

一种描述逻辑与 DL-safe 规则复合逻辑中的规则抽取方法^{*})

张 阔 李涓子 唐 杰 王克宏

(清华大学计算机系知识工程研究室 北京 100084)

摘 要 OWL 本体描述语言包含相对丰富的概念定义符,但提供的关系定义符描述能力则较差。将 OWL 与规则相结合是克服这种表达能力限制的一种常用方法。Motik 提出了一种可判定的方法,将 SHIQ(D)描述逻辑与规则结合,所结合的规则限定在一种叫做 DL-safe 规则的范围内。本文提出了一种抽取 DL-safe 规则的方法,方法利用 SHIQ(D)逻辑中的 UNION 定义符满足一些规则在前件中包含析取逻辑运算符的需求,从而扩展了规则的描述能力。
关键词 描述逻辑, DL-safe 规则, OWL, 语义网

Rule Extraction in Combining Description Logics with DL-safe Rules

ZHAGN Kuo LI Juan-Zi TANG Jie WANG Ke-Hong

(Knowledge Engineering Group, Department of Computer, Tsinghua University, Beijing 100084)

Abstract Although the ontology languages like OWL include a relatively rich set of class constructors, it provides much weaker constructors for roles. One way to overcome this expressive limitation of OWL would be to extend it with rules. Motik presented a decidable combination of SHIQ(D) with rules, where decidability is due to restricting the rules to so-called DL-safe ones. In the paper, an approach for extracting new DL-safe rules is proposed. The approach makes use of the disjunctive constructor in SHIQ(D) to satisfy some kinds of rules with disjunction in their antecedents. And the new rules are DL-safe ones too.

Keywords Description logics, DL-safe rule, OWL, Semantic web

1 引言

当今的网络极大地促进了电子文档的应用性,但是同时它巨大的成功以及其潜在的增长使网络上的信息越来越难于搜索,访问,显示以及维护。此外,组成当今的网络的页面中的信息大多以自然语言文本以及图像的形式存在,这些内容仅仅限于人来观看和理解。因此, Tim Berners-Lee 首先提出了语义网络的概念,语义网络根据数据中所包含的机器可处理的语义信息及元数据中的启发式信息提供信息的自动访问及处理。

在语义网络中,隐藏在数据,网页以及其它资源中的语义信息被以本体形式存在的元数据所明确地表示出来。所以,它可以提供面向各种异构的以及分布的数据源的智能访问。

Tim Berners-Lee 提出了语义网络中所包含的各类技术的层次结构(图 1)。其中 XML 已经得到了因特网应用领域的广泛认可,并且拥有越来越多的应用程序。然而,遵循 XML 语法的文档只是机器可读的,并不能够做到机器可理解。以 RDF/RDFS 形式存在的信息可以直接的映射到一个没有歧义的非集中式的模型中,而且存在很多通用的解析器可以达到这个目的。建立在 RDF/RDFS 技术层之上的是本体层,本体层由一些表达能力很强的本体描述语言如 DAML + OIL^[1] 和 OWL^[2] 组成(DAML + OIL 和 OWL 分别由 DARPA 和 W3C 设计而成)。然而,描述逻辑做为这些本体语言的逻辑基础,其表达能力仍然不够强大,因为缺少对于概念与角色之间关系建模的能力。

尽管像 OWL 这样的本体语言拥有相对丰富的概念定义

符,但是它提供的角色定义符描述能力则较差。克服这种 OWL 表达能力缺陷的一个可行方法是与规则相结合。因为 Horn 规则与描述逻辑是正交的两种一阶谓词逻辑的子集合^[3],所以一些应用程序可以利用 Horn 规则与描述逻辑的复合逻辑的表达能力解决一些实际问题。

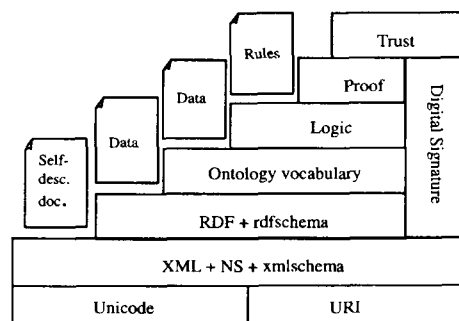


图 1 语义网络技术层次结构

在本文中,我们提出了一种在规则与描述逻辑的复合逻辑中抽取新规则的方法。在一般的规则中,前件只允许合取逻辑运算符的存在,然而在我们的方法所抽取出的新规则中,满足一定条件的 DL 项的析取关系仍然可以得到很好的表达与处理。

论文第 2 部分介绍一些相关工作并对其优缺点进行简要的讨论。在第 3 部分中,我们用一个例子来展示新规则抽取的必要性与优越性。方法的具体细节与原理在第四部分中给出。最后给出结论。

^{*}) 自然科学基金项目编号:60443002。张 阔 博士研究生。

2 相关工作

2.1 描述逻辑与规则的结合工作

实际上,向描述逻辑中增加规则并不是一个全新的想法。一些早期的基于描述逻辑的系统(如,CLASSIC^[4],LOOM^[5])已经包含相应的规则语言组件。然而,在这些系统中的规则所能表达的语义甚至比 sub-class 和 super-class 所能表达的语义还要弱。形式仅仅限定为 $C(x) \Rightarrow D(x)$, 其中 C 和 D 都是概念。此外,这些规则仅仅可以应用于实例,不能用于进行基于概念的推理(比如计算概念之间的层次关系)。

AL-log^[6]对基本的描述逻辑 ALC 与递归 DATALOG 规则进行了合并。规则的前件中仅可以包含以 ALC 中的概念作为一元谓词的项,而且它需要规则中的所有变量都必须出现在普通谓词中(所谓普通谓词即描述逻辑以外的谓词)。所以,所有的规则只能适用于在知识库中明确存在的实例。

CARIN^[7]语言系列对于 Horn 规则与描述逻辑进行了更加紧密的结合。CARIN 的知识库包含一个 Horn 规则集合以及描述逻辑的概念及角色定义。不仅仅普通谓词项可以出现在规则的前件中,甚至概念与角色谓词项都可以出现在规则的前件中。为了建立完备的推理过程,CARIN 语言分别对于公式进行了两种类型的限制。第一种限制为对描述逻辑概念定义符的限制,其中描述逻辑中的最大子集 ALCNR 可以和任意类型的递归无函数的 Horn 规则相结合而不影响推理过程的完备性。第二类是对于 Horn 规则的限制,要求在角色项中出现的变量中至少有一个要出现在一个普通谓词中。另外 CARIN 语言的一个很大的限制就是概念与角色谓词项只能出现在规则的前件中,而不能出现在后件中。

OWL 规则语言(ORL)^[8]将 OWL-DL 与 Horn 规则相结合,在这种结合方法中,概念与角色谓词项允许出现在规则的前件与后件中而没有任何限制。这种方法的最强大之处在于它的简单性以及它与现存的本体语言 OWL 结合的紧密性。由于 Horn 规则的特性,在规则中不允许出现逻辑析取符号以及逻辑非符号,也不存在内嵌的数学谓词。谓词的范围限定在描述逻辑中的概念与角色,这样使谓词的最高元数限定为 2。但是正如其作者指出,这种合并的推理过程是不可判定的。

2.2 SHIQ(D)描述逻辑与 DL-safe 规则结合

SHIQ(D)提供了 OWL-DL 的基本描述逻辑。OWL-DL 所应用的描述逻辑为 SHOIN(D),它与 SHIQ(D)的主要区别在于对于 NOMINALS 的支持。NOMINALS 提供一种概念的定义(这种概念是由确定的几个实例所构成)。文[9]提出了一种复合 SHIQ(D)描述逻辑与规则的方法,它通过限制所复合的规则为 DL-safe 规则而得到推理的可判定性。DL-safe 的定义如下:

定义 1(DL-safe 规则^[9]) 设 KB 为一个 SHIQ(D)知识库, N_c 为知识库中的概念集合, N_{R_0} 为抽象角色的名字集合, N_R 为类型角色的集合, N_P 为谓词符号的集合, $\{\approx\} \cup N_c \cup N_{R_0} \cup N_R \subseteq N_P$ 。一个 DL 项的形式如 $A(s)$, 其中 $A \in N_c$, 或者形如 $R(s, t)$, 其中 $R \in N_{R_0} \cup N_R$ 。规则 r 形式如下:

$$A_1 \vee A_2 \vee \dots \vee A_m \leftarrow B_1, B_2, B_3, \dots, B_N \quad (1)$$

规则 r 的后件中可以包含逻辑析取运算符,且谓词符号均来源于 N_P 集合。规则 r 的前件中只能包含逻辑合取运算符,谓词符号来源于 N_P 集合与非 DL 项谓词,并且所有在 r 的前件中的变量必须至少出现在一个非 DL 项的谓词中。

实际上,他们的方法不局限在与 Horn 规则的合并中,它允许在规则的后件中出现逻辑析取符号。我们的方法可以使

逻辑析取符号不仅仅局限在规则的后件中,在满足一定的条件下,充分利用描述逻辑的析取表达能力,可以使某种形式的析取表达式出现在规则的前件中。

3 需求实例

为了使讨论更加清晰,我们通过一个实例来说明抽取规则的必要性。实例为一个简单的知识库 KB:

知识库定义如下:

概念 1: *Human*;

概念 2: *Animal*;

角色 1: *parent*;

断言 1: $(Human \sqcup Animal)(Jack)$;

断言 2: $parent(Jack, Johnson)$;

规则 1: $Human(y) \leftarrow Human(x), parent(x, y), O(x), O(y)$;

规则 2: $Animal(y) \leftarrow Animal(x), parent(x, y), O(x), O(y)$;

事实 1: $O(Jack)$;

事实 2: $O(Johnson)$;

推论: $(Human \sqcup Animal)(Johnson)?$

规则 1 的意思为一个人的父母仍然是一个动物。规则 2 的意思为一个动物的父母仍然为一个动物。断言 1 表示实例 Jack 是概念表达式 $Human \sqcup Animal$ 的一个实例。断言 2 表示 Johnson 是 Jack 的父母。事实 1 和事实 2 是非 DL 谓词项,它们表示 Johnson 和 Jack 都是知识库中 Abox 中明确存在的实例。

Jack 是属于 *Human* 概念的实例,或者是属于 *animal* 概念的实例,但是我们不能确定具体是属于哪个概念。不管是具体属于哪个概念,都可以通过规则 1 或规则 2 推理得到 Johnson 是概念表达式 $Human \sqcup Animal$ 的一个实例。但是, $Human(Jack)$ 和 $Animal(Jack)$ 都没有在知识库中明确提到,所以没有一个规则可以通过直接从 DL 部分获取知识而得到执行。

我们的方法通过抽取新规则的方式来解决这个问题。并且需要建立新的原子概念。因为规则的谓词只允许原子概念出现。

新概念: $Nconcept = Human \sqcup Animal$;

新规则: $Nconcept(y) \leftarrow Nconcept(x), parent(x, y), O(x), O(y)$;

很明显,通过对于新规则和新概念的应用,可以获得断言 $(Human \sqcup Animal)(Johnson)$ 。这种方法充分利用了描述逻辑中 UNION 概念定义符的表达能力。在下一段,将详细介绍抽取新规则需要满足的条件以及如何抽取新规则的算法。

4 结合 SHIQ(D)描述逻辑的 DL-safe 规则抽取

在这一段,首先给出方法的基本原理,随后给出一个快速的新规则抽取算法。

4.1 原理

文[9]中指出基于 SHIQ(D)的知识库可以被约简为析取的 DATALOG 规则。更进一步,DL-safe 规则可以与通过转换得到的析取程序进行合并,从而增加逻辑的表达能力,且不影响推理的可判定性。

假设存在以下两个 DL-safe 规则:

$$A_1 \vee A_2 \vee \dots \vee A_m \leftarrow B_1, B_2, \dots, B_n \quad (2)$$

$$C_1 \vee C_2 \vee \dots \vee C_l \leftarrow D_1, D_2, \dots, D_j \quad (3)$$

我们的目标就是得到与式(4)具有相同逻辑的合并结果,

并且使其仍然可以遵循 DL-safe 规则的表示形式。

$$A_1 \vee \dots \vee A_m \vee C_1 \vee \dots \vee C_i \\ \leftarrow (B_1, B_2, \dots, B_n) \vee (D_1, D_2, \dots, D_j) \quad (4)$$

下面给出一些定义以及算法细节。

定义 2(可合并谓词项) 假设存在两个谓词项 A 和 B , 如果满足以下两个条件, 则 A 和 B 为可合并的谓词项:

I. 谓词项 A 为形如 $C(x)$ 的谓词, 其中 $C \in N_c$, x 为实例变量。谓词项 B 为形如 $D(y)$ 的谓词, 其中 $D \in N_c$, y 为实例变量。

II. x 和 y 不是相异的常量实例。

面提出对于可合并谓词项的合并算法。算法首先根据定义 2 对谓词项的可合并性进行判断, 如满足以上定义, 则算法对谓词项进行合并, 否则, 算法停止。

谓词项合并算法:

输入: 谓词项 A 和 B

输出: 合并结果项

```
(1) begin
(2) if  $A$  和  $B$  不是可合并的谓词项, then 停止
(3) 使  $A = C(x)$ , 且  $B = D(y)$ 
(4) if  $x$  和  $y$  是实例常量, 且  $x = y$ 
(5) then 建立新概念  $E = C \cup D$ , return  $E(x)$ 
(6) else if  $x$  是一个实例变量, 用  $y$  替换  $x$ 
(7) then 建立新概念  $E = C \cup D$ , return  $E(y)$ 
(8) end
```

当两个谓词项在第(5)行和第(7)行被合并时, 需要利用描述逻辑中的 UNION 概念定义符建立一个新概念, 并作为合并结果返回。

4.2 规则抽取算法

在这一段, 介绍规则合并算法的具体细节。算法首先对输入规则前件中的概念谓词项进行搜索, 并根据搜索结果判断规则的可合并性以及概念谓词项合并的处理过程。随后对角色谓词项进行匹配, 并根据匹配结果判断规则是否可合并。最后生成新的规则并返回。

规则抽取算法:

输入:

规则 1: $(W_{A1} \vee \dots) \leftarrow (C_{A1}, \dots, C_{Am}, P_{A1}, \dots, P_{An}, O_{A1}, \dots)$

规则 2: $(W_{B1} \vee \dots) \leftarrow (C_{B1}, \dots, C_{Bn}, P_{B1}, \dots, P_{Bj}, O_{B1}, \dots)$

$C_A = \{C_{A1}, \dots, C_{Am}\} \subseteq N_c$, $C_B = \{C_{B1}, \dots, C_{Bn}\} \subseteq N_c$, $P_A = \{P_{A1}, \dots, P_{An}\} \subseteq R$, $P_B = \{P_{B1}, \dots, P_{Bj}\} \subseteq R$, $W_A = \{W_{A1} \vee \dots \vee W_{Ak}\} \subseteq N_p$, $W_B = \{W_{B1} \vee \dots \vee W_{Bk}\} \subseteq N_p$.

输出:

所抽取的新规则: $(W_1 \vee \dots) \leftarrow (E_1, \dots, E_l, P_1, \dots, P_h, O_1, \dots)$, $E = \{E_1, \dots, E_l\} \subseteq N_c$, $P = \{P_1, \dots, P_h\} \subseteq R$, $W = \{W_1 \vee \dots\} \subseteq N_p$.

Step1 初始化谓词项集合 $E = \Phi$, $P = \Phi$.

Step2 对谓词项集合 C_A 和 C_B 进行搜索, 如果在这些谓词项中存在两个不同的常量实例, 则停止 (这两个规则不能合并)。

Step3 如果在谓词项集合 C_A 和 C_B 中只存在一个常量实例 a , 合并分别来自集合 C_A 和 C_B 中的每个谓词项对, 比如, C_{A1} 和 C_{B1} , C_{A1} 和 C_{Bk} , 然后将每个谓词项对合并的结果谓词项加入集合 E 。将所有在这两个规则中出现的变量 (包括出现在概念谓词项和角色谓词项中的变量) 都用常量 a 代替。转向 Step5。

Step4 如果在谓词项集合 C_A 和 C_B 中没有出现常量实例。从

规则中的变量中随意选择一个变量 x 。用变量 x 来替换其他所有规则中出现的其他变量。然后合并分别来自集合 C_A 和 C_B 中的每个谓词项对, 并将每个谓词项对合并的谓词项结果加入谓词项集合 E 。转向 Step5。

Step5 对两个谓词项集合 $(P_{A1}, \dots, P_{An})$ 和 $(P_{B1}, \dots, P_{Bj})$ 进行匹配。如果对于每个谓词项都可以在另一个谓词项集合中找到一个相同的谓词项, 则匹配成功。将 $(P_{A1}, \dots, P_{An})$ 加入到谓词项集合 P 中。反之则停止 (这两个规则不可合并)。

Step6 将 W_A 和 W_B 中的谓词项直接加入到集合 W 中。并保留规则前件中所有的谓词项 O 项。返回新生成的规则。

很明显, 所有抽取出的新规则都仍然是 DL-safe 规则。首先, 算法对于每个合并的谓词项生成了新的概念, 所以在规则中只存在简单概念 (有明确定义的概念)。其次, 它保留了所有的谓词项 O 。此外, 在逻辑上新生成的规则为对原有规则的补充, 并不能代替原有规则。

算法复杂度: 两个规则的大小可以用如下参数进行表示: m (第一个规则前件中的概念谓词项的数量), n (第一个规则前件中角色谓词项的数量), i (第二个规则前件中的概念谓词项的数量), j (第二个规则前件中角色谓词项的数量)。合并两个规则的时间复杂度为 $\Omega(m * i + n * j)$, 在多项式范围内。这说明当规则中所包含的谓词项数量很大时, 算法仍然会拥有很好的性能。

结论 文[9]提出了一种可判定的合并描述逻辑 SHIQ (D) 与 DL-safe 规则的方法。它允许描述逻辑中的概念谓词项和角色谓词项同时出现在规则中, 而且, 这些谓词项同样可以出现在规则后件中。因为 DL-safe 规则不允许在前件中出现逻辑析取运算符, 我们的算法利用描述逻辑中 UNION 概念定义符的表达能力来满足在某些规则前件中表达析取含义的需求。这样可以使规则直接应用在从知识库 KB 中得到的信息上。算法抽取的新规则仍然为 DL-safe 规则, 所以用我们的方法抽取的新规则不影响复合逻辑的可判定性。

致谢 感谢德国 Karlsruhe 大学 FZI 信息技术研究中心的 Boris Motik 先生对于一些疑难问题的回答。

参考文献

- Connolly D, et al. DAML + OIL (March 2001) reference description. W3C Note, 2001. <http://www.w3.org/TR/daml+oil-reference>, 2001, 12
- McGuinness D L, Harmelen F v. OWL Web ontology language overview. W3C Recommendation, 2004. <http://www.w3.org/TR/2004/REC20040210/>, 2004, 2
- Borgida, Alexander. On the relationship between description logic and predicate logic. In: Proc. of CIKM-94, 1994
- Brachman R J, Brogida A, McGuinness D L, Patel-Schneider P F, Resnick L A. Living with CLASSIC: When and how to use a KL-ONE-like language. In: Sowa, John, ed. Principles of Semantic Networks. Morgan Kaufmann, San Mateo, CA, 1991. 401~456
- MacGregor R M. A deductive pattern matcher. In: Proc. of AAAI-88, 1988
- Donini F M, Lenzerini M, Nardi D, Schaerf A. AL-log: Integrating Datalog and Description Logics. J. of Intelligent Information Systems, 1998, 10(3): 227~252
- Alon Y L, Marie-Christine R. CARIN: a representation language integrating rules and description logics. In: Proc. of ECAI-96, 1996
- Horrocks I, Patel-Schneider P F. A Proposal for an OWL Rules Language. In: Proc. of the Thirteenth Int. World Wide Web Conf. (WWW 2004). ACM, 2004
- Motik B, Sattler U, Studer R. Query Answering for OWL-DL with Rules. To be published in Proc. of the 3rd Intl. Semantic Web Conf. (ISWC2004), 2004