

程序报告

学号：2213041

姓名：李雅帆

一、问题重述

1、问题概述：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个房子里，房子颜色各不相同，每个人喜欢的饮料和所养的宠物也各不相同。根据 14 条已知信息（不再详述），判断那所房子里的人养的斑马，那所房子里的人喜欢喝矿泉水。本实验还要求输出正确的五条匹配信息，且格式为(国家，工作，饮料，宠物，颜色)。

2、条件汇总：

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、医生
宠物：狗、蜗牛、狐狸、马、斑马
饮料：茶、牛奶、咖啡、橘子汁、矿泉水

二、设计思想

1.采用方法：使用 `kanren` 库中的逻辑规则推理来解决斑马问题。它使用了逻辑变量和规则来描述房子的不同属性，然后通过定义逻辑规则来求解符合条件的房子特征，最后通过提取结果来展示斑马和矿泉水的归属

伪代码如下：。

```
# 定义房子和相关属性的变量
units = var()
# 定义逻辑规则
rules_zebraproblem = lall(
    # 约束条件 1
    ...
    # 约束条件 2
    ...
    # 其他约束条件
    ...
)
# 使用规则求解器求解
solutions = run(0, units, rules_zebraproblem)
# 展示解的结果
for i in solutions[0]:
    print(i)
# 提取解释器的输出
output = [house for house in solutions[0] if '斑马' in house][0][4]
print("\n{}房子里的人养斑马".format(output))
```

```
output = [house for house in solutions[0] if '矿泉水' in house][0][4]
print('{}房子里的人喜欢喝矿泉水'.format(output))
```

2.对于改进，可以考虑以下方向：

- (1) 参数调整：可以尝试不同的规则定义和逻辑推理方法，以提高求解效率和准确性。
- (2) 框架调整：考虑采用其他逻辑编程框架或者搜索算法来解决类似的问题，例如采用基于约束的逻辑编程库或者深度学习方法。
- (3) 方法局限性和常见问题：kanren 库虽然能够解决类似的逻辑推理问题，但没有考虑到规则的复杂性和求解的效率。在处理大量数据时，可能会出现性能瓶颈。

三、代码内容

1.自定义之后会用到的函数： left、right、和 next 函数

构建方法核心：

将房子列表错位 zip，使得每个房子都和它旁边的房子对应打包为元组，然后返回由这些元组组成的列表。此时元组中左右元素即为相邻房子的左右顺序，使用 kanren 中的 membero，即包含逻辑关系，赋予参数 x、y 左右的位置关系。

```
def left(x,y,units):
    groups=zip(units,units[1:])
    # units 为原来的房子序列，通过切片 units[1:]使其错位一个房子，然后用 zip 打包。
    return membero((x,y),groups)
def right(x,y,units):
    return left(y,x,units)
def next(x,y,units):
    return conde([left(x, y, units)], [right(x, y, units)])
```

2.构建智能体类对象 agent，类中定义逻辑变量 units

```
self.units = var()
(eq, (var(), var(), var(), var(), var()), self.units)
# 相当于 self.units = (var, var, var, var, var)
units 中包含 5 个房子的逻辑变量，而每个房子的逻辑变量 var 又包括 5 个逻辑变量 (国家，工作，饮料，宠物，颜色)
agent 中还定义了 rules_zebraproblem 和 solutions，分别用来定义规则和存储结果。
```

3.智能体类中定义规则函数：

使用 kanren 包中的 lall 函数定义规则

membero 表示 包含关系，下例表明 红色的、住着英国人的房子 var 包含在 units 里。

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

eq 表示 相等关系，下例表明 挪威人住在左边的第一个房子里

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

4.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)
# 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),
(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)
)

```

四、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')

系统测试

 main.py

 main.ipynb

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	✓	10s	测试成功!

五、总结

就结果而言该程序实现了目标预期，以正确的格式和答案回答了斑马问题。实验运用 `kanren` 的逻辑包进行逻辑推理，最终得到结果，问题关键在于理解 `kanren` 包的使用规则和理解利用切片和 `zip` 构造 `left`、`right`、`next` 位置关系。