Learn Prolog Now 翻译 - 第四章 - 列表 - 第一节，列表定义和使用

内容提要

列表定义；

合一在列表中的使用；

匿名变量；

列表定义

正如名字暗示的，列表就是多个元素组成的集合。更精确地说，是元素的有限序列。在Prolog中的列表，有如下的一些具体例子：

[mia, vincent, jules, yolanda]

[mia, robber(honey_bunny), X, 2, mia]

[ ]

[mia, [vincent, jules], [butch, girlfriend(butch)]]

[[ ], dead(z), [2, [b, c]], [ ], Z, [2, [b, c]]]

我们可以从上面的例子中，学习到下面一些重要的内容：

1. 在Prolog中，使用英文字符中括号封装一个列表（即"[ ]"）；列表中的元素使用英文字符逗号进行分割（即“,”）；比如，上面的第一个列表，[mia, vincent, jules, yolanda]，

是一个包含了4个元素的列表，分别是mia，vincent，jules，和yolanda；一个列表的长度是它包含的元素的个数，所以第一个例子的列表的长度是4。

2. 从第二个例子，[mia, robber(honey_bunny), X, 2, mia]，我们可以知道所有Prolog中的类型都可以是列表的元素；例子中第一个元素是mia，是一个原子；第二个元素是一个

复杂语句；第三个元素是X，是一个变量；第四个元素是2，是一个数字。而且，我们也可以进一步知道相同的元素可能在一个列表中出现多次，比如，第五个元素是面，

和第一个元素是相同的原子。

3. 第三个例子展示了一个特殊的列表，空列表。空列表（正如名字暗示的）是一个没有包含任何元素的列表。那么空列表的长度呢？当然是0。

4. 第四个例子揭示了一个十分重要的事情：列表能够包含其他列表作为其元素。比如这个例子中第二个元素是：[vincent, jules]，第三个元素是：[butch, girlfriend(butch)]。

那么第四个例子的列表长度是多少？答案是3。如果你认为是5（或者其他），那么你思考列表的方式可能就不正确。列表的元素是最外层的中括号封装

的，使用逗号分割的。

所以这个列表有3个元素：第一个是mia，第二个是[vincent, jules]，第三个是[butch, girlfriend(butch)]。

5. 最后一个例子将前面所有的概念都合并在一起。所以我们就有一个列表包含了空列表（事实上，包含了两个），复杂语句：dead(Z)，两个相同的子列表：[2, [b, c]]，

和一个变量Z。请注意第三个（和最后一个）元素是列表，并且其中还包含了子列表（即[b, c]）。

接下来介绍十分重要的点，任何非空的列表都可以认为是两部分构成的：头部和尾部（head and tail）。头部简单地说就是列表的第一个元素，尾部就是列表剩余的部分。

更精确地定义，尾部是指列表去掉第一个元素后的列表。即，一个列表的尾部还是一个列表。比如，如下列表：

[mia, vincent, jules, yolanda]

的头部是mia，它的尾部是：[vincent, jules, yolanda]。类似地，列表：

[[], dead(z), [2, [b, c]], [], Z, [2, [b, c]]]

的头部是[]，尾部是[dead(z), [2, [b, c]], [], Z, [2, [b, c]]]。那么列表[dead(z)]的头部和尾部分别是什么？因为头部是列表的第一个元素，所以是dead(z)，尾部是去掉第一个元素

后的列表，在这里，就是空列表：[]。

那么空列表呢？它没有头部和尾部。即，空列表没有内部结构；对于Prolog而言，[]是特殊的，有独特作用的列表。后面我们将会学习使用递归进行列表操作，空列表的事实

在Prolog编程中会扮演重要的角色。

合一在列表中的使用

Prolog有一个重要的内置操作符｜（英文字符竖线），可以将列表分解为它的头部和尾部。知道如何使用｜非常重要，因为这个符号是进行Prolog列表操作的核心工具。

｜最重要的作用是从列表中提取信息。我们通过使用｜和合一来达成这个目标。比如，为了获取列表：[mia, vincent, jules, yolanda]的头部和尾部，我们可以进行如下的查询：

```
?- [Head | Tail] = [mia, vincent, jules, yolanda].

Head = mia

Tail = [vincent, jules, yolanda]

true
```

即，列表的头部已经和Head绑定，列表的尾部已经和Tail绑定。注意这里到Head和Tail没有什么特别，它们只是简单的变量。我们可以也使用如下的查询：

```
?- [X | Y] = [mia, vincent, jules, yolanda].
```

X = mia

Y = [vincent, jules, yolanda]

true

正如我们之前提到的，只有非空的列表才有头部和尾部，如果我们使用｜去解析[]，Prolog会回答false：

?- [X | Y] = [].

false

即，Prolog将空列表视为特殊列表进行处理，这个结果十分重要，我们将会在后面中的学习看到。

让我们看另外的例子。我们能够从下面到列表中提取出头部和尾部，如我们之前看到的：

?- [X | Y] = [[], dead(z), [2, [b,c]], [], Z].

X = []

Y = [dead(z), [2, [b,c]], [], _7800]

Z = _7800

true

解释一下：列表的头部和X绑定，尾部和Y绑定。（我们还可以看到Z和一个中间变量_7800进行了绑定）

但是我们可以使用多个｜，它是一个很灵活大工具。比如，假设我们希望能够获取列表的头两个元素，及其第二个元素后面的剩余列表，我们可以进行如下查询：

?- [X, Y | W] = [[], dead(z), [2, [b,c]], [], Z].

X = []

Y = dead(z)

W = [[2, [b,c]], [], _8327]

Z = _8327

true

即，列表的头部和X进行绑定，第二个元素和Y进行绑定，列表的剩余部分和W进行绑定（W也是一个列表，即原始列表去掉了头两个元素后的）。所以｜不仅仅

能够将列表分成头部和尾部，还可以从任意点将列表进行分解，｜的左边指示出我们希望从前部提前多少个元素，｜的右边指示出我们希望列表剩余的部分。

匿名变量

是时候介绍匿名变量了。假设我们对如下列表对第二个和第四个元素感兴趣：

[[], dead(z), [2, [b, c]], [], Z].

我们可以使用如下的方式得出结果：

```
?- [X1, X2, X3, X4 | Tail] = [[], dead(z), [2, [b, c]], [], Z].

X1 = []

X2 = dead(z)

X3 = [2, [b, c]]

X4 = []

Tail = [_8910]

Z = _8910

true
```

是的，我们已经获取了我们希望得到的数据，我们感兴趣的信息已经分别和X2，X4进行了绑定，但是同时我们也获取了很多其他的信息，比如X1，X3

和Tail，也许这些是我们不感兴趣的。而且为了处理这样的需求而引入三个无关的变量是一件比较愚蠢的做法。事实上，我们可以通过进行如下的查询达

到目标：

```
?- [_, X, _, Y | _] = [[], dead(z), [2, [b,c]], Z].

X = dead(z)

Y = []

Z = _9523

true
```

英文字符下划线（“_”）代表匿名变量。当我们需要使用一个变量，但又补关心这个变量到底需要绑定什么值时，就可以使用匿名变量。正如你在上面这个

例子中看到的，_不会和任何的值进行绑定。而且，请注意每个_都是独立的：每一个都代表一些不同的变量。这种情况在普通的变量中是不被允许的，但是匿

名变量不是普通的变量，它只是Prolog用于在特殊位置需要变量，每一个都是和其他变量独立的。

让我们看最后一个例子。上面列表的第三个元素也是一个列表（即[2, [b,c]]），假设我们想要获取这个内置列表的尾部，而且对于其他的信息都不感兴趣，我

们可以进行如下的查询：

```
?- [_, _, [_, |X] | _] = [[], dead(z), [2, [b,c]], Z, [2, [b,c]]].

X = [[b,c]]

Z = _10087

true
```

分类：Prolog

标签：合一，Lists，列表，匿名变量

seaman.kingfall
关注 - 1
粉丝 - 4
+加关注

0 0

« 上一篇：Learn Prolog Now 翻译 - 第三章 - 递归 - 第四节，更多的实践和练习
» 下一篇：Learn Prolog Now 翻译 - 第四章 - 列表 - 第二节，列表成员

posted on 2015-07-10 16:45 seaman.kingfall 阅读(469) 评论(0) 编辑 收藏