**程序报告**

学号：2013750 姓名：管昀玫

1. **问题重述**

**斑马问题：** 5 个不同国家（英国、西班牙、日本、意大利、挪威）且工作各不相同（油漆工、摄影师、外交官、小提琴家、医生）的人分别住在一条街上的 5 所房子里，  
每所房子的颜色不同（红色、白色、蓝色、黄色、绿色），每个人都有自己养的不同宠物（狗、蜗牛、斑马、马、狐狸），喜欢喝不同的饮料（矿泉水、牛奶、茶、橘子汁、咖啡）。

根据以下提示，你能告诉我哪所房子里的人养斑马，哪所房子里的人喜欢喝矿泉水吗？

1. 英国人住在红色的房子里
2. 西班牙人养了一条狗
3. 日本人是一个油漆工
4. 意大利人喜欢喝茶
5. 挪威人住在左边的第一个房子里
6. 绿房子在白房子的右边
7. 摄影师养了一只蜗牛
8. 外交官住在黄房子里
9. 中间那个房子的人喜欢喝牛奶
10. 喜欢喝咖啡的人住在绿房子里
11. 挪威人住在蓝色的房子旁边
12. 小提琴家喜欢喝橘子汁
13. 养狐狸的人所住的房子与医生的房子相邻
14. 养马的人所住的房子与外交官的房子相邻
15. **设计思想**

使用kanren逻辑变成包来完成该次程序设计。

1. 首先需要初始化一个智能体self.units，单个unit变量指代一座房子的信息(国家，工作，饮料，宠物，颜色)，例如('英国人', '油漆工', '茶', '狗', '红色')。需要用lall包定义逻辑规则，并使用self.solutions来存储结果。

2. 由于推理中有描述像左边、右边、中间等位置信息的语句，所以我们需要定义left、right、next函数。伪代码如下：

def left(q, p, list):

return membero((q,p), 位置信息二元组列表)

def next(q, p, list):

return conde(位置信息二元组列表1，位置信息二元组列表2)

3. 根据题目所给的信息，定义逻辑规则。这里需要使用membero函数，示例：

英国人住在红色的房子里

(membero,('英国人', var(), var(), var(), '红色'), self.units),

而对于xx在xx的右边，如：

绿房子在白房子的右边

(right,(var(), var(), var(), var(), '绿色'),

(var(), var(), var(), var(), '白色'), self.units),

将14个信息一一翻译为代码；

4. 补充基于问题的信息。由于题目中所提到的每一项信息，如“斑马”、“蓝色”、“矿泉水”等，并没有所有都在14条提示中出现过，所以我们需要将他们提炼出来加上：

如：(membero,(var(), var(), var(), '斑马', var()), self.units),

1. **代码内容**

#定义left/ right/ next

def left(q, p, list):

return membero((q,p), zip(list, list[1:]))

def right(q, p, list):

return membero((q,p), zip(list[1:], list[:-1]))

def next(q, p, list):

return conde([left(q, p, list)], [left(p, q, list)]) ====================================================================

self.rules\_zebraproblem = lall(

(eq, (var(), var(), var(), var(), var()), self.units), # self.units共包含五个unit成员，即每一个unit对应的var都指代一座房子(国家，工作，饮料，宠物，颜色)

# 各个unit房子又包含五个成员属性: (国家，工作，饮料，宠物，颜色)

(membero,('英国人', var(), var(), var(), '红色'), self.units),

(membero,('西班牙人', var(), var(), '狗', var()), self.units),

(membero,('日本人', '油漆工', var(), var(), var()), self.units),

(membero,('意大利人', var(), '茶', var(), var()), self.units),

(eq,(('挪威人', var(), var(), var(), var()), var(), var(), var(), var()), self.units),

(right,(var(), var(), var(), var(), '绿色'),

(var(), var(), var(), var(), '白色'), self.units),

(membero,(var(), '摄影师', var(), '蜗牛', var()), self.units),

(membero,(var(), '外交官', var(), var(), '黄色'), self.units),

(eq,(var(), var(), (var(), var(), '牛奶', var(), var()), var(), var()),self.units),

(membero,(var(), var(), '咖啡', var(), '绿色'), self.units),

(next,('挪威人', var(), var(), var(), var()),

(var(), var(), var(), var(), '蓝色'), self.units),

(membero,(var(), '小提琴家', '橘子汁', var(), var()), self.units),

(next,(var(), var(), var(), '狐狸', var()),

(var(), '医生', var(), var(), var()), self.units),

(next,(var(), var(), var(), '马', var()),

(var(), '外交官', var(), var(), var()), self.units),

#补充信息

(membero,(var(), var(), var(), '斑马', var()), self.units),

(membero,(var(), var(), '矿泉水', var(), var()), self.units),

(membero,(var(), var(), var(), var(), '蓝色'), self.units)

1. **实验结果**

绿色房子里的人养斑马

黄色房子里的人喜欢喝矿泉水

('挪威人', '外交官', '矿泉水', '狐狸', '黄色')

('意大利人', '医生', '茶', '马', '蓝色')

('英国人', '摄影师', '牛奶', '蜗牛', '红色')

('西班牙人', '小提琴家', '橘子汁', '狗', '白色')

('日本人', '油漆工', '咖啡', '斑马', '绿色')

1. **总结**

模型已达到预期效果。

我遇到的一些困难：

1. 一开始，我并不知道有关于位置信息的描述应该如何转到代码语言，如“绿房子在白房子的右边”、“养狐狸的人所住的房子与医生的房子相邻”，提示说写left、right函数也没有理解，后来查阅资料才发现，left实际上就是定义（p,q）这个二元组的相对位置，如果A在B的左边，则A是p，B是q；right和next也同理，只不过next无所谓左边和右边。

2. 对于房子相对位置的定义：一开始我想用list来表示，受到他人启发之后我认为用list与二元组结合起来表示会更合适。假如A在B的左边，则（A,B）为（0,1）（1，2）（2,3）（3,4）这四种可能，这也能应用到left和right函数中去。

3. 需要根据提示补充信息。一开始我并没有写补充的信息，如“斑马”，运行出来之后就会显示~\_63。这是因为14条提示中没有一条提及，那么运行的结果将会是不确定的值。因此在后续处理此类问题时，需要补充题目含有但提示没有的值，确保每一个值在语句中至少出现一次。