

公告

昵称: seaman.kingfall  
园龄: 4年3个月  
粉丝: 4  
关注: 1  
[+加关注](#)

搜索

常用链接

[我的随笔](#)  
[我的评论](#)  
[我的参与](#)  
[最新评论](#)  
[我的标签](#)

我的标签

[练习题\(6\)](#)  
[合一\(3\)](#)  
[递归\(3\)](#)  
[中断\(2\)](#)  
[类型变量\(2\)](#)  
[数字\(2\)](#)  
[列表\(2\)](#)  
[Haskell\(2\)](#)  
[recursive\(2\)](#)  
[比较\(2\)](#)  
[更多](#)

随笔分类

[Haskell\(2\)](#)  
[Prolog\(32\)](#)

随笔档案

[2015年8月 \(7\)](#)  
[2015年7月 \(22\)](#)  
[2015年6月 \(5\)](#)

最新评论

1. Re:Learn Prolog Now 翻译 - 第一章 - 事实，规则和查询 - 第一节，一些简单的例子  
学习!  
--深蓝医生

2. Re:Learn Prolog Now 翻译 - 第一章 - 事实，规则和查询 - 第一节，一些简单的例子  
翻译了这么多了，而且每天一篇，不能望其项背啊。

Learn Prolog Now 翻译 - 第五章 - 数字运算 - 第三节，整数的比较

内容提要

Prolog中如何进行整数的比较

整数比较的实际应用

Prolog中如何进行整数的比较

一些Prolog的运算谓词可以实际地进行运算（即，不需要通过“is”协助），这些运算谓词都是进行整数比较的操作符。

运算实例	Prolog表达式
$x < y$	<code>X &lt; Y.</code>
$x \leq y$	<code>X = &lt; Y.</code>
$x = y$	<code>X =:= Y.</code>
$x \neq y$	<code>X =\= Y.</code>
$x \geq y$	<code>X &gt;= Y.</code>
$x > y$	<code>X &gt; Y.</code>

这些操作符有明确的含义，可以直接在Prolog中使用：

```
?- 2 < 4.  
  
true  
  
?- 2 = < 4.  
  
true  
  
?- 4 = < 4.  
  
true  
  
?- 4 =:= 4.  
  
true  
  
?- 4 =\= 5.  
  
true  
  
?- 4 >= 4.  
  
true  
  
?- 4 > 2.
```

--Benjamin Yan

### 阅读排行榜

1. Learn Prolog Now 翻译 - 第三章 - 递归 - 第一节, 递归的定义(1168)
2. Learn Prolog Now 翻译 - 第一章 - 事实, 规则和查询 - 第一节, 一些简单的例子(1087)
3. Learn Prolog Now 翻译 - 第一章 - 事实, 规则和查询 - 第二节, Prolog语法介绍(781)
4. Haskell学习笔记二: 自定义类型(767)
5. Learn Prolog Now 翻译 - 第六章 - 列表补遗 - 第一节, 列表合并(753)

### 评论排行榜

1. Learn Prolog Now 翻译 - 第一章 - 事实, 规则和查询 - 第一节, 一些简单的例子(2)

### 推荐排行榜

1. Haskell学习笔记二: 自定义类型(1)
2. Learn Prolog Now 翻译 - 第三章 - 递归 - 第四节, 更多的实践和练习(1)

true

而且, 这些操作符可以促使其左右两端的参数进行计算:

```
?- 2 < 4 + 1.
```

true

```
?- 2 + 1 < 4.
```

true

```
?- 2 + 1 < 3 + 2.
```

true

需要注意, `:=`和`=`是不同的操作符, 比如:

```
?- 4 = 4.
```

true

```
?- 2+2 = 4.
```

false

```
?- 2+2 := 4.
```

true

即, `=`会尝试对参数进行合一, 而不是进行运算, `:=`就会进行运算然后比较。

而且在使用这些操作符时, 必须注意变量是已经被初始化的, 比如, 下面的所有例子都因为变量没有初始化而失败:

```
?- X < 3.
```

false

```
?- 3 < Y.
```

false

```
?- X := X.
```

false

而且, 变量必须要被初始化为整数, 比如:

```
?- X = 3, X < 4.
```

true

```
?- X = b, X > 4.
```

false

### 实际应用

下面我们通过一个实际的例子来学习如何将Prolog的比较数字能力运用到程序中。我们将会定义一个谓词, 其第一个参数是由非负整数组成的列表, 最后一个参数是返回的列表中的

最大值。同样, 我们将会使用累加器。当遍历列表的过程中, 累加器会保持当前找到的最大值, 如果找到了更大的值, 累加器会更新为新的最大值。当我们调用程序时, 会将累加器的

初始值设置为0。下面是代码, 请注意存在两个递归子句:

```
accMax([H|T], A, Max) :- H > A, accMax(T, H, Max).

accMax([H|T], A, Max) :- H =< A, accMax(T, A, Max).

accMax([], A, A).
```

第一个子句测试如果列表的头元素大于当前找到的最大值的情况, 如果是, 就会设置累加器为新的最大值, 然后对列表的尾部进行递归操作。第二个子句测试列表的头元素等于小于当

前找到最大值的情况, 如果是, 就会继续使用当前累加器对列表尾部进行递归操作。最终, 基础子句将第二个参数和第三个参数合一, 将遍历列表后找到的最大值传递给第三个参数作为

结果。下面是查询的例子:

```
?- accMax([1, 0, 5, 4], 0, Max).
```

首先accMax的第一个子句会起作用, 得到如下的新目标:

```
?- accMax([0, 5, 4], 1, Max).
```

注意累加器的值已经更新为1。然后, accMax的第二个子句会起作用, 因为0 (列表当前的头元素) 比1小。这个过程会重复直到列表为空:

```
?- accMax([5, 4], 1, Max).

?- accMax([4], 5, Max).

?- accMax([], 5, Max).
```

最后, 第三个子句会起作用, 将变量Max和累加器合一:

```
Max = 5.

true
```

同样的, 我们可以再定义一个谓词调用之前的谓词, 并给出累加器; 初始化的值。但是等等, 我们应该将累加器的初始值赋值为什么? 如果说是0, 那么意味着我们假设列表中所有的数字

都是正数。但是如果有负数的列表, 比如:

```
?- accMax([-11, -2, -7, -4, -12], 0, Max).

Max = 0

true
```

这个结果就不是我们期望的, 因为列表中最大值应该是: -2。我们使用0作为累加器的初始值, 但是它比列表中所有的数字都大。

有一个简单的方式解决这个问题: 由于输入的列表是要求非空的整数组成的, 所以将累加器的初始值设置为列表头元素的值。通过这种方式我们可以确保累加器初始值为列表中的数字之一,

如下是具体的实现:

```
?- max(List, Max) :- List = [H|_], accMax(List, H, Max).
```

所以, 如果我们查询:

```
?- max([1, 2, 46, 53, 0], X).
```

```
X = 53.
```

```
true
```

如果我们查询:

```
?- max([-11, -2, -7, -4, -12], X).
```

```
X = -2.
```

```
true
```

能够得到期望的结果。

分类: Prolog

标签: 比较, 整数

好文要顶

关注我

收藏该文



seaman.kingfall

关注 - 1

粉丝 - 4

+加关注

0

0

« 上一篇: Learn Prolog Now 翻译 - 第五章 - 数字运算 - 第二节, 数字运算与列表

» 下一篇: Learn Prolog Now 翻译 - 第五章 - 数字运算 - 第四节, 练习题和答案

posted on 2015-07-17 11:16 seaman.kingfall 阅读(474) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

**注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。**

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【活动】看雪2019安全开发者峰会, 共话安全领域焦点

【培训】Java程序员年薪40W, 他1年走了别人5年的路

**最新新闻:**

- 微信公开课聚焦“增长”: 墨迹天气小程序DAU环比增100%
  - 知否 | 太空垃圾如何清理? 卫星测试用鱼叉击中太空垃圾碎片
  - 一线 | “美团配送”品牌发布: 对外开放配送平台 共享配送能力
  - 苍蝇落在食物上会发生什么? 让我们说的仔细一点
  - 科学家研究板块构造变化对海洋含氧量影响
- » 更多新闻...

Copyright © seaman.kingfall  
Powered by: .Text and ASP.NET  
Theme by: .NET Monster