软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(1):64−77 [doi: 10.13328/j.cnki.jos.004403]

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

基于 OBDD 的描述逻辑$\mathcal{EL}$循环术语集推理*

古天龙, 吕思菁, 常 亮, 徐周波

(广西可信软件重点实验室(桂林电子科技大学),广西 桂林 541004)

通讯作者: 古天龙, E-mail: cctlgu@guet.edu.cn

摘 要: 循环术语集推理是描述逻辑研究中面临的难点问题,尚未得到很好的解决.有序二叉决策图(ordered binary decision diagram,简称 OBDD)是一种对布尔函数进行紧凑表示和高效操作的数据结构,适用于表示和处理大规模问题.将 OBDD 应用于描述逻辑循环术语集的推理.首先,针对描述逻辑$\mathcal{EL}$中的循环术语集,给出了描述图上关于最大模拟关系的重要性质,并借助集合表示和集合运算对该性质进行了表述和证明.在此基础上,应用布尔函数对描述图进行编码,给出了基于OBDD求解最大模拟关系的方法,进而给出了最大不动点语义下基于OBDD对概念包含关系进行判定的算法;接下来,基于 OBDD 给出了求解描述图中可以到达循环路径的所有结点的方法,进而给出了最小不动点语义下基于OBDD对概念包含关系进行判定的算法;最后,对算法的正确性、复杂度等进行了分析和证明,并对算法进行了编程实现,给出了关于计算性能的实验结果.该工作为循环术语集的推理提供了一条有效途径,也为 OBDD 在逻辑推理中的应用提供了新的案例.

关键词: 描述逻辑$\mathcal{EL}$;循环术语集;有序二叉决策图;概念包含关系;不动点语义

中图法分类号: TP181 **文献标识码**: A

中文引用格式: 古天龙,吕思菁,常亮,徐周波.基于 OBDD 的描述逻辑$\mathcal{EL}$ 循环术语集推理.软件学报,2014,25(1):64−77. http://www.jos.org.cn/1000-9825/4403.htm

英文引用格式: Gu TL, Lü SJ, Chang L, Xu ZB. OBDD-Based reasoning for terminological cycles of the description logic $\mathcal{EL}$. Ruan Jian Xue Bao/Journal of Software, 2014,25(1):64−77 (in Chinese). http://www.jos.org.cn/1000-9825/4403.htm

OBDD-Based Reasoning for Terminological Cycles of the Description Logic $\mathcal{EL}$

GU Tian-Long, LÜ Si-Jing, CHANG Liang, XU Zhou-Bo

(Guangxi Key Laboratory of Trusted Software (Guilin University of Electronic Technology), Guilin 541004, China)

Corresponding author: GU Tian-Long, E-mail: cctlgu@guet.edu.cn

Abstract: Reasoning about terminological cycles of description logics is a hard problem that needs to be solved. Ordered binary decision diagram (OBDD) is a data structure suitable for dealing with large-scale problems since through the diagram Boolean functions can be represented compactly and computed efficiently. In this paper, OBDD is adopted to reason about terminological cycles of description logics. First, for terminological cycles of the description logic $\mathcal{EL}$, with the help of set representation and operation, a property about the greatest simulation relationship on description graphs of terminological cycles is specified and proved. Second, by encoding description graphs of terminological cycles as Boolean functions, an OBDD-based procedure for computing the greatest simulation relationship is proposed, and based on this procedure an algorithm is presented for deciding the subsumption relationship between concepts under the greatest fix-point semantics. Third, based on OBDD, a procedure for calculating all the nodes which can reach cycle paths in a description graph is proposed, and based on this procedure an algorithm for deciding the subsumption relationship under the least fix-point semantics is also presented. Finally, the correctness and time complexity of both algorithms are proved; the computing performance of both algorithms are also demonstrated by a set of experiments. This work provides an effective approach for reasoning about terminological cycles of description logics; it also provides a new case for applying OBDD in the fields of logical reasoning.

* 基金项目: 国家自然科学基金(60963010, 60903079, 61100025, 61262030, 61363030); 广西自然科学基金(2012GXNSFBA053169)

收稿时间: 2012-09-06; 定稿时间: 2013-03-27

Key words: description logic εL; terminological cycles; ordered binary decision diagram; concept subsumption; fix-point semantics

作为一类用于知识表示的形式化工具,描述逻辑在信息系统、软件工程、自然语言处理等领域得到了成功应用,尤其是在语义 Web 中扮演着关键角色,成为了语义 Web 本体语言 OWL 的逻辑基础.描述逻辑的主要特点在于提供了命题逻辑所无法比拟的刻画能力,同时保证了相关推理问题的可判定性,此外还拥有高效的推理机制作为支撑[1].

循环术语集是描述逻辑长期以来的研究难点,其语义定义问题和推理算法问题尚未得到很好的解决.因此,在目前对描述逻辑的研究和应用中,一般都假设所考察的知识库中不含有循环定义.但实际上,循环定义不仅可以扩充描述逻辑的表达能力,而且在许多应用中是不可避免的[2,3].此外,循环定义还能够方便用户建立描述逻辑知识库,使所刻画的知识符合人们的直觉;如果不使用循环定义,则会使得最终建立的知识库非常复杂,用户也难以理解[4,5].例如,在考察一个有向图时,令 Node 和 edge 分别对应于图中的所有结点和有向边,则我们可以将图中出现在无限长的通路上的所有结点刻画为[6]:Inode≡Node⊓∃edge.Inode;由于 Inode 出现在自身的定义式中,从而形成了循环定义.通常也将含有循环定义的知识库称为循环术语集[6].

Nebel[4,5]最早对循环术语集进行了研究,提出了循环术语集的 3 种语义:描述语义、最大不动点语义和最小不动点语义.其中,描述语义与一般情况下(即不含有循环定义时)描述逻辑中采用的语义相同;但是,由于循环定义的出现,使得一个循环术语集可能具有多个基于描述语义的解释.最大不动点语义和最小不动点语义以定义在解释结构上的偏序关系为基础,分别将基于该偏序关系的最大不动点和最小不动点作为唯一的语义解释.在多数情况下,采用最大不动点语义或最小不动点语义能够更好地符合人们刻画知识库的初衷.

在 Nebel 给出的语义解释的基础上,许多研究者对循环术语集的推理问题进行了研究.其中,针对由描述逻辑 FL_0 刻画的循环术语集,Baader[2]借助有穷状态自动机给出了对概念的可满足性和概念之间包含关系进行判定的算法.针对由描述逻辑εL 刻画的循环术语集,Baader[6]借助描述图给出了对概念的可满足性和概念之间包含关系进行判定的算法.之后,蒋运承等人[7]将 Baader 的工作扩展到描述逻辑εLN,给出了εLN 循环术语集在不动点语义下关于概念可满足性和概念包含关系的推理算法.王驹等人[8]借助描述图和模拟关系,给出了描述逻辑 vL 循环术语集关于可满足性和概念包含关系的推理算法.但到目前为止,研究者尚未对这些算法加以实现.目前,著名的描述逻辑推理机也不能在不动点语义下对描述逻辑循环术语集进行推理.

有序二叉决策图(ordered binary decision diagram,简称 OBDD)[9]是一种对布尔函数进行紧凑表示和高效操作的数据结构.由于其可以有效地缓解乃至避免对布尔函数进行处理时可能出现的状态组合爆炸问题,因而在硬件电路综合与验证、组合优化、智能调度、模型检测等领域都得到了成功应用[9,10].能否在逻辑推理领域发挥出 OBDD 的优点,是相关研究者关注的问题.许多研究者已在这方面进行了探索,其中,Uribe 和 Stickel[11]将 OBDD 应用于命题公式的可满足性判定,显示了 OBDD 在命题逻辑公式可满足性判定方面具有巨大的应用潜力.Groote[12]将 OBDD 用于一阶谓词逻辑的推理,将 OBDD 对命题逻辑的推理提升到一阶谓词逻辑层面.Marrero[13]将 OBDD 运用于分支时态逻辑 CTL 的推理,给出了基于 OBDD 的 CTL 可满足性判定算法.Pan 等人[14]将 OBDD 应用于模态逻辑的可满足性判定,实验结果表明,该方法在逻辑公式中出现模态词较多时具有明显的性能优势.

针对描述逻辑,Keller[15]最先提出基于 OBDD 进行逻辑推理的思路,但并未给出算法.之后,Rudolph 等人[16]针对描述逻辑 ALCIb 首次给出了基于 OBDD 的推理算法.由于可以将描述逻辑 SHIQ 的一致性问题在可满足性等价的情况下转换为描述逻辑 ALCIb 的一致性问题,因而,Rudolph 等人实际上也基于 OBDD 给出了描述逻辑 SHIQ 的一致性判定算法[17].但是,Rudolph 等人的算法不支持循环术语集;虽然 Rudolph 等人[16,17]在论文中没有明确地假设不含有循环定义,但其论文中考察的描述逻辑在语义定义上采用的是描述语义,因而相应的算法不支持循环定义所要求的最大不动点语义和最小不动点语义.此外,Rudolph 等人尚未对算法加以实现,不能从实际计算性能的角度对基于 OBDD 的推理算法进行考察.

本文将 OBDD 技术引入到对描述逻辑循环术语集的推理之中,针对由描述逻辑εL 刻画的循环术语集,在

Baader 等人[6]给出的基于描述图的判定方法的基础上,本文借助 OBDD 分别给出最大不动点语义下和最小不动点语义下对概念之间包含关系进行判定的算法.首先,针对最大不动点语义,本文借助集合运算给出描述图上关于最大模拟关系的一个重要性质,并基于该性质给出应用 OBDD 求解最大模拟关系的方法,进而得到最大不动点语义下应用 OBDD 判断概念包含关系的算法;其次,针对最小不动点语义,本文借助集合运算给出计算描述图中所有可以到达循环通路的结点的方法.在此基础上,得出最小不动点语义下应用 OBDD 判断概念包含关系的算法;最后,本文对上述两种算法进行优化实现,给出具体的实验结果.

本文第 1 节对背景知识进行介绍.第 2 节和第 3 节分别给出最大不动点语义下和最小不动点语义下基于 OBDD 对概念包含关系进行判定的算法.第 4 节对两种算法的正确性和时间复杂度进行分析,并与描述逻辑推理机 Pellet 进行实验比较.第 5 节对相关工作进行考察.第 6 节总结全文.

1 背景知识

本文涉及的背景知识主要包括 3 个方面:描述逻辑εL 循环术语集,Baader 等人给出的关于循环术语集推理的两个定理以及有序二叉决策图.

1.1 描述逻辑εL循环术语集

描述逻辑εL 的基本符号包括由角色名组成的集合N_R和由概念名组成的集合N_C.从这些符号出发可以递归地构造出角色和概念:R 是εL 中的角色当且仅当 $R\in N_R$;令 $A\in N_C$,R 为角色,则εL 中的概念由如下产生式生成:

$$C,D::=\top|A|C\sqcap D|\exists R.C.$$

εL 的解释 $I=(\Delta^I,\bullet^I)$由解释域Δ^I和解释函数$\bullet^I$构成.解释函数$\bullet^I$将每个角色名解释为Δ^I上的一个关系,将每个概念名解释为Δ^I的一个子集,递归定义如下:

(1) $\top^I=\Delta^I$;

(2) $A^I\subseteq\Delta^I$;

(3) $R^I\subseteq\Delta^I\times\Delta^I$;

(4) $(C\sqcap D)^I=C^I\cap D^I$;

(5) $(\exists R.C)^I=\{x\in\Delta^I|\exists y\in\Delta^I$ 使得$(x,y)\in R^I$ 并且 $y\in C^I\}$.

εL 的 TBox 是由形如 $C\equiv D$ 的概念定义式组成的有限集合,其中,$C\in N_C$,D 为εL 概念.给定一个 TBox T,如果概念名 C 出现在 T 中概念定义式的左边,则称 C 为被定义的概念名;否则,称 C 为原始概念名.

令 C 是一个被定义的概念名,其概念定义式为 $C\equiv D$.如果某个概念名 B 出现在 D 中,则称 C 直接使用了 B.将关系“直接使用”的传递闭包称为“使用”.在此基础上,如果 TBox 中存在某个被定义的概念名在概念定义式中“使用”了自身,则称该 TBox 中存在循环定义,也称该 TBox 为循环术语集.

文献中给出了循环术语集的 3 种语义定义[4,5].其中,由于描述语义与一般情况下(即不含有循环定义时)描述逻辑中采用的语义相同,本文不再详述.下面引入解释结构上的偏序关系,进而给出εL 循环术语集的最大不动点语义和最小不动点语义:

- 首先,给定描述逻辑εL 的任一 TBox T,令 N_{role},N_{prim},N_{def}是由出现在 T 中的所有角色名、原始概念名和被定义的概念名组成的集合,并且令 $N_{role}=\{R_1,\ldots,R_n\}$,$N_{prim}=\{P_1,\ldots,P_m\}$和 $N_{def}=\{C_1,\ldots,C_k\}$;
- 其次,给定一个解释 $J=(\Delta^J,\bullet^J)$,其中,对于任一 $R_i\in N_{role}$ 都有 $R_i^J\subseteq\Delta^J\times\Delta^J$,对于任一 $P_i\in N_{prim}$ 都有 $P_i^J\subseteq\Delta^J$.在此基础上,对于任一解释 $I=(\Delta^I,\bullet^I)$,如果$\Delta^I=\Delta^J$,并且对于任一 $R_i\in N_{role}$ 和任一 $P_i\in N_{prim}$ 都有 $R_i^I=R_i^J$ 和 $P_i^I=P_i^J$, 则称解释 J 是解释 I 的原始解释.相应地,称解释 I 是解释 J 的扩充,或者称 I 是基于 J 的解释;
- 最后,给定一个解释 J,令集合 $Int(J):=\{I|I$ 是基于 J 的解释$\}$,在 $Int(J)$上定义偏序关系$\preccurlyeq_J$如下:$I_1\preccurlyeq_J I_2$ 当且仅当对于任一 $C_i\in N_{def}$都有 $C_i^{I_1}\subseteq C_i^{I_2}$.

显然,由集合 $Int(J)$和偏序关系$\preccurlyeq_J$构成的二元组$\langle Int(J),\preccurlyeq_J\rangle$是一个完全格.由 Tarski 不动点定理可知,如果存

在单调函数 O:$Int(J)\to Int(J)$,使得对于任意 $I_1,I_2\in Int(J)$来说,当 $I_1\preccurlyeq_J I_2$ 时必然有 $O(I_1)\preccurlyeq_J O(I_2)$,则函数 O 必然存在不动点.

给定任一 TBox $T=\{C_1\equiv D_1,\ldots,C_k\equiv D_k\}$和解释 J,对函数 $O_{T,J}$:$Int(J)\to Int(J)$定义如下:$O_{T,J}(I_1)=I_2$ 当且仅当对于任一 $1\leqslant i\leqslant k$ 都有 $C_i^{I_2}\equiv D_i^{I_1}$. 容易证明,$O_{T,J}$ 是一个单调函数,并且存在最小不动点和最大不动点.

定义 1. 令 T 是εL 的任一 TBox.如果存在某个解释 $I=(\Delta^I,\bullet^I)$使得对于任一概念定义式 $C\equiv D\in T$ 都有 $C^I=D^I$,则称 I 是 T 的一个模型.如果 I 是 T 的模型,并且存在一个解释 J 使得 $I\in Int(J)$而且 I 是函数 $O_{T,J}$ 的最大不动点(最小不动点),则称 I 是 T 的最大不动点模型(最小不动点模型),简称为 *gfp*-模型(*lfp*-模型).

最大不动点语义仅接受最大不动点模型.相应地,最小不动点语义仅接受最小不动点模型.

定义 2. 令 T 是εL 的任一 TBox,A,B 是 T 中被定义的概念,则:

(1) 在最大不动点语义下 A 包含于 B,记为 $A\sqsubseteq_{gfp,T}B$,当且仅当对于 T 的任一 *gfp*-模型 I 都有 $A^I\subseteq B^I$;

(2) 在最小不动点语义下 A 包含于 B,记为 $A\sqsubseteq_{lfp,T}B$,当且仅当对于 T 的任一 *lfp*-模型 I 都有 $A^I\subseteq B^I$.

1.2 描述逻辑εL循环术语集的推理定理

为了对循环术语集中的概念包含关系进行判定,Baader 等人借助描述图给出了两个定理[6].下面首先引入一些符号和术语,然后对这两个定理进行介绍.本文的算法将建立在这两个定理的基础之上.

给定任一 TBox $T=\{C_1\equiv D_1,\ldots,C_k\equiv D_k\}$,称 T 是正规化的当且仅当对于任一 $C_i\equiv D_i(1\leqslant i\leqslant k)$来说,$D_i$ 具有如下形式:

$$P_1\sqcap\ldots\sqcap P_t\sqcap\exists R_1.B_1\sqcap\ldots\sqcap\exists R_s.B_s,$$

其中,$t,s\geqslant 0,P_1,\ldots,P_t\in N_{prim},R_1,\ldots,R_s\in N_{role},B_1,\ldots,B_s\in N_{def}$.如果 $t=s=0$,则 $D_i=\top$.

给定描述逻辑εL 的任一 TBox,可以在多项式时间内将其转化为在最大不动点语义(最小不动点语义)下等价的正规化的 TBox[6].在本文接下来的内容中,我们假设所讨论的 TBox 都是正规化的.

定义 3. 给定εL 的任一 TBox T,其描述图是一个三元组 $G_T=\langle V_T,E_T,L_T\rangle$,其中,

(1) $V_T=N_{def}$,是将 T 中每个被定义的概念作为一个结点后形成的结点集合;

(2) $E_T\subseteq V_T\times N_{role}\times V_T$,是用 T 中的角色名进行标记的有向边的集合;

(3) L_T:$V_T\to 2^{N_{prim}}$,为每个结点标记 T 中的若干个原始概念;

(4) L_T 和 E_T 满足如下性质:对于任一 $C\in V_T$,如果 C 在 T 中的概念定义式为 $C\equiv P_1\sqcap\ldots\sqcap P_m\sqcap\exists R_1.B_1\sqcap\ldots\sqcap\exists R_n.B_n$,则有 $L_T(C)=\{P_1,\ldots,P_m\}$和$\{(C,R_1,B_1),\ldots,(C,R_n,B_n)\}\subseteq E_T$.

例 1:考察 TBox $T=\{C_1\equiv P_1\sqcap P_2\sqcap P_3\sqcap\exists R_1.B_1\sqcap\exists R_2.B_2\sqcap\exists R_1.B_3,C_2\equiv P_2\sqcap P_3\sqcap\exists R_2.B_2\sqcap\exists R_1.B_3,C_3\equiv P_2\sqcap P_3\sqcap\exists R_2.B_2\sqcap\exists R_1.B_3,B_1\equiv\exists R_2.C_3,B_2\equiv\exists R_1.C_1,B_3\equiv P_1\sqcap P_2\}$,其描述图如图 1 所示.

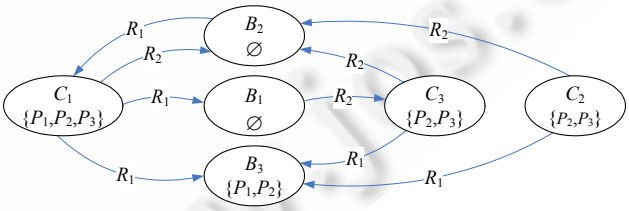


Fig.1 Description graph corresponding to the TBox T in Example 1

图 1 例 1 中 TBox T 的描述图

定义 4. 给定任意两个描述图 $G_1=\langle V_1,E_1,L_1\rangle$和 $G_2=\langle V_2,E_2,L_2\rangle$,将二元关系 $Z\subseteq V_1\times V_2$ 称为 G_1 到 G_2 的模拟关系,当且仅当 Z 满足以下条件:

(1) 如果$(v_1,v_2)\in Z$,则有 $L_1(v_1)\subseteq L_2(v_2)$;

(2) 如果$(v_1,v_2)\in Z$,且存在一个结点$v_1'\in V_1$使得$(v_1,R,v_1')\in E_1$,则必存在某个结点$v_2'\in V_2$使得$(v_2,R,v_2')\in E_2$和$(v_1',v_2')\in Z$.

如果Z是G_1到G_2的模拟关系,则记为$Z:G_1\approx G_2$.

借助描述图和模拟关系,可以在最大不动点语义下对概念之间的包含关系进行判定.判定定理如下:

定理 1[6]. 给定$\mathcal{EL}$的任一 TBox T和其中两个被定义的概念A,B,下面两个命题是互相等价的:

(1) $A\sqsubseteq_{gfp,T}B$;

(2) 对于T的描述图G_T,存在G_T到G_T的模拟关系Z,使得$(B,A)\in Z$.

在最小不动点语义下,也可以借助描述图和模拟关系进行推理.在给出推理定理之前需引入如下概念:

令T是$\mathcal{EL}$的任一 TBox,$G_T=\langle N_{def},E_T,L_T\rangle$是$T$的描述图.对于任意两个结点$A,B\in N_{def}$,如果图$G_T$中存在从$A$到$B$的一条路径,则记为$A\xrightarrow{*}_T B$;如果存在从$A$到$B$的一条非空路径,则记为$A\xrightarrow{+}_T B$.在此基础上,定义$Cyc_T$如下:$Cyc_T:=\{A|A\in N_{def}$,并且存在结点$B\in N_{def}$使得$A\xrightarrow{*}_T B\xrightarrow{+}_T B\}$.

定理 2[6]. 令T是$\mathcal{EL}$的任一 TBox,$G_T=\langle N_{def},E_T,L_T\rangle$是$T$的描述图,$G_T'$是从图$G_T$中删除所有出现在$Cyc_T$中的结点以及所有与这些结点相连的边之后得到的描述图,则$A\sqsubseteq_{lfp,T}B$当且仅当满足下列两种情况之一:

(1) $A\in Cyc_T$;

(2) $A\notin Cyc_T,B\notin Cyc_T$,并且存在$G_T'$到$G_T'$的模拟关系$Z$,使得$(B,A)\in Z$.

以这两个定理为基础,本文将借助 OBDD 分别给出最大不动点语义下和最小不动点语义下对概念包含关系进行判断的算法.

1.3 有序二叉决策图

有序二叉决策图(OBDD)是一种对布尔函数进行紧凑表示和高效操作的数据结构.给定含有n个命题变元$x_1,x_2,\ldots,x_n$的任一布尔函数$f(x_1,x_2,\ldots,x_n)$,其 OBDD 表示是一个十元组$W=(N,root,1,0,low,high,Var,\pi,\lambda,\varphi)$,其中,

(1) N是由结点组成的有限集合;

(2) $root\in N$为根结点;

(3) $0,1\in N$为终端结点;

(4) low和$high$是从$N\backslash\{0,1\}$到$N\backslash\{root\}$的函数,分别为每个非终端结点指定左孩子结点和右孩子结点;

(5) $Var=\{x_1,x_2,\ldots,x_n\}$;

(6) π是对Var中的各个变量指定一种排序,称为变量序;

(7) λ是从$N\backslash\{0,1\}$到Var的函数,为每个非终端结点赋予一个变量,并且使得从根结点到终端结点的任一路径上的变量均以变量序π所规定的次序依次出现;

(8) φ为每个结点赋予一个布尔函数,并且满足如下条件:

① 根结点对应的布尔函数为$\varphi(root)=f(x_1,x_2,\ldots,x_n)$;

② 终端结点对应的布尔函数分别为$\varphi(0)$=false 和$\varphi(1)$=true;

③ 对于每个非终端结点u,其对应的布尔函数为$\varphi(u)=(\neg\lambda(u)\wedge\varphi(low(u)))\vee(\lambda(u)\wedge\varphi(high(u)))$.

将布尔函数表示为 OBDD 之后,还可以对 OBDD 进行简化,得到紧凑的并且唯一的图形表示.在此基础上,对于布尔函数的各种操作(例如:否定、析取、合取、存在量化、全称量化等)都可以直接通过对 OBDD 的操作来高效地加以实现[9].

从应用的角度来看,OBDD 为集合提供了一种紧凑表示和高效操作的方式.首先,给定任一有限集合$S=\{s_1,s_2,\ldots,s_n\}$,我们可以利用二进制编码的形式,将S中的每个元素编码为$l=\lceil\log_2 n\rceil$位长的二进制串.在此基础上,将各个二进制串看成是含有l个命题变量$x_1,x_2,\ldots,x_l$的布尔函数的成真赋值,则可以将S中的每个元素唯一地表示为由l个命题变量构成的极小项.例如,若s_1编码为 00…0,s_2编码为 00…1,则元素s_1可以由极小项$m_0=\neg x_1\wedge\neg x_2\wedge\ldots\wedge\neg x_{l-1}\wedge\neg x_l$唯一表示,元素$s_2$可以由极小项$m_1=\neg x_1\wedge\neg x_2\wedge\ldots\wedge\neg x_{l-1}\wedge x_l$唯一表示.最后,将$S$中各个元素对应的极小项进行析取,便得到集合$S$所对应的布尔函数$\Delta(x_1,x_2,\ldots,x_l)$.

接下来,可以根据布尔函数构造出相应的 OBDD,使得对于任一极小项 m_i 来说:m_i 对应的元素 s_i 包含在集合 S 中当且仅当 OBDD 中存在对应于 m_i 的从根结点到终端结点 1 的一条路径.

最后,将集合隐式地表示为 OBDD 之后,关于集合的各种运算以及对集合包含等关系的判断都可以由相应的 OBDD 操作来高效地加以实现.例如,给定集合 S 和 T,令相应的布尔函数分别为$\Delta_S(x_1,x_2,\ldots,x_l)$和$\Delta_T(x_1,x_2,\ldots,x_l)$,则集合 S 和 T 的并、交和差运算分别对应于布尔运算$\Delta_{S\cup T}(x_1,x_2,\ldots,x_l):=\Delta_S(x_1,x_2,\ldots,x_l)\vee\Delta_T(x_1,x_2,\ldots,x_l)$,$\Delta_{S\cap T}(x_1,x_2,\ldots,x_l):=\Delta_S(x_1,x_2,\ldots,x_l)\wedge\Delta_T(x_1,x_2,\ldots,x_l)$和$\Delta_{S-T}(x_1,x_2,\ldots,x_l):=\Delta_S(x_1,x_2,\ldots,x_l)\wedge\neg(\Delta_T(x_1,x_2,\ldots,x_l))$,并且有 $S\subseteq T$ 当且仅当$\Delta_{S-T}(x_1,x_2,\ldots,x_l)$=false,而这些布尔运算都可以由相应的 OBDD 操作来高效地加以实现.

在接下来的内容中,我们将从集合的角度来研究循环术语集的推理问题,进而借助 OBDD 实现相关推理.

2 最大不动点语义下基于 OBDD 的εL 循环术语集推理

给定εL 的任一 TBox T,令 $G_T=\langle N_{def},E_T,L_T\rangle$是 T 的描述图,$N_{role}=\{R_1,\ldots,R_n\}$是出现在 T 中的所有角色.在给出基于 OBDD 的推理算法之前,下面首先引入一些集合,并给出关于模拟关系的一个性质.

首先,为每个角色 $R_i\in N_{role}$ 构造相应的集合 $S_{R_i}=\{(u,v)\in N_{def}\times N_{def}|(u,R_i,v)\in E_T\}$.

在此基础上,依次进行以下操作:

(1) 构造集合 $S_{old}:=\{(v_1,v_2)\in N_{def}\times N_{def}|L_1(v_1)\subseteq L_2(v_2)\}$;

(2) 对于每个角色 $R_i(1\leqslant i\leqslant n)$,依次求出如下两个集合:

- $E_{R_i}:=\{(v_1',v_2)\in N_{def}\times N_{def}|$存在 $v_2'\in N_{def}$ 使得 $(v_2,v_2')\in S_{R_i}$ 并且 $(v_1',v_2')\in S_{old}\}$;
- $A_{R_i}:=\{(v_1,v_2)\in N_{def}\times N_{def}|$对于任意 $v_1'\in N_{def}$:如果 $(v_1,v_1')\in S_{R_i}$,则必然有 $(v_1',v_2)\in E_{R_i}\}$;

(3) 令集合 $B_T:=S_{old}\cap A_{R_1}\cap\ldots\cap A_{R_n}$;

(4) 如果 $B_T=S_{old}$,则返回 B_T,操作结束;否则,令 $S_{old}:=B_T$,跳转到步骤(2).

定理 3. 令 B_T 是上述算法返回的集合,则 B_T 是 G_T 到 G_T 的最大模拟关系,即:

(1) B_T 是 G_T 到 G_T 的模拟关系;

(2) 对于 G_T 到 G_T 的任一模拟关系 Z,都有 $Z\subseteq B_T$.

证明:令 $B_{T\_k},S_{old\_k},E_{R_i\_k},A_{R_i\_k}$ $(k\geqslant 1)$分别是步骤(3)第 k 次执行结束时集合 B_T,S_{old},E_{R_i} 和 A_{R_i} 的值.显然,$B_{T\_k}\subseteq B_{T\_k-1}\subseteq\ldots\subseteq B_{T\_1}\subseteq S_{old\_1}$.即,对于任一元素$(v_1,v_2)\in B_{T_k}$都有 $L_1(v_1)\subseteq L_2(v_2)$.

令算法停止时对步骤(3)总共执行了 m 次,则有 $B_T=B_{T\_m}=S_{old\_m}$,因此有 $B_T=B_T\cap A_{R_1\_m}\cap\ldots\cap A_{R_n\_m}$,进而有 $B_T\subseteq A_{R_1\_m}\cap\ldots\cap A_{R_n\_m}$,其中,

- $A_{R_i_m}=\{(v_1,v_2)\in N_{def}\times N_{def}|$对于任意 $v_1'\in N_{def}$:如果 $(v_1,v_1')\in S_{R_i}$,则必然有 $(v_1',v_2)\in E_{R_i\_m}\}$;
- $E_{R_i_m}=\{(v_1',v_2)\in N_{def}\times N_{def}|$存在 $v_2'\in N_{def}$ 使得 $(v_2,v_2')\in S_{R_i}$ 并且 $(v_1',v_2')\in B_T\}$.

因此,对于任一元素$(v_1,v_2)\in B_T$,必然有 $L_1(v_1)\subseteq L_2(v_2)$;并且,对于任一角色 $R_i(1\leqslant i\leqslant n)$和任一结点 $v_1'\in N_{def}$,如果$(v_1,R_i,v_1')\in E_T$,则必然存在某个结点 $v_2'\in N_{def}$ 使得$(v_2,R_i,v_2')\in E_T$ 并且 $(v_1',v_2')\in B_T$.根据定义 4,B_T 是 G_T 到 G_T 的一个模拟关系.

令 Z 是从 G_T 到 G_T 的任一模拟关系,根据定义 4,必然有 $Z\subseteq\{(v_1,v_2)\in N_{def}\times N_{def}|L_1(v_1)\subseteq L_2(v_2)\}$,即 $Z\subseteq S_{old\_1}$.假设 Z 不包含于 B_T,则必然存在某个正整数 $l\geqslant 1$ 使得 $Z\subseteq S_{old\_l}$ 但 Z 不包含于 S_{old_l+1},进而必然存在某个二元组 $(v_1,v_2)\in Z$ 使得$(v_1,v_2)\in S_{old_l}$但$(v_1,v_2)\notin S_{old_l+1}$.又根据算法有 $S_{old\_l+1}=S_{old\_l}\cap A_{R_1\_l}\cap\ldots\cap A_{R_n\_l}$,因此必然存在某个角色 R_i 使得$(v_1,v_2)\notin A_{R_i_l}$,进而必然存在某个 $v_1'\in N_{def}$ 使得 $(v_1,v_1')\in S_{R_i}$ 但 $(v_1',v_2)\notin E_{R_i\_l}$,即存在某个 $v_1'\in N_{def}$ 使得 $(v_1,R_i,v_1')\in E_T$ 但不存在结点 $v_2'\in N_{def}$ 使得$(v_2,R_i,v_2')\in E_T$ 和 $(v_1',v_2')\in S_{old\_l}$.又由于 $Z\subseteq S_{old\_l}$,因此对于二元组 $(v_1,v_2)\in Z$,存在某个 $v_1'\in N_{def}$ 使得$(v_1,R_i,v_1')\in E_T$,但不存在结点 $v_2'\in N_{def}$ 使得$(v_2,R_i,v_2')\in E_T$和 $(v_1',v_2')\in Z$.根据定义 4,Z 不是 G_T 到 G_T 的模拟关系,产生矛盾.因此得证 $Z\subseteq B_T$. □

基于定理 3,我们可以给出最大不动点语义下借助 OBDD 判断概念包含关系的算法.

算法 1. 给定εL 的任一 TBox T,按如下步骤判断 T 中被定义的概念之间是否存在最大不动点语义下的包含

关系:

(1) 构造出 T 的描述图 $G_T=\langle N_{def},E_T,L_T\rangle$;

(2) 通过以下步骤求出 G_T 到 G_T 的最大模拟关系:

① 令集合 N_{def} 的元素个数为 k,用 $l=\lceil \log_2 k\rceil$ 位长的二进制串对集合 N_{def} 中的元素进行编码,进而用含有 l 个命题变量的极小项对集合 N_{def} 中的各元素进行唯一的表示;

② 针对每个角色 $R_i\in N_{role}$,分别用命题变量 $x_1,x_2,\ldots,x_l$ 构成的极小项和命题变量 $y_1,y_2,\ldots,y_l$ 构成的极小项来唯一表示集合 $S_{R_i}=\{(u,v)\in N_{def}\times N_{def}|(u,R_i,v)\in E_T\}$ 中各元素的第 1 元和第 2 元,进而将集合 S_{R_i} 表示为布尔函数 $\Delta_{R_i}(x_1,\ldots,x_l,y_1,\ldots,y_l):\{0,1\}^l\times\{0,1\}^l\to\{0,1\}$,使得:

$$\Delta_{R_i}(x_1,\ldots,x_l,y_1,\ldots,y_l)=\begin{cases}1, & \text{若}(u,R_i,v)\in E_T\\ 0, & \text{其他}\end{cases};$$

③ 基于上述编码方式对集合 $S_{old}=\{(v_1,v_2)\in N_{def}\times N_{def}|L_T(v_1)\subseteq L_T(v_2)\}$ 中的元素进行编码,并构造相应的极小项,得到集合 S_{old} 的布尔函数 $\Delta_{old}(x_1,\ldots,x_l,y_1,\ldots,y_l):\{0,1\}^l\times\{0,1\}^l\to\{0,1\}$,使得:

$$\Delta_{old}(x_1,\ldots,x_l,y_1,\ldots,y_l)=\begin{cases}1, & \text{若}\ L_T(v_1)\subseteq L_T(v_2)\\ 0, & \text{其他}\end{cases};$$

④ 针对每个角色 R_i,求出与集合 E_{R_i} 对应的布尔函数:

$$\Delta_{E\_R_i}(z_1,\ldots,z_l,y_1,\ldots,y_l):=\exists(u_1,\ldots,u_l)(\Delta_{R_i}(y_1,\ldots,y_l,u_1,\ldots,u_l)\wedge\Delta_{old}(z_1,\ldots,z_l,u_1,\ldots,u_l));$$

在此基础上,求出与集合 A_{R_i} 对应的布尔函数:

$$\Delta_{A\_R_i}(x_1,\ldots,x_l,y_1,\ldots,y_l):=\forall(z_1,\ldots,z_l)(\Delta_{R_i}(x_1,\ldots,x_l,z_1,\ldots,z_l)\to\Delta_{E\_R_i}(z_1,\ldots,z_l,y_1,\ldots,y_l));$$

⑤ 求出布尔函数:

$$\Delta_T(x_1,\ldots,x_l,y_1,\ldots,y_l):=\Delta_{old}(x_1,\ldots,x_l,y_1,\ldots,y_l)\wedge\Delta_{A\_R_1}(x_1,\ldots,x_l,y_1,\ldots,y_l)\wedge\ldots\wedge\Delta_{A\_R_n}(x_1,\ldots,x_l,y_1,\ldots,y_l);$$

⑥ 若 $\Delta_T(x_1,\ldots,x_l,y_1,\ldots,y_l)\neq\Delta_{old}(x_1,\ldots,x_l,y_1,\ldots,y_l)$,则令 $\Delta_{old}(x_1,\ldots,x_l,y_1,\ldots,y_l):=\Delta_T(x_1,\ldots,x_l,y_1,\ldots,y_l)$,并且跳转到步骤④继续执行;

(3) 对于任意两个被定义的概念 A,B,将集合 $\{(B,A)\}$ 表示为布尔函数 $\Delta_{\{(B,A)\}}(x_1,\ldots,x_l,y_1,\ldots,y_l)$.如果 $\Delta_{\{(B,A)\}}(x_1,\ldots,x_l,y_1,\ldots,y_l)\wedge\neg\Delta_T(x_1,\ldots,x_l,y_1,\ldots,y_l)=\text{false}$,则返回"$A\sqsubseteq_{gfp,T}B$";否则,返回"$A\not\sqsubseteq_{gfp,T}B$".

上述算法中,对布尔函数的表示和运算都直接通过 OBDD 来实现.

下面通过一个例子来说明上述算法.

例 2:对于 TBox $T=\{A\equiv P_1,B\equiv P_1\sqcap P_2,C\equiv P_3\sqcap\exists R.B\sqcap\exists R.D,D\equiv P_3\sqcap P_4\sqcap\exists R.C\sqcap\exists R.A\}$,在最大不动点语义下判断 D 和 C 之间是否存在包含关系,B 和 A 之间是否存在包含关系.

首先构造出 T 的描述图 $G_T=\langle N_{def},E_T,L_T\rangle$,如图 2 所示.

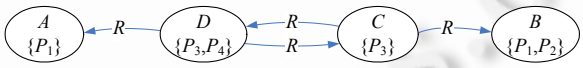


Fig.2 Description graph corresponding to the TBox T in Example 2

图 2 例 2 中 TBox T 的描述图

由于 TBox 中总共有 4 个被定义的概念,可以用长度为 $l=\lceil\log_2 4\rceil=2$ 的二进制串对 4 个被定义的概念进行编码.令 A 编码为 00(对应的极小项为 $\neg x_1\wedge\neg x_2$),B 编码为 01(对应的极小项为 $\neg x_1\wedge x_2$),C 编码为 10(对应的极小项为 $x_1\wedge\neg x_2$),D 编码为 11(对应的极小项为 $x_1\wedge x_2$).

由于 TBox 中只有一个角色 R,由描述图 G_T 可得 $S_R=\{(D,A),(C,D),(C,B),(D,C)\}$.对集合 S_R 中元素的第 1 元和第 2 元分别用命题变量 x_1,x_2 构成的极小项和命题变量 y_1,y_2 构成的极小项表示,可以得到:

$$\Delta_R(x_1,x_2,y_1,y_2):=(x_1\wedge x_2\wedge\neg y_1\wedge\neg y_2)\vee(x_1\wedge\neg x_2\wedge y_1\wedge y_2)\vee(x_1\wedge\neg x_2\wedge\neg y_1\wedge y_2)\vee(x_1\wedge x_2\wedge y_1\wedge\neg y_2)$$
$$:=(x_1\wedge x_2\wedge\neg y_2)\vee(x_1\wedge\neg x_2\wedge y_2).$$

由描述图 G_T 可得 $S_{old}=\{(v_1,v_2)\in N_{def}\times N_{def}|L_1(v_1)\subseteq L_2(v_2)\}=\{(A,A),(B,B),(C,C),(D,D),(C,D),(A,B)\}$,转换为布尔

函数并化简后可得:

$$\Delta_{old}(x_1,x_2,y_1,y_2):=(\neg x_1\wedge\neg x_2\wedge\neg y_1)\vee(\neg x_1\wedge x_2\wedge\neg y_1\wedge y_2)\vee(x_1\wedge\neg x_2\wedge y_1)\vee(x_1\wedge x_2\wedge y_1\wedge y_2).$$

对于角色 R,依次构造出与集合 E_R 和集合 A_R 相对应的布尔函数Δ_{E_R}和Δ_{A_R},如下所示:

$$\Delta_{E\_R}(z_1,z_2,y_1,y_2):=\exists u_1\exists u_2(\Delta_R(y_1,y_2,u_1,u_2)\wedge\Delta_{old}(z_1,z_2,u_1,u_2)):=(\neg z_2\wedge y_1\wedge y_2)\vee(y_1\wedge\neg y_2);$$

$$\Delta_{A\_R}(x_1,x_2,y_1,y_2):=\forall z_1\forall z_2(\Delta_R(x_1,x_2,z_1,z_2)\rightarrow\Delta_{E\_R}(z_1,z_2,y_1,y_2)):=\neg x_1\vee(x_2\wedge y_1)\vee(y_1\wedge\neg y_2).$$

接下来计算Δ_T:

$$\Delta_T(x_1,x_2,y_1,y_2):=\Delta_{old}(x_1,x_2,y_1,y_2)\wedge\Delta_{A\_R}(x_1,x_2,y_1,y_2)$$
$$:=(\neg x_1\wedge\neg x_2\wedge\neg y_1)\vee(\neg x_1\wedge x_2\wedge\neg y_1\wedge y_2)\vee(x_1\wedge\neg x_2\wedge y_1\wedge\neg y_2)\vee(x_1\wedge x_2\wedge y_1\wedge y_2).$$

由于$\Delta_T\neq\Delta_{old}$,所以令$\Delta_{old}:=\Delta_T$,跳转到步骤④进行第 2 次迭代.

第 2 次执行步骤④后可得:

$$\Delta_{E\_R}(z_1,z_2,y_1,y_2):=(\neg z_2\wedge y_1\wedge y_2)\vee(z_2\wedge y_1\wedge\neg y_2)\vee(\neg z_1\wedge\neg z_2\wedge y_1\wedge\neg y_2),$$

$$\Delta_{A\_R}(x_1,x_2,y_1,y_2):=\neg x_1\vee(x_2\wedge y_1\wedge y_2)\vee(\neg x_2\wedge y_1\wedge\neg y_2).$$

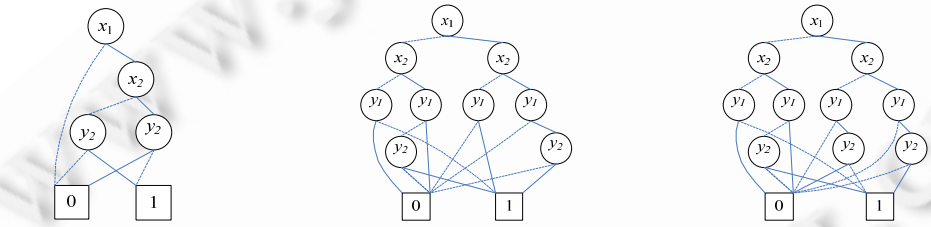
接下来执行步骤⑤后可得:

$$\Delta_T(x_1,x_2,y_1,y_2):=((\neg x_1\wedge\neg x_2\wedge\neg y_1)\vee(\neg x_1\wedge x_2\wedge\neg y_1\wedge y_2)\vee(x_1\wedge\neg x_2\wedge y_1\wedge\neg y_2)\vee(x_1\wedge x_2\wedge y_1\wedge y_2)).$$

此时有$\Delta_T=\Delta_{old}$,不再进行循环,算法将执行步骤(3).

假设需要判断概念 C 在最大不动点语义下是否包含于概念 D.根据步骤(3),首先将集合$\{(D,C)\}$表示为布尔函数得到$\Delta_{\{(D,C)\}}(x_1,x_2,y_1,y_2):=x_1\wedge x_2\wedge y_1\wedge\neg y_2$;由于$\Delta_{\{(D,C)\}}(x_1,x_2,y_1,y_2)\wedge\neg\Delta_T(x_1,x_2,y_1,y_2)=x_1\wedge x_2\wedge y_1\wedge\neg y_2\neq$false,因此得到结论"$C\not\sqsubseteq_{gfp,\mathcal{T}}D$".类似地,对于本例还可以得到 $A\not\sqsubseteq_{gfp,\mathcal{T}}B$,$B\sqsubseteq_{gfp,\mathcal{T}}A$ 和 $D\not\sqsubseteq_{gfp,\mathcal{T}}C$ 等结论.

为了表述简洁,上面例子中仅从布尔函数的层面对计算过程进行了考察.在实际计算时,对布尔函数的表示和操作实际上都是借助 OBDD 来实现的,其中,与布尔函数Δ_R,Δ_{old}和Δ_T对应的 OBDD 如图 3 所示.



(a) Δ_R 对应的 OBDD (b) Δ_{old} 对应的 OBDD (c) Δ_T 对应的 OBDD

Fig.3 OBDDs corresponding to Δ_R, Δ_{old} and Δ_T

图 3 Δ_R,Δ_{old}和Δ_T对应的 OBDD

3 最小不动点语义下基于 OBDD 的$\mathcal{EL}$循环术语集推理

给定$\mathcal{EL}$的任一 TBox $\mathcal{T}$,令 $G_{\mathcal{T}}=\langle N_{def},E_{\mathcal{T}},L_{\mathcal{T}}\rangle$是 $\mathcal{T}$ 的描述图.由定理 2 可知,为了在最小不动点语义下判断概念之间的包含关系,首先需要求出集合 $Cyc_{\mathcal{T}}$.下面借助 OBDD 对集合 $Cyc_{\mathcal{T}}$ 进行求解.

首先,根据 $Cyc_{\mathcal{T}}$ 的定义,我们关注的仅仅是图 $G_{\mathcal{T}}$ 中结点之间的连接关系,而对 $G_{\mathcal{T}}$ 中各条边上标记的角色名并不关心.因此,我们可以去掉 $E_{\mathcal{T}}$ 中出现的角色名,得到集合 $E=\{(v,v')|$存在某个角色 R 使得$(v,R,v')\in E_{\mathcal{T}}\}$.应用第 2 节的编码方法,将集合 E 表示为布尔函数$\Delta_E(x_1,...,x_l,y_1,...,y_l)$.

其次,对于任一集合$S\subseteq N_{def}$,我们用 $Pre(S)$表示集合 S 中的元素在图 $G_{\mathcal{T}}$中的直接前驱结点,即 $Pre(S)=\{v\in N_{def}|$存在$v'\in S$使得$(v,v')\in E\}$.令对集合 S编码后得到布尔函数$\Delta_S(z_1,...,z_l)$,则与 $Pre(S)$相对应的布尔函数$\Delta_{Pre(S)}(x_1,...,x_l)$可由下述公式计算出来:

$$\Delta_{Pre(S)}(x_1,...,x_l):=\exists(z_1,...,z_l)(\Delta_S(z_1,...,z_l)\wedge\Delta_E(x_1,...,x_l,z_1,...,z_l)).$$

第三,对于任一结点 $v\in N_{def}$,我们用集合 $Front(v)$表示图 $G_{\mathcal{T}}$中可以到达 v 的所有结点,用 $Back(v)$表示图 $G_{\mathcal{T}}$

中从 v 出发可以到达的所有结点,即 $Front(v):=\{v'\in N_{def}\mid v'\xrightarrow{+}_T v\}$,$Back(v):=\{v'\in N_{def}\mid v\xrightarrow{+}_T v'\}$.令结点 v 编码后对应的布尔函数为 $\Delta_{\{v\}}(z_1,...,z_l)$,则与集合 $Front(v)$ 对应的布尔函数 $\Delta_{Front(v)}(x_1,...,x_l)$ 可由下列操作计算出来:

① $\Delta_{Temp}(x_1,...,x_l):=\exists(z_1,...,z_l)(\Delta_{\{v\}}(z_1,...,z_l)\wedge\Delta_E(x_1,...,x_l,z_1,...,z_l))$;

② $\Delta_{Front(v)}(x_1,...,x_l):=\Delta_{Temp}(x_1,...,x_l)\vee\exists(z_1,...,z_l)(\Delta_{Temp}(z_1,...,z_l)\wedge\Delta_E(x_1,...,x_l,z_1,...,z_l))$;

③ 若 $\Delta_{Front(v)}(x_1,...,x_l)\neq\Delta_{Temp}(x_1,...,x_l)$,则令 $\Delta_{Temp}(x_1,...,x_l):=\Delta_{Front(v)}(x_1,...,x_l)$,并跳转到步骤②;
否则,返回 $\Delta_{Front(v)}(x_1,...,x_l)$.

类似地,与集合 $Back(v)$ 对应的布尔函数 $\Delta_{Back(v)}(x_1,...,x_l)$ 可通过以下操作求出:

① $\Delta_{Temp}(x_1,...,x_l):=\exists(z_1,...,z_l)(\Delta_{\{v\}}(z_1,...,z_l)\wedge\Delta_E(z_1,...,z_l,x_1,...,x_l))$;

② $\Delta_{Back(v)}(x_1,...,x_l):=\Delta_{Temp}(x_1,...,x_l)\vee\exists(z_1,...,z_l)(\Delta_{Temp}(z_1,...,z_l)\wedge\Delta_E(z_1,...,z_l,x_1,...,x_l))$;

③ 若 $\Delta_{Back(v)}(x_1,...,x_l)\neq\Delta_{Temp}(x_1,...,x_l)$,则令 $\Delta_{Temp}(x_1,...,x_l):=\Delta_{Back(v)}(x_1,...,x_l)$,并跳转到步骤②;
否则,返回 $\Delta_{Back(v)}(x_1,...,x_l)$.

第四,定义集合 $Scc:=\{B\mid B\in N_{def}$ 并且 $B\xrightarrow{+}_T B\}$.令对集合 N_{def} 编码后得到布尔函数 $\Delta_N(x_1,...,x_l)$,则可通过以下操作求出与集合 Scc 对应的布尔函数 $\Delta_{Scc}(x_1,...,x_l)$:

① $\Delta_{Scc}(x_1,...,x_l):=$false;

② $\Delta_{Temp}(x_1,...,x_l):=\Delta_N(x_1,...,x_l)$;

③ 设 S 是 $\Delta_{Temp}(x_1,...,x_l)$ 表示的集合,从中随机选取一个结点 v,求出与集合 $\{v\}$ 对应的布尔函数 $\Delta_{\{v\}}(x_1,...,x_l)$,令 $\Delta_{Temp}(x_1,...,x_l):=\Delta_{Temp}(x_1,...,x_l)\wedge\neg\Delta_{\{v\}}(x_1,...,x_l)$;

④ 求出 $\Delta_{Front(v)}(x_1,...,x_l)$ 和 $\Delta_{Back(v)}(x_1,...,x_l)$;若 $\Delta_{Front(v)}(x_1,...,x_l)\wedge\Delta_{Back(v)}(x_1,...,x_l)\neq$false,则令

$$\Delta_{Scc}(x_1,...,x_l):=\Delta_{Scc}(x_1,...,x_l)\vee\Delta_{\{v\}}(x_1,...,x_l)\vee(\Delta_{Front(v)}(x_1,...,x_l)\wedge\Delta_{Back(v)}(x_1,...,x_l)),$$

并且令

$$\Delta_{Temp}(x_1,...,x_l):=\Delta_{Temp}(x_1,...,x_l)\wedge\neg(\Delta_{Front(v)}(x_1,...,x_l)\wedge\Delta_{Back(v)}(x_1,...,x_l));$$

⑤ 若 $\Delta_{Temp}(x_1,...,x_l)\neq$false,则跳转到步骤③;否则,算法结束,返回 $\Delta_{Scc}(x_1,...,x_l)$.

最后,可以通过以下步骤计算出与集合 Cyc_T 对应的布尔函数 $\Delta_{Cyc}(x_1,...,x_l)$:

① $\Delta_{Cyc}(x_1,...,x_l):=\Delta_{Scc}(x_1,...,x_l)$;

② $\Delta_{Temp}(x_1,...,x_l):=\Delta_{Scc}(x_1,...,x_l)$;

③ 设 S 是 $\Delta_{Temp}(x_1,...,x_l)$ 表示的集合,求出 $\Delta_{Pre(S)}(x_1,...,x_l)$,令 $\Delta_{Cyc}(x_1,...,x_l):=\Delta_{Cyc}(x_1,...,x_l)\vee\Delta_{Pre(S)}(x_1,...,x_l)$;

④ 若 $\Delta_{Cyc}(x_1,...,x_l)\neq\Delta_{Temp}(x_1,...,x_l)$,则令 $\Delta_{Temp}(x_1,...,x_l):=\Delta_{Cyc}(x_1,...,x_l)$,跳转到步骤③;
否则,算法结束,返回 $\Delta_{Cyc}(x_1,...,x_l)$.

在上述操作的基础上,基于定理 2,我们可以给出最小不动点语义下借助 OBDD 判断包含关系的算法.

算法 2. 给定 εL 的任一 TBox T,按如下步骤判断 T 中被定义的概念之间是否存在最小不动点语义下的包含关系:

(1) 构造出 T 的描述图 $G_T=\langle N_{def},E_T,L_T\rangle$;

(2) 令集合 N_{def} 的元素个数为 k,用 $l=\lceil\log_2 k\rceil$ 位长的二进制串对集合 N_{def} 中的元素进行编码,进而用含有 l 个命题变量的极小项对集合 N_{def} 中的各元素进行唯一表示;在该编码的基础上,借助 OBDD 实现以下操作:

① 求出与集合 Cyc_T 对应的布尔函数 $\Delta_{Cyc}(x_1,...,x_l)$;

② 针对每个角色 $R_i\in N_{role}$,首先利用编码的形式构造出与集合 $S_{R_i}=\{(u,v)\in N_{def}\times N_{def}\mid(u,R_i,v)\in E_T\}$ 对应的布尔函数 $\Delta_{R_i}(x_1,...,x_l,y_1,...,y_l)$.在此基础上,求出布尔函数 $\Delta'_{R_i}(x_1,...,x_l,y_1,...,y_l):=\Delta_{R_i}(x_1,...,x_l,y_1,...,y_l)\wedge\neg\Delta_{Cyc}(x_1,...,x_l)\wedge\neg\Delta_{Cyc}(y_1,...,y_l)$,其中的 $\Delta_{Cyc}(x_1,...,x_l)$ 和 $\Delta_{Cyc}(y_1,...,y_l)$ 表示将 $\Delta_{Cyc}(z_1,...,z_l)$ 中的变量分别置换为 $(x_1,...,x_l)$ 和 $(y_1,...,y_l)$;

③ 利用编码的形式构造出与集合 $S_{old}:=\{(v_1,v_2)\in N_{def}\times N_{def}\mid L_1(v_1)\subseteq L_2(v_2)\}$ 对应的布尔函数 $\Delta_{old}(x_1,...,x_l,y_1,...,y_l)$.在此基础上,求出布尔函数

$$\Delta'_{old}(x_1,...,x_l,y_1,...,y_l):=\Delta_{old}(x_1,...,x_l,y_1,...,y_l)\wedge\neg\Delta_{Cyc}(x_1,...,x_l)\wedge\neg\Delta_{Cyc}(y_1,...,y_l);$$

④ 针对每个角色 R_i,首先求出布尔函数

$$\Delta'_{E\_R_i}(z_1,...,z_l,y_1,...,y_l):=\exists(u_1,...,u_l)(\Delta'_{R_i}(y_1,...,y_l,u_1,...,u_l)\wedge\Delta'_{old}(z_1,...,z_l,u_1,...,u_l)),$$

然后求出布尔函数 $\Delta'_{A\_R_i}(x_1,...,x_l,y_1,...,y_l):=\forall(z_1,...,z_l)(\Delta'_{R_i}(x_1,...,x_l,z_1,...,z_l)\rightarrow\Delta'_{E\_R_i}(z_1,...,z_l,y_1,...,y_l))$;

⑤ 求出布尔函数

$$\Delta'_T(x_1,...,x_l,y_1,...,y_l):=\Delta'_{old}(x_1,...,x_l,y_1,...,y_l)\wedge\Delta'_{A\_R_1}(x_1,...,x_l,y_1,...,y_l)\wedge...\wedge\Delta'_{A\_R_n}(x_1,...,x_l,y_1,...,y_l);$$

⑥ 如果 $\Delta'_T(x_1,...,x_l,y_1,...,y_l)\neq\Delta'_{old}(x_1,...,x_l,y_1,...,y_l)$,则令 $\Delta'_{old}(x_1,...,x_l,y_1,...,y_l):=\Delta'_T(x_1,...,x_l,y_1,...,y_l)$,并且跳转到步骤④继续执行;

(3) 对于 T 中任意两个被定义的概念 A,B,将集合$\{A\},\{B\},\{(B,A)\}$分别表示为布尔函数$\Delta_{\{A\}}(z_1,...,z_l)$,$\Delta_{\{B\}}(z_1,...,z_l)$和$\Delta_{\{(B,A)\}}(x_1,...,x_l,y_1,...,y_l)$;如果$\Delta_{\{A\}}(z_1,...,z_l)\wedge\neg\Delta_{Cyc}(z_1,...,z_l)$=false,或者$\Delta_{\{A\}}(z_1,...,z_l)\wedge\neg\Delta_{Cyc}(z_1,...,z_l)\neq$false 并且有$\Delta_{\{B\}}(z_1,...,z_l)\wedge\neg\Delta_{Cyc}(z_1,...,z_l)\neq$false 和$\Delta_{\{(B,A)\}}(x_1,...,x_l,y_1,...,y_l)\wedge\neg\Delta'_T(x_1,...,x_l,y_1,...,y_l)$=false,则返回"$A\sqsubseteq_{lfp,T}B$",其他情况都返回"$A\not\sqsubseteq_{lfp,T}B$".

上述算法中,对布尔函数的表示和运算都直接通过 OBDD 来实现.

下面通过一个例子来说明上述算法.

例 3:对于例 2 给出的 TBox T,在最小不动点语义下判断 D 和 C 之间是否存在包含关系,B 和 A 之间是否存在包含关系.

首先,构造出 T 的描述图 $G_T=\langle N_{def},E_T,L_T\rangle$,与例 2 中相同;其次,采用与例 2 相同的形式对 N_{def} 中的元素进行编码;接下来,计算与集合 Cyc_T 对应的布尔函数Δ_{Cyc}.

在计算Δ_{Cyc} 的过程中,首先将得到与集合 N_{def} 对应的布尔函数$\Delta_N(x_1,x_2):=$true,以及与集合 $E=\{(v,v')|$存在某个角色 R 使得$(v,R,v')\in E_T\}=\{(D,A),(D,C),(C,D),(C,B)\}$对应的布尔函数$\Delta_E(x_1,x_2,y_1,y_2):=(x_1\wedge x_2\wedge\neg y_2)\vee(x_1\wedge\neg x_2\wedge y_2)$.通过多次迭代,将得到与集合 $Scc:=\{B|B\in N_{def}$ 并且 $B\xrightarrow{+}_T B\}$对应的布尔函数$\Delta_{Scc}(x_1,x_2):=x_1$.最终,可以得到与集合 Cyc_T 对应的布尔函数$\Delta_{Cyc}(x_1,x_2):=x_1$.

针对角色 R,与例 2 相同,可以将集合 S_R 表示为布尔函数$\Delta_R(x_1,x_2,y_1,y_2):=(x_1\wedge x_2\wedge\neg y_2)\vee(x_1\wedge\neg x_2\wedge y_2)$.在此基础上,得到布尔函数 $\Delta'_R(x_1,x_2,y_1,y_2):=\Delta_R(x_1,x_2,y_1,y_2)\wedge\neg\Delta_{Cyc}(x_1,x_2)\wedge\neg\Delta_{Cyc}(y_1,y_2):=$false.

对于集合 $S_{old}:=\{(v_1,v_2)\in N_{def}\times N_{def}|L_1(v_1)\subseteq L_2(v_2)\}$,首先得到与例 2 相同的布尔函数:

$$\Delta_{old}(x_1,x_2,y_1,y_2):=(\neg x_1\wedge\neg x_2\wedge\neg y_1)\vee(\neg x_1\wedge x_2\wedge\neg y_1\wedge y_2)\vee(x_1\wedge\neg x_2\wedge y_1)\vee(x_1\wedge x_2\wedge y_1\wedge y_2).$$

在此基础上,得到布尔函数

$$\Delta'_{old}(x_1,x_2,y_1,y_2):=\Delta_{old}(x_1,x_2,y_1,y_2)\wedge\neg\Delta_{Cyc}(x_1,x_2)\wedge\neg\Delta_{Cyc}(y_1,y_2):=(\neg x_1\wedge\neg x_2\wedge\neg y_1)\vee(\neg x_1\wedge x_2\wedge\neg y_1\wedge y_2).$$

对于角色 R,依次构造出布尔函数 Δ'_{E_R} 和 Δ'_{A_R},如下所示:

$$\Delta'_{E\_R}(z_1,z_2,y_1,y_2):=\exists u_1\exists u_2(\Delta'_R(y_1,y_2,u_1,u_2)\wedge\Delta'_{old}(z_1,z_2,u_1,u_2)):=\text{false};$$

$$\Delta'_{A\_R}(x_1,x_2,y_1,y_2):=\forall z_1\forall z_2(\Delta'_R(x_1,x_2,z_1,z_2)\rightarrow\Delta'_{E\_R}(z_1,z_2,y_1,y_2)):=\text{true}.$$

接下来,计算出布尔函数 Δ'_T:

$$\Delta'_T(x_1,x_2,y_1,y_2):=\Delta'_{old}(x_1,x_2,y_1,y_2)\wedge\Delta'_{A\_R}(x_1,x_2,y_1,y_2):=(\neg x_1\wedge\neg x_2\wedge\neg y_1)\vee(\neg x_1\wedge x_2\wedge\neg y_1\wedge y_2).$$

由于 $\Delta'_T=\Delta'_{old}$,不再进行循环,算法将执行步骤(3).

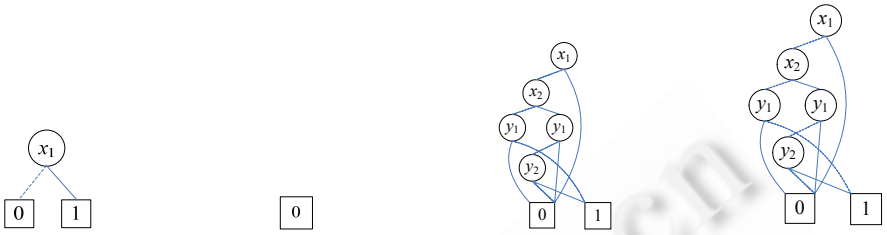
假设需要判断概念 C 在最小不动点语义下是否包含于概念 D.根据步骤(3),将集合$\{C\}$表示为布尔函数,得到$\Delta_C(x_1,x_2):=x_1\wedge\neg x_2$;由于$\Delta_C(x_1,x_2)\wedge\neg\Delta_{Cyc}(x_1,x_2)=(x_1\wedge\neg x_2)\wedge\neg x_1=$false,因此得到结论"$C\sqsubseteq_{lfp,T}D$".类似地,可以得到结论"$D\sqsubseteq_{lfp,T}C$".

假设需要判断最小不动点语义下概念 A 与概念 B 之间的包含关系.根据步骤(3),将集合$\{A\},\{B\},\{(A,B)\},\{(B,A)\}$分别表示为布尔函数后得到$\Delta_A(x_1,x_2):=\neg x_1\wedge\neg x_2$,$\Delta_B(x_1,x_2):=\neg x_1\wedge x_2$,$\Delta_{\{(A,B)\}}(x_1,x_2,y_1,y_2):=\neg x_1\wedge\neg x_2\wedge\neg y_1\wedge y_2$和$\Delta_{\{(B,A)\}}(x_1,x_2,y_1,y_2):=\neg x_1\wedge x_2\wedge\neg y_1\wedge\neg y_2$.

由于$\Delta_A(x_1,x_2)\wedge\neg\Delta_{Cyc}(x_1,x_2)=\neg x_1\wedge\neg x_2\neq$false,$\Delta_B(x_1,x_2)\wedge\neg\Delta_{Cyc}(x_1,x_2)=\neg x_1\wedge x_2\neq$false,$\Delta_{\{(A,B)\}}(x_1,x_2,y_1,y_2)\wedge\neg\Delta'_T(x_1,$

x_2,y_1,y_2)=false 和$\Delta_{\{(B,A)\}}(x_1,x_2,y_1,y_2)\wedge\neg\ \Delta'_T\ (x_1,x_2,y_1,y_2)=\neg x_1\wedge x_2\wedge\neg y_1\wedge\neg y_2\neq$false,因此可以得到结论“$B\sqsubseteq_{lfp,T}A$”和“$A\not\sqsubseteq_{lfp,T}B$”.

为了表述简洁,上面例子中,从布尔函数的层面对计算过程进行了考察.在实际计算时,对布尔函数的表示和操作都是借助 OBDD 来实现的,其中,与布尔函数Δ_{Cyc}, Δ'_R, Δ'_{old} 和 Δ'_T 对应的 OBDD 如图 4 所示.



(a) Δ_{Cyc} 对应的 OBDD (b) Δ'_R 对应的 OBDD (c) Δ'_{old} 对应的 OBDD (d) Δ'_T 对应的 OBDD

Fig.4 OBDDs corresponding to Δ_{Cyc}, Δ'_R, Δ'_{old} and Δ'_T

图 4 Δ_{Cyc}, Δ'_R, Δ'_{old} 和 Δ'_T 对应的 OBDD

4 算法分析

本节首先对算法的正确性进行论证;然后,对算法的时间复杂度进行考察;最后,通过开发推理机对算法的实际计算性能进行考察.

定理 4. 给定$\mathcal{EL}$ 的任一 TBox T 和 T 中任意两个被定义的概念 A,B,算法 1 返回“$A\sqsubseteq_{gfp,T}B$”当且仅当在最大不动点语义下 A 包含于 B.

证明:令 B_T 是算法 1 结束时布尔函数$\Delta_{\{(B,A)\}}(x_1,...,x_l,y_1,...,y_l)$所表示的集合,由定理 3 可得,$B_T$ 是 G_T 到 G_T 的最大模拟关系.在此基础上,

- 一方面,如果算法 1 返回“$A\sqsubseteq_{gfp,T}B$”,则有$\Delta_{\{(B,A)\}}(x_1,...,x_l,y_1,...,y_l)\wedge\neg\Delta_T(x_1,...,x_l,y_1,...,y_l)$=false,即$\{(B,A)\}\subseteq B_T$.由定理 1 可得,在最大不动点语义下 A 包含于 B;
- 另一方面,如果最大不动点语义下 A 包含于 B,则根据定理 1,存在模拟关系 Z 使得$(B,A)\in Z$;又由于 B_T 是最大模拟关系,因此有 $Z\subseteq B_T$,进而有$\{(B,A)\}\subseteq B_T$.因此,算法将返回“$A\sqsubseteq_{gfp,T}B$”. □

定理 5. 给定$\mathcal{EL}$ 的任一 TBox T 和 T 中任意两个被定义的概念 A,B,算法 2 返回“$A\sqsubseteq_{lfp,T}B$”当且仅当在最小不动点语义下 A 包含于 B.

证明:首先考察布尔函数$\Delta_{Scc}(x_1,...,x_l)$和$\Delta_{Cyc}(x_1,...,x_l)$的计算过程.对于任一结点 $v\in N_{def}$,根据对集合 $Front_T(v)$ 和 $Back_T(v)$的定义,$\{v\}\cup(Back_T(v)\cap Front_T(v))$(即$\Delta_{\{v\}}(x_1,...,x_l)\vee(\Delta_{Front(v)}(x_1,...,x_l)\wedge\Delta_{Back(v)}(x_1,...,x_l))$)恰好是由含有 v 的且不是孤立点的强连通分支中所有结点组成的集合.因此,对$\Delta_{Scc}(x_1,...,x_l)$的计算过程恰好求出了集合 $Scc=\{B|B\in N_{def}$并且 $B\xrightarrow{+}_T B\}$,而对$\Delta_{Cyc}(x_1,...,x_l)$的计算过程则恰好求出了集合 Cyc_T.在此基础上,与算法 1 相比,算法 2 在步骤②和步骤③中通过合取布尔函数$\neg\Delta_{Cyc}(x_1,...,x_l)\wedge\neg\Delta_{Cyc}(y_1,...,y_l)$删除了描述图中出现在 Cyc_T 中的所有结点以及与这些结点关联的所有边.基于定理 2 和定理 4,显然定理 5 成立. □

定理 6. 给定$\mathcal{EL}$ 的任一 TBox T,令其描述图为 $G_T=\langle N_{def},E_T,L_T\rangle$,则算法 1 的时间复杂度为 $O(|N_R|\times|N_{def}|^2)$.

证明:首先,集合 $S_{old}=\{(v_1,v_2)\in N_{def}\times N_{def}|L_T(v_1)\subseteq L_T(v_2)\}$的元素个数最多为$|N_{def}|^2$,因此,算法 1 中步骤④与步骤⑥之间的循环最多为$|N_{def}|^2$ 次;其次,每次执行步骤④时,需要为每个角色 R_i 都计算一次布尔函数 $\Delta_{E_R_i}(z_1,...,z_l,y_1,...,y_l)$和 $\Delta_{A\_R_i}(x_1,...,x_l,y_1,...,y_l)$;最后,对于布尔函数的析取、合取、量化等操作,可以借助 OBDD 在恒定时间内完成.因此,总的来说,算法 1 的时间复杂度为 $O(|N_R|\times|N_{def}|^2)$. □

定理 7. 给定$\mathcal{EL}$ 的任一 TBox T,令其描述图为 $G_T=\langle N_{def},E_T,L_T\rangle$,则算法 2 的时间复杂度为 $O(|N_R|\times|N_{def}|^2)$.

证明:与算法 1 相比,算法 2 增加的时间开销主要体现在对布尔函数$\Delta_{Cyc}(x_1,...,x_l)$的计算上,而计算$\Delta_{Cyc}(x_1,...,$

x_l)的时间又主要花费在对布尔函数$\Delta_{Scc}(x_1,...,x_l)$的计算上.下面来考察$\Delta_{Scc}(x_1,...,x_l)$的计算过程.首先,步骤③与步骤⑤之间的循环最多为$|N_{def}|$次;其次,每次执行步骤④计算$\Delta_{Front(v)}(x_1,...,x_l)$(或$\Delta_{Back(v)}(x_1,...,x_l)$)时,需要调用布尔函数$\Delta_{Front(v)}(x_1,...,x_l)$(或$\Delta_{Back(v)}(x_1,...,x_l)$)的计算过程,在该计算过程中,最多又需要进行$|N_{def}|$次循环.因此总的来说,计算$\Delta_{Scc}(x_1,...,x_l)$的时间复杂度为 $O(|N_{def}|^2)$.因此,计算$\Delta_{Cyc}(x_1,...,x_l)$的时间复杂度也为 $O(|N_{def}|^2)$.再综合定理 6 可得,算法 2 的时间复杂度为 $O(|N_R|\times|N_{def}|^2)$. □

上述关于时间复杂度的结论与 Baader 等人[6]给出的结论相一致.

为了从实际计算性能的角度对基于 OBDD 的推理算法进行考察,我们采用 Java 语言并且调用 JavaBDD (http://javabdd.sourceforge.net/)类库中关于 OBDD 的操作函数实现了本文给出的两种算法,开发了基于 OBDD 的εL 循环术语集推理机.推理机以 OWL 语言表示的εL 知识库作为输入,计算出被定义的概念之间存在的所有包含关系,同时给出整个过程所花费的时间.

由于找不到既含有循环定义又不超过描述逻辑εL 刻画能力的本体测试用例,我们以本文例 2 中的循环术语集作为基本测试用例.例 2 的循环术语集只含有 8 个概念名和 1 个角色.为了增大本体规模,我们为该循环术语集不断生成新的副本,将副本中的概念和角色重新命名(例如,对于概念名 A,在各个副本中依次命名为 A_1, A_2,…等),然后将这些副本和原来的术语集一起构成新的本体.我们将例 2 中循环术语集的规模看作 1 个单位,每加入 1 个副本都增加 1 个单位.在此基础上,实验结果如图 5 所示,其中的横坐标为本体规模,当本体规模为 10 时,表示该本体由例 2 的循环术语集及其 9 个副本组成,从而具有 80 个概念名和 10 个角色,依此类推.纵坐标为推理结束所需要的时间.为了实验数据的准确性,我们针对每个本体都采用运行 5 次后求平均值的方式得到运行时间.图中标识为“EL_gfp”的曲线是最大不动点语义下(即算法 1)的实验结果,标识为“EL_lfp”的曲线是最小不动点语义下(即算法 2)的实验结果.实验平台为 Thinkpad X200 PC 机、Windows7 64bit 操作系统、Intel Core 2 Duo P8600 CPU、4GB 内存、JDK 版本为 1.6.

为了得到关于算法性能的直观评价,我们还在同样的配置环境下用著名的描述逻辑推理机 Pellet 运行了各个测试用例,得到图中标识为“Pellet”的实验数据.

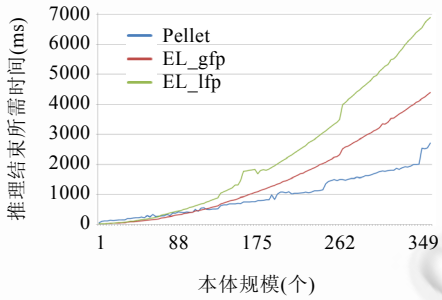


Fig.5 Experimental comparison with Pellet

图 5 与 Pellet 的实验比较

需要强调的是:Pellet 推理机并不支持最大不动点语义下和最小不动点语义下对循环术语集的推理;将测试用例作为输入之后,Pellet 推理机给出的仅仅是描述语义下的推理结果.因此,从时间性能的角度将本文的算法与 Pellet 推理机作比较是不合适的.但通过该实验至少可以看出,本文的算法在时间性能上是可以接受的.甚至于当本体规模小于 115 个单位(即 920 个概念名和 115 个角色)时,本文最大不动点语义下的推理算法所需要的时间低于目前性能最好的 Pellet 推理机.此外,本文基于 OBDD 的算法在时间性能上还有较大的提升空间:一方面,目前我们是先用显式的集合的方式计算和刻画出描述图 G_T,然后在算法过程中才部分地借助 OBDD 来表示和实现.如果一开始就借助 OBDD 对描述图进行求解和表示,则算法性能可以得到更大的提高;另一方面,我们在实现算法时使用了 JavaBDD 类库中关于 OBDD 的现有操作和函数,这些操作和函数的计算性能并不是最优的,从而也在一定程度上影响了本文的实验结果.

5 相关工作比较

蒋运承等人[7]将 Baader[6]的工作扩展到描述逻辑εLN,将描述图和模拟关系引入到描述逻辑εLN 循环术语集中;在不动点语义下,蒋运承等人研究了概念可满足性问题和概念包含问题,给出了基于描述图和模拟关系的判定算法.与该工作相比,本文采用 OBDD 研究循环术语集的推理问题.为了能够使用 OBDD,在 Baader 和蒋运承等人工作的基础上,本文首先给出一条关于描述图上最大模拟关系的性质,借助集合表示和集合运算对该性质进行了表述和证明.在此基础上,本文应用布尔函数对描述图进行编码,进而基于 OBDD 给出了不动点语义下对概念包含关系进行判定的算法.本文的工作主要是针对 Baader[6]提出的描述逻辑εL 循环术语集来进行的,但实际上,在蒋运承等人工作的基础上,可以比较容易地将本文的算法扩展到处理描述逻辑εLN 循环术语集.

Pan 等人[14]将 BDD 应用于模态逻辑 K 的可满足性判定.该方法可以比较容易地扩展到处理含有多个模态词的模态逻辑 $K_{(m)}$.由于 $K_{(m)}$与描述逻辑 ALC 之间存在一一对应关系,因此,Pan 等人[14]的工作实际上给出了一种对描述逻辑 ALC 的概念可满足性进行判定的算法.之后,Keller[15]首次明确提出了将 OBDD 应用于描述逻辑推理的思路:首先,将描述逻辑知识库转化为与之可满足性等价的命题逻辑公式;然后,再借助 OBDD 对命题公式的可满足性进行判定.然而,Keller 并未给出具体的算法.Rudolph 等人[16,17]首次给出了将 OBDD 应用于描述逻辑推理的算法.针对描述逻辑 SHIQ 刻画的知识,Rudolph 等人首先构造出一种称为多米诺集的结构,借助多米诺集,Rudolph 等人将描述逻辑知识库转化为布尔函数表示形式,进而给出了基于 OBDD 的推理算法.Rudolph 等人从理论上证明了算法的正确性,并对算法的复杂度进行了分析.上述 3 项工作为描述逻辑推理提供了一种全新的思路和途径,但与描述逻辑领域的大部分研究工作相似,他们都假设了所考察的知识库中不含有循环定义.因此与这些工作相比,本文的特点是首次将 OBDD 应用于描述逻辑循环术语集的推理.

6 结束语

在文献中借助描述图和模拟关系解决循环术语集推理问题的基础上,本文借助集合表示和集合运算给出描述图上关于最大模拟关系的一个重要性质,基于该性质给出了应用 OBDD 求解最大模拟关系的方法,进而依次给出了最大不动点语义下和最小不动点语义下借助 OBDD 对概念包含关系进行判定的算法.据我们所知,本文是首次将 OBDD 应用于描述逻辑循环术语集推理的文献.本文的工作一方面为循环术语集的推理提供了一条有效的途径,另一方面为 OBDD 与描述逻辑推理以及逻辑定理证明的结合提供了新的思路和案例.

下一步的工作是继续优化本文的算法.同时,对这些算法进行扩展,借助 OBDD 对表达能力更强的描述逻辑的循环术语集进行推理.

References:

[1] 史忠植,董明楷,蒋运承,张海俊.语义 Web 的逻辑基础.中国科学(E 辑:信息科学),2004,34(10):1123−1138. [doi: 10.3969/j.issn.1674-7259.2004.10.004]

[2] Baader F. Using automata theory for characterizing the semantics of terminological cycles. Annals of Mathematics and Artificial Intelligence, 1996,18(2-4):175−219. [doi: 10.1007/BF02127747]

[3] Kusters R. Characterizing the semantics of terminological cycles in ALN using finite automata. In: Cohn AG, Schubert L, Shapiro SC, eds. Proc. of the 6th Int'l Conf. on Principles of Knowledge Representation and Researching. San Francisco: Morgan Kaufmann Publishers, 1998. 499−511.

[4] Nebel B. Terminological cycles: Semantics and computational properties. In: Sowa JF, ed. Proc. of the Principles of Semantic Networks. San Francisco: Morgan Kaufmann Publishers, 1991. 331−362.

[5] Nebel B. Reasoning and Revision in Hybrid Representation Systems. New York: Spring-Verlag, 1990.

[6] Baader F. Terminological cycles in a description logic with existential restrictions. In: Gottlob G, Walsh T, eds. Proc. of the 18th Int'l Joint Conf. on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 2003. 325−330.

[7] Jiang YC, Wang J, Shi ZZ, Tang Y. Fixpoint semantics and reasoning terminological cycles in description logic $\mathcal{ELN}$. Ruan Jian Xue Bao/Journal of Software, 2009,20(3):477−490 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3215.htm [doi: 10.3724/SP.J.1001.2009.03215]

[8] 王驹,蒋运承,申宇铭.描述逻辑系统 vL 循环术语集的可满足性及推理机制.中国科学(F 辑:信息科学),2009,39(2):205−211. [doi: 10.1007/s11432-008-0101-6]

[9] Gu TL, Xu ZB. Ordered Binary Decision Diagram and Its Applications. Beijing: Science Press, 2009 (in Chinese).

[10] Yan H, Zhang W, Zhao HY, Mei H. BDD-Based approach to the verification of feature models. Ruan Jian Xue Bao/Journal of Software, 2010,21(1):84−97 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3525.htm [doi: 10.3724/SP.J.1001.2010.03525]

[11] Uribe TE, Stickel ME. Ordered binary decision diagrams and the Davis-Putnam procedure. In: Jouannaud JP, ed. Proc. of the 1st Int'l Conf. on Constraints in Computational Logics. LNCS 845, Berlin: Springer-Verlag, 1994. 34−49. [doi: 10.1007/BFb0016843]

[12] Groote JF, Tveretina O. Binary decision diagrams for first-order predicate logic. Journal of Logic and Algebraic Programming, 2003,57(1-2):1−22. [doi: 10.1016/S1567-8326(03)00039-0]

[13] Marrero W. Using BDDs to decide CTL. In: Halbwachs N, Zuck LD, eds. Proc. of the 11th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. LNCS 3440, Berlin: Springer-Verlag, 2005. 222−236. [doi: 10.1007/978-3-540-31980-1_15]

[14] Pan GQ, Sattler U, Vardi MY. BDD-Based decision procedures for the modal logic K. Journal of Applied Non-Classical Logics, 2006,16(1-2):169−208. [doi: 10.3166/jancl.16.169-207]

[15] Keller U. Towards novel techniques for reasoning in expressive description logics based on binary decision diagrams. In: Simperl EB, Diederich J, eds. Proc. of the 4th Annual European Semantic Web Conf. 2007. 435−450.

[16] Rudolph S, Krötzsch M, Hitzler P. Terminological reasoning in SHIQ with ordered binary decision diagrams. In: Fox D, Gomes CP, eds. Proc. of the 23rd National Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 2008. 529−534.

[17] Rudolph S, Krötzsch M, Hitzler P. Type-Elimination-Based reasoning for the description logic SHIQb_s using decision diagrams and disjunctive datalog. Logical Methods in Computer Science, 2012,8(1):1−38. [doi: 10.2168/LMCS-8(1:12)2012]

附中文参考文献:

[7] 蒋运承,王驹,史忠植,汤庸.描述逻辑$\mathcal{ELN}$ 循环术语集的不动点语义及推理.软件学报,2009,20(3):477−490. http://www.jos.org.cn/1000-9825/3215.htm [doi: 10.3724/SP.J.1001.2009.03215]

[9] 古天龙,徐周波.有序二叉决策图及应用.北京:科学出版社,2009.

[10] 闫华,张伟,赵海燕,梅宏.基于二分决策图的特征模型验证方法.软件学报,2010,21(1):84−97. http://www.jos.org.cn/1000-9825/3525.htm [doi: 10.3724/SP.J.1001.2010.03525]

古天龙(1964−),男,山西芮城人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为形式化方法,符号计算.

E-mail: cctlgu@guet.edu.cn

常亮(1980−),男,博士,教授,CCF 高级会员,主要研究领域为知识表示与推理,智能规划,形式化方法.

E-mail: changl@guet.edu.cn

吕思菁(1983−),男,硕士,主要研究领域为描述逻辑,符号计算.

E-mail: darrenlv@msn.com

徐周波(1976−),女,博士,副教授,CCF 会员,主要研究领域为符号计算,智能规划,约束满足问题求解.

E-mail: xzbli_11@guet.edu.cn