

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。
--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)

Learn Prolog Now 翻译 - 第二章 - 合一和证明搜索 - 第三节, 练习题和答案

练习题 2.1 下面各组语句中, 哪些组是能够合一的? 如果能够合一, 请给出变量初始化的信息。

1. bread = bread.
2. 'Bread' = bread.
3. 'bread' = bread.
4. Bread = bread.
5. bread = sausage.
6. food(bread) = bread.
7. food(bread) = X.
8. food(X) = food(bread).
9. food(bread, X) = food(Y, sausage).
10. food(bread, X, beer) = food(Y, sausage, X).
11. food(bread, X, beer) = food(Y, kahuna_burger).
12. food(X) = X.
13. meal(food(bread), drink(beer)) = meal(X, Y).
14. meal(food(bread), X) = meal(X, drink(beer)).

我的答案及其解释:

1. bread = bread. 可以合一, 相同的原子;
2. 'Bread' = bread. 不能合一, 因为是不同的原子;
3. 'bread' = bread. 可以合一, 相同的原子;
4. Bread = bread. 可以合一, 变量Bread初始化为bread;
5. bread = sausage. 不能合一, 因为是不同的原子;
6. food(bread) = bread. 不能合一, 一个是复杂语句, 一个是原子;
7. food(bread) = X. 可以合一, 变量X初始化为food(bread);
8. food(X) = food(bread). 可以合一, 变量X初始化为bread;
9. food(bread, X) = food(Y, sausage). 可以合一, 变量X初始化为sausage, 变量Y初始化为bread;
10. food(bread, X, beer) = food(Y, sausage, X). 不能合一, 无法初始化X;
11. food(bread, X, beer) = food(Y, kahuna_burger). 不能合一, 两个复杂语句的元数不同;
12. food(X) = X. 可以合一, 根据我机器的实现, 变量X初始化为food(X);
13. meal(food(bread), drink(beer)) = meal(X, Y). 可以合一, 变量X初始化为food(bread), 变量Y初始化为drink(beer);
14. meal(food(bread), X) = meal(X, drink(beer)). 不能合一, 无法初始化X。

练习题 2.2 给出一个知识库, 及其基于这个知识库的查询, 请选择那些查询是能够满足的, 如果能够满足, 请给出相关的变量初始化信息。

知识库如下:

- 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)

```
house_elf(dobby).

witch(hermione).
witch('McGonagall').
witch(rita_skeeter).

magic(X) :- house_elf(X).
magic(X) :- wizard(X).
magic(X) :- witch(X).
```

查询如下：

- ?- magic(house_elf).
- ?- wizard(harry).
- ?- magic(wizard).
- ?- magic('McGonagall').
- ?- magic(Hermione).

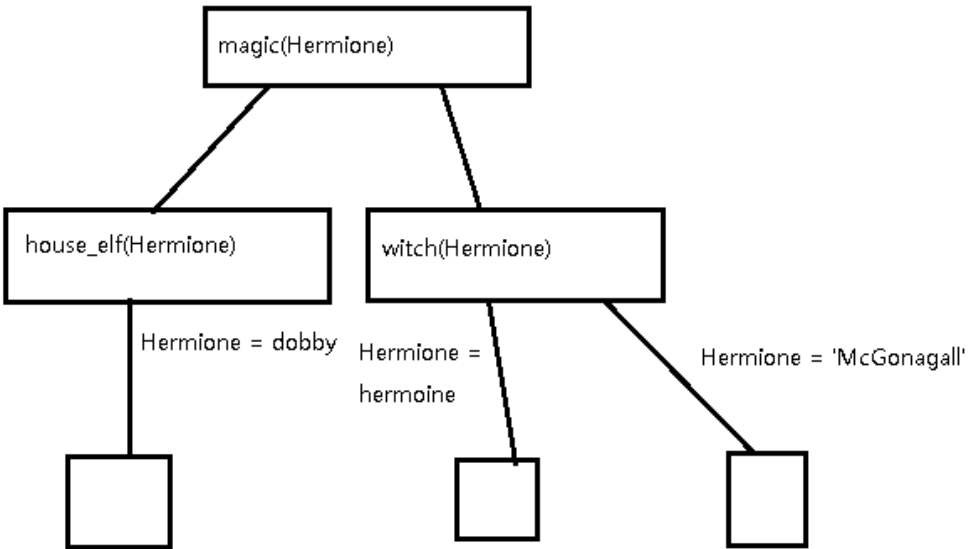
并且绘制出magic(Hermione)的搜索树。

我的答案和解释：

- 1. ?- magic(house_elf). 我的Prolog实现对这个问题会报错，因为wizard这个谓词没有具体实现；但是这个查询整体来说是无效的，因为它既不是事实，有不能被推导出；
- 2. ?- wizard(harry). 我的Prolog实现会报错，因为wizard谓词没有在知识库中定义；
- 3. ?- magic(wizard). 我的Prolog实现会报错，因为wizard谓词没有定义；
- 4. ?- magic('McGonagall'). Prolog回答true，因为magic('McGonagall') -> witch('McGonagall')，而witch('McGonagall')是知识库中定义的事实；
- 5. ?- magic(Hermione). 注意这点的Hermione是首字母大写的，即这是一个变量，所以这里的答案是：


```
Hermione = dooby;
Hermione = hermione;
Hermione = 'McGonagall'.
```

magic(hermione)的搜索树如下：




练习题 2.3 有如下的微型词库（即，一些个体单词的信息），和由一个句法规则（定义了一个句子由如下的顺序构成：一个量词，一个名词，一个动词，一个量词，一个名词）组成的微型语法。如果要在这个知识库中找到合理的句子，应该如何进行查询？列出所有符合语法规则的句子。

知识库如下：



```
word(determiner, a).
word(determiner, every).
word(noun, criminal).
word(noun, 'big kahuna burger').
word(verb, eats).
word(verb, likes).

sentence(Word1, Word2, Word3, Word4, Word5) :-
    word(determiner, Word1),
    word(noun, Word2),
    word(verb, Word3),
    word(determiner, Word4),
    word(noun, Word5).
```



我的答案和解释：

可以简单地进行如下的查询，然后使用分号“；”：“查询出每个可能的组合值：

```
?- sentence(A, B, C, D, E).
```

将A, B, C, D, E每次的值组合一下（便于阅读），既可以得出所有的答案如下：（限于篇幅，就不一一列举完）

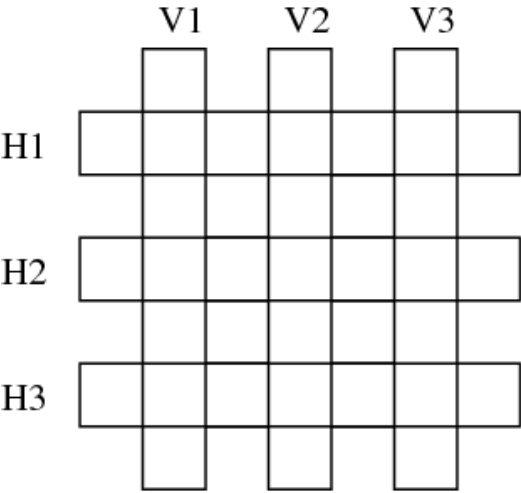
```
a criminal eats a criminal
a criminal eats a 'big kahuna burger'
a criminal eats every criminal
a criminal eats every 'big kahuna burger'
a criminal likes a criminal
a criminal likes a 'big kahuna burger'
a criminal likes every criminal
a criminal likes every 'big kahuna burger'
...
```

练习题 2.4 完成下面的填字游戏

有下面6个拉丁单词：

```
astente, astoria, baratto, cobalto, pistola, statale.
```

它们被分配到下面的谜题宫格中：



如下的知识库是关于这些单词的简单词库：

```
word(astante, a,s,t,a,n,t,e).
word(astoria, a,s,t,o,r,i,a).
word(baratto, b,a,r,a,t,t,o).
word(cobalto, c,o,b,a,l,t,o).
word(pistola, p,i,s,t,o,l,a).
word(statale, s,t,a,t,a,l,e).
```

请构建一个名字为crossword/6的谓词，完成谜题宫格的填写。其中的前三个参数代表从左到右的列上的单词，后三个参数代表从上到下的行上的单词。

我的答案和解释：

crossword/6谓词的定义是这样思考的，首先6个单词，是必须符合知识库中word定义的；其次根据行列交叉的位置，应该是同一字母的原则来进行；最后加入一条后面会涉及的知识，

每个单词不能相等：

```
crossword(V1, V2, V3, H1, H2, H3) :-
    word(V1, _, V1H1,_,V1H2,_,V1H3,_),
    word(V2, _, V2H1,_,V2H2,_,V2H3,_),
    word(V3, _, V3H1,_,V3H2,_,V3H3,_),
    word(H1, _, V1H1,_,V2H1,_,V3H1,_),
    word(H2, _, V1H2,_,V2H2,_,V3H2,_),
    word(H3, _, V1H3,_,V2H3,_,V3H3,_),
    V1 \= H1,
    V2 \= H2,
    V3 \= H3.
```

运行查询的结果如下：

```
?- crossword(V1, V2, V3, H1, H2, H3).

V1 = astanto, V2 = cobalto, V3 = pistola, H1 = astoria, H2 = baratto, H3 = statale.
```

填入结果后的谜题宫格如下图所示：

| | | | | | | | | | |
|----|----|---|---|----|---|---|----|---|--|
| | V1 | | | V2 | | | V3 | | |
| | | a | | | c | | | p | |
| H1 | a | s | t | o | r | i | a | | |
| | | t | | b | | s | | | |
| H2 | b | a | r | a | t | t | o | | |
| | | n | | l | | o | | | |
| H3 | s | t | a | t | a | l | e | | |
| | | e | | o | | a | | | |

这道题可以对Prolog编程解决问题有一点直观的感觉：我只是描述和构建事实，包括规则也是基于我观察到的现实，并没有告诉计算机如果具体找出这些单词；

—— 救赎之道就在其中。

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 - 第二章 - 合一和证明搜索 - 第二节, 证明搜索

» 下一篇: Learn Prolog Now 翻译 - 第三章 - 递归 - 第一节, 递归的定义

posted on 2015-07-03 14:44 seaman.kingfall 阅读(418) 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