

# 带值限制的本体描述语言研究

李志平<sup>1</sup>, 孙 瑜<sup>2</sup>

LI Zhi-ping<sup>1</sup>, SUN Yu<sup>2</sup>

1. 云南师范大学 现代教育技术中心, 昆明 650092

2. 云南师范大学 计算机科学与信息技术学院, 昆明 650092

1.Center of Modern Education Technology, Yunnan Normal University, Kunming 650092, China

2.Institute of Computer Science and Information Technology, Yunnan Normal University, Kunming 650092, China

E-mail: sunyu\_km@hotmail.com

**LI Zhi-ping, SUN Yu. Research of ontology description languages with value restrictions. Computer Engineering and Applications, 2009, 45(30): 143-145.**

**Abstract:** Based on F-logic and O-logic, a formal description language of ontologies with value restrictions is proposed. An ontology includes class frames, slot frames, class-slot frames, and individual frames. Value restriction specification on each class is discussed in ontologies. In slot frames, general value restrictions of slots are described. For each slot and each class, there is a class-slot frame representing the specific value restrictions of the slot when describing the class. After giving the formal syntax of ontologies, the inheritance mechanism of ontologies are discussed, and a sample ontology is given to illustrate the inheritance in ontologies.

**Key words:** ontology; value restriction; knowledge representation; inheritance mechanism

**摘 要:** 基于 F-逻辑和 O-逻辑, 提出了一种带值限制的本体形式表示语言。一个本体包括类框架、槽框架、类槽框架以及个体框架。本体通过槽的框架来说明对该槽的一般值限制。本体中的一个类和一个槽对应着一个类槽; 并且通过类槽的框架来说明用该槽来对这个类进行描述时的特定的值限制。在给出了本体的形式语法后, 讨论了本体的继承机制; 并且通过具体示例来对本体以及继承机制进行阐述。

**关键词:** 本体; 值限制; 知识表示; 继承机制

**DOI:** 10.3778/j.issn.1002-8331.2009.30.044 **文章编号:** 1002-8331(2009)30-0143-03 **文献标识码:** A **中图分类号:** TP18

## 1 引言

本体是一个概念化的明确说明(an explicit specification of a conceptualization)<sup>[1-3]</sup>。McGuinness<sup>[4]</sup>指出具有特性: 有限受控的词汇, 对类和术语关系的无歧异解释, 以及类之间的严格分层结构的规范说明(specification)可以被看作是一个简单的本体; 以及特性: 每个类上的性质说明, 本体中包括个体, 以及每个类上的值限制说明对一个本体来说是典型的。

在每个类(class)上的值限制说明可以在本体中进行描述, 也可以在本体之外进行描述。如果值限制在本体中进行描述, 那么可以很方便地在本体中相对于值限制对该本体的一致性(consistency)进行检验。事实上, 值限制与类是相关的。当用槽(slot)来对不同的类进行描述时, 该槽的性质是不同的。例如: 一般情况下, 槽年龄有最小值 0 和最大值  $\infty$ 。然而, 用槽年龄来对人这个类进行描述时, 其最小值为 0, 最大值为 200。因此, 定义一个类槽(class-slot)人\_年龄, 并且在人\_年龄的框架(frame)中表示用年龄来描述人时的特殊的值限制。

基于 F-逻辑和 O-逻辑<sup>[5-7]</sup>, 以及在构建本体的过程中遇到的具体问题, 提出了一种带值限的本体描述语言。本体中包括 4 类框架: 类框架、槽框架、类槽框架以及个体框架。本体通过槽的框架来说明对该槽的一般值限制; 同时, 通过类槽框架来描述该槽在类上的特定的值限制; 而在 F-逻辑和 O-逻辑中没有槽框架和类槽框架, 并且在 O-逻辑中只讨论了值类型这个值限制。对多重继承(multiple inheritance), 允许类框架中的槽从该类的多个超类中继承缺省值, 并且在继承过程中对该槽的缺省值进行修正。从下面所给出的例 1, 可以看出这种类型的修正是必要的。然而, 在 O-逻辑中, 类框架中的槽只能从该类的具有最高优先序的那个超类来继承缺省值。

提出了一种本体的形式表示语言。本体具有 McGuinness<sup>[4]</sup>所指出的简单本体所具有的特性, 同时, 还具有本体应有的所有典型特性。在本体的表示中, 对值限制, 例如: 槽和类槽的最大值(maximum value), 值类型(value type)等进行了描述。本体的继承机制考虑了覆盖(overriding), 修正(revising)和冲突消

**基金项目:** 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60903131); 云南省自然科学基金(the Natural Science Foundation of Yunnan Province of China under Grant No.2004F0017Q, No.2005F0022Q, No.2009ZC052W); 云南省教育厅重点基金(the Key Project Foundation of the Education Bureau of Yunnan Province under Grant No.07Z10661)。

**作者简介:** 李志平(1972-), 男, 副教授, 主要研究领域: 人工智能、网络工程; 孙瑜(1974-), 女, 博士, 教授, 主要研究领域: 智能信息处理。

**收稿日期:** 2008-06-11 **修回日期:** 2008-09-12

解(conflicts resolution)。

## 2 本体的语法

在本章中,给出本体的形式语法。假设存在下列两两不交集的集合:

- (1)类的名称(记为 $c, c_1, \dots$ )的集合 $C$ ;
- (2)槽的名称(记为 $s, s_1, \dots$ )的集合 $S$ ;
- (3)子槽(sub-slot)的名称(记为 $sl, sl_1, \dots$ )的集合 $M$ ;
- (4)个体的名称(记为 $t, v, \dots$ )的集合 $I$ ;其中集合 $C$ 包括:顶层(top class,所有的类的超类);数值类(自然数,整数,实数等);字符串类;个体类以及符号类。集合 $I$ 包括所有类的所有实例。

接着,从类框架,槽和类槽框架,以及个体框架这三部分来介绍本体的语法。

### 2.1 类框架

对每一个类都有一组用来对其进行描述的槽的集合。在类框架中,槽有两种类型的值:确定值和缺省值(default value)。在0-逻辑中,用来描述类的槽只有缺省值。

为了描述类和类之间的从属(subsumption)关系,引入直接子类关系 $isa$ ,并且用 $isa^*$ 表示 $isa$ 的传递闭包。假定子类关系是无循环的。同时,定义超类断言 $c isa c_1, \dots, c_n$ ,它说的是 $c_1, \dots, c_n$ 是 $c$ 的直接子类,来表示类上基于 $isa$ 的层次结构。为了解决在多重继承问题,假定在 $c$ 的超类 $c_1, \dots, c_n$ 中从左到右有一个优先序 $<$ ,即, $c_1 < c_2 < c_3 \dots < c_n$ ,其中 $c_1 < c_2$ 表示 $c_1$ 比 $c_2$ 具有更高的优先序。

设 $v_1, \dots, v_m (m > 0)$ 为槽或子槽的值,定义槽值断言 $c[s=v_1, \dots, v_m]$ 以及缺省槽值断言 $c[s=_d v_1, \dots, v_m]$ ,它们分别表示用来描述类 $c$ 的槽 $s$ 具有确定值 $v_1, \dots, v_m$ 和具有缺省值 $v_1, \dots, v_m$ 。例如,鸟[腿的数目=2]是表示鸟有两条腿的槽值断言;哺乳动物[栖息地=\_d 陆地]是表示缺省的情况下哺乳动物生活在陆地上的缺省槽值断言。对缺省槽值断言 $c[s=_d v_1, \dots, v_m]$ ,假定在 $s$ 的缺省值 $v_1, \dots, v_m$ 中从左到右有一个优先序 $<$ ,即, $v_1 < v_2 < \dots < v_m$ ,其中 $v_1 < v_2$ 表示 $v_1$ 比 $v_2$ 具有更高的优先序。例如,对缺省值断言鸟[喜欢的食物=\_d 昆虫,种籽],有‘昆虫’ $<$ ‘种籽’,它表示昆虫比种籽具有更高的优先序,即,鸟更喜欢吃昆虫。

基于上述讨论,给出一个类的类框架的定义。设 $c[V_1], \dots, c[V_k]$ 为槽值断言; $c[DV_1], \dots, c[DV_p]$ 为缺省槽值断言( $k, p > 0$ ),那么 $c isa c_1, \dots, c_n [V_1; \dots; V_k; DV_1; \dots; DV_p]$ 是一个类框架。

### 2.2 槽和类槽的框架

对任意的类 $c$ 和任意的槽 $s$ ,定义类槽 $cs$ 的框架来说用 $s$ 来对槽 $c$ 进行描述时的特定的值限制。设 $CS = \{cs : c \in C, s \in S\}$ 为类槽的集合。对任意的槽 $s$ ,称 $cs$ 为其所对应的类槽。引入类槽的目的是为了区分对槽 $s$ 的一般值限制和对用来描述某个类 $c$ 的槽 $s$ 的特定的值限制。

槽和类槽是由子槽来进行描述的。假定在本体中子槽是原子的。集合 $M$ 包括下面的子槽:

- (1)定义域:对槽或类槽的作用范围进行限制。定义域的每一个值表示一个类;
- (2)值类型:对槽或类槽的值的类型限制。值类型的每一个值表示一个类;
- (3)最大取值个数和最小取值个数:分别表示槽或类槽的最大和最小取值个数。这两个子槽的每个值都是非负整数;
- (4)最大值和最小值:分别表示槽或类槽的最大和最小值;它们的每个值都是数。

定义子槽断言 $s[sl=v_1, \dots, v_m]$ 以及缺省子槽断言 $s[sl=_d v_1, \dots, v_m]$ ,它们分别表示用来描述槽 $s$ 的子槽 $sl$ 具有确定值 $v_1, \dots, v_m$ 和具有缺省值 $v_1, \dots, v_m$ 。类似的,可以定义类子槽断言 $cs[s=v_1, \dots, v_m]$ 。例如,腿的数目[值类型=整型]是用来表示槽腿的数目的取值类型是整型的子槽断言;

企鹅\_喜欢的食物[最大取值个数=3]是表示类槽企鹅\_喜欢的食物最多可以取3个值类子槽断言。设 $s[S_1], \dots, s[S_n]$ 为子槽断言和缺省子槽断言; $cs[S_1], \dots, cs[S_n]$ 为类子槽断言( $n > 0$ );那么 $s[S_1; \dots; S_n]$ 为一个槽框架; $cs[S_1; \dots; S_n]$ 为一个类槽框架。在类槽框架中,子槽只有确定值,而没有缺省值。

### 2.3 个体

为了描述个体和类之间的实例(instance)关系,引入直接实例关系:,并且 $t:c$ 表示 $t$ 是 $c$ 的直接实例。对任意的个体 $t$ ,假定它只是一个类的直接实例。

定义个体槽值断言 $t[s=v_1, \dots, v_m]$ ,它表示用来描述个体 $t$ 的槽 $s$ 具有确定值 $v_1, \dots, v_m$ 。设 $t[T_1], \dots, t[T_k] (k > 0)$ 为个体槽值断言,那么 $t:c[T_1, \dots, T_k]$ 是一个个体框架。用来描述个体的槽没有缺省值。

## 3 本体及其继承机制

在本章中,先给出本体的定义,然后讨论本体的继承机制。

### 3.1 本体

在本节中,给出本体的定义和一个示例本体。基于第2章的讨论,定义本体如下:

**定义1** 一个本体 $O$ 是由类框架、槽框架、类槽框架以及个体框架所组成的集合。

**例1** 一个简单的本体

类 鸟

{腿的个数=2;会飞=\_d T.;

喜欢的食物=\_d 昆虫,种籽}

类 企鹅 isa 鸟

{会飞=\_d F.;喜欢的食物=\_d 鱼}

槽 喜欢的食物

{定义域=鸟;值类型=字符串型;

最大取值个数=\_d 2;

最小取值个数=\_d 1}

槽 会飞

{定义域=鸟;值类型=布尔型;

最大取值个数=1;

最小取值个数=1}

槽 腿的个数

{定义域=鸟;值类型=整型;

最大取值个数=1;

最小取值个数=1;

最大值=100;最小值=0}

类槽 企鹅\_喜欢的食物

{最大取值个数=3}

个体 tweety:企鹅{ }

### 3.2 本体的继承机制

本节对本体的继承机制进行讨论。

在一个槽框架中,子槽 $sl$ 的确定值被继承到其对应的类槽作为该类槽中 $sl$ 的确定值;并且它的缺省值被继承到其对应的类槽作为该类槽框架中 $sl$ 的确定值,除非这些缺省值被这些类槽框架中 $sl$ 的确定值所覆盖。

**定义 2** 设  $P_1 = s[sl =_d v_1, \dots, v_m]$ ; 并且  $P_2 = c[sl = v'_1, \dots, v'_n](m, n > 0)$ , 那么  $P_2$  覆盖  $P_1$ 。

从直观上说, 定义在类槽  $cs$  上的关于子槽  $sl$  的类子槽断言覆盖定义在槽  $s$  上的关于子槽  $sl$  的子槽断言。在例 1 中, 企鹅\_喜欢的食物[最大取值个数=3]覆盖喜欢的食物[最大取值个数= $_d$  2]。

类似的, 在一个类框架中, 槽  $s$  的确定值被继承到其直接实例作为该实例框架中  $s$  的确定值; 并且它的缺省值被继承到其直接实例作为该实例框架中  $s$  的确定值, 除非这些缺省值被该实例框架中  $s$  的确定值所覆盖。

**定义 3** 设  $P_1 = t[s = v_1, \dots, v_m]$ ; 并且  $P_2 = c[s =_d v'_1, \dots, v'_n](m, n > 0)$  满足  $t : c$ , 那么  $P_1$  覆盖  $P_2$ 。

从直观上说, 如果个体  $t$  是  $c$  的直接实例, 那么定义在  $t$  上的关于槽  $s$  的个体槽值断言覆盖定义在  $c$  上的关于  $s$  的槽值断言。

在一个类框架中, 槽  $s$  的确定值被继承到其对应的子类作为该子类中  $s$  的确定值; 并且它的缺省值被继承到其对应的子类作为该子类中  $s$  的缺省值, 除非这些缺省值被该子类中  $s$  的确定值或者缺省值所覆盖, 或者被修正。

**定义 4** 缺省槽值断言的覆盖定义如下:

(1) 设  $P_1 = c[s = v_1, \dots, v_m]$ ; 并且  $P_2 = c'[s =_d v'_1, \dots, v'_n](m, n > 0)$  满足  $c \text{ isa } c'$ , 那么  $P_1$  覆盖  $P_2$ 。

(2) 设  $P_1 = c[s =_d v_1, \dots, v_m]$ ; 并且  $P_2 = c'[s =_d v'_1, \dots, v'_n](m, n > 0)$  满足  $c \text{ isa } c'$ , 那么  $P_1$  覆盖  $P_2$  当且仅当  $cs$  [最大取值个数= $k$ ], 并且  $k = m$ 。

从直观上说, 如果类  $c$  是类  $c'$  的直接子类, 那么定义在  $c$  上的关于槽  $s$  的槽值断言覆盖定义在  $c'$  上的关于槽  $s$  的缺省槽值断言; 同时, 定义在  $c$  上的关于槽  $s$  的缺省槽值断言覆盖定义在  $c'$  上的关于槽  $s$  的缺省槽值断言, 如果  $c$  是  $c'$  的直接子类并且类槽  $cs$  的最大取值个数与用来描述类  $c$  的槽  $s$  的缺省值的个数是相同的。在例 1 中, 鸟[会飞= $_d$  T.]被企鹅[会飞= $_d$  F.]所覆盖, 由于企鹅是鸟的直接子类, 并且企鹅\_会飞的最大取值个数为 1。

**定义 5** 缺省槽值断言的修正定义如下:

设  $cs$  [最大取值个数= $k$ ],  $P_1 = c[s =_d v_1, \dots, v_m]$ ; 和  $P_2 = c'[s =_d v'_1, \dots, v'_n](m, n > 0)$  满足 (1)  $c \text{ isa } c'$ , 和 (2)  $P_1$  没有覆盖  $P_2$ 。同时, 设  $u_1, \dots, u_p (p < n)$  为从  $v'_1, \dots, v'_n$  中选出的满足 (1)  $u_i <_d \dots <_d u_p$  以及 (2) 对任意的  $0 < i < p + 1, u_i$  不同于  $v_1, \dots, v_m$  中任何一个值的缺省值序列。于是, 将  $P_1$  修正为满足  $v^1 = v_1, \dots, v^m = v_m, v^{m+1} = u^1, \dots, v^{m+k} = u^k (k \leq p, m+k \leq k)$  的缺省槽值断言  $c[s =_d v^1, \dots, v^m, v^{m+1}, \dots, v^{m+k}]$ 。

从直观上说, 定义在  $c$  上的关于槽  $s$  的缺省槽值断言被修正以包含一部分用来描述  $c'$  的槽  $s$  的缺省值, 如果  $c$  是  $c'$  的直接子类并且类槽  $cs$  的最大取值个数大于描述类  $c$  的槽  $s$  的缺省值的个数。在例 1 中, 由于鸟[喜欢的食物= $_d$  昆虫, 种籽], 以及鸟\_喜欢的食物[最大取值个数=3], 因此, 通过对缺省值断言企鹅[喜欢的食物= $_d$  昆虫, 种籽]进行修正可以得到企鹅[喜欢的食物= $_d$  鱼, 昆虫, 种籽]。在此, 如果鸟\_喜欢的食物[最大取值个数=2], 那么可以得到企鹅[喜欢的食物= $_d$  鱼, 昆虫]。

类似的, 对多重继承, 在子类  $c$  有多个直接超类  $c_1, \dots, c_n$  的情况下, 可以根据  $c_1, \dots, c_n$  之间的优先序  $<_c$  来定义对缺省槽值断言的修正。

在例 1 中, 由于鸟没有超类, 因此, 在使用本节所给出的继承机制后, 鸟的框架和原来是一样的。然而, 企鹅有超类鸟, 在

使用继承机制后, 企鹅的框架如下:

类 企鹅 *isa* 鸟  
{腿的个数=2; 会飞= $_d$  F.;  
喜欢的食物= $_d$  鱼, 昆虫, 种籽}。

同样, 在使用继承机制后, 类槽企鹅\_喜欢的食物的框架如下:

槽 企鹅\_喜欢的食物  
{定义域=鸟; 值类型=字符串型;  
最大取值个数=3;  
最小取值个数=1};

个体 tweety 的框架变成:

个体 tweety: 企鹅  
{腿的个数=2; 会飞=F.;  
喜欢的食物=鱼, 昆虫, 种籽}。

## 4 结论

提出的带值限制的个体描述语言有以下两个特点: (1) 通过槽框架和类槽框架来在个体中讨论值限制; 以便在个体中相对于值限制来对个体的一致性进行检测。例如, 由于在类槽  $cs$  的框架中说明了用  $s$  来对类  $c$  进行描述时  $s$  的最大取值个数, 因此, 可以在个体中检测  $s$  在描述  $c$  时的取值个数是否满足该值限制; (2) 在个体的继承机制中, 考虑了根据最大取值个数和一个类的直接超类之间的优先序  $<_c$  对缺省值断言的修正。

另外, 在个体中所讨论的槽和子槽的缺省值以及类槽框架在 Web 个体语言 OWL<sup>[9]</sup> 中都没有进行阐述。

尽管目前在个体表示中公理还没有被广泛使用, 但由于在寻找信息时新的知识会被推导出来; 以及个体的不一致性会被检测出来等, 公理将成为个体表示中的一个重要因素。因此, 基于一阶逻辑在个体中建立公理来限制槽的取值, 验证个体的一致性以及得出新的信息是下一步的研究课题之一。此外, 个体中的推理规则, 断言在模型中的可满足性等也是进一步的研究课题。

## 参考文献:

- [1] Guarino N, Welty C. Ontological analysis of taxonomic relationship[C]// Proc of International Conference on Conceptual Modeling, 2000: 210-224.
- [2] Smith B. Ontology and information systems[EB/OL]. (2001). [http://ontology.buffalo.edu/ontology\(PIC\).pdf](http://ontology.buffalo.edu/ontology(PIC).pdf).
- [3] Gruber T. A translation approach to portable ontology specifications[J]. Knowledge Acquisition, 1993, 5(2): 199-220.
- [4] McGuinness D. Ontologies come of age[M]// Fensel D. The Semantic Web: Why, What, and How.[S.l.]: MIT Press, 2002.
- [5] Cao C. Progress in the development of national knowledge infrastructure[J]. Journal of Computer Science & Technology, 2002, 17: 1-16.
- [6] Kifer M, Lausen G, Wu J. Logical foundations of object-oriented and frame-based languages[J]. Journal of ACM, 1995, 42: 741-843.
- [7] Kifer M, Wu J. A logic for programming with complex objects[J]. Journal of Comput Syst Sci, 1993, 47: 77-120.
- [8] Liu M, Dobbie G, Ling T. W. A logical foundation for deductive object-oriented databases[J]. ACM Transactions on Database Systems, 2002, 27: 117-151.
- [9] McGuinness D L, van Harmelen F. OWL Web ontology language: Overview[EB/OL]. <http://www.w3.org/TR/owl-features/>.