

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

付雪峰¹, 漆桂林², 张 勇²

(1. 南昌工程学院信息工程学院, 江西南昌 330099; 2. 东南大学计算机科学与工程学院, 江苏南京 210096)

摘 要: 演变中的本体常出现不一致性问题, 这将导致标准推理失效. 针对不一致性问题, 最小不可满足保持子集能够提供本体中概念不可满足的解释. 计算最小不可满足保持子集是本体工程中的一项重要非标准推理任务, 但多数计算方法须借助外部的推理机, 导致计算的效率不高. 为了减少对推理机的依赖, 本文提出了一种基于图的最小不可满足保持子集的计算方法. 新的方法面向 DL-Lite 描述逻辑家族, 将 DL-Lite 本体转换成图, 将本体中的最小不可满足保持子集转换成图上的最小不可满足保持路径对. 对比实验表明, 基于图的方法提高了计算的效率和稳定性.

关键词: 本体; 描述逻辑; 不一致处理; 最小不可满足保持子集

中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2016)09-2040-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2016.09.002

A Graph-Based Approach for Calculating Minimal Unsatisfiability-Preserving Subsets of Ontology in DL-Lite

FU Xue-feng¹, QI Gui-lin², ZHANG Yong²

(1. School of Information Engineering, Nanchang Institute of Technology, Nanchang, Jiangxi 330099, China;

2. School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu 210096, China)

Abstract: Inconsistency often occurs during ontology evolution, and leads to the invalidity of standard reasoning. Minimal unsatisfiability-preserving sub-TBox (MUPS) can provide an explanation of the unsatisfiability of a concept in an ontology. Finding all MUPS is an important nonstandard reasoning task in ontology engineering. Most of the approaches for calculating MUPS are built on external description logic reasoners. However, a reasoner-based method can hardly achieve positive efficiency and stability. In this paper, we propose a reasoner-independent approach to calculating MUPS using graph representation. We first transform DL ontologies to graphs, and find MUPS by computing the minimal unsatisfiability-preserving path-pair (MUPP) based on the transformed graphs. We implement and evaluate our approach. The experimental results demonstrate that our approach performs well in efficiency and stability.

Key words: ontology; description logic; inconsistency handling; minimal unsatisfiability-preserving sub-TBox

1 引言

在实际应用中, 基于描述逻辑的本体常处于扩展、重用、融合等演变过程中, 这可能导致本体中出现不一致^[1], 也因此使得本体推理的失败^[2]. 本体调试是一种常用的处理不一致问题的方法, 它旨在寻求本体中导致冲突的最小不一致子集^[3]. 对于导致不一致的原因, Schlobach 等认为是由于本体中存在不可满足概念或角

色, 他们提出了最小不可满足保持子集 (Minimal Unsatisfiability-Preserving Subset, MUPS) 的概念来解释概念或角色不可满足的原因^[4]. 对于 MUPS 的计算, Kalyanpur 等提出基于碰集树的计算方法^[5], 但碰集树算法复杂度太高. 为了提高计算效率, Suntasirivarnorn 等通过模块抽取的方法来优化 MUPS 的计算^[6], 但该方法抽取的模块过大导致计算效率提升不明显且需要反复调用外部推理机. 为减少对推理机的依赖, 本文寻求借助于图

收稿日期: 2015-06-24; 修回日期: 2015-12-03; 责任编辑: 覃怀银

基金项目: 国家“八六三”高技术研究发展计划基金项目 (No. 2015AA015406); 国家自然科学基金 (No. 61272378); 江西省教育厅青年科学基金项目 (No. GJJ12643)

的方法来完成 MUPS 的计算. 基于图的方法常用于查询应答, 如 Qin 等提出了一种基于图的查询重写方法^[7]. 在分类任务方面, Lembo 等提出了基于图的本体分类的方法^[8], 但该方法仅涉及肯定包含公理. 在不一致处理方面, Qi 等提出了一种基于图的 ABox 修正方法^[9], 但它没有考虑 TBox 中的不一致性问题. 为此, 本文提出了一种基于图 MUPS 计算方法. 在方法中 DL-Lite 本体被转换成有向图, 在图上定义了与 MUPS 相对应的概念 MUPP(最小不可满足保持路径), 并证明了两者在表达能力上是等价的. 最后, 基于图数据库实现了本文的计算方法, 对比实验表明基于图的计算方法在运行效率和稳定性上要明显优于其他基于推理机的计算方法.

2 理论基础

本文的研究对象是描述逻辑中的 DL-Lite 家族, 它们能保证推理在多项式时间内完成, 同时又能够捕获实体关系模型以及统一建模语言中类的大部分知识^[10]. DL-Lite 的概念和角色的形式化定义如下^[10]:

$$B::=A|\exists R \quad R::=P|P^- \quad C::=B|\neg B \quad E::=R|\neg R$$

其中, A 表示原子概念, P 表示原子角色, P^- 是原子角色的逆, B 表示基本概念, 其语法形式为原子概念或者为 $\exists R$; R 表示基本角色, 其语法形式为原子角色或者原子角色的逆; C 表示一般概念, 其语法形式为基本概念或者基本概念的否定; E 表示一般角色, 其语法形式为基本角色或者基本角色的否定.

一个 DL-Lite 本体由术语断言集合与实例断言集合组成, 记为 $O = \langle T, A \rangle$. T 是术语断言集合, 其语法形式为 $B \sqsubseteq C$. A 是实例断言集合, 包括概念实例断言和角色实例断言. 在 DL-Lite 的子语言中, DL-Lite_R 扩充了角色包含公理, 其语法形式为 $R \sqsubseteq E$; DL-Lite_F 扩充了函数断言, 其语法形式为 (funct R). 在本体中, 形如 $B_1 \sqsubseteq B_2$ 或 $R_1 \sqsubseteq R_2$ 的公理我们称为肯定包含公理, 形如 $B_1 \sqsubseteq \neg B_2$ 或 $R_1 \sqsubseteq \neg R_2$ 的公理我们称为否定包含公理. Calvanese 等在文献^[10]中定义了否定包含公理的演绎闭包 $\text{cln}(T)$ 和肯定包含公理的演绎闭包 $\text{clp}(T)$, 记 $\text{cl}(T) = \text{clp}(T) \cup \text{cln}(T)$. 描述逻辑通过解释来描述语义, 一个解释 $I = \langle \Delta^I, \cdot^I \rangle$, 其中 Δ^I 为非空的解释域, \cdot^I 为解释域上的解释函数. 解释函数将概念 A 解释为集合 $A^I (A^I \sqsubseteq \Delta^I)$, 将原子角色 P 解释为关系 $P^I (P^I \sqsubseteq \Delta^I \times \Delta^I)$. 对于一个 DL-Lite 公理 ϕ 和一个解释 I , 如果满足 $I \models \phi$, 则 I 为公理 ϕ 的一个模型; 如果解释 I 满足本体 O 中任意的公理, 则 I 为本体 O 的一个模型, 如果本体 O 的所有模型也是公理 ϕ 的模型, 则称本体 O 蕴含 ϕ , 记为 $O \models \phi$. 接下来将引入一些概念来描述本体中的不一致问题.

定义 1 不可满足概念或者角色^[4]. 如果一个概念 C (或角色 R), 对于 TBox T 的任意一个模型 I , 都有 C^I

$= \emptyset$ (或者 $R^I = \emptyset$), 那么概念 C (或角色 R) 就被称为 T 的一个不可满足概念 (或角色).

对一个不可满足概念或角色而言, 其 MUPS 并不是唯一的, 我们将 TBox T 中关于 C 的所有 MUPS 记为 $\text{mups}(C, T)$. 如果本体中存在不可满足概念或角色, 则该本体是不协调的, 在本体不协调的问题中, 最小不可满足保持子集与不可满足概念密切相关.

定义 2 最小不可满足保持子集^[4]. 设 $C(R)$ 是 TBox T 中的任意一个不可满足概念 (或者角色), 一个 TBox $T' \sqsubseteq T$ 是 T 中关于概念 C (或者角色 R) 的最小不可满足保持子集 (MUPS) 当且仅当 C (或者角色 R) 在 T' 中是不可满足的, 且对于 T' 的任何一个真子集 $T'' \subset T'$, 概念 C (或者角色 R) 在 T'' 上是可满足的.

3 本体到图的转换

为了实现基于图 MUPS 计算, 需要将 TBox 转换成有向图. 给定一个 DL-Lite 的 TBox T 以及 T 上的符号集合 Σ_T ; 一个与 TBox 对应的有向图, 记为 $G_T = \langle V, E \rangle$ (V 是图上顶点的集合, E 是图上边的集合), 可以通过如下的规则构造^[11]:

- (1) 对于 Σ_T 中的任意的概念 A , V 中包含节点 A .
- (2) 对于 Σ_T 中的任意的角色 P , V 中包含节点 $P, P^-, \exists P, \exists P^-$.
- (3) 如果在 T 中存在公理 $B \sqsubseteq C$, 则在有向图 G_T 中有边 $\langle B \rightarrow C \rangle \in E$ 并且节点 $B, C \in V$.
- (4) 如果在 T 中存在公理 $B \sqsubseteq \neg C$, 则在有向图 G_T 中有边 $\langle B \rightarrow \neg C \rangle \in E$ 并且节点 $B, \neg C \in V$.
- (5) 如果在 T 中存在公理 $R_1 \sqsubseteq R_2$, 则在有向图 G_T 中有边 $\langle R_1 \rightarrow R_2 \rangle \in E, \langle \exists R_1 \rightarrow \exists R_2 \rangle \in E, \langle \exists R_1^- \rightarrow \exists R_2^- \rangle \in E$, 并且节点 $R_1, R_2, \exists R_1, \exists R_2, \exists R_1^-, \exists R_2^- \in V$.

在规则中, 符号 $\langle A \rightarrow B \rangle$ 表示从节点 A 到节点 B 的有向边, 图中的结点对应于 TBox 中的概念或者角色, 图中的有向边对应于 TBox 中的包含公理, 图 G_T 的传递闭包记为 $G_T^* = \langle V, E^* \rangle$, 其中 $E^* = \{ \langle A \rightarrow B \rangle : \text{图 } G_T \text{ 中存在一条从 } A \text{ 到 } B \text{ 的通路} \}$.

例 1 给定一个 DL-Lite 的 TBox $T = \{ A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D, B \sqsubseteq \neg C, R_1 \sqsubseteq R_2, B \sqsubseteq \exists R_1, \exists R_2 \sqsubseteq \neg D \}$, 其中 A, B, C, D 为概念, R_1 和 R_2 为角色. 根据图的构造规则, 我们得到 T 对应的有向图, 如图 1 所示.

在本体到图的转换中, 由于 TBox 中可能存在等价关系或相互包含的概念或角色, 这样在转换后的有向图中将存在环路, 即在有向图中存在强连通分支. 而强连通分支的存在将增加路径计算的复杂性, 因而有必要对有向图中的强连通分支做进一步的处理, 图 2 所示为处理强连通图分支的一个示例.

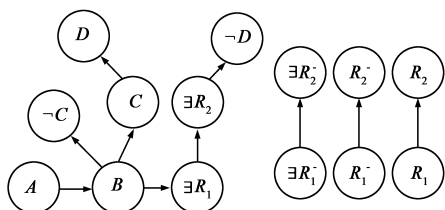


图1 例1中的TBox对应的有向图

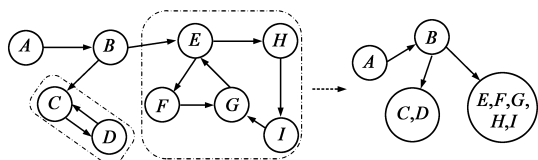


图2 有向图上环的处理

图2的左侧部分为原始的有向图,虚线框中为强连通分支,强连通分支的获取可以应用 Kosaraju-Sharir 算法^[12]来完成.在获取了强连通分支后,使用一个节点来代替该强连通分支就能实现对有向图中环的消除,如图2右侧所示.

接下来我们将讨论本体中的包含公理与图上节点的连通性之间的关系.对于 TBox $T = \{A_1 \sqsubseteq A_2, A_2 \sqsubseteq A_3\}$,根据推理规则, $T \models A_1 \sqsubseteq A_3$.应用图的构造规则, T 转换后的图 G_T 中存在节点 A_1, A_2, A_3 和有向边 $\langle A_1 \rightarrow A_2 \rangle, \langle A_2 \rightarrow A_3 \rangle$,根据图的传递闭包,节点 A_1 可达节点 A_3 .从中可看出本体中包含公理的蕴含与有向图上节点间的可达性存在对应关系.为了描述这种关系,我们将文献[8]中关于本体蕴含的定理以及包含公理与图上可达性的对应关系的定理扩展到否定包含公理^[11].

定理1 给定一个 DL-Lite TBox T ,若 S_1, S_2 为原子概念或者原子角色,则 $T \models S_1 \sqsubseteq \neg S_2$ 当且仅当下面至少一个条件成立:

- (1) 存在肯定包含公理集合 $P \sqsubseteq T$ (P 可以为空) 和一个否定包含公理 $\phi \in T$,使得 $P \cup \{\phi\} \models S_1 \sqsubseteq \neg S_2$;
- (2) $T \models S_1 \sqsubseteq \neg S_1$.

定理2 给定一个 DL-Lite TBox $T, G_T = \langle V, E \rangle$ 是 T 通过构建规则得到的有向图,图中的节点对应 T 中的一般概念或者角色,图中的有向边对应 T 中的包含公理. $\text{cl}(T)$ 是 T 的闭包, $G_T^* = \langle V, E^* \rangle$ 是有向图 G_T 的传递闭包,则 $m \sqsubseteq n \in \text{cl}(T)$ 成立当且仅当边 $\langle m \rightarrow n \rangle \in E^*$.

4 基于图的 MUPS 计算方法

为了实现图上完成 MUPS 的计算,我们在图上对 MUPS 做相应的描述.

定义3 最小的不可满足保持路径对.假设 T 是一个 DL-Lite TBox, T 对应的有向图记为 $G_T = \langle V, E \rangle$.对于 V 中的任意结点 C ,如果在 G_T 中存在一条从结点 C 到结点 D 的路径(无环)和一条从结点 C 到结点 $\neg D$ 的路径(无环),则称这两条路径为关于结点 C 的最小的不可满足保持路径对 (Minimal Unsatisfiability-Preserving Path-Pair, MUPP),图 G_T 中关于结点 C 的 MUPP 的集合记为 $\text{mupp}(C, G_T)$.

图上的 MUPP 与本体中的不可满足概念之间存在对应的关系.

定理3 假设 T 是一个 DL-Lite TBox, $G_T = \langle V, E \rangle$ 是 T 对应的有向图.若 C (或 R) 为 T 中的概念(或角色),则 C (或 R) 是不可满足的当且仅当 G_T 中至少存在一个关于结点 C (或 R) 的 MUPP.

证明 (\Leftarrow) 若 G_T 中存在一个关于结点 C 的 MUPP,不妨假定该 MUPP 中的路径对为 $\{\text{path}\{C, D\}, \text{path}\{C, \neg D\}\}$,这样在 G_T^* 中必定存在 $\langle C \rightarrow D \rangle \in E^*$ 和 $\langle C \rightarrow \neg D \rangle \in E^*$.根据定理2,若 G_T^* 中包含 $\langle C \rightarrow D \rangle$ 和 $\langle C \rightarrow \neg D \rangle$,则 $C \sqsubseteq D \in \text{cl}(T)$ 且 $C \sqsubseteq \neg D \in \text{cl}(T)$,因而 $T \models C \sqsubseteq \neg C$,即 C 是一个不可满足概念.

(\Rightarrow) 如果 C 是一个不可满足的概念,则存在一对不相交概念,(不失一般性,用 D 和 $\neg D$ 表示这两个概念), T 中存在两个公理集合满足 $T \models C \sqsubseteq D$ 且 $T \models C \sqsubseteq \neg D$ ^[13],即公理 $C \sqsubseteq D$ 和 $C \sqsubseteq \neg D$ 都属于 $\text{cl}(T)$.根据定理2可得, G_T^* 中必定包含有向边 $\langle C \rightarrow D \rangle$ 和 $\langle C \rightarrow \neg D \rangle$,因此在有向图 G_T 中存在两条路径 $\{\text{path}\{C, D\}, \text{path}\{C, \neg D\}\}$,即存在关于 C 的一个 MUPP.对于角色 R 可类似证明.

根据图的构建规则,图 G_T 中的有向边都对应到包含公理,因而 G_T 中的路径 P 对应到 T 中的公理的集合,不妨将此公理集合记为 T_p ,显然 $T_p \sqsubseteq \text{cl}(T)$.如果 T 存在一个子集 S 满足 $S \not\sqsubseteq T_p$,并且对于任一 S 的真子集 $S' \subset S, S' \not\sqsubseteq T_p$.我们称 S 为 T 中对应于有向图中的路径 P 的最小公理集合.

定理4 假设 T 是一个 DL-Lite TBox, $G_T = \langle V, E \rangle$ 为 T 对应的有向图.若 G_T 中存在关于结点 C 的 MUPP,我们 TBox 中与该 MUPP 相对应的最小公理集合记为 $S_{T \rightarrow C, \text{MUPP}}$,则 $S_{T \rightarrow C, \text{MUPP}}$ 是 T 中关于概念 C 的一个 MUPS,且 $\text{mupp}(C, G_T)$ 中所有关于结点 C 的 MUPP 所对应的最小公理集合 $S_{T \rightarrow C, \text{MUPP}}$ 为 $\text{mups}(C, T)$.

证明 G_T 中存在结点 C 的 MUPP,根据定理3 结点 C 对应的概念是不可满足的.接下来证明对于 $S_{T \rightarrow C, \text{MUPP}}$ 中任意的一个真子集 $S' \subset S_{T \rightarrow C, \text{MUPP}}, S' \not\sqsubseteq C \sqsubseteq \neg C$. 结点 C 的 MUPP 中的每边都对应到包含公理,这里将 MUPP 对应的公理集合记为 \mathcal{M} .因为 $S_{T \rightarrow C, \text{MUPP}}$ 是该 MUPP 所对应的 T 中最小公理集合,因此 $S_{T \rightarrow C, \text{MUPP}} \not\sqsubseteq \mathcal{M}$,并且 $S' \not\sqsubseteq \mathcal{M}$,这样由 S' 构建的图 $G_{S'}$ 中不存在关于 C 的 MUPP.根据定理3,概念 C 在 S' 是可满足的,即 $S' \not\sqsubseteq C \sqsubseteq \neg C$,根据定义1,可得 $S_{T \rightarrow C, \text{MUPP}}$ 是 T 中关于概念 C 的一个 MUPS.

对于 $\text{mupp}(C, G_T)$ 中任意一个节点 C 的 MUPP, $S_{T \rightarrow C, \text{MUPP}}$ 是 T 中对应到 MUPP 的最小公理集合. 由于 $S_{T \rightarrow C, \text{MUPP}}$ 是 T 中关于概念 C 的一个 MUPS, 因此 $S_{T \rightarrow C, \text{MUPP}} \in \text{mups}(C, T)$. 对于 $\text{mups}(C, T)$ 中任意一个关于概念 C 的 MUPS, 假定该 MUPS 对应的公理集合为 \mathcal{M} . 根据定义 1, $\mathcal{M} \models C \sqsubseteq \neg C$, 且 \mathcal{M} 的任意一个真子集 $\mathcal{M}' \subset \mathcal{M}$, $\mathcal{M}' \not\models C \sqsubseteq \neg C$. 因此, 根据图的构建规则 \mathcal{M} 对应的图 G_M 中存在关于结点 C 的 MUPP, 而 G_M 中不存在关于结点 C 的 MUPP, 即 \mathcal{M} 为 T 中对应到图上节点 C 的 MUPP 的最小集合, \mathcal{M} 包含于集合 $S_{T \rightarrow C, \text{MUPP}}$.

定理 4 建立了图上的 MUPP 与本体中的 MUPS 之间的联系, 在此基础上, 我们给出基于图的 MUPS 计算方法 CalMUPSonGraph, 如算法 1 所示.

算法 1 CalMUPSonGraph

```

输入: 不协调的 TBox  $T$ 
输出: 所有的  $\text{mups}(C, T)$ 
1.  construct  $G_T = \langle V, E \rangle$ ;
2.   $\text{mupp} := \emptyset; \text{mups} := \emptyset$ ;
3.  for each  $(\neg S, S) \in V$  do
4.       $\text{unsatSet} := \text{Descendants}(\neg S, G_T) \cap \text{Descendants}(S, G_T)$ 
5.      for each  $n \in \text{unsatSet}$  do
6.           $\text{mupp} := \text{mupp} \cup \{ \text{path}(n, \neg S), \text{path}(n, S) \}$ ;
7.  for each  $\text{mu}$  in  $\text{mupp}$  do
8.       $\text{axiomset} := \emptyset$ ;
9.      for each path  $p_i \in \text{mu}$  do
10.         for each edge  $\langle n_1 \rightarrow n_2 \rangle \in p_i$  do
11.             if  $n_1 = \exists R_1$  and  $n_2 = \exists R_2$  and  $\langle R_1 \rightarrow R_2 \rangle \in G_T$  then
12.                  $\text{axiomset} := \text{axiomset} \cup \{ R_1 \sqsubseteq R_2 \}$ ;
13.             else
14.                  $\text{axiomset} := \text{axiomset} \cup \{ n_1 \sqsubseteq n_2 \}$ ;
15.   $\text{mups} := \text{mups} \cup \text{axiomset}$ ;
16.  return  $\text{mups}$ ;

```

例 2 考虑例 1 中的 TBox T , 应用算法 CalMUPSonGraph 计算 T 中所有的 MUPS.

CalMUPSonGraph 首先将 T 构造成为如图 1 所示有向图; 在图中有两对不相交的节点 $\{C, \neg C\}$ 和 $\{D, \neg D\}$; 接下来计算不相交节点对的公共子类, 可得节点对 $\{C, \neg C\}$ 的公共子节点是 $\{A, B\}$ 、节点对 $\{D, \neg D\}$ 的公共子节点是 $\{A, B\}$, 节点 A, B 即对应不可满足概念(角

色); 之后, 计算节点 A, B 到不相交节点 $\{C, \neg C\}$ 的路径对以及节点 A, B 到不相交节点 $\{D, \neg D\}$ 的路径对, 根据定义 3, 这些路径对就是 MUPP. 例 2 中的 MUPP 结果如图 3 所示, 图的左边为 $\text{mupp}(B, G_T)$, 右边为 $\text{mupp}(A, G_T)$. 将 MUPP 转换成 MUPS, 将得到: $\text{mups}(B, T) = \{ \{ B \sqsubseteq C, B \sqsubseteq \neg C \} \{ B \sqsubseteq C, C \sqsubseteq D, B \sqsubseteq \exists R_1, R_1 \sqsubseteq R_2, \exists R_2 \sqsubseteq \neg D \} \}$, $\text{mups}(A, T) = \{ \{ A \sqsubseteq B, B \sqsubseteq C, B \sqsubseteq \neg C \}, \{ A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq D, B \sqsubseteq \exists R_1, R_1 \sqsubseteq R_2, \exists R_2 \sqsubseteq \neg D \} \}$.

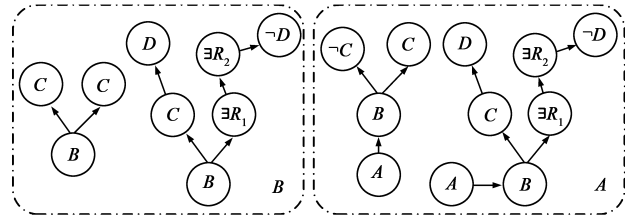


图 3 有向图上的 MUPP

基于图的 MUPS 的计算方法的主要过程是对图的遍历, 图上计算一个 MUPP 能在多项式时间内完成, 但在最坏的情况下可能存在指数多个 MUPS^[14], 因而对应于图上也可能存在指数多个 MUPP, 这导致算法达到指数级别的复杂度.

5 实验评估

本节将本文的方法与其他基于主流推理机的计算方法作性能上的对比, 实验中的本体来源不同的学科, 这些本体并不完全遵循 DL-Lite 的语法规则, 为此我们对其做特定的转换处理, 如对形如 $A \cup B \sqsubseteq C$ 的公理, 转换为 $A \sqsubseteq C$ 和 $B \sqsubseteq C$. 另外, 源本体大多是协调的, 预处理中将向本体注入导致某些概念或者角色不满足的公理, 如对形如 $\{A \sqsubseteq B, B \sqsubseteq C, B \sqsubseteq D\}$ 的 TBox, 向其添加公理 $\{C \sqsubseteq \neg D\}$ 使概念 A, B 不可满足. 上述处理可以参考文献[15]中提供的不一致注入工具 (<http://jfdulimewebs.com/dbo-debug/tools.html#injector>). 实验软硬件环境如下: 处理器为 Intel Core i5 主频为 3.1GHz, JAVA 堆的大小为 6GB; 各推理机为: Pellet 2.3.1, Hermit1.3.8, FaCT++ 1.6.3, JFaCT1.2.1.

实验中本体到图的转换需要耗费一定的计算时间, 但就 MUPS 的计算任务而言, 图转换的过程能够以离线的方式完成并且只需运行一次, 为了更详细记录实验, 我们列出本体到图的转换时间, 如表 1 所示:

表 1 本体到图的转换时间(单位: ms)

AEO	CL	DOLCE	Economy	Fly-Anatomy	FMA	GO	Plant	Terrorism	Transportation
225	685	108	168	812	15167	1447	342	127	319

文中的实验采用两种模式,一种模式在各类本体上展开计算实验,记录时间来比较各方法的执行效率;另一模式是通过向同一个本体中不断增加不可满足概念或者角色的数量,比较在冲突信息增量的情况下各个方法计算时间的变化。

表 2 中记录的是第一种模式下各个方法的实验结果(CalMUPSonGraph 缩写为 onGraph)。

表 2 各方法计算 MUPS 所耗费时间(单位:ms)

本体	Pellet	Hermit	FaCT++	JFaCT	onGraph
AEO	1284	1959	788	1517	389
CL	1428	2366	1148	2263	728
DOLCE	757	1038	472	809	365
Economy	1206	1364	616	1076	503
Fly-Anatomy	2013	3314	1873	4895	1159
FMA	47255	49438	45552	91742	12057
GO	3761	5367	3792	9310	1625
Plant	1540	2006	932	1880	664
Terrorism	383	441	382	448	167
Transportation	1768	2727	1330	4705	995

从表 2 提供的实验结果来看,基于图的 MUPS 计算方法的效率要好于其他方法,特别是在大规模的本体上,如 GO 和 FMA,算法的执行时间有显著的差异;在小本体上,各方法的计算速度都较快,性能差异并不明显。总体说来,本体规模越小以及 MUPS 的数量越少,计算 MUPS 所耗费的时间就越少。

冲突信息增量模式下的实验结果如表 3 所示。由于 GO 本体规模较大且在前一个实验中各方法在 GO 上的运行时间差异比较明显,便于效率的对比,因此增量实验选择 GO 本体做载体。

表 3 增量实验中计算 MUPS 所耗费时间(单位:ms)

不可满足概念 (角色)数	Pellet	Hermit	FaCT++	JFaCT	onGraph
100	4352	6086	3987	8873	1868
110	4637	6203	4337	10858	2076
120	4976	7150	4942	12629	2245
130	6602	8764	6418	17152	2292
140	8069	10124	7297	65967	2387
150	9439	12071	8704	内存溢出	2515
160	10100	13053	9619	内存溢出	2692
200	12870	16553	12483	内存溢出	3181

从表 3 同样可以发现,基于图的计算方法的执行效率上要优于各对比方法。由于基于 JFaCT 的方法出现内存溢出的情况,在后续的讨论中将剔除该方法。为

了便于观察,我们将增量实验的结果表示成图 4 所示。

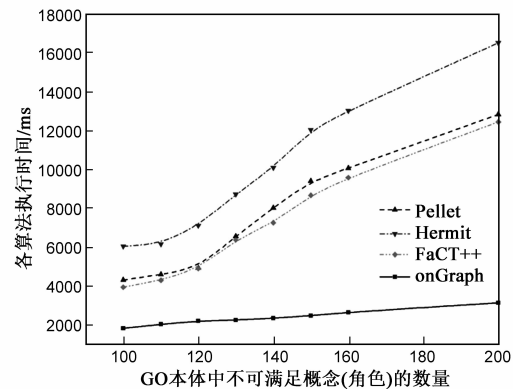


图 4 增量模式下各算法执行时间对比图

从图中可以看出,随本体中不可满足概念和角色的增加,所有方法的执行时间都随之增加。其中,基于 Pellet 和 FaCT++ 的计算方法耗费的时间很接近,增量的曲线也很相似;基于 Hermit 的计算方法耗费的时间最多并且增量曲线最陡,这表明该方法最不稳定。基于图的方法耗费的时间最少,并且执行时间的增量曲线最为平稳,表明该方法在冲突信息增量的情况下,计算时间的增量比较均匀,算法的稳定性较好。

6 结束语

本文提出了一种基于图的 MUPS 计算方法,该方法依据构建规则将本体转换成有向图并在图上定义最小不可满足保持路径对。实验结果表明,新方法有效的提高了 MUPS 计算的效率。下一步我们考虑将该方法推广到表达力更强的描述逻辑语言以及考虑应用并行的技术扩展基于图的 MUPS 计算方法。

参考文献

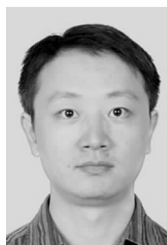
- [1] Console M, Lenzerini M. Data quality in ontology-based data access; the case of consistency [A]. Proceedings of the 28th AAAI Conference on Artificial Intelligence [C]. Québec: AAAI Press, 2014. 1020 - 1026.
- [2] Carnielli W A, Marcos J. Ex contradictione non sequitur quodlibet [J]. Bulletin of Advanced Reasoning and Knowledge, 2001, 27(1): 89 - 109.
- [3] de la Banda M J, Stuckey P J, Wazny J. Finding all minimal unsatisfiable subsets [A]. Proceedings of the 5th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming [C]. Uppsala: ACM, 2003. 32 - 43.
- [4] Schlobach S, Cornet R. Non-standard reasoning services for the debugging of description logic terminologies [A]. Proceedings of the 18th International Joint Conf. on Artificial Intelligence [C]. San Francisco: Morgan Kaufmann Pub-

- lishers, 2003. 355 – 362.
- [5] Kalyanpur A, Parsia B, Horridge M, et al. Finding all justifications of OWL DL entailments [A]. Proceedings of the 6th International Semantic Web Conference [C]. Berlin: Springer, 2007. 267 – 280.
- [6] Suntisrivaraporn B, Qi G, Ji Q, et al. A modularization-based approach to finding all justifications for OWL DL entailments [A]. Proceedings of the 3rd Asian Semantic Web Conference [C]. Berlin: Springer, 2008. 1 – 15.
- [7] Qin B, Wang S, Du X, et al. Graph-based query rewriting for knowledge sharing between peer ontologies [J]. Information Science, 2008, 178(18): 3525 – 3542.
- [8] Lembo D, Santarelli V, Savo D F. Graph-based ontology classification in OWL 2 QL [A]. Proceedings of the 26th International Workshop on Description Logics [C]. Ulm: CEUR-WS, 2013. 747 – 759.
- [9] Qi G, Wang Z, Wang K, et al. Approximating model-based ABox revision in DL-Lite: theory and practice [A]. Proceedings of the 29th AAAI Conference on Artificial Intelligence [C]. Menlo Park: AAAI Press, 2015. 254 – 260.
- [10] Calvanese D, De Giacomo G, Lembo D, et al. Tractable reasoning and efficient query answering in description logics: the DL-Lite family [J]. Journal of Automated Reasoning, 2007, 39(3): 385 – 429.
- [11] Fu X, Zhang Y, Qi G. A graph-based approach to ontology debugging in DL-Lite [A]. Proceedings of the 4th Joint International Conference on Semantic Technology [C]. Berlin: Springer, 2014. 33 – 46.
- [12] Sharir M. A strong-connectivity algorithm and its applications in data flow analysis [J]. Computers and Mathematics with Applications, 1981, 7(1): 67 – 72.
- [13] 周丽平, 黄厚宽, 漆桂林, 等. 一种在 DL-Lite 中计算本体最小不可满足保持子集的算法 [J]. 计算机研究与发展, 2011, 48(12): 2334 – 2342.
Zhou Li-ping, Huang Hou-kuang, Qi Gui-lin, et al. An algorithm for calculating minimal unsatisfiability-preserving subsets of ontology in DL-Lite [J]. Journal of Computer Research and Development, 2011, 48(12): 2334 – 2342. (in Chinese)
- [14] Baader F, ALoza R P N, Suntisrivaraporn B. Pinpointing in the description logic EL [A]. Proceedings of the 2007 International Workshop on Description Logics [C]. Berlin: Springer, 2007. 52 – 67.
- [15] Du J, Qi G. Decomposition-based optimization for debugging of inconsistent OWL DL ontologies [A]. Proceedings of the 4th International Conference on Knowledge Science, Engineering and Management [C]. Berlin: Springer, 2010. 88 – 100.

作者简介



付雪峰 男, 江西高安人, 1978 年出生. 南昌工程学院信息工程学院讲师, 主要研究方向: 本体调试与修正, 不一致性处理.
E-mail: fxf@seu.edu.cn



漆桂林 男, 江西宜丰人, 1977 年出生. 东南大学计算科学与工程学院教授, 博士生导师, 主要研究方向: 知识表达, 语义网, 不一致推理等.
E-mail: gqi@seu.edu.cn



张勇 男, 山东蓬莱人, 1989 年出生. 东南大学计算机科学与工程学院硕士研究生, 主要研究方向: 本体映射与调试.
E-mail: zhangyong@seu.edu.cn