

程序报告

学号：2314022

姓名：袁田

一、问题重述

(简单描述对问题的理解，从问题中抓住主干，必填)

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）。

在已知该问题中的某些关系后，利用给定的关系编写代码，通过 Python 编程将所有人的(国籍，职业，宠物，爱好饮料，房子颜色)一一对应，解决问题“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？”，也就是将结果输出。

二、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)

通过将所有给出的条件(例如：英国人住在红色的房子里)借助 Python 的一个逻辑编程包 Kanren，以代码的形式表现出来(例如：(membero('英国人',var(),var(),var(),'红色'),self.units)),将所有给出的显示条件(直接给出的条件)和隐式条件(题目中隐含的条件)全部用代码表示出来。在该问题中

三、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)

```
1. self.rules_zebraproblem = lall(
    (membero('英国人',var(),var(),var(),'红色'),self.units),
    (membero('西班牙人',var(),var(),var(),'狗',var()),self.units),
    (membero('日本人','油漆工',var(),var(),var()),self.units),
    (membero('意大利人',var(),var(),var(),'茶',var(),var()),self.units),
    (eq, (('挪威人',var(),var(),var(),var()), var(), var(), var(), var()), self.units),
    (on_right((var(),var(),var(),var(),'绿色'),(var(),var(),var(),var(),'白色'),self)),
    (membero(var(),'摄影师',var(),var(),var()),self.units).....)(部分代码省略)
这一部分是将问题中所有给出的条件用 Kanren 库中给出的成员关系，等价关系等用代码表示出来，并定义为规则集 self.rules_zebraproblem
```

```

2. def on_left(x,y,self):
    return conde([eq((x,y,var(),var(),var()),self.units)],
                 [eq((var(),x,y,var(),var()),self.units)],
                 [eq((var(),var(),x,y,var()),self.units)],
                 [eq((var(),var(),var(),x,y),self.units)])

def on_right(x,y,self):
    return conde([eq((y,x,var(),var(),var()),self.units)],
                 [eq((var(),y,x,var(),var()),self.units)],
                 [eq((var(),var(),y,x,var()),self.units)],
                 [eq((var(),var(),var(),y,x),self.units)])

def on_next(x,y,self):
    return conde([on_left(x,y,self)],[on_right(x,y,self)])

```

由于在给定的关系中，除了可用 Kanren 库中本来就有的 `membero` 和 `eq` 关系外，还需要自己写一些特定的关系函数，`on_left(x,y)`指：x 在 y 的左边；`on_right(x,y)`指：x 在 y 的右边。以 `on_left` 函数为例，此处利用(逻辑和/或的目标构造函数)这部分中的代码，将所有可能的情况或起来作为返回值，`on_right` 函数同理。而 `on_next(x,y)`函数指的是 x 和 y 相邻，也就包含两种可能的情况：`on_left(x,y)`或者 `on_right(x,y)`，因此同样利用 `conde` 函数进行逻辑或。

四、实验结果

(实验结果，必填)

```

2025-03-18 20:21:18.086500 SYSTEM: Preparing env...
2025-03-18 20:21:18.478600 SYSTEM: Running...
2025-03-18 20:21:19.335700
2025-03-18 20:21:19.335700 绿色房子里的人养斑马
2025-03-18 20:21:19.335700 黄色房子里的人喜欢喝矿泉水
2025-03-18 20:21:19.335700 ('挪威人', '外交官', '矿泉水', '狐狸', '黄色')
2025-03-18 20:21:19.335700 ('意大利人', '医生', '茶', '马', '蓝色')
2025-03-18 20:21:19.335700 ('英国人', '摄影师', '牛奶', '蜗牛', '红色')
2025-03-18 20:21:19.335700 ('西班牙人', '小提琴家', '橘子汁', '狗', '白色')
2025-03-18 20:21:19.335700 ('日本人', '油漆工', '咖啡', '斑马', '绿色')

```