**程序报告**

学号： 2111033 姓名：艾明旭

1. **问题重述**

（简单描述对问题的理解，从问题中抓住主干，必填）

====================================================================

五个不同国家且工作各不相同的人分别住在一条街上的五所房子里，每所房子颜色不同，每个人都有自己养的不同的宠物，喜欢喝不同的饮料。根据提示，判断哪所房子里的人养斑马，哪所房子里的人喜欢喝矿泉水。

条件：国籍：英国、西班牙、日本、意大利、挪威

房子颜色：红色、绿色、白色、蓝色、黄色

工作：油漆工、摄影师、外交官、小提琴家、医师

宠物：狗、蜗牛、狐狸、马、斑马

饮料：茶、牛奶、咖啡、橘子汁、矿泉水

1. **设计思想**

（所采用的方法，有无对方法加以改进，该方法有哪些优化方向（参数调整，框架调整，或者指出方法的局限性和常见问题），伪代码，理论结果验证等… **思考题，非必填**）

====================================================================

一、由条件5，（1，1）为挪威，删除条件5；

二、由条件9，（5，3）为牛奶，删除条件9；

三、由条件11，（2，2）为蓝色，删除条件11；

四、由条件6，（2，1）为黄色；

五、由条件8，（3，1）为外交官，删除条件8；

六、由条件14，（4，2）为马，删除条件14；

七、由条件4、10、12，（5，1）为矿泉水；

八、由条件6，（2，3）为白色，（2，4）为绿色，（2，5）为红色，或者（2，3）为红色，（2，4）为白色，（2，5）为绿色；

不妨设（2，3）为白色，（2，4）为绿色，（2，5）为红色：

① 由条件1，（1，5）为英国，删除条件1、6；

② 由条件10，（5，4）为咖啡，删除条件10；

③ 由条件4、12，（5，5）为橘子汁，（3，5）为小提琴家，删除条件12；

④ （5，2）为茶，由条件4，（1，2）为意大利，删除条件4；

⑤ 由条件3、7，（3，2）为医师；

⑥ 由条件2、7、13，（4，5）为斑马；

⑦ 由条件3，西班牙人是摄影师，于是条件2与条件7矛盾；

因此（2，3）为红色，（2，4）为白色，（2，5）为绿色；

九、由条件1，（1，3）为英国，删除条件1、6；

十、由条件10，（5，5）为咖啡，删除条件10；

十一、（5，2）为橘子汁或茶，不妨设（5，2）为橘子汁：

① （5，4）为茶，由条件4，（1，4）为意大利，删除条件4；

② 由条件12，（3，3）为小提琴家，删除条件12；

③ 由条件2，（1，5）为西班牙，（1，2）为日本，条件3与条件12矛盾；

因此（5，2）为茶，（5，4）为橘子汁；

十二、由条件4，（1，2）为意大利，删除条件4；

十三、由条件12，（3，4）为小提琴家，删除条件12；

十四、由条件3、7，（3，3）为摄影师，（4，3）为蜗牛，删除条件7；

十五、由条件3，（1，5）为日本，（3，5）为油漆工，删除条件3；

十六、（1，4）为西班牙，由条件2，（4，4）为狗，删除条件2；

十七、由条件13，（4，1）为狐狸，（3，2）为医师，删除条件13；

十八、（4，5）为斑马。

1. **代码内容**

（能体现解题思路的主要代码，有多个文件或模块可用多个"===="隔开，必填）

====================================================================

from kanren import run, eq, membero, var, conde # kanren一个描述性Python逻辑编程系统

from kanren.core import lall # lall包用于定义规则

import time

###############################################################################

#### 可在此处定义自己所需要用到的自定义函数(可选) ####

#### 提示：定义左邻近规则left(), 定义右邻近规则right(),定义邻近规则next() ####

###############################################################################

# #

def Left(q,p,list):

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

def Right(p,q,list):

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

def next1(q,p,list):

return conde([Left(q,p,list)],[Left(p,q,list)])

# #

###############################################################################

################# 非必要性工作 ##################

###############################################################################

class Agent:

"""

推理智能体.

"""

def \_\_init\_\_(self):

"""

智能体初始化.

"""

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)

# 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

# 请基于给定的逻辑提示求解五条正确的答案

self.rules\_zebraproblem = None # 用lall包定义逻辑规则

self.solutions = None # 存储结果

def define\_rules(self):

"""

定义逻辑规则.

"""

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)

##############################################################################

#### 请在以下区域中添加逻辑规则，感受逻辑约束问题 ####

#### 输出：五条房子匹配信息('英国人', '油漆工', '茶', '狗', '红色') ####

##############################################################################

# #

(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),

(Left, (var(), var(), var(), var(), '白色'), (var(), var(), var(), var(), '绿色'),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),

(next1, ('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),

(next1, (var(), var(), var(), '狐狸', var()), (var(), '医生', var(), var(), var()),self.units),

(next1, (var(), var(), var(), '马', var()),(var(), '外交官', var(), var(), var()), self.units),

# 示例：基于问题信息可以提炼出，有人养斑马，有人喜欢和矿泉水等信息

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

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

# #

##############################################################################

################# 完成后请记得提交作业 #################

##############################################################################

)

def solve(self):

"""

规则求解器(请勿修改此函数).

return: 斑马规则求解器给出的答案，共包含五条匹配信息，解唯一.

"""

self.define\_rules()

self.solutions = run(0, self.units, self.rules\_zebraproblem)

return self.solutions

1. **实验结果**

（实验结果，必填）

====================================================================



1. **总结**

（自评分析（是否达到目标预期，可能改进的方向，实现过程中遇到的困难，从哪些方面可以提升性能，模型的超参数和框架搜索是否合理等），**思考题，非必填**）

====================================================================