

# 人工智能 Prolog 语言的基本概念

项湜伍

(广西商业专科学校)

通常的过程式的语言,如 Basic, Fortran, Pascal 语言在解决问题时,程序员给出一系列计算机要执行的计算步骤,即告诉计算机“如何做”。

而人工智能的语言——非过程式语言(Prolog),它不必告诉计算机如何做,它能依靠自动搜索、模式匹配和回溯来求解问题,所以称为超高级语言——人工智能的语言。此语言能描述问题本身外,它还有下面三个特点:

- 1) 数据和项的程序结构统一。它的程序和项是以项为基本单位,并且都是树型结构。
- 2) 自动完成模式匹配和回溯,这些是人工智能系统中最常使用的、最基本的操作。
- 3) 递归是它的重要特点。由于此特性,一个大的数据结构能由一个小小的程序来处理。

所以在智能程序,自然语言处理、专家系统、定理证明、智能数据库等方面,Prolog 语言有它十分重要的用途,日本已把它作为第五代计算机的核心语言。

## 1、简单的例子

设有一有向图,如图 1 所示。

若用  $connected(x, y)$  表示  $x$  到点  $y$  的一条有向边。则事实如下:

$connected(a, b).$   
 $connected(a, d).$   
 $connected(d, c).$   
 $connected(b, c).$

现定义两点之间有通道的概念。点  $x$  到点  $y$  有通路是指:

- ①  $x$  到  $y$  有一条有向边
- ② 存在一点  $z$ ,  $x$  到  $z$  有一条有向边,并且  $z$  到  $y$  有一条有向边。这二条规则用

Prolog 规则来描述。

$Path(x, y): if\ connected(x, y)$

$Path(x, y): if\ connected(x, z)\ and\ Path(z, y).$

第一条规则就是说,如果  $x$  到  $y$  有一条有向边,由  $x$  到  $y$  有通路。第二条规则表示,如果  $x$  到  $z$  有一条有向边,且  $z$  到  $y$  有通路,则  $x$  到  $y$  有通路。

输入了上述事实 and 规则以后,就可以向计算机询问图中点之间的关系。它会自动地给出答案。

goal:  $\neg path(a, b)$   $a$  到  $b$  有通路吗?

yes	有
goal: $\text{-path}(b, a)$	b 到 a 有通路吗?
no	没有
goal: $\text{-path}(a, c)$	a 到 c 有通路吗?
yes	有
goal: $\text{-path}(d, X)$	d 到哪一点有通路? (这里 X 是大写, 表示变量)
X=c	d 到 c 有通路?
goal: $\text{-path}(b, X), \text{path}(d, X)$	是否有一点 X 存在, 使得 b 和 c 都有通路到这一点 X 存在
X=c	这一点 c.

从上面例子中, 可以看出仅仅用事实、规则和询问三个基本语句, 即可组成人工智能语言。

## 2. 事实

从上面可以看到, 为了告诉 Prolog 从点 a 到 b 有一条有向边这一事实, 写了 connect (a, b)。这个事实由二个对象 a 到 b 以及一个称为 connected(连接)的关系所组成。

事实的一般形式为:

- 21 首先写关系, 接着是括在一对圆括号内的若干对象(变元), 每个对象之间用逗号隔开。
- 22 所有的对象和关系名必须以小写字母开始。如 b, a, connected 等。
- 23 在事实的末尾必须用“.”结束。

在 prolog 语言中, 关系 connected 被称为谓词, 括号中的 a, b 被称为变元。要注意圆括号中的书写次序。

下面是一些事实的例子及它们的解释。

on (book, desk)	书在桌子上面
owns(john, book)	john 有书
play(jane, jin, badminton)	jan 和 jin 打羽毛球
valuable(gold)	金是值钱的

当这个名字, 一经使用, 就指一个特定的对象。我们使用名字时, 必须决定如何来解释这个名字。只要在一给定问题中所使用的名字解释一致, 不会有任何问题。

## 3. 规则

假如我们叙述这样的事实: jan 喜欢所有运动。表示这一事实的一种办法是写出下列所有事实:

likes(jan, football).	jan 喜欢足球
likes(jan, swim)	jan 喜欢游泳

likes ( jan , baseball )      jan 喜欢棒球

.....

这里 football, swing baseball 等等都是运动中的一个项目, 如果有几十个运动项目, 显然这种表示方法是不方便的。因此表示 jan 喜欢所有运动的另一种方法是把它说成: “jan 喜欢任一对象, 只要这对象是运动项目。” 这个事实可以用下面的规则表示。

likes (jan, X): if sport(X)

它的含义是: 如果 X 是运动项目, jan 就喜欢它。

(注意: 这里 X 是大写字母 以大写字母开头的字母、数字串代表变量)

也就是说, 当描述一个事实依赖于其它一组事实时, 则用规则来表示。我们用“如果, 则”来表示。

例如: 例 1、如果 X 是一动物, 且有羽毛, 则 X 是鸟。

bird(X): if animal(X) and feathers(X)

例 2、如果 X 是女的, 则 X 和 y 有相同的父母亲, 则 X 是 y 的姐妹。

sister(X, y): if female (X) and parents(X, F, M) and parents(Y, F, M)

这里 female(X) 表示 X 是女的, parents(X, F, M) 表示 X 的父母是 F, M。这个规则意义是: 如果 X 是女的, 并且 X 的父母是 F, M, 而且 y 的父母也是 F, M, 则 X 和 y 是姐妹。

通常一个谓语句(如 likes, sister)能用事实和规则的混合来定义。例如

likes(john, food)

likes(john, X), if person(X)

这些事实和规则称为子句—— clause。以后我们每当涉及一事实和一规则时, 用子句来统称。

我们再给一些规则的例子:

例 1、john 喜欢任何讲话和游泳的人。

likes ( john , X ): if person ( X ) likes ( X , talk ) and likes ( X , swim )

例 2、父母关系、(外)祖父母关系

parents ( X , Y ): if mother ( X , Y )

parents ( X , Y ): if father ( X , Y )

grand parent(X, Y): if parent ( X , Z ) and parent ( Z , Y )

### 4、询问

上述事实、规则称为数据库, 它和询问一齐构成了人工智能语言的三大语句: 事实、规则、询问。

用户在任何时候可以提问题。一般先把事实和规则提供给计算机, 然后再询问问题。

询问看上去象事实, 除了多一个 goal: 以外。

实际上当打入询问: goal: connected(a, b) 后, 它到已取得的事实和规则的数据库去找是否有这样一个事实和规则。其寻找方法是把询问的目标去匹配数据库中的 connected 子句。寻找过程是从上到下, 从左到右的扫描数据库。如果问题中的目标和数据库的某一子句匹配成功, 则表示该目标成功, 回答 yes, 否则回答 no。

仅仅询问这样的问题, 获得 yes 和 no 的回答是不够的。询问如 john 有什么东西, 谁与谁打羽毛球等问题。

例如, 有这样一个事实, 即有一个电话数据库

clauses

```
reference("Albert", "01-123456").
reference("Betty", "01-569767").
reference("Carol", "01-267400").
reference("Dorothy", "01-191051").
```

询问 Betty 的电话号码为多少?

goal: refernce("Betty", X)

当收到这个问题后, 它便到先前得到的数据库中找相应的事实, 在找到目标的匹配事实 refernce("Betty", "01-569767"). 后, 它们的变元一一对应, X 被实例化为 01-569767, 因此允许一个未实例化的变量匹配任何一个对象。

## 5. 基本语法

为了提高编译目标质量, 还要加上一系列说明性的语句。

### 5.1 域说明语句 domains

要规定谓语中参数所属域。

例如: domains

```
person, activity =symbol
```

规定了 person 和 activity 为符号类型。

谓语参加可以取以下五种类型:

基本类型: 字符(char)、整数(integer)、实数(real)、  
符号(symbol)、字符串(string)

多复合域类型

表类型: < 名字 > \*

指针类型: reference < 右部 >

文件类型: file = < 名字表 >

它的语法规则如下:

```
< domain 语份 > :: = [global] domains { < 域定义 > }
< 域定义 > :: = < 名字表 > = [reference] < 右部 > | file = < 名字表 >
< 右部 > :: = < 域基本类型 > | < 名字 > * | < 多复合项类型 >
< 域基本类型 > :: = char | integer | symbol | string | real
```

### 5.2 谓词说明语句(predicates) 和数据库语句(data base)

谓词可以是一个或多个子句的定义。还必须规定谓词中参数所属域。

例如: domains person, activity=symbol

```
predicates likes ( person, activity)
```

它规定 likes 带二个参数, 这二个参数属于一个符号域, 它的语法规则如下:

```

< data base 部分 > :: = database { < 谓语句定义 > }
< predicate 部分 > :: = predicates { < 谓语句定义 > }
                        | global predicates { < 全局谓语句定义 > }
< 谓语句定义 > :: = < 名字 > [ ( [ 名字表 ] ) ] [ , ] ]

```

全局谓语句说明及全局谓语句定义, 用于与其它语言(如 C, Pascal, Asm)连接中。

一个完整的 Prolog 程序的基本结构形式为:

```

domains /* 域说明语句 */
predicates /* 谓语句说明语句 */ ] turbo prolog 语言增加部
clauses: /* 语句(事实和规则) */

```

## 6、应用实例

我们在看实例以前, 还补充二个定义:

### 6.1 谓词和原子:

```

< 谓词名 > :: = < 原子 >
< 原子 > :: = < 小写字母 > { < 尾部 > | < 谓词符号 > | "任何字符串" }
< 小写字母 > :: = a, b, c, ..., z

```

6.2 汽车咨询系统。这一个十分简单的汽车咨询系统, 但是它是程序是完整的, 包括域说明语句。例:

```

domains
    brand, color = symbol
    age, price   = integer
    mileage      = real

predicates
    car(brand, mileage, age, color, price).

clauses
    car(chrysler, 130 000, 3, red, 12 000).
    car(ford, 90 000, 4, gray, 25 000).
    car(datsun, 8000, 1, red, 3000).

```

在这个程序中, 谓词 car 的参数中, 汽车寿命和价格的取值域为整数, 汽车的行驶公里数为实型数, 汽车的牌子和颜色为符号型。

询问时要注意它的领域。

### 6.2.1 goal, car (M, O, Y, B, 25 000)

即询问价格为 25 000 元的车辆是什么型号? 颜色? 寿命? 行驶公里数?

对此目标回答为:

```

M = ford (福特型号), O = 9000 (行驶 9000 km)
Y = 4 (使用寿命为 4 年), B = gray (颜色为灰色)

```

6.2.2 如果询问时, 领域类型有错, 则它拒绝回答。

goal: car (12, 8000, 1, red, 30 000).

出错, 由于领域汽车型号(符号型)使用了数字型。

6.2.3 复合询问目标: 如何以询问价格小于 25 000 为什么型号车? goal: car (M, —, —, —, —) and cost < 27 000.

回答 M = chrysler 型号为这种牌子。

在这里用了无名变量\_ , 即不需要了解和询问的变量

6.3 家族关系咨询系统。下面是一个完整的家族关系咨询系统。它给出了父母、兄弟、姐妹、叔父、祖(外)母关系。

首先它给出了域值 person 为符号型。

另外它给出了谓词: male(person)

female(person)

.....

接着它给子句: 子句中事实为

male(alan) alan 男

female(fay) fay 女

mother(alan, sally) alan 的母亲为 sally

father(fay, bob) fay 的父亲为 bob

子句的规则为下:

BROTHERS(X, Y) if male(y) and parent(x, p) and parent(y, p)  
and x < > y

即如果 Y 是男的, X 的父母和 Y 的父母皆为 P, 且 X 不等于 Y, 则 Y 是 X 的兄弟, 程序如下:

Person =symbol

predicates

male(person)

female(person)

father(person, person)

mother(person, person)

sister(person, person)

parent(person, person)

brother(person, person)

uncle(person, person)

grandfather(person, person)

clauses

male(alan).

male(charles).

male(bob).

male(ivan).

female(berly).

female(fay).

female(marilyn).

female(sally).

mother(marilyn, berly).

mother(alan, sally)

father(alan, bob).

father(berly, charles).

father(fay, bob)

father(marilyn, alan)

parent( X, Y) if mother( X, Y) .  
parent(X, Y) if father(X, Y).

brother(X, Y) if       /\*The brother of X is Y if \*/  
  male(Y) and         /\*Y is a male and         \*/  
  parent(X, P) and    /\*the parent of X is P and \*/  
  parent(Y, P) and    /\*the parent of Y is P and \*/  
  X ≠ Y.             /\*X and Y are not the same \*/

sister(X, Y) if       /\*The sister of X is Y if     \*/  
  female(Y) and      /\*Y is female and           \*/  
  parent(X, P) and    /\*the parent of X is P and\*/  
  parent(Y, P) and    /\*the parent of Y is P and\*/  
  X <> Y.            /\*X and Y are not the same \*/

uncle(X, U) if        /\*The uncle of X is U if     \*/  
  mother(Y, P) and    /\*the mother of X is P U   \*/  
  brother(P, U).      /\*the brother of P is and   \*/  
uncle(X, U) if        /\*The uncle of X is U if     \*/  
  father(X, P) and    /\*the father of X is P and   \*/  
  brother(P, U)       /\*the brother of P is U     \*/

grandfather(X, G) if   /\*The grandfather of X is G \*/  
  father(P, G) and     /\*if the father of P is G     \*/  
  mother(X, P).       /\*and the mother of X is P. \*/  
grandfather(X, G) if   /\*The grandfather of X is G \*/  
  father(X, P) and     /\*if the father of X is P     \*/  
  father(P, G).       /\*the father of P is G       \*/

对于这问题系统, 我们可以询问

goal: brother(ivan, alan)   alan 是ivan 的兄弟吗?  
      uncle(X, Y)            有哪些人是叔侄关系  
      grandfather(marilyn, X)   marilyn 的祖父是谁  
      .....

为此我们介绍了人工智能语言的基本核心大多数内容。主要是

说明有关对象的事实           例   likes(paul, win)  
用规则的形式表示关系           例   likes(john, X): if person(X)  
询问有关事实和规则的问题       例   goal: likes(paul, X)  
使用变量及域、谓词语句。

参考文献

- [1] Melish programming in prolog W. F. clocksin
- [2] How to solve with prolog cootho
- [2] 人工智能程序设计 南京大学

# THE BASIC CONCEPT OF ARTIFICIAL LANGUAGE PROLOG

Xiang Diwu

( *Guangxi Commercial Training School* )