

Seaman.h.zhang

博客园 :: 首页 :: 新随笔 :: 联系 :: 订阅 [XML](#) :: 管理 34 Posts :: 0 Stories :: 2 Comments :: 0 Trackbacks

公告

昵称: seaman.kingfall

园龄: 4年3个月

粉丝: 4

关注: 1

+加关注

搜索

找找看

谷歌搜索

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

阅读排行榜

评论排行榜

推荐排行榜

Learn Prolog Now 翻译 - 第九章 - 语句深究 - 第一节, 语句的比较

内容提要

- 判断两个语句是否相等
- 判断两个语句是否不等

判断两个语句是否相等

Prolog中有一个用于判断两个语句是否相等的重要谓词, 即`==/2`。这个谓词可以用来判断两个语句是否相等, 但是不能用于初始化变量, 所以这一点是和谓词`=/2`有所区别的。让我们来看一些具体的例子:

```
?- a == a.
true

?- a == b.
false

?- a == 'a'
true
```

Prolog回答上面查询的答案的原因是很明确的, 但是特别提一下最后一个查询, 因为`a`和`'a'`在Prolog中被认为是同一个原子。

接下来, 我们看一些包含了变量的例子, 也可以明确比较谓词`==/2`和合一谓词`=/2`的不同:

```
?- X == Y.
false

?- X = Y.
X = _2808
Y = _2808
true
```

在上面的查询中, `X`和`Y`都是没有初始化的变量, 我们没有给它们任何值。所以这一个查询结果是正确的: 因为`X`和`Y`不是相同的对象, 所以`==`验证返回`false`。另一方面, 使用`=`可以成功, 因为`X`和`Y`能够合一。

让我们继续看包含了初始化值的变量的查询:

```
?- a = X, a == X.
X = a
true
```

查询的第一个部分, `a=X`, 将`X`和`a`合一, `X`就被初始化为`a`。所以当`a == X`运算时, 左右两边都是相同的Prolog对象, 所以检查通过。

类似的情况也发生在下面的查询中:

```
?- X = Y, X == Y.
X = _4500
```

```
Y = _4500
true
```

查询的第一个子句, $X = Y$, 将变量X和Y合一, 它们共享同一个Prolog对象, 然后使用 $X == Y$ 检查是否相等, 就能够成功。

可以看出, $=$ 和 $==$ 是明确不同的, 尽管如此, 它们之间有一个重要的关联: $==$ 可以看作是比 $=$ 更强的验证两个语句相等的方式, 因为, 如果term1和term2都是Prolog的语句, 并且查询 $term1 == term2$ 能够成功, 那么查询 $term1 = term2$ 也必然能够成功。

判断两个语句是否不等

另外一个值得一提的谓词是: \neq , 这个谓词的作用和 $==$ 正好相反, 当 $==$ 失败时成功。即, 当两个语句不相等时, \neq 返回为真, 看例子:

```
?- a \== a.
false

?- a \== b.
true

?- a \== 'a'
false
```

上面的查询及其结果都很好理解: 这些都是使用 \neq 的相反结果。思考下一个例子:

```
?- X \== a.
X = _3719
true
```

为什么会有这样的答案? 我们知道之前查询 $X == a$ 是 false, 所以查询 $X \neq a$ 是 true, Prolog也确实是这样运行的。

类似地:

```
?- X \== Y.
X = _798
Y = _799
true
```

同样地, 我们知道之前查询 $X == Y$ 失败了, 所以查询 $X \neq Y$ 就能够成功。

分类: Prolog

标签: 比较, 语句, terms

好文要顶

关注我

收藏该文



seaman.kingfall

关注 - 1

0

0

粉丝 - 4

+加关注

« 上一篇: Learn Prolog Now 翻译 - 第六章 - 列表补遗 - 第三节, 练习题和答案

» 下一篇: Learn Prolog Now 翻译 - 第九章 - 语句深究 - 第二节, 特殊表示法的语句

posted on 2015-07-24 16:51 seaman.kingfall 阅读(430) 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

相关博文:

- [Learn Prolog Now 翻译 - 第二章 - 合一和证明搜索 - 第一节, 合一](#)
- [Learn Prolog Now 翻译 - 第九章 - 语句深究 - 第四节, 操作符](#)
- [Learn Prolog Now 翻译 - 第五章 - 数字运算 - 第三节, 整数的比较](#)
- [Learn Prolog Now 翻译 - 第九章 - 语句深究 - 第三节, 语句的检查](#)
- [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](#)

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