

# 程序报告

学号：

姓名：

## 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）

本问题是逻辑推理问题，情景为 5 个不同国家且工作不同的人分别住在一条街上的 5 所房子里，题目中给出了 14 条提示，要求是给出哪所房子里的人养斑马，哪所房子里的人喜欢喝矿泉水，题目中使用 `kanren` 包进行推理。

## 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）

采用的方法主要是依次用 `kanren` 中提供的函数对给出的 14 条自然语言进行逻辑上的翻译，主要会使用到 `eq`、`membero` 等逻辑关系，本题主要的难点在于如何用逻辑描述房子之间的位置。

注意到一开始样例代码中给出了 `(eq, (var(), var(), var(), var(), var()), self.units)` 这一 `eq` 逻辑，说明此处的 `units` 代表的是 5 所房子，里面的每个 `var()` 又代表着房子相关的 5 种属性，这里就隐含了 5 所房子的位置关系。笔者尝试用索引来查找 `units` 但对于 `kanren` 定义的 `var()` 并不行，于是笔者也通过 `eq` 关系，列出 5 个 `var()`，并将邻居关系对应上去（例如若 A 和 B 是邻居，那么 `(A, B, var(), var(), var())` 以及其他 7 种情况都可能是 A 和 B 的邻居关系，所以笔者用“或”将这些 `eq` 关系连接起来，成功描述了邻居关系）。

可以看到这种方法描述邻居关系，需要提供 8 种 `eq` 关系的或，这使得计算开销较大，可以尝试使用其他方式来描述这种关系。

## 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）

关于邻居关系的定义：

```
def left(A, B, units):
    return conde([eq((A, B, var(), var(), var()), units)],
                  [eq((var(), A, B, var(), var()), units)],
                  [eq((var(), var(), A, B, var()), units)],
                  [eq((var(), var(), var(), A, B), units)]
    )

def right(A, B, units):
    return conde([eq((B, A, var(), var(), var()), units)],
                  [eq((var(), B, A, var(), var()), units)],
```

```
[eq((var(), var(), B, A, var()), units)],
[eq((var(), var(), var(), B, A), units)]
)

def next(A, B, units):
    return conde([left(A, B, units)], [right(A, B, units)])
=====
对 14 条关系的翻译:
(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),
    (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),
    (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

(实验结果，必填)

测试成功

测试详情

测试点	状态	时长	结果
测试结果	✓	65s	测试成功!

确定

五、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**)

---

本次实验达到了目标预期，但是运行时间较长，未来可能的改进方向是优化对邻居关系的描述来提高性能。

实现过程中主要的困难是对邻居关系的描述，一开始使用索引来获取 `units` 的各个元素，但是发现这种做法对于 `var()`来说是不行的，最终重新考虑思考了实现的方式，成功描述出斑马问题。