

斑马问题-实训报告

2312966 林晖鹏

一、问题重述

题目解读：

- 有五个来自不同国家不同职业的人住在不同颜色的房子，养不同品种的宠物以及爱喝不同种类的饮品。五间房子呈线性排列，即有左右关系。（就是有五个对象，每个对象的属性值不同。）
- 现在提供多条信息（给出属性值之间关系），要求求出这五个对象的所有属性值是什么。
- 使用kanren包来进行这个逻辑推理，掌握kanren这个包的基本使用，**用提供的信息来规定逻辑规则**，最终解出结果。

作业要求：

- a. 基本掌握逻辑编程的思想，了解逻辑编程与命令式编程的区别
- b. 能够依据给定的事实以及规则编写代码，解决逻辑约束问题（CLP）
- c. 使用 Python 语言(可参考kanren包)

二、设计思想

按照kanren包的语法规则，在lall中根据题目所给的条件制定规则。

三、代码内容

代码分析：

1. 定义 `self.units` 的格式

这里用eq把units的变量定义为一个元组，元组里面包含五个变量（实际上这里的变量还可以进一步清晰为含五个变量的元组）。

```
1 (eq,(var(), var(), var(), var(), var()), self.units),
```

2. 利用题目提供的条件来制定规则

lall是用来规定规则集合的，用来多方程求解答案。这里我们将题目提供的条件设置为规则写在lall里面：

```
1  (membero,('英国人', var(), var(), var(), '红色'), self.units),
2  (membero,('西班牙人', var(), var(), '狗', var()), self.units),
3  (membero,('日本人', '油漆工', var(), var(), var()), self.units),
4  (membero,('意大利人', var(), '茶', var(), var()), self.units),
5  (pos_, self.units, ('挪威人', var(), var(), var(), var()), 1),
6  (nextto_,self.units, (var(), var(), var(), var(), '白色'), (var(), var(),
    var(), var(), '绿色')),
7  (membero,(var(), '摄影师', var(), '蜗牛', var()), self.units),
8  (membero,(var(), '外交官', var(), var(), '黄色'), self.units),
9  (pos_,self.units, (var(), var(), '牛奶', var(), var()), 3),
10 (membero,(var(), var(), '咖啡', var(), '绿色'), self.units),
11 (conde,[(nextto_, self.units, ('挪威人', var(), var(), var(), var()), (var(),
    var(), var(), var(), '蓝色'))],
12         [(nextto_, self.units, (var(), var(), var(), var(), '蓝色'), ('挪威
    人', var(), var(), var(), var()))]),
13 (membero,(var(), '小提琴家', '橘子汁', var(), var()), self.units),
14 (conde,[(nextto_, self.units, (var(), '医生', var(), var(), var()), (var(),
    var(), var(), '狐狸', var()))],
15         [(nextto_, self.units, (var(), var(), var(), '狐狸', var()), (var(),
    '医生', var(), var(), var()))]),
16 (conde,[(nextto_, self.units, (var(), '外交官', var(), var(), var()), (var(),
    var(), var(), '马', var()))],
17         [(nextto_, self.units, (var(), var(), var(), '马', var()), (var(), '外
    交官', var(), var(), var()))]),
18 (membero, (var(), var(), '矿泉水', var(), var()), self.units), #限定定义域
19 (membero, (var(), var(), var(), '斑马', var()), self.units)
20 )
```

3. 定义位置函数：

在有的条件中，涉及到屋子之间的位置关系（例如：中间那个房子的人喜欢喝牛奶），这种规则代码篇幅较大，我们设计 `pos_` 和 `nextto_` 函数来简化代码。

```
1  def nextto_(units,l,r):
2      return membero(units,(
```

```

3         (l,r,var(),var(),var()),
4         (var(),l,r,var(),var()),
5         (var(),var(),l,r,var()),
6         (var(),var(),var(),l,r)))
7 def pos_(units,it,i):
8     if i == 1:
9         return eq(units,(it,var(),var(),var(),var()))
10    elif i == 2:
11        return eq(units,(var(),it,var(),var(),var()))
12    elif i == 3:
13        return eq(units,(var(),var(),it,var(),var()))
14    elif i == 4:
15        return eq(units,(var(),var(),var(),it,var()))
16    elif i == 5:
17        return eq(units,(var(),var(),var(),var(),it))

```

在查阅资料后发现，其实在类似于 `(membero,(var(), '摄影师', var(), '蜗牛', var()), self.units)` 这种语句，第一个参数其实是一个函数，后面的变量为这个函数所需要的参数。所以在制定规则的时候，我们也采用了这种写法：`(pos_,self.units, (var(), var(), '牛奶', var(), var()), 3)`。

4. 限定定义域

在条件中，有的属性并没有被提到，即使可以排除法得出结果，但我们仍然需要限定一下剩下的属性。

```

1 (membero, (var(), var(), '矿泉水', var(), var()), self.units), #限定定义域
2 (membero, (var(), var(), var(), '斑马', var()), self.units)

```

完整代码

```

1 from kanren import run, eq, membero, var, conde          # kanren一个描述性Python
  逻辑编程系统
2 from kanren.core import lall                             # lall包用于定义规则
3 import time
4
5 def nextto_(units,l,r):
6     return membero(units,(
7         (l,r,var(),var(),var()),

```

```

8         (var(),l,r,var(),var()),
9         (var(),var(),l,r,var()),
10        (var(),var(),var(),l,r)))
11 def pos_(units,it,i):
12     if i == 1:
13         return eq(units,(it,var(),var(),var(),var()))
14     elif i == 2:
15         return eq(units,(var(),it,var(),var(),var()))
16     elif i == 3:
17         return eq(units,(var(),var(),it,var(),var()))
18     elif i == 4:
19         return eq(units,(var(),var(),var(),it,var()))
20     elif i == 5:
21         return eq(units,(var(),var(),var(),var(),it))
22
23 class Agent:
24
25     def __init__(self):
26         self.units = var() # 单个unit变量指代一座房子的信息(国家,工
作, 饮料, 宠物, 颜色)
27
28         # 例如('英国人', '油漆工', '茶', '狗',
'红色')即为正确格式, 但不是本题答案
29
30         # 请基于给定的逻辑提示求解五条正确的答案
31         self.rules_zebraproblem = None # 用lall包定义逻辑规则
32         self.solutions = None # 存储结果
33
34     def define_rules(self):
35
36         self.rules_zebraproblem = lall(
37         (eq,(var(), var(), var(), var(), var()), self.units),
38
39         (membero,('英国人', var(), var(), var(), '红色'), self.units),
40         (membero,('西班牙人', var(), var(), '狗', var()), self.units),
41         (membero,('日本人', '油漆工', var(), var(), var()), self.units),
42         (membero,('意大利人', var(), '茶', var(), var()), self.units),
43         (pos_, self.units, ('挪威人', var(), var(), var(), var()), 1),
44         (nexttto_,self.units, (var(), var(), var(), var(), '白色'), (var(),
var(), var(), var(), '绿色')),
45         (membero,(var(), '摄影师', var(), '蜗牛', var()), self.units),
46         (membero,(var(), '外交官', var(), var(), '黄色'), self.units),
47         (pos_,self.units, (var(), var(), '牛奶', var(), var()), 3),
48         (membero,(var(), var(), '咖啡', var(), '绿色'), self.units),
49         (conde,[(nexttto_, self.units, ('挪威人', var(), var(), var(), var()),
(var(), var(), var(), var(), '蓝色'))],
50         [(nexttto_, self.units, (var(), var(), var(), var(), '蓝色'),
('挪威人', var(), var(), var(), var()))]),
51         (membero,(var(), '小提琴家', '橘子汁', var(), var()), self.units),

```

```

50         (conde,[(nexttto_, self.units, (var(), '医生', var(), var(), var()),
          (var(), var(), var(), '狐狸', var()))],
51             [(nexttto_, self.units, (var(), var(), var(), '狐狸', var()),
          (var(), '医生', var(), var(), var()))]),
52         (conde,[(nexttto_, self.units, (var(), '外交官', var(), var(), var()),
          (var(), var(), var(), '马', var()))],
53             [(nexttto_, self.units, (var(), var(), var(), '马', var()),
          (var(), '外交官', var(), var(), var()))]),
54         (membero, (var(), var(), '矿泉水', var(), var()), self.units), #限定定义
域
55         (membero, (var(), var(), var(), '斑马', var()), self.units)
56     )
57
58     def solve(self):
59         self.define_rules()
60         self.solutions = run(0, self.units, self.rules_zebraproblem)
61         return self.solutions
62

```

四、代码尝试改进

感性理解，条件之间应该亦有高下。

比如，位置条件应该是比属性条件更强的，属性条件只能规定某个对象之内的属性捆绑，但是这个对象的位置是未知的，在逻辑推理的时候，导致迭代更慢。而位置条件是固定的，一方面能把某个位置的对象属性固定，另一方面也能将带有该属性的对象固定在某个变量上（外层变量，五个房子那个，条件更加强）。

再者，一些条件蕴含的信息更多。

如果我们将这些强一点的条件放在推理链的前面，这样在进行后面的逻辑推理的时候能快速找到正确对象，从而加快速度。

但是，在多次修改顺序的尝试下，时间仍然不变，可能需要优化的是我前面函数部分。下面是我尝试的顺序，原本打算结合人工推理下的思路，加快求解，但是结果都是4s。

```

1     (pos_, self.units, ('挪威人', var(), var(), var(), var()), 1),
2         (nexttto_,self.units, (var(), var(), var(), var(), '白色'),
          (var(), var(), var(), var(), '绿色')),
3     (conde,[(nexttto_, self.units, ('挪威人', var(), var(), var(), var()), (var(),
          var(), var(), var(), '蓝色'))],

```

```

4      [(nexttto_, self.units, (var(), var(), var(), var(), '蓝色'), ('挪威
      人', var(), var(), var(), var()))]),
5      (pos_,self.units, (var(), var(), '牛奶', var(), var()), 3),
6
7      (membero,('英国人', var(), var(), var(), '红色'), self.units),
8      (membero,('西班牙人', var(), var(), '狗', var()), self.units),
9      (membero,('日本人', '油漆工', var(), var(), var()), self.units),
10     (membero,('意大利人', var(), '茶', var(), var()), self.units),
11     (membero,(var(), '摄影师', var(), '蜗牛', var()), self.units),
12     (membero,(var(), '外交官', var(), var(), '黄色'), self.units),
13     (membero,(var(), var(), '咖啡', var(), '绿色'), self.units),
14     (membero,(var(), '小提琴家', '橘子汁', var(), var()), self.units),
15
16     (conde,[(nexttto_, self.units, (var(), '医生', var(), var(), var()), (var(),
      var(), var(), '狐狸', var()))],
17     [(nexttto_, self.units, (var(), var(), var(), '狐狸', var()), (var(), '医生',
      var(), var(), var()))]),
18     (conde,[(nexttto_, self.units, (var(), '外交官', var(), var(), var()), (var(),
      var(), var(), '马', var()))],
19     [(nexttto_, self.units, (var(), var(), var(), '马', var()), (var(), '外
      交官', var(), var(), var()))]),
20
21     (membero, (var(), var(), '矿泉水', var(), var()), self.units), #限定定义域
22     (membero, (var(), var(), var(), '斑马', var()), self.units)

```

五、实验结果

系统测试

 main.py

接口测试

✔ 接口测试通过。

用例测试

测试点	状态	时长	结果
测试结果	✔	4s	测试成功!

提交结果

六、总结

- 在本次实验中，我进一步熟悉了python的使用及代码书写。
- 学习了kanren包的使用方式，并利用kanren包实现了斑马问题的求解。
- 尝试调换逻辑条件的顺序来加快速度，但是没有太大提升。