

Seaman.h.zhang

博客园 :: 首页 :: 新随笔 :: 联系 :: 订阅  :: 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

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 - 第四章 - 列表 - 第四节, 练习题和答案

练习题4.1

Prolog将会如何回答下面的查询?

1. `[a, b, c, d] = [a, [b, c, d]]`.
2. `[a, b, c, d] = [a | [b, c, d]]`.
3. `[a, b, c, d] = [a, b, [c, d]]`.
4. `[a, b, c, d] = [a, b | [c, d]]`.
5. `[a, b, c, d] = [a, b, c, [d]]`.
6. `[a, b, c, d] = [a, b, c | [d]]`.
7. `[a, b, c, d] = [a, b, c, d, []]`.
8. `[a, b, c, d] = [a, b, c, d | []]`.
9. `[] = _`.
10. `[] = [_]`.
11. `[] = [_ | []]`.

我的答案和解释:

1. false.
2. true; 因为a是头部, `|` 是列表解析符号, `[b, c, d]`是尾部, 结合在一起, 就是`[a, b, c, d]`.
3. false.
4. true; 理由同2.
5. false.
6. true; 理由同2.
7. false;
8. true; 理由同2.
9. true, 因为`_`代表匿名变量, 这个查询相当于将一个空列表和匿名变量进行合一, 是能够成功的。
10. false.
11. false.

练习题4.2

下面哪些是正确的列表? 如果是列表, 其中有多少个元素?

--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)

1. [1 | [2, 3, 4]]
2. [1, 2, 3, | []]
3. [1 | 2, 3, 4]
4. [1 | [2 | [3 | [4]]]]
5. [] | []
6. [[1, 2] | 4]
7. [[1,2], [3, 4] | [5, 6, 7]]

我的答案和解释:

1. 是列表, 长度是4; [1, 2, 3, 4].
2. 是列表, 长度是3; [1, 2, 3].
3. 不是列表。
4. 是列表, 长度是4; [1, 2, 3, 4].
5. 是列表, 长度是4; [1, 2, 3, 4].
6. 是列表, 长度是1; [[]].
7. 不是列表。
8. 是列表, 长度是5; [[1, 2], [3, 4], 5, 6, 7].

练习题4.3

写出一个谓词second(X, List), 可以检查X是否是List的第二个元素。

我的答案:

```
second(X, [_ , X | _]).
```

一些测试查询:

```
?- second(2, [1, 2, 3, 4]).
```

```
true
```

```
?- second(X, [1, 2, 3, 4]).
```

```
X = 2
```

```
?- second(1, [1, 2, 3, 4]).
```

```
false
```

```
?- second(1, []).
```

```
false
```

练习题4.4

写出一个谓词`swap12(List1, List2)`, 检查两个列表除了头两个元素是互相交换的, 其他元素都是相等的。

我的答案:

```
swap12([H1, H2 | T], [H2, H1 | T]).
```

一些测试:

```
?- swap12([1, 2, 3, 4], [2, 1, 3, 4]).
```

true

```
?- swap12([1, 2], [2, 1]).
```

true

```
?- swap12([1,2,3,4], [1,2,3,4]).
```

false

练习题4.5

假设存在由如下事实构成的知识库:



```
tran(eins, one).  
tran(zwei, two).  
tran(drei, three).  
tran(vier, four).  
tran(fuenf, five).  
tran(sechs, six).  
tran(sieben, seven).  
tran(acht, eight).  
tran(neun, nine).
```



写一个谓词`listtran(G, E)`, 可以将德文的数字列表, 翻译为英文的数字列表。比如:

```
?- listtran([eins, neun, zwei], X).
```

Prolog会回答:

```
X = [one, nine, two].
```

谓词可以从另外一个方向也起作用, 比如查询:

```
?- listtran(X, [one, seven, six, two]).
```

Prolog会回答:

```
X = [eins, sieben, sechs, zwei]
```

我的答案和一些测试:

```
listtran([], []).
```

```
listtran([H1 | T1], [H2 | T2]) :- tran(H1, H2), listtran(T1, T2).
```

测试:

```
?- listtran([eins, neun, zwei], X).
```

```
X = [one, nine, two].
```

```
?- listtran(X, [one, seven, six, two]).
```

```
X = [eins, sieben, sechs, zwei]
```

练习题4.6

请写出一个谓词, `twice(In, Out)`, 其中第一个参数是List, 第二个参数也是List, 其中的元素是第一个List元素的两个, 比如:

```
?- twice([a, 4, buggle], X).
```

```
X = [a, a, 4, 4, buggle, buggle].
```

```
?- twice([1, 2, 1, 1], X).
```

```
X = [1, 1, 2, 2, 1, 1, 1, 1].
```

我的答案:

```
twice([], []).
```

```
twice([H|T1], [H, H|T2]) :- twice(T1, T2).
```

练习题4.7

画出下面三个查询的搜索树:

```
?- member(a, [c, b, a, y]).
```

```
?- member(x, [a, b, c]).
```

```
?- member(X, [a, b, c]).
```

此题限于篇幅我不再提供答案, 读者如果感兴趣, 可以自己尝试画出搜索树。

分类: Prolog

标签: 练习题

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 - 第四章 - 列表 - 第三节, 递归遍历列表

» 下一篇: Learn Prolog Now 翻译 - 第五章 - 数字运算 - 第一节, Prolog中的数字

运算

posted on 2015-07-15 15:43 seaman.kingfall 阅读(381) 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