断系统.

3 理由与增量本体序列

在本节中, 我们首先介绍了理由的基本概念和目前探求理由的主流方法, 然后从黑盒法的公理扩张阶段观察公理变化情况, 定义了本体链和增量本体序列的概念, 并引入选择函数计算公理、公理集合的相关性.

3.1 理由的探求

理由是本体中的某一公理子集, 该公理集能够为所指定的蕴涵关系做出合理的精确解释. 例如, A 和 B 是本体 O 中涉及的两个概念, 并且在 O 下 A 是 B 的子概念, 即 $O \models A \subseteq B$. 那么关于 O 蕴涵 $A \subseteq B$ 的理由, 就是对此蕴涵关系负责的 O 中的一个极小小公理集合. 如果理由中的任何一个公理被移除, 则对应的蕴涵关系不再成立.

定义 1(理由)^[3]. 存在某一蕴涵关系 $O \models \alpha_{\text{ent}}$, 其中 α_{ent} 是目标公理, O 是一致的本体. 一个本体片段 $O' \subseteq O$ 是关于 α_{ent} 在 O 下的理由, 如果 $O' \models \alpha_{\text{ent}}$, 但是对任意的 $O'' \subset O'$ 都有 $O'' \not\models \alpha_{\text{ent}}$. 记为 $\text{Just}(\alpha_{\text{ent}}, O)$.

求解理由的黑盒法是基于“扩张-收缩”框架的探求方法. 在扩张阶段, 算法首先生成一个初始为空的集合 Σ , 接下来从 O 中随机选择某个公理添加进 Σ 中, 每次添加之后都要调用推理机进行一次 $\Sigma \models \alpha_{\text{ent}}$ 的判断, 直到满足 $\Sigma \models \alpha_{\text{ent}}$ 为止. 在收缩阶段, 则从 S 中任意删除某个公理 α , 每次删除之后都要判断是否继续保持 $\Sigma \models \alpha_{\text{ent}}$. 如果 $\Sigma \models \alpha_{\text{ent}}$ 仍然成立, 则表明 α 属于 $O \models \alpha_{\text{ent}}$ 的理由, 此时需要将 α 放回 Σ 中. 否则, 直接删除即可. 遍历完 Σ 中所有的公理之后得到的新的集合 Σ' 就是的 $O \models \alpha_{\text{ent}}$ 的理由, 它满足 $\Sigma' = \text{Just}(\alpha_{\text{ent}}, O)$.

例 1. 若有本体 $O = \{\alpha_1 : A \subseteq C_1 \cap C_2, \alpha_2 : C_1 \subseteq D_1, \alpha_3 : D_1 \cup C_2 \subseteq B, \alpha_4 : D_1 \subseteq E_1, \alpha_5 : C_2 \subseteq E_2\}$, 目标公理为 $\alpha_{\text{ent}} = A \subseteq B$. 则黑盒法探求蕴涵关系 $O \models \alpha_{\text{ent}}$ 的求解过程如下:

首先生成初始集合 $\Sigma = \emptyset$. 接下来假设随机选择了 α_2 添加进 Σ , 由于 $\Sigma = \{\alpha_2\} \not\models \alpha_{\text{ent}}$, 则需要继续添加. 假设依次添加的公理分别是 $\alpha_1, \alpha_5, \alpha_3$, 此时 $\Sigma = \{\alpha_2, \alpha_1, \alpha_5, \alpha_3\} \models \alpha_{\text{ent}}$, 扩张阶段结束. 然后进入收缩阶段, 假设选择了公理 α_1 进行删除, 此时 $\Sigma = \{\alpha_2, \alpha_5, \alpha_3\} \not\models \alpha_{\text{ent}}$, 这说明 α_1 属于 $O \models \alpha_{\text{ent}}$ 的理由, 不

能删除, 于是将其放回. 遍历完 Σ 中所有的公理后得到 $\Sigma' = \{\alpha_1, \alpha_2, \alpha_3\}$, 它就是 $O \models \alpha_{\text{ent}}$ 的理由. 即 $\text{Just}(\alpha_{\text{ent}}, O) = \{\alpha_1, \alpha_2, \alpha_3\}$.

3.2 本体链与增量本体序列

黑盒法探求理由的主要代价在于频繁调用本体推理机判定不断变化的公理子集与目标公理之间的蕴涵关系. 我们从添加公理的角度, 将判定过程中不断变化的公理子集形式化定义为如下的本体序列.

定义 2(本体链). 集合 S 是本体 O 的幂集 2^O 的某一子集(即 $S \subseteq 2^O$). 如果公理集合间的真包含关系 \subset_{axioms} 是 S 上的拟序关系^①, 并且 S 中的任意两个元素在该关系下是可比的, 则称 S 是关于 \subset_{axioms} 的一条本体链.

如果 S 是一条本体链, 则 S 在拟序关系 \subset_{axioms} 存在最大元. 我们称该最大元为最大本体, 记为 $O_{\text{great}} = \text{great}(S, \subset_{\text{axioms}})$.

对于例 1 中的本体 O , 它的部分子集 $O_1 = \{\alpha_2\}$, $O_2 = \{\alpha_2, \alpha_1\}$, $O_3 = \{\alpha_2, \alpha_1, \alpha_5\}$, $O_4 = \{\alpha_2, \alpha_1, \alpha_5, \alpha_3\}$, $O_5 = \{\alpha_2, \alpha_1, \alpha_5, \alpha_3, \alpha_4\}$ 满足 $O_1 \subset O_2 \subset O_3 \subset O_4 \subset O_5$. 则 $S = \langle O_1, O_2, O_3, O_4, O_5 \rangle$ 就是关于 \subset_{axioms} 的一条本体链. 它的最大元为 O_5 .

定义 3(增量本体序列). 如果在一条本体链 S 中, 只有它的最大元蕴涵公理 α_{ent} , 而其它元素都不蕴涵 α_{ent} , 此时我们称 S 是一个增量本体序列, 记为 S_{inc} . 该序列中的任意本体被称作增量本体.

如上例所得到的 S 不是一个增量本体序列, 因为 O_4 与 O_5 都蕴涵 α_{ent} . 若 $S = \langle O_1, O_2, O_3, O_4 \rangle$, 则只有最大元 O_4 蕴涵 α_{ent} . 此时 $S = \langle O_1, O_2, O_3, O_4 \rangle$ 是一个增量本体序列.

由定义 2、3 可知, 一个增量本体序列 S_{inc} 是一条特殊的本体链 S . 所以任意 S_{inc} 都存在它的最大本体 O_{great} , 称为最大增量本体, 并且它蕴涵公理 α_{ent} .

由定义可知, 增量本体序列的生成过程主要涉及两方面因素: (1) 后继本体对先驱本体的有效扩充(必须保证是拟序关系); (2) 对本体链中的本体是否蕴涵目标公理的推理判定(必须保证当且仅当最大本体蕴涵目标公理). 为保证生成的本体集合是拟序集, 并兼顾蕴涵推理判定的有效性, 我们采用选择函数(Select Function)^[23] 选择待添加的公理集合, 通过对初始本体的公理递归扩充, 生成一系列后继本体. 选择函数是基于公理的语法相关性定义的.

① 设 R 是集合 S 上的一个二元关系, 若 R 满足反自反性与传递性, 则 R 是 S 上的拟序关系.

对于一个给定的公理 α , 使用 $I(\alpha), C(\alpha), R(\alpha)$ 分别表示出现在该公理中的实例名称、概念名称和角色名称的集合。

定义 4(直接符号相关)^[23]. 两个公理 α_1 与 α_2 是直接相关的, 当且仅当存在一个公共名称同时出现在两个公理中. 即 $I(\alpha_1) \cap I(\alpha_2) \neq \emptyset \vee C(\alpha_1) \cap C(\alpha_2) \neq \emptyset \vee R(\alpha_1) \cap R(\alpha_2) \neq \emptyset$.

定义 5(直接相关公理集)^[23]. 如果 α_1 与公理集 Σ 直接相关, 当且仅当存在一个公理 $\alpha \in \Sigma$, 使得 α_1 与 α 是直接相关的。

选择函数的核心思想就是基于公理之间的直接符号相关选择相关公理集. 这种符号的“相关性”能够有效避免在扩充公理时的盲目性, 从而使得公理的选择具有“目标导向”, 以此提高后继本体对先驱本体扩充过程中的精确度。

例 2. 考虑例 1 中的 $O = \langle \alpha_1 : A \subseteq C_1 \cap C_2, \alpha_2 : C_1 \subseteq D_1, \alpha_3 : D_1 \cup C_2 \subseteq B, \alpha_4 : D_1 \subseteq E_1, \alpha_5 : C_2 \subseteq E_2 \rangle$, 目标公理为 $\alpha_{ent} = A \subseteq B$.

首先易得 $I(\alpha_{ent}) = \{A, B\}, I(\alpha_1) = \{A, C_1, C_2\}, I(\alpha_2) = \{C_1, D_1\}, I(\alpha_3) = \{D_1, C_2, B\}, I(\alpha_4) = \{D_1, E_1\}, I(\alpha_5) = \{C_2, E_2\}$.

由目标公理开始, 由于 $I(\alpha_{ent}) \cap I(\alpha_1) = \{A\} \neq \emptyset, I(\alpha_{ent}) \cap I(\alpha_3) = \{B\} \neq \emptyset$. 根据定义 4, α_{ent} 与 α_1 以及 α_{ent} 与 α_3 都是直接符号相关的, 这样就得到一个直接相关公理集 $\Sigma = \{\alpha_1, \alpha_3\}$ (见定义 5), 将 Σ 扩充进 O_1 中得到 $O_1 = \{\alpha_1, \alpha_3\}$. 接下来, 由于 $I(\alpha_1) \cap I(\alpha_2) = \{C_1\} \neq \emptyset, I(\alpha_1) \cap I(\alpha_2) = \{C_2\} \neq \emptyset$, 于是又将 O_1 扩充为 $O_2 = \{\alpha_1, \alpha_3, \alpha_2, \alpha_5\}$. 因为 O_2 蕴涵 α_{ent} , 所以扩充结束. 得到增量本体序列为 $S = \langle O_1, O_2 \rangle = \langle \{\alpha_1, \alpha_3\}, \{\alpha_1, \alpha_3, \alpha_2, \alpha_5\} \rangle$.

类似地, Huang 进一步定义了关于公理 α_1 以及本体 O 相关度为 k 的相关公理和相关公理集^[23]. 在我们给出的定义 2 中, 采用此选择函数首先递归计算初始本体的 k 相关公理集, 依据相关度由直接到间接的顺序形成公理序列 $\langle \Sigma_1, \Sigma_2, \dots, \Sigma_k, \dots \rangle$, 然后根据此公理序列向本体添加公理集 Σ_i , 直到发现第一个蕴涵目标公理 α_{ent} 的本体 O_{great} 为止, 生成相应的增量本体序列 $\langle O_1, O_2, \dots, O_i, \dots, O_{great} \rangle$.

4 增量本体序列中的半模型增量推理

在第 2 节中, 我们讨论了通过选择函数和标准的蕴涵判定, 可以得到一个增量本体序列. 本节我们将讨论如何通过定义一个新的推理任务——“半模型

增量推理”, 来更有效地生成一个增量本体序列。

4.1 理由超集的探求方法

根据定义 3 中增量本体与目标公理之间的蕴涵关系可知, 增量本体序列具有关于目标公理理由的如下性质。

性质 1. 增量本体序列 S_{inc} 中的最大增量本体 O_{great} 是关于目标公理 α_{ent} 和本体 O 的理由的一个超集, 而序列中的其它任何增量本体都不包含 α_{ent} 的一个完整理由。

性质 1 表明存在某个关于 O 和 α_{ent} 的理由 $JUST(O, \alpha_{ent})$, 使得 $JUST(O, \alpha_{ent}) \subseteq O_{great}$; 但对于任意非最大增量本体 O_i 而言 (即 $i \neq |S_{inc}|$), $JUST(O, \alpha_{ent}) \not\subseteq O_i$. 本体 O 必然蕴涵目标公理 α_{ent} , 所以 O 自然是目标公理理由的一个超集. 一般来说一个增量本体序列的最大增量本体 O_{great} 是一个比本体 O 小的理由超集. 也就是说, 如果我们找到了关于 O 和 α_{ent} 的一个增量本体序列 S_{inc} 及其最大增量本体 O_{great} , 那么就找到一个更小范畴意义上的理由的超集. 下面通过递归调用蕴涵关系判定 (即算法 1 中的第 4 步), 给出获取增量本体序列及其最大增量本体的算法。

函数 1. $Gen_Great(O, O', O'', S, \alpha_{ent})$.

输入: 本体 O , 初始本体 O' , 待扩展本体 O'' , 增量本体序列 S , 目标公理 α_{ent}

输出: 最大增量本体 O_{great}

1. IF ($O' \neq \alpha_{ent}$)
2. $S \leftarrow append(O')$;
3. $O'' \leftarrow Fun_select(O', 2^{(O/O')})$;
4. WHILE ($O' \cup O'' \neq \alpha_{ent}$)
5. $S \leftarrow append(O' \cup O'')$;
6. $O' \leftarrow O' \cup O''$;
7. $O'' \leftarrow Fun_select(O', 2^{(O/O')})$;
8. RETURN $O_{great} \leftarrow O' \cup O''$;

在函数 1 中, 首先判定初始本体 $O' \subseteq O$ 是否蕴涵 α_{ent} (即 Step 1). 如果成立则初始本体即为最大增量本体, 返回 $O' \cup O''$ (Step 8). 否则将初始本体 O' 放入序列 S (Step 2), 并利用 3.2 节叙述的选择函数 “ Fun_select ” 计算待扩展本体 O'' (Step 3). 进一步地, 判定扩展后的本体是否蕴涵 α_{ent} (Step 4). 如果成立则返回 $O' \cup O''$ 为最大增量本体 (Step 8). 否则将 $O' \cup O''$ 作为最新的增量本体添加到 S 中 (Step 5), 同时更新 O' 并扩展 O'' (Steps 6, 7). 循环这一判定过程, 直到蕴涵成立为止. 程序结束, 返回 $O' \cup O''$ 为最大增量本体 (Step 8). 最大增量本体 $O' \cup O''$ 同时也是 $JUST(O, \alpha_{ent})$ 的一个超集。

4.2 探求过程中的半模型增量推理

增量本体序列探求过程中的主要开销在于增量本体与目标公理之间的蕴涵关系判定(函数 1 中的 Step4). 并且在这一过程中, 随着本体规模的扩大, 每次蕴涵关系判定的推理代价也逐步增大. 如果能够发现增量本体间在蕴涵关系判定问题上的内在关联, 利用当前增量本体的判定结论优化其后继增量本体的蕴涵判定过程, 有可能会降低推理开销, 从而提升增量本体序列探求的性能和效率.

我们首先通过引理 1, 将本体与目标公理之间的蕴涵判定问题转换成基于本体的不可满足概念问题.

引理 1^[24]. 目标公理 α_{ent} 形为 $A \subseteq B$. 本体 $O \models A \subseteq B$, 当且仅当基于 O 的概念 $A \cap \neg B$ 是不可满足的.

引理 2. 给定当前本体 O 和待扩展本体 O' . 当 $O \not\models A \subseteq B$ 时, 如果存在概念 $A \cap \neg B$ 关于 O 的一个模型 M , 使得 M 在 O' 下是一致的, 则 $O \cup O' \not\models A \subseteq B$.

证明. 根据引理 1, 如果 $O \not\models A \subseteq B$, 则概念 $A \cap \neg B$ 是关于 O 可满足的, 所以一定存在关于 $A \cap \neg B$ 和 O 的某一模型 M . 进一步地, 如果 M 是在 O' 下一致的, 则一定存在关于 M 和 O' 的某一模型 M' , 并且有 $M \subseteq M'$.

因为 $M \subseteq M'$, 并且 M 是 O 的模型, 所以 M' 也是 O 的模型. 又因为 M' 是关于 M 和 O' 的模型, 所以 M' 既是 O 的模型又是 O' 的模型, 即 M' 是 $O \cup O'$ 的模型.

综上, M' 是 $A \cap \neg B$ 关于 $O \cup O'$ 的一个模型, 即 $A \cap \neg B$ 关于 $O \cup O'$ 是可满足的. 故根据引理 1 有, $O \cup O' \not\models A \subseteq B$. 证毕.

引理 3. 引理 2 中, 如果存在关于概念 $A \cap \neg B$ 和 O 的某一模型 M , 在 O' 下是不一致的, 则未必有 $O \cup O' \models A \subseteq B$.

证明. 如果存在关于概念 $A \cap \neg B$ 和 O 的某一模型 M , 则概念 $A \cap \neg B$ 是关于 O 可满足的. 根据引理 1 有 $O \not\models A \subseteq B$. 如果此时该模型 M 是关于 O' 不一致的, 那么就不存在一个解释 I' 使得 M 、 $A \cap \neg B$ 、 O 和 O' 同时可满足 $(O \cup O')_M \models A \subseteq B$.

但是使得概念 $A \cap \neg B$ 是关于 O 可满足的模型并不唯一, 可能存在不同于 M 的其它模型 M' , 在 O' 下是一致的. 那么根据引理 2 有, $(O \cup O')_{M'} \models A \subseteq B$.

综上, 如果存在关于概念 $A \cap \neg B$ 和 O 的某一模

型 M , 在 O' 下是不一致的, 未必有 $O \cup O' \models A \subseteq B$. 证毕.

定义 6(半模型). 在增量本体序列中, 关于当前增量本体 O 和概念 $A \cap \neg B$ 可满足的模型 M , 称为后继增量本体 $O \cup O'$ 的半模型.

之所以称 M 为后继增量本体 $O \cup O'$ 的半模型, 是因为由引理 2、引理 3 可知: 在模型 M 下, 当存在一个解释 I 使得 M 和 O' 同时可满足时, 可以判定 $O \cup O' \not\models A \subseteq B$. 但当不存在一个解释 I 使得 M 和 O' 同时可满足时, 有可能存在关于 O 和 $A \cap \neg B$ 的其它模型 M' , 使得 M' 在 O' 下是可满足的. 此时无法根据 M 关于 O' 的不可满足性, 去判定 M 与 $A \subseteq B$ 之间的蕴涵关系是否成立. 故称模型 M 是后继增量本体 $O \cup O'$ 的半模型.

4.3 基于半模型的增量推理算法

我们在 4.1 节给出了探求增量本体序列中最大增量本体的一般算法——函数 1. 利用 4.2 节给出的增量本体的相关增量推理性质, 可以设计一个更高效的等价判定方法, 代替函数 1 中 Step4 的增量本体 $O' \cup O''$ 和目标公理 $\alpha_{ent}: A \subseteq B$ 间蕴涵关系的判定. 定义判定函数如下.

函数 2. IncrementEntailment(M, O'', α_{ent}).

输入: M 是概念 $A \cap \neg B$ 关于 O' 的一个模型, O'' 是待扩展的本体, 目标公理 $\alpha_{ent}: A \subseteq B$

输出: $IsEntailment$

1. IF (M is consistant with O'')
2. $M \leftarrow getModel(M, O'');$
3. $IsEntailment == False;$
4. ELSE
5. IF($A \cap \neg B$ is satisfiable for $O' \cup O''$)
6. $IsEntailment == False;$
7. $M \leftarrow getModel(O' \cup O'', A \cap \neg B);$
8. ELSE
9. $IsEntailment == True;$
10. RETURN $IsEntailment, M;$

定理 1. 利用判定函数 IncrementEntailment(M, O'', α_{ent}) 替换函数 1 的 Step4 中的后继增量本体与目标公理蕴涵关系的判定, 得到关于函数 1 的更新函数 1'. 函数 1' 的返回值也是增量本体序列的最大增量本体.

证明. 由函数 1 中 Step4 在 Step1 的 if 判定条件可知, 当执行函数 1 中 Step4 时, 一定有 $O' \not\models \alpha_{ent}$. 根据引理 1, 公理 α_{ent} 对应的概念表达式 $A \cap \neg B$ 是关于 O' 可满足的. 所以存在关于 O' 和 $A \cap \neg B$ 的某一模型 M (即函数 2 的输入 M). 根据定义 6 有, M

也称为后继本体 $O' \cup O''$ 的半模型。

如果半模型 M 是关于待扩展本体 O'' 一致的 (即 Step1), 根据引理 2 有 $O' \cup O'' \models A \subseteq B$. 此时 $\text{IncrementEntailment}(O', O'', \alpha_{\text{ent}})$ 返回值为 False (即 Step3). 判定函数返回值与后继本体和目标公理间蕴涵关系一致. 同时, 使用 M 在 O'' 下一致的模型 $\text{getModel}(M, O'')$ 更新模型 M (即 Step2).

如果半模型 M 是关于 O'' 不一致的 (即 Step4). 根据引理 3 可知, 此时无法判定 $O' \cup O''$ 与 $A \subseteq B$ 之间的蕴涵关系是否成立. 所以只好进一步判定 $A \cap \neg B$ 关于 $O' \cup O''$ 的可满足性. 根据引理 1 可知, 当 $A \cap \neg B$ 关于 $O' \cup O''$ 可满足时, $O' \cup O'' \models A \subseteq B$. 此时 $\text{IncrementEntailment}(M, O'', \alpha_{\text{ent}})$ 返回值为 False (Step6), 判定函数返回值与后继本体和目标公理间蕴涵关系一致. 同时更新当前模型为半模型 (Step7). 否则, 当 $A \cap \neg B$ 关于 $O' \cup O''$ 不可满足时 (Step8), $O' \cup O'' \models A \subseteq B$. 此时 $\text{IncrementEntailment}(M, O'', \alpha_{\text{ent}})$ 返回值为 True (Step9), 判定函数返回值亦与后继本体和目标公理间蕴涵关系一致.

综上, 判定函数 $\text{IncrementEntailment}(O', O'', \alpha_{\text{ent}})$ 的返回值与函数 1 中 Step4 中 O' 是否蕴涵 $A \subseteq B$ 的判定一致. 所以, 进一步有函数 1' 和函数 1 等价. 故函数 1' 返回值也是增量本体序列的最大增量本体. 证毕.

4.4 算法复杂度及性能分析

利用函数 2 (即基于半模型的公理增量推理算法) 替换函数 1 (即理由超集探求算法) 中 Step4 中的蕴涵判定, 并没有改变理由超集探求算法本身的复杂度. 对于一个长度为 m 的增量本体序列, 替换后的函数 1' 与函数 1 的复杂度都是 m . 从这一层面的复杂度分析上, 是看不出函数 1' 性能差异的. 函数 1 和函数 1' 的性能差异主要体现在对蕴涵关系判定的调用上, 即直接调用推理机判定蕴涵关系与采用基于半模型的公理增量推理判定蕴涵关系的性能差异.

虽然函数 2 和函数 1 中的蕴涵关系判定在结果是等价的 (如 4.3 节中所述), 但是它们的执行性能是不同的. 在函数 1 中, 蕴涵关系是通过直接调用推理机进行判定的, 调用次数与函数 1 的复杂度相同, 都是 m . 而函数 1' 中的蕴涵关系是通过调用 4.3 节中的函数 2 进行判定的, 它的情况比较复杂.

最好的情况是函数 1' 每次调用函数 2 都只执行到 Step3, 即仅判定当前模型 M 与 O'' 的一致性就退

出判定. 这样的调用, 最好情况是 $m-1$ 次, 最后一次第 m 次是蕴涵公理 α_{ent} 的判定. 此次判定遍历函数 2 的全部代码, 除了进行一致性检测, 还进行了 Step5 中概念 $A \cap \neg B$ 的可满足检测 (直接调用推理机). 可见, 相比函数 1, 除最后一次外, 前 $m-1$ 次均不用进行概念 $A \cap \neg B$ 的可满足检测. 相应的公理增量推理检测效率要远高于概念 $A \cap \neg B$ 的可满足检测.

最坏的情况是函数 1' 每次调用函数 2 都只执行到 Step7, 即不仅判定当前模型 M 与 O'' 的一致性, 还要进行概念 $A \cap \neg B$ 的可满足检测. 与函数 1 相比, 同样是 m 次的调用, 但每次都多了一致性检测推理过程. 所以, 最坏情况下, 函数 1' 的效率不如函数 1. 但是最坏情况这种极端例子在实际本体调试中极为罕见. 一般情况下, 在探求理由超集的过程中, 调用函数 2 的推理效率均高于直接调用推理机判定蕴涵关系的效率. 对性能的具体测试分析, 见第 7 节实验部分.

综上, 从最好情况看, 每一次我们都利用基于半模型的增量推理代替传统的蕴涵判定; 从最坏情况看, 每一次基于半模型的增量推理都失败, 仍需调用传统的蕴涵判定. 半模型增量推理的增量体现在: 保留上一次得到的模型作为下一次判定的初始条件之一, 从而在避免的传统蕴涵判定中, 每一次都完全重构模型的冗余计算.

类似的增量思想, 在该领域已经广泛应用. 例如增量式求解 MUPS 中^[23], Huang 通过关联公理增量扩展公理 (在 3.2 节亦有相关引用), 亦称为 incremental expansion; 再如文献[25]所用的增量推理 incremental reasoning 是使用局部语法模块 (Locality-Based Modules) 优化概念间包含关系判定 (Incremental classification). 与已有工作中针对公理的增量和针对模块化的增量不同, 本文所提出的方法是针对 (半) 模型的增量. 所以我们将其命名为半模型增量推理方法, 以示与同类思想的其它方法进行区别.

5 扩缩框架下的半模型增量推理

本节首先介绍基于黑盒技术的理由探求一般方案——“扩-缩”探求方法 (见 5.1 节). 然后建立基于半模型增量推理的最大增量本体与“扩-缩”探求方法的联系, 讨论基于半模型增量推理的扩张过程,

给出新的“扩-缩”理由探求算法(见 5.2 节).

5.1 “扩-缩”探求方法

黑盒法的基本思想是先将公理蕴涵的理由探求问题转换成等价概念不可满足问题(由引理 1 保证),然后再计算该不可满足概念的理由. 在计算不可满足概念的理由过程中,目前主流的方法是采用“扩-缩”方法^[3-4]. 该方法的基本思想是:给定某一概念 C 是关于本体 O 不可满足的. 从 O 中选择公理添加到一个临时生成的本体 O' 中,直到发现 O' 使得 C 不可满足,停止 O' 的扩张. 然后我们删减 O' 中冗余的公理,直到约简成使得 C 不可满足的一个最小的解释集合,即相应公理蕴涵的理由. 在这两个阶段中: O' 的“扩展”是为了找到理由的一个超集;而 O' 的“收缩”是为了找到最终的理由. 该框架的算法形式化表述由 Kalyanpur 等人在文献[3]中给出,如下算法 1.

算法 1. “扩-缩”探求算法.

输入: 本体 O , 不可满足概念 C

输出: 理由 O'

1. $O' \leftarrow \emptyset$;
2. WHILE (C is satisfiable w. r. t. O')
3. Select a set of axioms $s \subseteq O/O'$;
4. $O' \leftarrow O' \cup s$;
5. FOR EACH axiom $k' \in O'$
6. $O' \leftarrow O' - \{k'\}$;
7. IF (C is satisfiable w. r. t. O')
8. $O' \leftarrow O' \cup \{k'\}$;
9. RETURN O' ;

算法 1 的输入是本体 O 和关于目标公理 α_{ent} 等价的概念 C . 在第一个循环阶段(Steps 2~4),算法首先生成一个初始化为空的 O' , 并且从中选择公理迭代插入,直到 C 是关于 O' 不可满足的. 在第二个循环阶段(Steps 5~8),算法在每次迭代过程中,从 O' 中移除一个公理,并检测 C 是否关于 O' 由不可满足变为可满足. 凡是导致 C 由不可满足变为可满足的公理,再重新插入 O' . 直到 O' 中的所有公理都被测试过,第二循环阶段停止.

5.2 “基于半模型增量推理的扩-缩”探求方法

算法 1 中的扩张阶段(Steps 1~4)的目的是找到关于概念 C 的一个尽可能小的不可满足集合. 其中的 C 是关于理由探求中目标公理 $\alpha_{\text{ent}}: A \subseteq B$ 的一个等价概念,即 $C = A \cap \neg B$. 而这个尽可能小的不可满足集合就是 α_{ent} 的理由的一个超集. 我们在第 3 节中给出了同样能够探求到理由超集的一个一般方法(即函数 1)和基于半模型增量推理的方法(即函

数 $1'$, 是函数 1 的一个优化). 利用基于半模型增量推理的理由超集探求方法(即函数 $1'$),并结合“扩-缩”探求框架下的“收缩”阶段(算法 1 中的 Steps 5~8),可以得到一个“基于半模型增量推理的扩-缩”探求方法,该算法描述如下.

算法 2. “基于半模型增量推理的扩-缩”算法.

输入: 本体 O , 初始本体 O' , 待扩展本体 O'' , 增量本体序列 S , 目标公理 $\alpha_{\text{ent}}: A \subseteq B$

输出: 理由 O'

1. $O' \leftarrow \emptyset$; $O'' \leftarrow \emptyset$; $S \leftarrow \emptyset$;
2. $O' \leftarrow \text{Gen_Great}'(O, O', O'', S, \alpha_{\text{ent}})$;
3. FOR EACH axiom $k' \in O'$
4. $O' \leftarrow O' - \{k'\}$;
5. IF ($A \cap \neg B$ is satisfiable w. r. t. O')
6. $O' \leftarrow O' \cup \{k'\}$;
7. RETURN O' ;

算法 2 的输入是原本体 O , 增量本体序列 S ,

增量本体序列中的初始本体 O' 和待扩展本体 O'' 以及目标公理 $\alpha_{\text{ent}}: A \subseteq B$. 首先对初始本体 O' 、待扩展本体 O'' 和增量本体序列 S 进行初始化(Step 1). 再利用 $\text{Gen_Great}'(O, O', O'', S, \alpha_{\text{ent}})$, 即函数 $1'$ 生成一个增量本体序列, 该函数的返回值是序列的最大增量本体 O_{great} . 它同时也是目标公理 α_{ent} 的一个理由的超集, 故将 O_{great} 赋值给“收缩”阶段的 O' (Step 2). 在此基础上进行 Steps 3~6 的冗余公理迭代删减, 直到生成目标公理 α_{ent} 的一个理由.

现有的“扩-缩”理由探求方法下的优化策略同样适合于“基于半模型的增量推理的扩-缩”理由探求策略. 因为在它们的扩张或收缩阶段都包括对推理机的一系列的可满足性检测的调用. 这些优化策略对现有框架的改进也都集中在如何最小化调用可满足检测的次数. 例如 Schlobach 等人在文献[4]中采用了选择函数, 从语法角度选择与当前公理或当前公理结合相关联的公理(集). 该策略在 2.2 节的本体序列生成过程分析中已采用, 参见定义 4 和 5. 再例如 Kalyanpur 等人在文献[3]采用的滑动窗口策略, 该策略保证扩张或收缩阶段每次选择探求的公理个数以某一概率保持增长, 从而加速扩张和收缩进程. 该策略同样适用于加速函数 $1'$ 中提出的本体序列的生成过程.

相比较基本的“扩-缩”理由探求方法, 本节提出的“基于半模型增量推理的扩-缩”理由探求方法的优势在于: 采用 4.3 节的函数 2(即增量推理函数)生成的函数 $1'$ 去替换扩张过程, 使得算法不仅能在调用次数上得到优化, 还能够在推理消耗上得到优

化. 推理耗时上的性能优势在 4.4 节的“算法复杂度及性能分析”中给出分析.

6 “双扩”框架下的半模型增量推理

本节首先讨论了一种与收缩过程相反的新的理由探求策略——“基于扩张”的理由探测方法(见 6.1 节). 采用“基于扩张”的理由探求过程替代“扩-缩”结构下的收缩阶段(见 6.2 节), 能够更大程度地发挥半模型增量推理的优势. 所以在 6.3 节, 进一步给出了“基于半模型增量推理的双扩”理由探求方法, 并讨论了“双扩”探求方法下的半模型增量推理优化.

6.1 关键公理

与收缩过程相反, 我们提出一个基于扩张的过程探求理由. 我们通过迭代的添加公理过程, 探测该公理集下的所有理由的公共元素. 利用探测到的所有公共元素构造目标公理的理由. 首先给出所有理由的公共元素的定义——关键公理.

定义 7(关键公理). 如果本体 O 中的公理 α 属于基于 O 和目标公理 α_{ent} 的所有理由, 则称 α 是关于 O 和 α_{ent} 的关键公理. 即 $\alpha \in each\ JUST(\alpha_{ent}, O)$.

根据定义 7 中关于关键公理的定义, 进一步给出基于本体 O 和目标公理 α_{ent} 的关键公理集合 O_{criti} , 记为 $O_{criti} = \{\alpha | \alpha \in each\ JUST(\alpha_{ent}, O)\}$.

实际上, 因为关键公理要求属于每个 $JUST(\alpha_{ent}, O)$, 所以一般很难存在关于整个本体的关键公理. 但是我们可以尝试着在 O 的一个足够小的子本体 O' 下探测关键公理. 相对于目标公理 α_{ent} 而言, 公理 α 不是关于 O 关键的, 但却是关于 O' 关键的. 关键公理的探测算法如下函数 3 所示.

函数 3. FindCriticalAxiom(O, O', α_{ent}).

输入: 本体 O , 目标公理 α_{ent} , 不蕴涵 α_{ent} 的子本体 $O' \subseteq O$

输出: 关于 O 和 α_{ent} 的关键公理 α

1. FOR each axiom α from O/O'
2. IF (α_{ent} is not entailed by $O' \cup \{\alpha\}$)
3. $O' \leftarrow O' \cup \{\alpha\}$
4. $O' \leftarrow O' \cup \{\alpha\}$
5. RETURN last appended(α)

假设已知本体 O 的一个子本体 O' 并不蕴涵目标公理 α_{ent} , 我们通过函数 3 迭代添加公理持续更新 O' , 直到使它蕴涵 α_{ent} 为止. 定位 O' 的基本思想是: 首先从 O/O' 中选择任一公理 α (即 Step 1), 检测 $O' \cup \{\alpha\}$ 是否蕴涵 α_{ent} . 如果 $O' \cup \{\alpha\} \models \alpha_{ent}$ 不成立

(即 Step 2), 向 O' 添加当前被选择的公理 α 来更新 O' (即 Step 3). 循环这一过程, 直到 $O' \cup \{\alpha\} \models \alpha_{ent}$ (转至 Step 4). 此时, 最后一个被添加测试的公理 α 就是 O' 的关键公理 (即 Step 5). 根据函数 3 的描述, 我们有如下引理 4 成立.

引理 4. 函数 3 最终向 O' 添加的公理 α 是关于 O' 和 α_{ent} 的关键公理.

证明. 有定义 7 显然知, 引理 4 是正确的. 证毕.

注意, 此处的当前本体 O' 与整个本体 O 不是同一概念.

6.2 基于扩张的探求方法

事实上, 理由是蕴涵目标公理的一个极小的特殊本体. 当我们把理由看作一个本体时, 该本体中的每个公理都是关于本体和目标公理关键的. 故有如下引理 5.

引理 5. $\forall \alpha \in O$ 是关于 O 和目标公理 α_{ent} 关键的, 当且仅当 O 本身是关于 O 和目标公理 α_{ent} 的一个理由, 即 $O = JUST(\alpha_{ent}, O)$.

证明. 首先证明必要性成立. 如果存在某一本体 O 本身就是一个关于自身 O 和目标公理 α_{ent} 的理由, 那么此时关于 O 和 α_{ent} 仅有一个理由 $JUST(\alpha_{ent}, O)$. 根据定义 7 可知, 每一个公理都属于这个唯一的理由 $JUST(\alpha_{ent}, O)$, 所以属于 O 的任一公理都是关于 O 和 α_{ent} 的关键公理. 必要性成立.

相反, 再证引理的充分性. 假设当 $\forall \alpha \in O$ 都是关于 O 和 α_{ent} 的关键公理时, 存在一个子本体 $O' \subseteq O$ 是关于 O 和 α_{ent} 的理由 $JUST(\alpha_{ent}, O)$. 那么根据定义 7 有, 属于 O' 的任一公理也一定属于 O . 所以 O' 和 O 是相等的, 即同一本体. 所以当 $\forall \alpha \in O$ 是关于 O 和目标公理 α_{ent} 关键的时, O 一定就是理由 $JUST(\alpha_{ent}, O)$.

综上, 引理 5 得证.

证毕.

根据引理 5 设计如下递归程序算法 3 探求关于某一本体和目标公理的理由. 该算法通过调用函数 3 反复探求所有关键公理去构造理由. 在每次调用函数 3 之前, O 和 O' 都分别被 O' 和 J 更新 (其中, J 为一个关于 α_{ent} 和 O' 的关键公理集合), 以便保证在更准确和更小的规模内找到关键公理.

算法 3. 基于扩张的理由探求算法.

输入: 本体 O 目标公理 α_{ent}

输出: 理由 J

1. $J \leftarrow \emptyset$
2. $O' \leftarrow \emptyset$
3. WHILE ($|J| < |O|$)

4. FindCriticalAxiom(O, O', α_{ent})
5. $J \leftarrow J \cup \{\alpha\}$
6. $O \leftarrow O'$
7. $O' \leftarrow J$
8. RETURN J

定理 2. 算法 3 返回的 J 是本体 O 蕴涵目标公理 α_{ent} 的理由。

证明. 算法 3 首先在 Step1 和 2 初始化理由 J 和关键公理集合 O' , 使 J 和 O' 为空. 当算法运行到循环体内时, 函数 FindCriticalAxiom(O, O', α_{ent}) 首先被调用, 即 Step4. 根据引理 4 有, 函数返回的公理 α 是关于 O' 的关键公理. 所以它也一定是关于 O' 的某一理由的元素, 故在 Step5 中将它添入 J . 然后在 Step6 和 7 中, 分别用 O' 代替 O , J 代替 O' . 使得下一次循环时, 在一个更小的本体范畴内定位新的理由的元素. 关于 Step3 的循环, 每一轮循环都会探测到关于 J 的一个新元素, 即关键公理. 如果在某一轮循环的开始判定到 J 蕴涵 α_{ent} , 那么在本轮循环结束前 O 将会和 J 相等. 算法在 O 等于 J 时终止. 根据引理 5 有, 此时的 J 就是关于 α_{ent} 和当前更新后的 O 的理由, 同时它也是 α_{ent} 关于和原始 O 的理由. 证毕.

6.3 “基于半模型增量推理的双扩”探求方法

在扩缩框架中的收缩阶段, 算法是通过通过对理由超集中的每个公理进行迭代检测, 在迭代检测过程中删减掉冗余的公理, 保留支持理由的公理. 对于一个包含公理个数为 m 的理由超集而言, 收缩阶段的计算迭代次数上限为 m (参见 5.1 节算法 1 的 Steps5~8). 与收缩阶段相比, 如果对同一个包含公理个数为 m 的理由超集, 采用 6.2 节提出的基于扩张的理由探求方法, 则最坏情况下的计算复杂度为 $m(m+1)/2$.

尽管最坏情况极少出现, 扩展过程不可能每次循环都遍历所有公理后才定位到关键公理. 但是, 从计算复杂度上来看, 基于扩展的探求策略相对传统的收缩策略在求解效率上并没有优势. 但如果我们采用半模型增量推理的思想改进基于扩张的理由探求策略, 则有可能通过降低推理消耗的途径, 使其在求解效率上胜出.

为了实现半模型增量推理对“双扩”框架的应用, 我们对算法 3 以及算法 3 调用的函数 3 进行了修改, 生成新的算法 3'. 我们将算法 3' 设计成算法 3 的一个重载函数. 算法 3 是关于二元参数 O 和 α_{ent} 输入的算法, 而算法 3' 是关于 O, O' 和 α_{ent} 三元参数

输入的算法, 算法 3' 直接以输入 O' 为初始化, 取消了算法 3 中对 O' 进行置空的 Step2. 另外, 在调用函数 3 时, 函数 3 的 Step2 中关于蕴涵关系的判定改为调用函数 2, 即调用半模型增量推理判定. 基于以上修改, 我们给出基于半模型增量推理的双扩理由探求算法——算法 4.

算法 4. 基于半模型增量推理的“双扩”理由探求算法.

输入: 本体 O , 不蕴涵 α_{ent} 的子本体 $O' \subseteq O$, 待扩展增量本 O'' , 增量本体序列 S , 目标公理 α_{ent}

输出: 理由 J

1. $O' \leftarrow \emptyset; O'' \leftarrow \emptyset; S \leftarrow \emptyset;$
2. $O \leftarrow Gen_Great'(O, O', O'', S, \alpha_{ent});$
//调用函数 1', 获得理由超集
3. $O' \leftarrow \emptyset;$
4. Algorithm 3'(O, O', α_{ent});
//调用算法 3', 探求理由

首先初始化子本体 O' (亦是增量本体序列中的初始本体)、待扩展本体 O'' 和增量本体序列 S (即 Step1). 然后利用函数 1', 即 $Gen_Great'(O, O', O'', S, \alpha_{ent})$ 生成关于本体的增量本体序列及其最大增量本体, 并利用函数 1' 的返回值替换原本体 O (即 Step2). 接下来初始化 O' 为空 (即 Step3). 由 Step2 和 3 所得到的 O 和 O' , 作为算法 3' 的输入传入基于扩展的理由探求过程, 最终得到关于原本体 O 和目标公理 α_{ent} 的一个理由. 之所以叫做“双扩”探求算法, 是因为 Step2 是一个探求理由超集的过程, 相当于“扩-缩”探求框架中的“扩展”阶段; 而 Step4 是利用基于扩展的探求方法从当前理由超集中获取最终理由的另一个“扩展”过程.

定理 3. 算法 4 的返回值是关于本体 O 蕴涵目标公理 α_{ent} 的一个理由.

证明. 由定理 1 和定理 2 可知, 定理 3 正确.

证毕.

7 实验与结果

本节首先介绍了实验所用的测试数据 (即 7.1 节), 然后对算法的实现进行了说明 (即 7.2 节). 最后介绍了测试环境, 对“扩-缩”理由探求框架下的探求方法、基于扩展的理由探求方法以及“双扩”理由探求方法进行了实验对比 (即 7.3 节).

7.1 测试数据

为测试理由探求方法的有效性, 我们选择了一些真实世界本体. 这些本体其中一些来自 WebProtege

知识库^[26],一些来自同类工作^[3-4,20]中的测试用例.测试本体及其具体参数细节在表 1 中给出.其中,第 1 列(即 Ontology 列)记录了本体名称,所涉及领域涵盖数据库百科(dbPedia)、地学(SWEET)、医学(Tambis)、大学(University)、经济(Economy)和交通(Transportation)等;第 2 列(即 DL 列)采用何种描述逻辑语言表达能力刻画待测本体;第 3 列(即 A 列)统计了相应本体所含公理的个数;第 4 列(即 C/P/I)分别记录了相应本体所含概念数/属性数/实例数;最后一列给出了待测蕴含关系的数目.

表 1 测试本体描述

	Ontology	DL	A	C/P/I	N
1	MadCow	ALCHOIN ^(D)	109	53/29/13	1
2	dbPedia-2014	ALCHI ^(D)	6772	828/3035/1	2
3	SWEET-JPL	ALCHOF ^(D)	3833	1537/121/50	30
4	Tambis	SHIN ^(D)	795	392/112/0	144
5	University	SHI ^(D)	117	43/44/1555	10
6	Economy	ALCH ^(D)	663	338/65/482	51
7	Transportation	ALCN ^(D)	1282	444/105/183	62

其中,待测蕴含关系测试用例的生成从以下两个方面确定:一方面表 1 中大多数本体如 MadCow, dbPedia-2014 等等,待测蕴含关系取自对应的本体不可满足概念.如前所述,本体调试任务下关于不可满足概念的 MUPS 探求是蕴含判定的特例;另一方面,经常出现在相关工作测试用例中的 SWEET-JPL 本体和 University 本体的当前可下载版本不存在不可满足概念的.所以我们随机选在相应本体中的有关概念构造目标公理,生成蕴含关系测试用例.其中 SWEET-JPL 本体随机生成 30 个测试用例,University 本体随机生成 10 个测试用例.

7.2 算法实现

理由探求过程主要涉及到不同的本体公理集合(即子本体)与目标公理间蕴含关系的判定.在基于黑盒的探求技术中,一般先将蕴含关系判定转换成相应的求解极小不可满足保持子集(MUPS)问题,再进一步转换成相应的概念可满足(Concept SAT)问题,最后转换成本体一致性检测(consistency checking).现有主流的本体推理机,如 Pellet、Fact++ 和 Racer++ 等,都已实现本体一致性检测接口、概念可满足性判定接口、甚至是 MUPS 求解问题接口.因此,公理集合(即子本体)与目标公理之间蕴含关系的判定任务可以通过直接(或者简单转换之后)调用上述提到的推理机的接口完成.

本文中的算法 1(即“扩-缩”框架的理由探求)

和算法 3(即基于扩展的理由探求方法)可以直接调用上述所提到的本体推理机中相应推理功能接口进行算法实现,而算法 2(即基于半模型增量推理的“扩-缩”框架理由探求)和算法 4(“双扩”框架理由探求)涉及到新的推理任务——半模型增量推理.我们将半模型增量推理中涉及到的半模型存储为本体的断言集合,将半模型增量推理任务转换成基于后继增量本体的断言集合的可满足性问题进行判定.上述所有代码均利用 OWL API 接口^[27]进行编码实现.为实现增量推理任务,在接口平台上进行二次设计,开发了半模型类.利用 OWL API 中的 assertion 类,定义半模型类中的半模型.此外,类成员还包括当前本体 ontology 类、增量本体 ontology 类、当前模型 model 类.半模型类的初始化由重载 Reasoner 类中的 Satisfiable 方法而构造.半模型类提供了访问半模型、判定增量本体一致性以及更新增量本体的方法.

7.3 实验与结果

以下所有实验在如下环境下运行:Dell 笔记本,具体配置为 2.3 GHz Inter(R) Core(TM) 2 Duo CPU, 4.0 GB 内存,操作系统为 32 位 Windows 7.

在对比的四个算法中,算法 1 是目前已有的理由探求框架,现有调试工具基本都是遵从于该框架.因此,我们在实验过程中以 pellet 推理机为例进行算法 1 的数据测试,与本文所提其它算法进行比较.首先我们对 7.1 节给出的 7 个待测本体和测试用例进行 5.1 节的“扩张-收缩”探求方法(即算法 1)和 5.2 节的基于半模型增量推理的“扩张-收缩”探求方法(即算法 2)的比较测试实验.表 2 给出了两种方法的实验结果对比.第一列本体号 1~7 分别对应表 1 中的 7 个领域本体.对每一个本体的测试,分别记录了该方法下的扩展阶段耗时、收缩阶段耗时以及总耗时.以第二行本体编号为 1 的 MadCow 本体为例,在扩缩框架下探求理由所记录的表格内容“484/43”表示在扩张阶段耗时 484 ms,在收缩阶段耗时 43 ms;同行粗体标记的内容是总耗时 527 ms.从实验结果来看:在扩张阶段,采用半模型增量推理策略的理由超集探求时间(第 2 列“/”的前项)明显优于未采用半模型增量推理策略的扩张时间(第 3 列“/”的前项).而关于“/”的后项对收缩阶段的时间对比,虽然由于前阶段生成的理由超集不尽相同、计算关于同一目标公理的不同理由等等,导致同一测试用例的收缩阶段耗时不尽相同,但因为两种方法

采取的收缩策略完全相同,所以耗时差别并不大.综合扩展阶段和收缩阶段的耗时,从整体耗时(第3列与第5列)对比来看,采用半模型增量推理策略的“扩-缩”框架要优于普通的“扩-缩”框架.

表 2 “扩-缩”框架 vs. 基于半模型增量推理的“扩-缩”框架

Ontology No.	“扩-缩”探求 (算法 1, 以 pellet 为例)		基于半模型增量推理的 “扩-缩”探求(算法 2)	
1	484/43	527	420/45	465
2	1273/87	1360	994/81	1075
3	18850/1124	19974	13921/1101	15022
4	63616/481	68427	53987/4841	58828
5	4828/609	5437	4102/587	4689
6	46468/512	46980	40917/541	41458
7	49880/2010	51890	43148/1942	45090

在测试完“扩-缩”方案的理由探求方法后,我们又对 6.2 节的基于扩展的理由探求方法(即算法 3)和 6.3 节的基于半模型增量推理的“双扩”理由探求方法(即算法 4)进行了测试对比.实验结果记录在表 3 中,表 3 中的内容项和表 2 的解释一样.从表 3 中算法 3 和算法 4 中扩展阶段耗时(表 3 第 2 列和第 4 列的“/”之前项)比较来看基本没有差别,因为它们都采用了基于半模型增量推理的扩展策略进行理由超集的探求.本阶段耗时也和表 2 中的基于半模型增量推理的“扩-缩”框架下的扩展阶段(表 2 第 4 列的“/”之前项)基本一致,它们在性能上都优于普通的扩展阶段(表 2 第 3 列的“/”之前项).表 3 中的性能差别主要体现在第二个扩展阶段(即表 3 第 2 列和第 4 列的“/”之后项).未加半模型增量推理优化策略的算法 3 在扩展阶段的时间不仅高于算法 4 相同阶段的时间,还高于表 2 中相应的收缩阶段的时间.这是因为基于扩展的探求方法虽然在探求理由的每个元素的无须遍历所有公理,但是对每个元素求解的多轮循环,导致整体耗时高于收缩策略.而通过半模型增量推理策略优化之后,效果要优于收缩阶段的求解性能.综上,基于半模型增量推理的“双扩”方法在求解性能上表现最优.

表 3 基于扩展的探求方法 vs. 基于增量推理的“双扩”探求方法

Ontology No.	基于扩展的探求 (算法 3)		基于半模型增量推理的 “双扩”探求(算法 4)	
1	417/59	476	420/39	459
2	1013/164	1177	989/69	1058
3	13798/2486	16284	13936/851	14787
4	53909/5939	59848	53962/3917	57879
5	4015/765	4480	4109/406	4516
6	40739/762	41501	40902/428	41330
7	44261/2813	47074	43133/1685	44818

8 总 结

本文对 OWL 本体理由探求问题中的推理过程进行了研究.通过定义增量本体序列和半模型等概念,将理由探求的一般推理过程定义为一个半模型增量推理过程.并将这一增量推理用于理由探求过程的优化,提出基于半模型增量推理的“扩-缩”理由探求框架.

为了进一步利用半模型增量推理过程,提出与收缩策略相反的基于扩展的理由探求方法,并将增量推理策略用于“双扩”过程的理由探求任务中,最大化地提升理由探求的求解性能.本文目前仅讨论了关于本体和目标公理的一个理由的求解方法,而结合 Hitting-tree 算法能够有效求解所有理由.我们相信:与“扩-缩”理由探求框架相比,如果采用基于半模型增量推理的理由探求方法与 Hitting-tree 算法有效结合的话,应该在求解性能上存在巨大提升空间.这些问题有待于我们在后续工作中进一步研究.

参 考 文 献

- [1] Baader F. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge, UK: Cambridge University Press, 2003
- [2] Schlobach S. Diagnosing terminologies//Proceedings of the 20th International Conference on Artificial Intelligence (AAAI-05). Pittsburgh, USA, 2005: 670-675
- [3] Kalyanpur A, Parsia B, Horridge M, Sirin E. Finding all justifications of OWL DL entailments//Proceedings of the 6th International Asian Semantic Web Conference. Busan, Korea, 2007: 267-280
- [4] Schlobach S, Huang Z, Cornet R, van Harmelen F. Debugging incoherent terminologies. Journal of Automated Reasoning, 2007, 39(3): 317-349
- [5] Kalyanpur A. Debugging and Repair of OWL Ontologies [Ph. D. dissertation]. University of Maryland at College Park, USA, 2006
- [6] Schlobach S, Cornet R. Non-standard reasoning services for the debugging of description logic terminologies//Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003). Acapulco, Mexico, 2003: 355-360
- [7] Nyssen R P. Axiom-Pinpointing in Description Logics and Beyond [Ph. D. dissertation]. Dresden University of Technology, Dresden, Germany, 2009

- [8] Baader F, Pealozza R. Axiom pinpointing in general tableaux. *Journal of Logic Computation*, 2010, 20(1): 5-34
- [9] Baader F, Pealozza R, Suntisrivaraporn B. Pinpointing in the description logic EL//*Proceedings of the 20th International Workshop on Description Logics*. Brixen/Bressanone, Italy, 2007: 171-178
- [10] Baader F, Suntisrivaraporn B. Debugging SNOMED CT using axiom pinpointing in the description logic EL//*Proceedings of the 3rd International KR-MED*. Phoenix, USA, 2008: 1-7
- [11] Horridge M, Parsia B, Sattler U. Explaining inconsistencies in OWL ontologies//*Proceedings of the 3rd International Conference on Scalable Uncertainty Management*. Washington, USA, 2009: 124-137
- [12] Horridge M, Parsia B, Sattler U. Extracting justifications from BioPortal ontologies//*Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*. Boston, USA, 2012: 287-299
- [13] Suntisrivaraporn B. Module extraction and incremental classification: A pragmatic approach for EL^+ ontologies//*Proceedings of the 5th European Semantic Web Conference*. Tenerife, Canary Islands, Spain, 2008: 230-244
- [14] Horridge M. Justification Based Explanation in Ontologies [Ph. D. dissertation]. University of Manchester, Manchester, UK, 2011
- [15] Lam J S C, Sleeman D, Pan J Z, Vasconcelos W. A fine-grained approach to resolving unsatisfiable ontologies//*Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Hong Kong, China, 2006: 428-434
- [16] Du Jianfeng, Qi Guilin, Fu Xuefeng. A practical fine-grained approach to resolving incoherent OWL 2 DL terminologies//*Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*. Shanghai, China, 2014: 919-928
- [17] Zhou Zhangquan, Qi Guilin, Suntisrivaraporn B. A new method of finding all justifications in OWL 2 EL//*Proceedings of the 12th IEEE/WIC/ACM International Conference on Web Intelligence*. Atlanta, USA, 2013: 213-220
- [18] Fu Xue-Feng, Qi Gui-Lin, Zhang Yong. A graph-based approach for calculating minimal unsatisfiability-preserving subsets of ontology in DL-Lite. *Acta Electronica Sinica*, 2016, 44(9): 2040-2045(in Chinese)
(付雪峰, 漆桂林, 张勇. 一种基于图的 DL-Lite 本体最小不可满足保持子集的计算方法. *电子学报*, 2016, 44(9): 2040-2045)
- [19] Fu Xuefeng, Qi Guilin, Zhang Yong, Zhou Zhangquan. Graph-based approaches to debugging and revision of terminologies in DL-Lite. *Knowledge-Based Systems*, 2016, 100(5): 1-12
- [20] Ji Qiu, Gao Zhiqiang, Huang Zhisheng, Zhu Man. Measuring effectiveness of ontology debugging systems. *Knowledge-Based Systems*, 2014, 71(11): 169-186
- [21] Ouyang Dan-Tong, Zhang Yu, Ye Yu-Xin. Research on evaluating ontology reasoners for calculating Mups. *Chinese Journal of Computers*, 2017, 40(6): 1422-1439(in Chinese)
(欧阳丹彤, 张瑜, 叶育鑫. 本体推理机求解 Mups 的性能评测研究. *计算机学报*, 2017, 40(6): 1422-1439)
- [22] Jannach D, Schmitz T, Shchekotykhin K M. Parallel model-based diagnosis on multi-core computers. *Journal of Artificial Intelligence Research*, 2016, 55: 835-887
- [23] Huang Zhisheng, van Harmelen F, ten Teije A. Reasoning with inconsistent ontologies: Framework, prototype, and experiment//Davies J, Studer R, Warren P eds. *Semantic Web Technologies: Trends and Research in Ontology-based Systems*. Amsterdam, Netherlands: Vrije Universiteit Amsterdam, 2006: 71-93
- [24] Horrocks I, Patel-Schneider P. Reducing OWL entailment to description logic satisfiability//*Proceedings of the 2nd International Conference on the Semantic Web*. Sanibel Island, USA, 2003: 17-29
- [25] Grau B C, Halaschek-Wiener C, Kazakov Y. History matters: Incremental ontology reasoning using modules//*Proceedings of the 6th International Semantic Web Conference*. Busan, Korea, 2007: 183-196
- [26] Musen M A. The Protégé project: A look back and a look forward. *AI Matters*, 2015, 1(4): 4-12
- [27] Horridge M, Bechhofer S. The OWL API: A java API for OWL ontologies. *Semantic Web Journal*, 2011, 2(1): 11-21



ZHANG Yu, born in 1982, Ph. D. candidate. His main research interest is ontology reasoning.

OUYANG Dan-Tong, born in 1968, Ph. D., professor. Her current research interest includes model-based diagnosis and automated reasoning.

CUI Xian-Ji, born in 1986, Ph. D., lecturer. Her research interests include semantic web and description logics.

YE Yu-Xin, born in 1981, Ph. D., associate professor. His research interests include semantic web and ontology reasoning.

Background

This study belongs to the problem of detecting justifications for ontologies in the field of knowledge reasoning. Because of the rapid growth of semantic web data and the continuous increase of scales of ontologies, existing semantic web reasoning strategy has been difficult to adapt to such environment.

Black box method for detecting justifications is based on the “expansion” stage and “contraction” stage. The “expansion” stage can achieve a superset of justification which entails the target axiom. The “contraction” stage prunes the superset to obtain a minimal set of axioms. However, the main cost of it is the frequent call to the ontology reasoners to determine the implication relationship between changing axioms and target axioms, thus affecting the efficiency of the detecting. In order to solve this problem, a new method to detect justifications which is in converse direction of shrinking process is proposed, named expansion based justification detection strategy. We exploit incremental reasoning to devise an incremental reasoning based on expansion-shrink framework and an incremental reasoning based on double-expansion framework

separately. Specifically, this paper defined concepts of the incremental sequence of ontology and the semi-model, and then promoted reason to solve the performance of detecting justification. based on the incremental reasoning “expansion-contraction” and “double expansion” influence framework,

Justification and their variants have also been known by other names, particular, MUPS (Minimal Unsatisfiability Preserving Sub-TBoxes) and MinAs (Minimal Axiom set) of an unsatisfiable concept. A MUPS or a MinAs is a minimal set of axioms that can explain an unsatisfiable concept. Thus, the problem of finding the MUPS for an unsatisfiable concept and that of finding all the justifications for an unsatisfiable entailment can be reduced to each other.

This work was supported by the National Natural Science Foundation of China (Nos. 61672261 and 61502199). The focus is on investigating the effective technique for knowledge representation and inference, which can help the knowledge experts to get further insights into the features of knowledge representation and inference.