ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.11, November 2010, pp.2826−2837
doi: 10.3724/SP.J.1001.2010.03690

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

可能性扩展规则的推理和知识编译*

殷明浩[1,2,3+], 孙吉贵[2,3], 林 海[2,3], 吴 瑕[2,3]

[1](东北师范大学 计算机学院,吉林 长春 130117)
[2](吉林大学 计算机科学与技术学院,吉林 长春 130012)
[3](吉林大学 符号计算与知识工程教育部重点实验室,吉林 长春 130012)

Possibilistic Extension Rules for Reasoning and Knowledge Compilation

YIN Ming-Hao[1,2,3+], SUN Ji-Gui[2,3], LIN Hai[2,3], WU Xia[2,3]

[1](School of Computer Science, Northeast Normal University, Changchun 130117, China)
[2](School of Computer Science and Technology, Jilin University, Changchun 130012, China)
[3](Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Changchun 130012, China)
+ Corresponding author: E-mail: mhyin@nenu.edu.cn

Yin MH, Sun JG, Lin H, Wu X. Possibilistic extension rules for reasoning and knowledge compilation. *Journal of Software*, 2010,21(11):2826−2837. http://www.jos.org.cn/1000-9825/3690.htm

Abstract: First, the possibilistic extension, a possible manifestation of the extention rule, is proposed. A reasoning mechanic in possibilistic logic, using the possibilistic extension rule, is then introduced, using the concept of the complementary factor to estimate its complexity. The definitions of entailment tractable and satisfiability tractable are extended to introduce the concepts of possibilistic entailment tractable and consistent-computing tractable. Based on possibilistic extension rules, EPPCCCL (each pair of possibilistic clauses contains complementary literals) theory is introduced, which is proved to be possibilistic entailment tractable and consistent-computing tractable; therefore, the theory can be regarded as a target language of possibilistic knowledge compilation.
Key words: extension rule; possibilistic logic; knowledge compilation; EPPCCCL (each pair of possibilistic clauses contains complementary literals) theory

摘 要: 在扩展规则的基础上提出了可能性扩展规则.给出了基于可能性扩展规则的可能性逻辑推理方法,利用互补因子的概念来估价推理问题的复杂度.扩展了经典逻辑的蕴含可控制类和可满足可控制类的定义,提出了可能性蕴含可控制类、不一致性程度计算可控制类的概念.在可能性扩展规则的基础上提出了 EPPCCCL(each pair of possibilistic clauses contains complementary literals)理论,并证明了该理论是在最优化形式蕴含可控制类和不一致性程度计算可控制类中的,可以作为可能性知识编译的目标语言.
关键词: 扩展规则;可能性逻辑;知识编译;EPPCCCL(each pair of possibilistic clauses contains complementary literals)理论
中图法分类号: TP181 **文献标识码**: A

---

* Supported by the National Natural Science Foundation of China under Grant Nos.60573067, 60803102 (国家自然科学基金)
Received 2008-07-18; Revised 2009-03-30; Accepted 2009-07-06

可能性逻辑是适用于给定知识中含有不确定因素或者给定知识不完全一致的条件下进行知识表示和推理的逻辑,在信念修正、非单调推理、不确定规划、诊断等众多领域都有着广泛的应用[1].在可能性逻辑中,最主要和最基本的推理规则是基于 Dubois 和 Prade 提出的可能性归结原理[2].

在文献[3]中,我们提出一种新的定理证明方法——扩展规则方法.该方法的主要思想是,沿着归结的逆向做定理证明并用容斥原理解决由此带来的空间复杂性问题.在此基础上,我们引入化简规则和启发式函数,提高了扩展规则的求解效率;完善了一阶逻辑扩展规则的推理方法,设计了基于扩展规则的一阶定理证明器;给出了基于扩展规则的模态逻辑定理证明方法,设计了基于模态逻辑的推理系统.证明了扩展规则不仅可以用于高效求解 SAT 问题,而且对于模型计数问题同样具有独特的优势[4−7].

知识编译是自动定理证明领域的一个重要论题,它使得自动推理得以在多项式时间内实现.Selman 通过找到和原子句集等价的 Horn 子句集,使得在此 Horn 子句集上的推理能够在线性时间内完成[8].但并非所有的子句集合都存在等价的 Horn 子句集,因此,该方法通常是不精确的.Val 提出了一种精确的知识编译方法,虽然一般情况下单元归结是不完备的,但是通过运行一个完备的求本原蕴含式的算法过程加入所有的合并归结来扩大原子句集的规模后,可以保证编译后的子句集上单元归结完备[9].文献[10]提出了一种基于理论本原蕴含式的知识编译方法.文献[11]提出了一种对偶的基于本原蕴含的方法.Darwiche 将子句形式转换为可分解否定范式,该方法是目前应用最为广泛的知识编译方法,并给出近年来知识编译的发展图谱[12].但是,上述方法大多是基于本原蕴含式的.在扩展规则的基础上中,我们提出了一种基于扩展规则的知识编译方法[13].与现有的方法都是基于归结原理不同,该方法在知识编译阶段和询问阶段都是基于扩展规则的,与现有知识编译方法具有对偶关系.在可能性逻辑方面,可能性知识编译的研究工作刚刚展开,据我们所知,目前只有 Prade 等人在文献[14]中提出的方法从本质上讲是基于可能性归结原理的.

本文的目的是将扩展规则推广到可能性逻辑中,主要工作如下:

(1) 提出了可能性扩展规则,该规则是经典逻辑中扩展规则的泛化形式.基于可能性扩展规则,提出可能性逻辑中一种新的推理框架,证明了该方法的有效性和完备性.该方法可以看作是可能性归结方法的补方法.

(2) 利用可能性扩展规则进行推理一般需要指数级别的时间.提出了基于可能性扩展规则的知识编译方法,该方法可以将任意可能性子句集转换为等价的 EPPCCCL(each pair of possibilistic clauses contains complementary literals)理论.利用可能性扩展规则,EPPCCCL 理论可以在多项式时间内完成一般的可能性推理问题.

1 可能性逻辑

本文只关注可能性逻辑的标准版本,即标准可能性逻辑(standard possibilistic logic,简称 SPL),有关可能性逻辑的详细介绍见文献[1].首先对本节使用的符号进行约定:用 $C,C_1,C_2,\ldots$表示经典逻辑中的单个子句,有时也把子句理解成文字集合的形式,用 $l,l_1,l_2,\ldots$表示文字,用$(C,\alpha),(C,\beta),\ldots$表示可能性子句.用$\Sigma,\Sigma_1,\Sigma_2,\ldots$表示可能性知识库,通常 M 表示一个子句集Σ中所有原子的集合.

定义 1. 设φ为经典逻辑公式,$\alpha\in(0,1)$,则二元组(φ,α)为可能性公式.其中,α表示必然性测度 N 的下界,即$N(\varphi)\geqslant\alpha$.我们称$\alpha$为该公式的值,记作 $val(\varphi)$.

定义 2. 设(C,α)为可能性公式,其中,C 为经典逻辑中析取范式的形式,则称(C,α)为可能性子句.

在可能性逻辑中,可能性归结是最基本也是最重要的推理规则,例如$(p\vee q,\alpha)\wedge(\neg p\vee r,\beta)\vdash(q\vee r,\min(\alpha,\beta))$.

定义 3. 设$\Sigma=\{(\varphi_1,\alpha_1),(\varphi_2,\alpha_2),\ldots,(\varphi_n,\alpha_n)\}$是一个可能性知识库,则$\Sigma$在经典逻辑中的投影$\Sigma^*$为忽略$\Sigma$中所有可能性公式的值得到的经典公式集合,即$\Sigma^*=\{\varphi_1,\varphi_2,\ldots,\varphi_n\}$.

定义 4. 设$\Sigma=\{(\varphi_1,\alpha_1),(\varphi_2,\alpha_2),\ldots,(\varphi_n,\alpha_n)\}$是一个可能性知识库,则$\Sigma$的$\alpha$-截集$\Sigma_\alpha$和严格$\alpha$-截集 $\Sigma_{\bar{\alpha}}$ 分别定义如下:

$$\Sigma_\alpha=\{(\varphi,\beta)\in\Sigma\},\ \beta\geqslant\alpha,$$

$$\Sigma_{\bar{\alpha}}=\{(\varphi,\beta)\in\Sigma\},\ \beta>\alpha.$$

对应的经典投影分别定义如下:

$$\Sigma_{\alpha}^{*}=\{\varphi\mid(\varphi,\beta)\in\Sigma\},\ \beta\geqslant\alpha,$$

$$\Sigma_{\bar{\alpha}}^{*}=\{\varphi\mid(\varphi,\beta)\in\Sigma\},\ \beta>\alpha.$$

定义 5. 设$\Sigma=\{(\varphi_1,\alpha_1),(\varphi_2,\alpha_2),\ldots,(\varphi_n,\alpha_n)\}$是一个可能性知识库,则$\Sigma$的不一致程度定义如下:

$$\_Inc(\Sigma)=\max\{\alpha\mid \Sigma_{\alpha}^{*}\vDash\bot\}.$$

从定义 4 可以看出,$\_Inc(\Sigma)=0$ 当且仅当Σ在经典逻辑中的投影Σ^*可满足.从这个意义上讲,计算可能性知识库的不一致程度是 SAT 问题的泛化.经典逻辑中的蕴含关系也可以被扩展到可能性逻辑中,要证明可能性公式(φ,α)为可能性知识库Σ的逻辑结果,只需要在Σ中添加新的可能性公式$(\neg\varphi,1)$,并证明$\Sigma\cup(\neg\varphi,1)\vDash(\bot,\alpha)$.与经典逻辑不同,可能性逻辑中需要考虑两类演绎问题:决策形式演绎问题和最优化形式演绎问题.给定一个可能性知识库Σ,前者需要判断一个可能性公式(φ,α)是否为Σ的逻辑结果;后者需要判断φ在多大程度上为Σ的逻辑结果.

定义 6. 设Σ为可能性知识库,φ为经典逻辑公式,决策形式标准演绎问题(SPL-DED)是指判断(φ,α)是否为Σ的逻辑结果的问题.

定义 7. 给定可能性知识库Σ,经典逻辑公式φ,公式φ作为Σ的逻辑结果的必然性程度的最优下确界为$\max\{\alpha|\Sigma\vDash(\varphi,\alpha)\}$.最优化形式标准演绎问题(SPL-OPT)是指判断φ在多大程度上为Σ的逻辑结果的问题,即计算$\max\{\alpha|\Sigma\vDash(\varphi,\alpha)\}$的值的问题.

需要注意的是,若$_Inc(\Sigma)\geqslant\alpha$,则$_Inc(\Sigma\cup(\neg\varphi,1))\geqslant\alpha$必然成立,$\Sigma\vDash(\varphi,\alpha)$也必然成立.因此,在可能性逻辑中通常只考虑决策形式非平凡演绎问题和最优化形式非平凡演绎问题,即要求$\alpha\geqslant_Inc(\Sigma)$.

2 可能性扩展规则

本节介绍可能性扩展规则.从本节起,如无特殊声明,可能性知识库Σ中的所有公式都是以可能性子句形式出现的.首先回顾扩展规则:

定义 8. 给定一个子句 C 和一个原子的集合 M,设 $D=\{C\vee a,C\vee\neg a\}$,其中,$a\in M$ 并且 a 和$\neg a$ 都不在 C 中出现,将从 C 到 D 中元素的推导过程称为扩展规则,D 中的元素称为应用扩展规则的结果.

在定义 8 的基础上,引入可能性包含规则和可能性扩展规则:

定义 9. 给定两个可能性子句集合$\Sigma=\{(C,\alpha),(C,\beta)\}$,其中,$\alpha\geqslant\beta$,$\Sigma'=\{(C,\alpha)\}$.我们将$\Sigma$到$\Sigma'$的推导过程称为可能性包含规则,记作(SPL-S)规则,Σ'中元素为Σ应用可能性包含规则的结果.

容易证明Σ和Σ'是逻辑等价的,因为$(C,\alpha)\wedge(C,\beta)\vDash(C,\alpha)$;$(C,\alpha)\vDash(C,\beta)$.

定义 10. 给定任意一个可能性子句(C,α)和一个原子集合 M,$D=\{(C\vee b,\alpha),(C\vee\neg b,\alpha)\}$,其中,$b\in M$ 且 b 和$\neg b$ 都不在 C 中出现.我们将从(C,α)到 D 中元素的推导过程称为可能性扩展规则,记作(SPL-ER)规则,D 中的元素为(C,α)应用可能性扩展规则的结果.

定理 1. 给定任意可能性子句(C,α),则(C,α)和它的可能性扩展规则结果逻辑等价.

证明:设 $D=\{(C\vee b,\alpha),(C\vee\neg b,\alpha)\}$是$(C,\alpha)$应用可能性扩展规则的结果.利用可能性归结原理,从 D 出发通过一步归结可以得到(C,α),因而 $D\vDash(C,\alpha)$.

另一方面,$N(C\vee b)\geqslant\max(N(C),N(b))=\max(\alpha,N(b))\geqslant\alpha$,其中,$N$ 为满足(C,α)的可能性分布诱导的必然性测度.因此,$(C,\alpha)\vDash(C\vee b,\alpha)$;类似地,我们可以证明$(C,\alpha)\vDash(C\vee\neg b,\alpha)$.因此,$(C,\alpha)\vDash D$.

由上可知,(C,α)和它的可能性扩展规则结果 D 等价. □

根据定理 1,若Σ为可能性知识库,则Σ和它经过有限次应用规则(SPL-ER)和规则(SPL-S)后的得到的新子句集合Σ'等价.因此,可能性扩展规则和可能性包含规则可以被看作是推理规则.因此,下述性质也成立:

(1) 对于Σ'中任意子句(C,α),$\Sigma\vDash(C,\alpha)$;

(2) Σ和Σ'的不一致性程度相同,即$_Inc(\Sigma)=_Inc(\Sigma')$.

3 基于可能性扩展规则的可能性推理

本节介绍如何利用可能性扩展规则在可能性逻辑中进行推理,即如何利用可能性扩展规则计算可能性知识库Σ的不一致程度.

注意下述事实成立:若可能性知识库Σ中的每个子句都只含有一个原子 A,则$_Inc(\Sigma)>0$ 当且仅当Σ是$\{(A,\alpha),(\neg A,\beta)\}$的形式.这时,$_Inc(\Sigma)=\min(\alpha,\beta)$;若$\Sigma$中的子句都含有两个相同的原子,那么$_Inc(\Sigma)>0$ 当且仅当它是$\{(A\vee B,\alpha_1),(\neg A\vee\neg B,\alpha_2),(A\vee\neg B,\alpha_3),(\neg A\vee B,\alpha_4)\}$.这时,$_Inc(\Sigma)=\min\{\alpha_i,i=1,2,3,4\}$.类似地,如果其中的每个可能性子句都含有 m 个相同的原子,则$_Inc(\Sigma)>0$ 当且仅当Σ的经典投影Σ^*含有 2^m 个经典子句.这时,$_Inc(\Sigma)$为这些子句的值的下确界.利用可能性扩展规则计算$_Inc(\Sigma)$的核心思想就是,尽量将Σ扩展成上述形式,然后再计算Σ的不一致程度.

为此,首先定义可能性极大项:

定义 11. 一个非重言式可能性子句(C,α)是原子集合 M 上的可能性极大项,当且仅当 C 中包含集合 M 上的所有原子或其否定.

定理 2. 设Σ为可能性知识库,M 为Σ中出现的所有原子的集合($|M|=m$).若Σ中的每个可能性子句都是可能性极大项,则:

(1) $\_Inc(\Sigma)>0$ 当且仅当Σ的经典投影Σ^*中含有 2^m 个互不相同的子句.

(2) 若$_Inc(\Sigma)>0$,设Σ'为Σ上不断应用(SPL-S)规则的结果,且Σ'上不能应用规则(SPL-S),则Σ的不一致性程度为

$$\_Inc(\Sigma)=\min\{\alpha_i|(C_i,\alpha_i)\in\Sigma'\}.$$

证明:(1) 若Σ的经典投影Σ^*中含有的子句数目小于 2^m,根据扩展规则定理证明的完备性,Σ^*可满足,故$_Inc(\Sigma)=0$.因此,$\_Inc(\Sigma)>0$ 当且仅当Σ^*含有 2^m 个不同子句.

(2) 根据文献[1],$\_Inc(\Sigma)=\max\limits_{\Sigma'\subseteq\Sigma,INC(\Sigma')>0}\min\{\alpha\,|\,(\varphi,\alpha)\in\Sigma'\}$.由于$\Sigma'$上不能应用(SPL-S)规则,且$_Inc(\Sigma)=_Inc(\Sigma')>0$,由定理 2(1)可知,$\Sigma'$中含有 2^m 个可能性子句,去掉Σ'中的任意子句都会导致其经典投影的个数小于 2^m,从而导致得到的新子句集合不一致性程度为 0.因此,$\_Inc(\Sigma)=\_Inc(\Sigma')=\min\{\alpha_i|(c_i,\alpha_i)\in\Sigma'\}$. □

因此,给定一个可能性知识库Σ,要计算Σ的不一致程度$_Inc(\Sigma)$,只需要不断应用(SPL-ER)规则和(SPL-S)规则将之扩展为与之等价的可能性极大项集合.若扩展出的子句集合里面有少于 2^m 个互不相同的子句,则$_Inc(\Sigma)=0$;反之,则根据定理 2 计算其不一致性程度.将上述过程称为基于可能性扩展规则的推理过程.

定理 3. 在可能性逻辑中,基于可能性扩展规则的推理过程是正确的和完备的.

该命题显然成立,因为在推理过程中使用的(SPL-ER)规则和(SPL-S)规则的结果都和原子句集等价.事实上,如果忽略可能性子句的值,定理 2 即转换为“若每个子句含有所有变量,则问题不可满足”,这就是文献[1]中提出的“canonical implicates”.

例 1:给定子句集合$\Sigma=\{C_1:(p\vee q,1),C_2:(\neg p\vee q,0.8),C_3:(\neg r\vee\neg q,0.4),C_4:(r,0.6),C_5:(\neg r,0.3)\}$,计算$_Inc(\Sigma)$.

首先给出一个可能性归结过程:

(1) $((p\vee q,1),(\neg p\vee q,0.8))\vDash(q,0.8)$.

(2) $((\neg r\vee\neg q,0.4),(q,0.8))\vDash(\neg r,0.4)$.

(3) $((r,0.6),(\neg r,0.4))\vDash(\bot,0.4)$.

通过检查,没有其他可能性归结能够得到更高的不一致值,因此,$\_Inc(\Sigma)=0.4$.

下面给出基于可能性扩展规则的推理过程:

首先通过应用(SPL-ER)规则给出Σ中各个子句扩展后的极大项集合:

(1) 扩展 C_1 得到的结果 $D_1:(p\vee q\vee r,1),(p\vee q\vee\neg r,1)$.

(2) 扩展 C_2 得到的结果 $D_2:(\neg p\vee q\vee r,0.8),(\neg p\vee q\vee\neg r,0.8)$.

(3) 扩展 C_3 得到的结果 $D_3:(p\vee\neg r\vee\neg q,0.4),(\neg p\vee\neg r\vee\neg q,0.4)$.

(4) 扩展 C_4 得到的结果 D_4:$(p\vee q\vee r,0.6),(\neg p\vee q\vee r,0.6),(p\vee\neg q\vee r,0.6),(\neg p\vee\neg q\vee r,0.6)$.

(5) 扩展 C_5 得到的结果 D_5:$(p\vee q\vee\neg r,0.3),(\neg p\vee q\vee\neg r,0.3),(p\vee\neg q\vee\neg r,0.3),(\neg p\vee\neg q\vee\neg r,0.3)$.

再使用(SPL-S)规则分别对 D_1,D_2,D_3,D_4,D_5 进行约简:

(1) D_1':$(p\vee q\vee r,1),(p\vee q\vee\neg r,1)$.

(2) D_2':$(\neg p\vee q\vee r,0.8),(\neg p\vee q\vee\neg r,0.8)$.

(3) D_3':$(p\vee\neg r\vee\neg q,0.4),(\neg p\vee\neg r\vee\neg q,0.4)$.

(4) D_4':$(p\vee\neg q\vee r,0.6),(\neg p\vee\neg q\vee r,0.6)$.

(5) D_5':$\varnothing$

由于Σ可以扩展出 $2^3=8$ 个极大项,因而根据定理 2,$\_Inc(\Sigma)=0.4$.

但是,直接采用上述的过程逐个扩展出所有极大项集合,并由此计算某个可能性子句集合的不一致性程度通常是不可行的,因为由此带来的时间代价和空间代价往往都是指数级别.事实上,不需要像例 1 那样列出所有可能的极大项,而只需计算出可能性极大项的个数,我们也可以计算可能性子句集合的一致性程度.因为只要知道扩展出的集合含有 2^m 个元素,我们就可以知道该集合为不一致的.事实上,通过利用著名的容斥原理就可以比较容易地计算出可能性知识库的不一致性程度.

定理 4(容斥原理). 集合 $A_1,A_2,\dots,A_n$ 并集的元素个数可以用如下公式计算:

$$|A_1\cup A_2\cup\dots\cup A_n|=\sum_{i=1}^{n}|A_i|-\sum_{1\leqslant i\leqslant j\leqslant n}^{n}|A_i\cap A_j|+\dots+(-1)^{n+1}|A_1\cap A_2\cap\dots\cap A_n|.$$

包含排斥原理的简单形式为大家所熟知:

$$|A\cup B|=|A|+|B|-|A\cap B|,$$

$$|A\cup B\cup C|=|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|+|A\cap B\cap C|.$$

为了使容斥原理可以用于计算极大项的个数,需要先引入下述定理:

定理 5. 给定任意两个可能性子句(C_1,α)和(C_2,β),D_1 和 D_2 分别为(C_1,α)和(C_2,β)扩展出的可能性极大项集合,对于任意可能性子句$(C_1',\alpha)\in D_1,(C_2',\beta)\in D_2$,我们无法使用(SPL-S)规则对$(C_1',\alpha)$和$(C_2',\beta)$进行约简当且仅当 C_1 和 C_2 中含有互补对.

证明:充分性.假设 C_1 和 C_2 中含互补对(不失一般性,设为 A 和$\neg A$),则在含 A 的子句扩展出来的极大项中,A 一定是以正的方式出现;在含$\neg A$ 的子句扩展出来的极大项中,A 一定是以负的方式出现.因此,对于 D_1 中的任意子句和 D_2 中的任意子句,其经典投影都不会相同.根据定义 10,我们必然无法对其中的任意一对子句使用(SPL-S)规则化简.

必要性.用反证法.假如两个子句的经典投影不含互补对,设(C_1,α)中出现的文字集合 $L=\{l_1,\dots,l_r\}$,出现在(C_2,β)中的文字集合 $L'=\{l_1',\dots,l_s'\}$,设 $L''=L\cup L'=\{l_1'',\dots,l_t''\}$.显然,由$(C_1,\alpha)$可以扩展出$(l_1''\vee\dots\vee l_t'',\alpha)$,由$(C_2,\beta)$可以扩展出$(l_1''\vee\dots\vee l_t'',\beta)$,我们可以在这两个新子句中运用规则(SPL-S)来删除一个子句.所以,若两个子句扩展出的极大项的集合无法使用(SPL-S)规则,则它们一定含有互补对. □

例 2:考虑子句$(p\vee q,1)$和$(\neg p\vee q,0.8)$,对于每个由$(p\vee q,1)$扩展的子句,必然含有 p;对于每个由$(\neg p\vee q,0.8)$扩展的子句,必然含有$\neg p$.因此,我们无法在它们扩展出的极大项集合上应用(SPL-S)规则.考虑子句$(p\vee q,1)$和$(r,0.6)$,前者可以扩展出$(p\vee q\vee r,1)$,后者可以扩展出$(p\vee q\vee r,0.6)$.可以通过应用规则(SPL-S)删除$(p\vee q\vee r,0.6)$.

引理 1. 给定任意两个可能性子句$(c_1,\alpha)=(l_1\vee\dots\vee l_r,\alpha),(c_2,\beta)=(l_1'\vee\dots\vee l_s',\beta)$,在$(c_1,\alpha)$和$(c_2,\beta)$中不含互补文字.设 M 为Σ中出现的文字集合,$|M|=m$.设 $L=\{l_1,\dots,l_r\}\cup\{l_1',\dots,l_s'\}=\{l_1'',\dots,l_t''\}$.若 C_1 为(c_1,α)扩展出的可能性极大项集合,C_2 为(c_2,β)扩展的所有可能性极大项集合,C_3 为在 C_1 和 C_2 上应用规则(SPL-S)删除的子句集合,则:

(1) $|C_1|=2^{m-r}$.

(2) $|C_2|=2^{m-s}$.

(3) $|C_3|=2^{m-t}$.

下面用一个例子说明引理 1.

例 3:设可能性子句集合如例 1 所示,显然,原子集合 $M=\{p,q,r\}$.我们可以看出:C_1 通过(SPL-ER)规则可以扩展的极大项个数为 $2^{|M|-|C_1|}=2^{3-2}=2$;C_4 可以扩展出 $2^{3-1}=4$ 个极大项.C_1 和 C_2 通过(SPL-S)规则没有约简任何极大项,C_1 和 C_4 通过(SPL-S)规则约简了 $2^{|M|-|C_1\cup C_4|}=2^{3-3}=1$ 个极大项.

给定任意两个子句,例如 $C_1=(p\vee q,1),C_2=(r,0.6)$.设 D_1 为 C_1 扩展出的极大项集合,D_2 为 C_2 扩展出的极大项集合.显然,$(p\vee q\vee r,1)\in D_1,(p\vee q\vee r,0.6)\in D_2$.尽管 $(p\vee q\vee r,1)\neq(p\vee q\vee r,0.6)$,但由于可以应用规则(SPL-S)对 $P_1\cup P_2$ 进行化简,在此过程中 $(p\vee q\vee r,0.6)$ 将被删除,而只有 $(p\vee q\vee r,1)$ 得以保留,因此在下面的公式中,我们将略微改变交集的定义,即 $D_i\cap D_j=\{(c_i,\beta)\}$,其中,$(c_i,\beta)\in D_i,(c_i,\alpha)\in D_j,\alpha\geqslant\beta$.如 $D_1\cap D_2=\{(p\vee q\vee r,0.6)\}$.需要在此说明的是,当问题化简为经典集合时,上述定义并没有改变交集定义.实际上,此处“交集”的语义在于指出哪些极大项将被通过规则(SPL-S)删除.

根据定理 5 和引理 1,下面我们介绍如何利用容斥原理计算可能性极大项的个数:

给定一个可能性子句集 $\Sigma=\{c_1,c_2,\dots,c_n\}$,$M$ 是出现在 Σ 中的所有原子的集合且 $|M|=m$.记 P_i 为 C_i 扩展出的所有可能性极大项的集合,令 S 为 Σ 能扩展出的不能应用规则(SPL-S)约简的所有极大项的集合的元素个数.则

$$S=|P_1\cup P_2\cup\dots\cup P_n|=\sum_{i=1}^{n}|P_i|-\sum_{1\leqslant i<j\leqslant n}|P_i\cap P_j|+\sum_{1\leqslant i<j<l\leqslant n}|P_i\cap P_j\cap P_l|-\dots+(-1)^{n+1}|P_1\cap P_2\cap\dots\cap P_n| \tag{1}$$

其中,

$$|P_i|=2^{m-|c_i|},$$

$$|P_i\cap P_j|=\begin{cases}0, & c_i,c_j\text{中含有互补对}\\ 2^{m-|c_i\cup c_j|}, & \text{否则}\end{cases}.$$

例 4:设子句集 $\Sigma=\{(p\vee q,1),(r,0.6),(\neg r,0.3)\}$,判断 $\_Inc(\Sigma)$ 是否为 0.

根据公式(1),Σ 可以扩展出的子句集合个数:

$$S=\sum_{i=1}^{n}|P_i|-\sum_{1\leqslant i<j\leqslant n}|P_i\cap P_j|+\sum_{1\leqslant i<j<l\leqslant n}|P_i\cap P_j\cap P_l|=2^{3-2}+2^{3-1}+2^{3-1}-2^{3-3}-2^{3-3}-0+0=8=2^3.$$

因此,$\_Inc(\Sigma)>0$.

根据公式(1),可以给出求解一致性程度的一般方法:给定可能性知识库 Σ,首先按照可能性子句的权值 α 对 Σ 从小到大进行排序,依次考虑 Σ 的不同 α 截集 Σ_α,根据公式(1)判断 Σ_α 是否可满足:若 Σ_α 不可满足则 Σ 的不一致程度为 α,否则考虑下一截集.

3.1 实验结果比较

在最坏情况下,可能性扩展规则方法的时间代价是指数级别的.但是在一些情况下,可以在多项式时间内找到解.比如,当给定的子句集中任意一对可能性子句都含有互补对时,任意两个子句扩展出的极大项集合的交集都为空,所以求解公式(1)中的前 n 项即可求出极大项的个数.事实上,在第 4 节中介绍的基于可能性扩展规则的知识编译方法就是基于此的.而对于基于可能性归结的方法,这时效率一般较低,因为存在着更多的归结选择.另一种极端情况是任意一对可能性子句都不含互补对,这时,为了计算可能性极大项的个数,必须计算公式(1)中的所有 2^m 项.这种情况下,可能性归结方法却可以很快计算出给定子句集的不一致性程度.

文献[3]引入了互补因子的概念来描述给定一个子句集合中含有的互补对多少的情况,本文也利用互补因子来比较可能性归结方法和可能性扩展规则方法.给定子句集 $\Sigma=\{c_1,c_2,\dots,c_n\}$,定义互补因子为 $S/(n\times(n-1)/2)$,其中,S 用于表示子句集中互补的子句对个数.

实验的目的在于,说明基于可能性扩展规则的方法和基于可能性归结方法由于互补因子大小的不同而导致的性能的差异.我们在一台 CPU 主频为 2.8G,内存为 512M 的 Dell PC 机上测试了我们的算法.问题样例由一个随机产生器产生,这个随机产生器有 3 个输入参数:变量的个数 n、子句的个数 m 以及每个子句的最大长度 k.每个子句都是随机地从 n 个变量中选取不大于 k 个得到的,并且每个变量为正或为负的概率都等于 0.5,子句的值为(0,1)之间的随机数.由于现实世界中的问题转化成样例之后不一定都是由相同长度的子句构成的,我们

的所有样例中每个子句长度并不固定.实验结果是 50 次实验的平均结果,有些舍去了小数点后的数字.对于 200 个子句、15 个变量、子句长度最大为 15 的情况,当互补因子较高时(比如 0.93),可能性扩展规则方法平均需要 0.062s 的时间,可能性归结方法平均需要 23.109s;当互补因子较低时(比如 0.25),可能性扩展规则方法平均需要 19.156s,而可能性归结只需要 0.73s.总体来说,互补因子高的问题可能性扩展算法效率更高;互补因子低的问题可能性归结原理的效率更高.由此可以得出以下结论:假设所有的可能性逻辑推理问题都在一个光谱上,光谱的一端(比如左端)是任意两个可能性子句都含互补对的情况,在光谱的另一端(右端)是任意两个可能性子句都不含互补对的情况,那么从理论上讲,在光谱的左端使用基于可能性扩展规则的方法效率更高,在光谱的右端可能性归结的方法效率更高.从这个意义上说,可能性扩展规则方法可以看作是任何可能性归结方法的补运算.

4 基于可能性扩展规则的可能性知识编译

通常,可能性推理问题是难以处理的,大规模的推理问题往往需要指数级别的时间花费.知识编译是近年来提出的一个新的研究方向,最初被用于处理通用推理问题的难求解性,目前已经被广泛地应用于人工智能的众多领域中.知识编译的基本思想是,将推理问题分为两个阶段,即离线编译(off-line compiling)阶段阶段和在线推理(on-line reasoning)阶段.在离线编译阶段,原有的理论被编译为与之等价的易处理的目标语言理论,这样就可以在多项式时间内完成其后的在线推理阶段的工作.知识编译的优点在于:虽然离线编译阶段需要指数级别的时间,但是由于任意一个知识库只需进行一次编译,因此编译阶段所耗费的时间可以通过对同一个知识库的多次查询得到补偿.在可能性扩展规则的基础上,本文提出了一种可能性知识编译方法.给定任意子句集Σ,通过知识编译可以将其转换成易处理的形式.这样,前文中所介绍的基于可能性扩展规则推理过程就可以在多项式时间内完成.

4.1 可能性知识编译

知识编译是将给定的知识转换为易于推理的表现形式的过程,转换的结果称为目标语言.在经典逻辑中作为知识编译的目标语言必须满足一定的性质,即需要其在蕴含可控制类中(即多项式时间内可以回答蕴含问题)[15].

定义 12. 设 L 为一个理论,对于任意子句集$\Sigma\in L$,如果Σ的可满足性可以在多项式时间内得到判定,则称理论 L 在可满足性可控制类中.

定义 13. 设 L 为一个理论,对于任意子句集$\Sigma\in L$,如果对于Σ的任意一个询问即“$\Sigma\vDash C$”是否成立都能在多项式时间内给出回答,则称理论 L 在蕴含可控制类中.

定义 14. 设 L 为一个理论,对于任意子句集$\Sigma\in L$,Σ可以作为知识编译的目标语言仅当 L 是在蕴含可控制类中.

对于可能性逻辑,问题要更为复杂.我们对可满足可控制和蕴含可控制进行扩展,引入一致性程度计算可控制类和泛化蕴含可控制类的定义.

定义 15. 设 L 为一个理论,对于任意子句集$\Sigma\in L$,如果可以在多项式时间内计算出Σ的一致性程度,则称理论 L 在一致性程度计算可控制类中.

可以看出,如果一个可能性逻辑问题属于一致性程度计算可控制类中,那么它的经典投影必然属于可满足可控制类中.因此,一致性程度计算可控制可以看作是可满足可控制的泛化.

文献[15]中给出了 P-反驳完备的定义,如果一个理论的每个子句集都是是 P-反驳完备的,那么该理论是在蕴含可控制类中的:

定义 16. 令$\vdash_p$是一个能够正确判定子句集可满足性的多项式过程.一个子句集Σ是 P-反驳完备的当且仅当对任何子句 C:$\Sigma\vDash C$ 当且仅当$(\Sigma\cup\neg C)\vdash_p\bot$.

重新观察定义 7 和定义 8,给定可能性知识库Σ,经典子句 C,决策形式演绎问题需要判断给定某个值α,(C,α)是否为Σ的逻辑结果;最优化形式演绎需要判断作为Σ的逻辑结果,子句 C 的值最大为多少.如果我们能够在多项

式时间内计算出作为Σ的逻辑结果的子句的值的上确界,那么也能在多项式时间内对决策形式演绎问题作出回答.这是因为在回答决策形式演绎时,只需比较给定子句的值是否大于该上确界即可.由此可知,给定任意理论Σ,如果可以在多项式时间内回答最优化形式演绎问题,我们就能在多项式时间内回答决策化形式演绎问题.因此,本文的后半部分只考虑最优化形式演绎问题.下面给出 P-反驳完备在可能性逻辑中的泛化.

定义 17. 设Σ为可能性知识库,C 为任意经典逻辑子句,令$\vdash_p$_$Inc(\cdot)$是一个能正确计算出子句集不一致性程度值的多项式过程.一个子句集Σ是 P-可能性反驳完备的当且仅当:

(1) $\vdash_p$_$Inc(\Sigma\cup\{(\neg C,1)\})$成立;

(2) 能够在多项式时间内判断_$Inc(\Sigma\cup\{(\neg C,1)\})\geqslant$_$Inc(\Sigma)$是否成立.

定义 18. 设 L 为一个理论,称 L 是在可能性蕴含可控制类中,当且仅当对于任意可能性子句集合$\Sigma\in L$,Σ是 P-可能性反驳完备的.

定义 19. 设 L 为一个理论,对于任意可能性子句集合$\Sigma\in L$,Σ可以作为可能性知识编译的目标语言,仅当 L 是在可能性蕴含可控制类中.

根据定义 18,如果一个可能性逻辑子句集合是在可能性蕴含可控制类中的,则它的投影是在蕴含可控制类中的.因此,可能性蕴含可控制类实际上是蕴含可控制类的可能性泛化.

相对于不一致程度计算可控制类,可能性蕴含可控制类的限制更强:对于给定理论的任意询问,它需要在多项式时间内计算出最优值,而 P-不一致性程度计算实际上是 P-可能性反驳完备的特例,即只需计算出询问⊥的最优值.下面讨论 P-不一致性程度计算和 P-可能性反驳完备之间的关系.

定理 6. 令 L 是理论类并且满足如下条件:

(1) 对于任意可能性子句集合$\Sigma\in L$ 不可满足当且仅当$\vdash_p Inc(\Sigma)$;

(2) 设 s 为经典命题单元子句,则对于任意 s,如果$\Sigma\in L$,那么$\Sigma\cup\{(s,1)\}\in L$.

这时,对于任意$\Sigma\in L$ 都是 P-可能性反驳完备的.

证明:设Σ为任意可能性子句集合,设φ为任意经典逻辑子句,其中,$\varphi=s_1\vee s_2\vee\ldots\vee s_m$,$m$ 为常量.由于

$$\Sigma\cup\{(\neg s_1\wedge\ldots\wedge\neg s_m,1)\}=\Sigma\cup\{(\neg s_1,1)\wedge\ldots\wedge(\neg s_m,1)\}=\Sigma\cup\{(\neg s_1,1)\}\wedge\{(\neg s_2,1)\}\wedge\ldots\wedge\{(\neg s_m,1)\},$$

根据条件(2),

$$\Sigma\cup\{(\neg s_1,1)\}\in L,\text{且}\Sigma\cup\{(\neg s_1,1)\}\wedge\{(\neg s_2,1)\}\wedge\ldots\wedge\{(\neg s_m,1)\}\in L,$$

因此,

$$\Sigma\cup(\neg\varphi,1)\in L,$$

根据条件(1),

$$\vdash_p Inc(\Sigma\cup(\neg\varphi,1)),$$

因此,任意的$\Sigma\in L$ 是 P-可能性反驳完备的. □

引理 2. 令 L 是理论类并且满足以下条件,则任意的$\Sigma\in L$ 是 P-可能性反驳完备的:

(1) 任意可能性子句集合$\Sigma\in L$ 不可满足当且仅当$\vdash_p Inc(\Sigma)$;

(2) 设 s 为经典命题单元子句,则对于任意 s,如果$\Sigma\in L$,那么$\Sigma\cup\{(s,1)\}\in L$,或者经过一个多项式时间等价处理后$\Sigma\cup\{(s,1)\}\in L$.

下面定义 EPPCCCL 理论,并证明该理论是在一致性程度计算可控制类和可能性蕴含可控制类中.

定义 20. EPPCCCL 理论是其中任意两个可能性子句的经典投影都含有互补对的子句集.

如前所述,EPPCCCL 理论中任意两个可能性子句扩展出的极大项的集合无法使用(SPL-S)规则进行约简,因此给定一个子句集$\Sigma=\{C_1,C_2,\ldots,C_n\}$,$M$ 是出现在Σ中的所有原子的集合且$|M|=m$.记 P_i 为 C_i 扩展出的所有可能性极大项的集合,令 S 为Σ能够扩展出的不能应用规则(SPL-S)约简的所有极大项的集合的元素个数.根据公式(1),$S=\sum_{i=1}^{n}|P_i|$,因此,以下定理成立:

定理 7. EPPCCCL 理论在一致性程度计算可控制类中.

EPPCCCL 理论是在一致性程度计算可控制类中,不能保证 EPPCCCL 理论可以作为可能性知识编译的目标语言,下面证明 EPPCCCL 理论是在可能性蕴含可控制类中的.首先介绍加强可能性包含规则 SPL-SS.

定义 21. 给定 SPL 公式集合 $C=\{(c,\alpha),(c',\beta)\}$,其中,$\alpha\geqslant\beta,c\vDash c',D=\{(c,\alpha)\}$.我们将 C 到 D 的推导过程称为加强可能性包含规则,C'中元素为 C 应用加强可能性包含规则的结果.

根据定义,C 和 C'之间是逻辑等价的.

定理 8. EPPCCCL 理论在可能性蕴含可控制类中.

证明:对于 EPPCCCL 理论中的任意一个可能性子句集,根据定理 7,其不一致性程度可以在多项式时间内得到计算,是在不一致性程度计算可控制类中,引理 2 中的第 1 个条件可以满足.下面证明引理 2 的第 2 个条件也可以被满足.加入任意一个单元子句$(s,1)$之后,由于Σ中的任意两个子句的经典投影已经含有互补对,只需要处理这样的一些可能性子句对:在这样的子句对中,一个子句是Σ中的可能性子句(c,α),另一个是$(s,1)$.考虑如下情况:(1) 若 c 中含有文字$\neg s$,则$(s,1)$和(c,α)已经含有互补对,这时不必进行任何操作;(2) 若 c 中含有文字 s,则根据加强可能性包含规则,我们将(c,α)删除;(3) 对于其他情况,在$\{s\}$上对(c,α)应用可能性扩展规则,得到两个子句$(c\vee s,\alpha),(c\vee\neg s,\alpha)$.根据加强可能性包含规则,将$(c\vee s,\alpha)$删除,这时剩下的可能性子句$(c\vee\neg s,\alpha)$和$(s,1)$含有互补对.因此,无论是哪种情况,子句集中的子句的数目都不会增加.这样,这个处理过程一定能够在子句集规模的线性时间内完成,且可以保证得到的结果是一个 EPPCCCL 理论.因此,引理 2 中的条件(2)也成立.由上可知,EPPCCCL 理论在可能性蕴含可控制类中. □

引理 3. 当给定知识为确定性知识时,EPPCCCL 理论在可满足可控制类和蕴含可控制类中.

以上证明了 EPPCCCL 理论可以作为可能性知识编译的目标语言,下面介绍给定任意子句集合,如何将其等价转换为 EPPCCCL 理论.我们给出了可能性知识编译算法 SPL-KCER 算法.算法的基本思想是基于桶删除(bucket elimination)方法.初始时,Σ中的每一个可能性子句相当于一个桶(bucket),算法依次处理每一个桶.当处理一个桶时,其相应的子句被从Σ_1移到Σ_2,用可能性扩展规则和加强可能性包含规则来保证每一个含Σ_1中的一个子句和Σ_2中的一个子句的子句对含互补对.这样,Σ_2中的子句就可以作为编译结果的一部分,被从Σ_2中移到Σ_3中.所有的桶都被处理完之后,算法即终止,Σ_3作为其输出.

算法. SPL-KCER.

输入:令子句集$\Sigma_1=\{C_1,C_2,\ldots,C_n\},\Sigma_2=\Sigma_3=\varnothing$.

```
While Σ1≠∅
Loop
    从Σ1 中选出一个子句,比如 C1,把 C1 加入到Σ2 中
    While i<Σ1 中子句的数目
    Loop
        While j<Σ2 中子句的数目
        Loop
        If Ci 和 Cj 的经典投影中含互补对 Then skip
        Else if Ci 和 Cj 可以应用 SPL-SS 规则 Then 应用 SPL-SS 规则删除一个子句
        Else 用可能性扩展规则扩展 Cj,将结果放入Σ2 中
        j:=j+1
        End loop
        i:=i+1
    End loop
    Σ3:=Σ3∪Σ2,Σ2:=∅
End loop
```

输出:Σ_3是可能性知识编译的结果.

例 4:对 $C=\{(p\vee q,1),(\neg p\vee q,0.8),(\neg r\vee\neg q,0.4),(r,0.6),(\neg r,0.3)\}$进行编译,步骤如下:

1. $\Sigma_1=\{(p\vee q,1),(\neg p\vee q,0.8),(\neg r\vee\neg q,0.4),(r,0.6),(\neg r,0.3)\},\Sigma_2=\Sigma_3=\varnothing$.

2. $\Sigma_1=\{(p\vee q,1),(\neg p\vee q,0.8),(\neg r\vee\neg q,0.4),(r,0.6)\},\Sigma_2=\{(\neg r,0.3)\},\Sigma_3=\varnothing$.
3. $\Sigma_1=\{(p\vee q,1),(\neg p\vee q,0.8),(\neg r\vee\neg q,0.4),(r,0.6)\},\Sigma_2=\varnothing,\Sigma_3=\varnothing$.
4. $\Sigma_1=\{(p\vee q,1),(\neg p\vee q,0.8),(r,0.6)\},\Sigma_2=\{(\neg r\vee\neg q,0.4)\},\Sigma_3=\varnothing$.
5. $\Sigma_1=\{(p\vee q,1),(\neg p\vee q,0.8),(r,0.6)\},\Sigma_2=\varnothing,\Sigma_3=\{(\neg r\vee\neg q,0.4)\}$.
6. $\Sigma_1=\{(p\vee q,1),(\neg p\vee q,0.8)\},\Sigma_2=\{(r,0.6)\},\Sigma_3=\{(\neg r\vee\neg q,0.4)\}$.
7. $\Sigma_1=\{(p\vee q,1),(\neg p\vee q,0.8)\},\Sigma_2=\{(p\vee\neg q\vee r,0.6),(\neg p\vee\neg q\vee r,0.6)\},\Sigma_3=\{(\neg r\vee\neg q,0.4)\}$.
8. $\Sigma_1=\{(p\vee q,1),(\neg p\vee q,0.8)\},\Sigma_2=\varnothing,\Sigma_3=\{(\neg r\vee\neg q,0.4),(p\vee\neg q\vee r,0.6),(\neg p\vee\neg q\vee r,0.6)\}$.
9. $\Sigma_1=\{(p\vee q,1)\},\Sigma_2=\{(\neg p\vee q,0.8)\},\Sigma_3=\{(\neg r\vee\neg q,0.4),(p\vee\neg q\vee r,0.6),(\neg p\vee\neg q\vee r,0.6)\}$.
10. $\Sigma_1=\{(p\vee q,1)\},\Sigma_2=\varnothing,\Sigma_3=\{(\neg r\vee\neg q,0.4),(p\vee\neg q\vee r,0.6),(\neg p\vee\neg q\vee r,0.6),(\neg p\vee q,0.8)\}$.
11. $\Sigma_1=\Sigma_2=\varnothing,\Sigma_3=\{(\neg r\vee\neg q,0.4),(p\vee\neg q\vee r,0.6),(\neg p\vee\neg q\vee r,0.6),(\neg p\vee q,0.8),(p\vee q,1)\}$.

在例 4 中,我们首先将子句集合中所有可能性子句放入Σ_1中(步骤 1),并从中任意选出一个可能性子句,例如,$(\neg r,0.3)$放入Σ_2(步骤 2),并开始处理Σ_1和Σ_2中的每个可能性子句对.对于$(p\vee q,1),(\neg r,0.3)$,由于两者之间没有互补文字,因此分别在 p 和 q 上扩展$(\neg r,0.3)$,得到子句集合$\{(\neg r\vee\neg p,0.3),(\neg r\vee p\vee\neg q,0.3)\}$,需要注意的是,$(\neg r\vee p\vee q,0.3)$被$(p\vee q,1)$蕴含,所以可通过(SPL-SS)规则删除.这时,$\Sigma_2$中的元素变为$\{(\neg r\vee\neg p,0.3),(\neg r\vee p\vee\neg q,0.3)\}$.再将其和$\Sigma_1$中剩下的元素逐一处理,将$\Sigma_2$中元素和$(\neg p\vee q,0.8)$处理后,$\Sigma_2=\{(\neg r\vee p\vee\neg q,0.3),(\neg r\vee\neg p\vee\neg q,0.3)\}$;再将$\Sigma_2$中元素和$(\neg r\vee\neg q,0.4)$进行处理,根据(SPL-SS)规则可以删除$(\neg r\vee p\vee\neg q,0.3)$和$(\neg r\vee\neg p\vee\neg q,0.3)$.因此,这时$\Sigma_2$中元素为空,因此开始下一次循环(以上为步骤3).将Σ_1中再选取一个子句$(\neg r\vee\neg q,0.4)$(步骤4),按照上述的方法处理后,Σ_2中最后剩下的子句为$(\neg r\vee\neg q,0.4)$,这时将$(\neg r\vee\neg q,0.4)$放入Σ_3中(步骤5).重复上述过程,直到Σ_1中所有子句都被处理完毕(步骤 11).这时,我们可以得到一个等价的 EPPCCCL 理论.

定理 9. Σ_3等价于Σ_1并且Σ_3是一个 EPPCCCL 理论.

证明:因为算法 SPL-KCER 中用的可能性扩展规则和 SPL-SS 规则都保证等价性,所以Σ_3等价于Σ_1.注意到以下事实成立:若两个子句已经含有互补对并且我们对其中一个子句进行扩展,扩展后的结果和另外一个子句之间一定也含互补对.在算法 SPL-KCER 中,一个子句在被移到Σ_3之前一定和其他的任意一个子句都含有互补对,尽管这些子句随着算法的运行有可能被进一步扩展,根据上面的事实,Σ_3中的子句一定和这些子句扩展的结果之间含互补对.因此,最终结果Σ_3一定是一个 EPPCCCL 理论. □

4.2 实验结果

需要指出的是,虽然通过可能性知识编译,原有的可能性知识库表示形式被转换成为易于推理的知识表示形式,例如,假设编译后的可能性知识库中含有 n 个子句,在集合 V 上取值,其中$|V|=v$,则对于任何一个可能性演绎问题,我们都可以在 $O(n\times v)$时间内得到答案,但是这并不意味可以在多项式时间内解决了原有的推理问题.一般来讲,编译后的可能性子句集要大于编译前的子句集规模,最坏情况下甚至是指数级别的.因此,编译后的子句集的规模直接影响推理时间.我们用一些随机生成的样例测试了可能性知识编译方法.问题样例由一个随机产生器产生,这个随机产生器有 3 个输入参数:变量的个数 n、子句的个数 m 以及每个子句的最大长度 k.对这些样例编译的结果在表 1 中给出.同样,结果是 50 次实验的平均结果.

我们还扩展了经典逻辑中的一些 benchmark 问题,实验结果见表 2.从表中可以看出,经过知识编译后,在线推理时间大多能够控制在 0.2s 以内,远少于直接使用可能性扩展规则进行推理的时间.但需要注意的是,在知识编译的过程中依然需要花费较长时间.总的来说,编译时间和在线推理的时间的总和大约等于直接进行推理的时间.但是,由于知识编译实际属于离线推理,用户实际关心的还是在线推理的时间,而且知识编译所花的时间可以通过对同一个知识库的多次询问得到补偿.

我们将灵活规划问题转换为可能性子句,并给定的子句进行编译,在此基础上进行求解.表 3 给出了两种方法实验结果的比较,从表中可以看出,通过知识编译可以加速规划问题的求解.更为重要的是,在规划问题中,由于知识编译的过程可以在脱机状态下执行,而实际的在线推理时间往往很短,因此在线获取规划的效率也因此会大幅度提升.

Table 1 Experimental results of random instances of possibilistic knowledge compilation

表 1 对于随机生成的实例的可能性知识编译实验结果

Number of variables	Number of clauses	Max length of each clause	Number of clauses after KC
20	40	10	2 767
20	60	10	6 712
20	80	10	9 760
30	40	10	4 508
30	60	10	7 827
30	80	10	13 029

Table 2 Experimental results of benchmark instances of possibilistic knowledge compilation

表 2 对于标准测试问题的可能性知识编译实验结果

Problems	Number of variables	Number of clauses	Numbers of clauses after KC	KC time	Online reasoning time	Offline reasoning time
Adder-3	13	43	169	0.21	0.02	0.12
Adder-4	17	57	681	0.28	0.02	0.14
Adder-5	21	71	2 729	0.21	0.09	0.27
Pipes-2	15	54	208	0.33	0.01	0.29
Pipes-3	21	82	483	0.86	0.03	1.12
Ppipes-4	27	110	767	84.46	0.11	82.06
Pigeon-4-5	20	74	4 430	73.21	0.17	66.4

Table 3 Experimental results of benchmark instances of flexible planning instances

表 3 灵活规划问题的实验结果

Problems	Length of optimal plan	Directly execution time	KC execution time
$A_1(3,2)$	9	23.609	7.327
$A_2(3,2)$	12	819.672	147.209
$B_1(4,2)$	9	1 031.468	679.254
$B_2(3,2)$	11	11 138.547	7 008.909
$C(5,3)$	11	35.703	35.127

目前,基于可能性知识编译的工作刚刚展开,只有文献[14]中做了一定的工作,由于我们没有获得Prade等人的程序,因此无法对两者之间的工作作出实验比较.但是我们相信,这两种方法都具有优缺点,在有些情况下,我们的方法的结果应该比 Prade 的方法的结果的指数级别要小.考虑如下情况:假设待编译的子句集合本身就是一个 EPPCCCL 理论,我们的方法编译后的结果就是它本身.由于 Prade 等人的方法是基于可能性归结原理的,结果一般将为指数级别.

5 总 结

本文将扩展规则推广到可能性逻辑中,提出了基于可能性扩展规则的推理方法;扩展了文献[15]中蕴含可控制和可满足可控制的概念,提出了可能性蕴含可控制、不一致性程度计算可控制的概念;基于可能性扩展规则,提出了 EPPCCCL 理论,并证明 EPPCCCL 理论是在最优化形式蕴含可控制类、不一致性程度计算可控制类中,并给出了将任意可能性子句集转换为与之等价的 EPPCCCL 理论的一般方法.需要指出的是,可能性扩展规则方法虽然需要考虑可能性知识库中的不一致性程度,但是在具体扩展过程中所采用的扩展方法和推理手段与经典扩展规则方法类似,无须考虑可能性逻辑中的其他特性.因此,本文给出的推广稍加修改后还可以被应用于概率逻辑和模糊逻辑中.

致谢 在此,感谢评审专家对本文提出的宝贵意见,并对本文的工作给予支持和建议的同行表示感谢.

References:

[1] Lang J, Possibilistic logic: Complexity and algorithms. In: Gabby DM, Semts P, Kohlas J, Moral S, eds. Handbook of Defeasible Reasoning and Uncertainty Management Systems. 1997. 179–220.

[2] Dubois D, Prade H. Possibilistic logic: A retrospective and prospective view. Fuzzy Sets and Systems, 2004,144(1):3–23. [doi:

10.1016/j.fss.2003.10.011]
[3] Lin H, Sun JG, Zhang YM. Theorem proving based on extension rule. Journal of Automated Reasoning, 2003,31(1):11−21. [doi: 10.1023/A:1027339205632]
[4] Wu X, Sun JG, Lin H, Feng SS. Modal extension rule. Progress in Natural Science, 2005,15(6):550−558. [doi: 10.1080/10020070512331342540]
[5] Yin MH, Sun JG. Counting models using extension rules. In: Proc. of the 22nd AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 2007. 1916−1917.
[6] Yin MH, Lin H, Sun JG. Solving #SAT using extension rules. Journal of Software, 2009,20(7):1714−1725 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3320.htm [doi: 10.3724/SP.J.1001.2009.03320]
[7] Li Y, Sun JG, Wu X, Zhu XJ. Extension rule algorithms based on IMOM and IBOHM heuristics strategies. Journal of Software, 2009,20(6):1521−1527 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3420.htm [doi: 10.3724/SP.J.1001.2009.03420]
[8] Selman B, Kautz HA. Knowledge compilation using horn approximation. In: Proc. of the 9th National Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 1991. 904−909.
[9] Del Val A. Tractable databases: How to make propositional unit resolution complete through compilation. In: Doyle J, Sandewall E, Torasso P, eds. Proc. of the 4th Int’l Conf. on Principles of Knowledge Representation and Reasoning. San Francisco: Morgan Kaufmann Publishers, 1994. 551−561.
[10] Marquis P. Knowledge compilation using theory prime implicates. In: Proc. of the 14th Int’l Joint Conf. on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 1995. 837−843.
[11] Schrag R. Compilation for critically constrained knowledge bases. In: Proc. of the 13th National Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 1996. 510−515.
[12] Darwiche A, Marquis P. A knowledge compilation map. Journal of Artificial Intelligence Research, 2002,17(1):229−264.
[13] Lin H, Sun JG. Knowledge compilation using extension rule. Journal of Automated Reasoning, 2004,32(2):93−102. [doi: 10.1023/B:JARS.0000029959.45572.44]
[14] Benferhat S, Prade H. Compiling possibilistic knowledge bases. In: Brewka G, Coradeschi S, Perini A, Traverso P, eds. Proc. of the ECAI 2006. Amsterdam: IOS Press, 2006. 337−341.
[15] Del Val A. On some tractable classes in deduction and abduction. Artificial Intelligence, 2000,116(1):297−313. [doi: 10.1016/S0004-3702(99)00088-0]

附中文参考文献:

[6] 殷明浩,林海,孙吉贵.一种基于扩展规则的#SAT 求解系统.软件学报,2009,20(7):1714−1725. http://www.jos.org.cn/1000-9825/20/1714.htm [doi: 10.3724/SP.J.1001.2009.03320]
[7] 李莹,孙吉贵,吴瑕,朱兴军.基于 IMOM 和 IBOHM 启发式策略的扩展规则算法.软件学报,2009,20(6):1521−1527. http://www.jos.org.cn/1000-9825/3420.htm [doi: 10.3724/SP.J.1001.2009.03420]

殷明浩(1979－),男,安徽安庆人,博士,讲师,CCF 会员,主要研究领域为自动推理,智能规划.

林海(1981－),男,博士,讲师,主要研究领域为自动推理.

孙吉贵(1962－2008),男,博士,教授,博士生导师,主要研究领域为自动推理.

吴瑕(1976－),女,博士,讲师,主要研究领域为自动推理.