

Visual Prolog 的接口作用域限定机制

田野, 雷英杰, 吉波, 孙晨
(空军工程大学 导弹学院, 陕西 三原 713800)

摘要:针对 Visual Prolog 中的接口这一特殊作用域,在考察其限定机制基本问题的基础上,详细分析它所特有的支持限定和开放限定等机制的功能与作用。首先阐述了接口与对象的基本概念,然后分析了开放限定和支持限定的作用机理,最后通过实例分析,揭示了接口作用域限定机制的本质特性和应用机理。

关键词:专家系统;编程语言;逻辑程序设计;Visual Prolog

中图分类号: TP182 **文献标识码:** A **文章编号:** 1009-3516(2005)06-0060-03

Visual Prolog 的接口(interface)是一种特殊的作用域^[1-2],其限定机制包含支持限定和开放限定^[3],本文系统分析这些机制的功能与作用机理。

1 接口与对象

Visual Prolog 中的接口是一种命名的对象类型^[4],它有一个名字且定义了一组命名对象谓词。在接口中声明的所有谓词都是接口对象类型的对象成员。接口也是一个全局作用域,在其中可以定义常量和论域。所以,在一个接口中所定义的常量和论域不是该接口指示的类型的一部分或具有该类型的对象。这样的论域和常量可以通过限定接口名或使用开放限定由其它作用域引用。其作用域限定(Scope Qualifications)只能是支持限定或开放限定两种类型。一个接口可以支持另外的接口。如果在构造器尾部的接口名存在,则必须与构造器开始的接口名相同。

在接口这个全局作用域中所使用的段只能是常量段(constants Section)、论域段(domains Section)、谓词段(predicates Section)、接口谓词段(predicates From Interface)、条件段(conditional Section)这样的类型。所有包含在条件段的部分也必须是这些类型的段。

如果一个接口没有明确地支持任何其他接口,那么它就隐含地支持内部预定义的接口对象。对象是一个空接口,即它不包含谓词的实现代码等内容。

2 开放限定

开放限定(Open Qualification)可使引用类层次的实体更加方便。开放段把一个作用域名代入另一作用域,以使这些名字可以在不受限制的情况下被引用。

开放限定符对于对象成员的名字不起作用,因为它们在任何情况下只能借助于一个对象进行访问。但是类成员名、论域、算符和常量可以在无限定符的情况下进行访问。开放段只会在它们所出现的作用域内产生影响,在一个类声明中的开放段不会影响到它的类实现。

当名字以这样的方式被带进一个作用域时,可能会使有些名字变得不明确。

收稿日期:2005-03-23

基金项目:国防科技预研基金资助项目(51406030104DZ0120)

作者简介:田野(1984-),女,山西洪洞人,硕士生,主要从事智能信息处理研究;

雷英杰(1956-),男,陕西渭南人,教授,博士生导师,主要从事智能信息处理与智能系统研究。

Visual Prolog 的名字只能在程序的某些区域内使用。这样的区域叫做名字的作用域。一个接口定义、一个类声明和一个类实现都是作用域。当类的构造器被调用和当该作用域的局部变量被初始化时,作用域决定了名字的可见性。作用域也决定了一个不表示全局作用域元素名字的“寿命”。作用域有4种:

1) 局部(Local)或私有作用域。一个在类实现中声明的名字只能在该类的实现中访问。这些名字有在实现中声明的论域、事实、谓词、常量和子句变量。

2) 类(Class)作用域。类成员名有类作用域,类成员可通过类名限定进行访问,例如 `className::ClassScopeName`。

3) 对象(Object)作用域。对象成员名有对象作用域。它们仅能在对象存在时被访问。对象成员仅可通过引用(在类实现内它被隐含)一个经限定的对象标识符进行访问,例如 `objectID: ObjectScopeName`。

4) 全局 Global 作用域。接口名和类名有全局作用域。全局作用域限定符“::”可用在这些名字前。预定义的论域(字符,字符串,符号,整数,无符号数,实数,二进制数,指针,布尔数,事实数据库)、常数和谓词也可以是全局的,它们也可以用“::”进行限制。

3 支持限定

支持限定(Supports Qualification)描述接口列表,这些接口由类实现私有地给予支持^[5]。支持限定只能在接口定义和类实现(class Implementation)中使用。支持限定用于以下两种情况:

1) 指定一个接口 A 扩展到另外一个接口 B,因此,对象类型 A 是对象类型 B 的子类型。

2) 声明一个特定类的对象“私下”具有比一个指定作为构造类型的类更多的对象类型。

支持限定具有传递关系:如果接口 A 支持接口 B,并且接口 B 支持接口 C,那么接口 A 也支持接口 C。就功能而言,一个接口直接或间接地支持一个特定的接口一次还是多次并没有差别,但是对于对象而言却有极大的不同。

当支持用于一个类的实现中时,结果是“This”不但可以与构造类型一起使用,而且还能与任何私有的所支持的对象类型一起使用。

如果接口有冲突的谓词,则它们就不能在一个支持限定内一起使用。如果谓词具有相同的名字和变元数,具有不同的原始接口,那就是冲突的。一个谓词的原始接口是该谓词文字上被声明的谓词的接口,同时它反对由支持限定间接声明的接口。因此如果同一接口在支持链中出现两次或更多次,它也不会发生冲突。

接口被构造在支持层次上,该结构是以接口对象为根的半格(semi-lattice)。如果一个对象有接口表示的类型,那么它也具有了任何支持接口的类型。因此,支持层次也是一种类型层次。也可以说对象支持接口。如果接口被命名为 X,那么就说对象是 X,或称 X 对象。

4 实例分析

考虑下面的定义和声明:

```
interface aaa
  predicates
    insert : (integer X) procedure (i)
end interface
interface bbb
  supports aaa
  predicates
    insert : (integer X, string Comment) procedure (i,i)
end interface
interface cc
  supports aaa
  predicates
```

```

    extract : () - > integer procedure ()
end interface
interface dd
    supports aaa, bbb, cc
    predicates
        extract : (string Comment) - > integer procedure (i)
end interface

```

这里是在 dd 中找到的所有谓词的列表(通过深度遍历查找^[6-7]):

```

predicates
    insert : (integer) procedure (i). % dd - > aaa
    insert : (integer) procedure (i). % dd - > bbb - > aaa
    insert : (integer, string) procedure (i,i). % dd - > bbb
    insert : (integer) procedure (i). % dd - > cc - > aaa
    extract : () - > integer procedure (). % dd - > cc
    extract : (string) - > integer procedure (i). % dd

```

其中有些谓词是一样的,因此总的来说,dd 将包含下列成员:

```

predicates
    insert : (integer) procedure (i). % 原始接口: aaa
    insert : (integer, string) procedure (i,i). % 原始接口: bbb
    extract : () - > integer procedure (). % 原始接口: cc
    extract : (string) - > integer procedure (i). % 原始接口: dd

```

考察下面的接口:

```

interface aaa
    predicates
        insert : (integer X) procedure (i).
end interface
interface bbb
    predicates
        insert : (integer X) procedure (i).
end interface
interface cc
    supports aaa, bbb % 相冲突的接口
end interface

```

接口 cc 是非法的,因为在 aaa 中所支持的 insert/1 以 aaa 为源,而在 bbb 中所支持的 insert/1 则是以 bbb 为源的。

5 结束语

针对 Visual Prolog 中的接口这一特殊作用域,在考察其限定机制基本问题的基础上,详细分析它所特有的支持限定和开放限定等机制的功能与作用。首先阐述了接口与对象的基本概念,然后分析了开放限定和支持限定的作用机理,最后通过实例分析,揭示了接口作用域限定机制的本质特性和应用机理。

参考文献:

- [1] 雷英杰,邢清华,孙金萍,等. Visual Prolog 智能集成开发环境评述[J]. 空军工程大学学报(自然科学版), 2002, 3 (5): 39-43.

(下转第 83 页)

4 结束语

以上深入分析了 FOIL 算法,讨论了利用 Visual Prolog 实现基于 FOIL 的一阶规则集学习器的设计方法。对于其他以 FOIL 算法为基础的改进算法,可以本文所设计的学习器为核心,进行适当改动来得到对应的学习器。

参考文献:

- [1] QUINLAN J R. Learning logic Definitions From Relations[J]. Machine Learning, 1990,5:239 - 266.
- [2] MITHELL T M. 机器学习[M]. 北京:机械工业出版社,2003.
- [3] 雷英杰. Visual Prolog 语言教程[M]. 西安:陕西科学技术出版社,2002.
- [4] 雷英杰. Visual Prolog 编程、环境及接口[M]. 北京:国防工业出版社,2004.
- [5] 雷英杰,王 涛,赵 晔. Visual Prolog 的回溯机制分析[J]. 空军工程大学学报(自然科学版),2004,5(5):80 - 84.
- [6] 雷英杰,王宝树,赵 晔,等. Visual Prolog 的搜索控制机制分析[J]. 计算机科学,2005,32(3):39 - 43.

(编辑:田新华)

Design of Learner for Sets of First - Order Rules Based on FOIL

XU Tong, ZHANG Li, XIE Bo

(The Missile Institute, Air Force Engineering University, Sanyuan, Shaanxi 713800, China)

Abstract: After a thorough analysis of the essence of FOIL algorithm, the paper discusses the way of programming a learner by using Visual Prolog and based on this algorithm, and at the same time demonstrates the key code and experiment results of this method in detail.

Key words: FOIL; first - order rules; sequential covering; horn clause; Visual Prolog

(上接第 62 页)

- [2] 雷英杰,张 雷,邢清华,等. Visual Prolog 语言教程[M]. 西安:陕西科学技术出版社,2002.
- [3] 雷英杰,邢清华,孙金萍,等. Visual Prolog 编程、环境及接口[M]. 北京:国防工业出版社,2004.
- [4] 雷英杰,王 涛,赵 晔. Visual Prolog 的回溯机制分析[J]. 空军工程大学学报(自然科学版),2004,5(5):80 - 84.
- [5] 雷英杰,邢清华,王 涛,等. 人工智能程序设计(面向对象语言)[M]. 北京:清华大学出版社,2005.
- [6] 雷英杰,王宝树,赵 晔,等. Visual Prolog 的搜索控制机制分析[J]. 计算机科学,2005,32(4):39 - 43.
- [7] 雷英杰,华继学,徐 彤,等. Visual Prolog 截断机制对回溯的作用机理[J]. 计算机工程,2005,31(8):39 - 43.

(编辑:田新华)

On the Mechanisms of Interface Score Qualifications in Visual Prolog

TIAN Ye, LEI Ying - jie, JI Bo, SUN Chen

(The Missile Institute, Air Force Engineering University, Sanyuan, Shaanxi 713800, China)

Abstract: To the particular scores of interfaces in Visual Prolog, the specific functions and effects of mechanisms of support qualifications and open qualifications are analyzed and investigated in detail on the basis of study of principal issues of mechanisms of these qualifications. First, the fundamental notions of interfaces and objects are expatiated. Then, the functional mechanisms of support qualifications and open qualifications are exposed with a detailed analysis. Finally, the essential characteristics and application mechanisms of the interface score qualifications are exposed and illustrated through an instance.

Key words: AI; expert systems; programming language; programming in logic; Visual Prolog