程序报告

学号：2211312　　姓名：贾景顺

一、问题重述

逻辑编程是一种编程典范，它设置答案须匹配的规则来解决问题，而非设置步骤来解决问题。过程是“事实+规则=结果”。kanren 是 python 的一个逻辑编程包，本次实验将利用 kanren 包，尝试自主撰写逻辑规则并解决斑马问题——五个人分别具有给定的不同的房子、职业、宠物、国籍和喜好，根据 14 条对这些属性的限制，求解得出唯一的可能组合。

二、设计思想

kanren 包支持使用封装好的函数增添简单的逻辑规则如等价（eq()），属于（membero()），或和且（conde()）等，并通过 lall()进行进一步完成对规则的封装。

在本问题中，首先将推理整体 self.units 定义为基本的推理变量 var()，其中包含的五个人的信息用元组串联五个 var()变量，再通过 eq 语句赋值给 self.units。元组中的每个 var()对应单个人的信息元组 unit，其包含五个信息——房子、职业、宠物、国籍和喜好。对于这些信息的约束使用 lall 功能封装。

对于一般的约束条件，可以通过 eq 语句和 membero 语句设计相等关系或者成员关系，较为简单的完成设计。如对一个人的多个属性加以绑定的时候，可以通过 membero 语句，对对应变量赋值，将其余属性设为 var（）；表达某一个房子的绝对位置的时候，可以通过 eq 语句，对其他四个位置设为 var（）；表达两个房子的邻接规则时，本次设计使用 def 定义好了左相邻/右相邻/邻近函数，在对应的规则上进行调用，并返回对应的逻辑规则函数。设计上使用 conde 语句表达或关系，判断两个房子的位置关系是否属于各类相邻情况组成的或集中。例如在左相邻 left 函数中，判断两个房子的绝对位置关系是否属于一二/二三/三四/四五这四类情况，而 right 和 next 函数可直接通过调用 left 并进一步修改来实现。

三、代码内容

```
#邻接函数定义
def left(x,y,units):
    return conde([eq((x, y, var(), var(), var()), units)],
                 def left(a,b,self_units):
    return membero(self_units,((a,b,var(),var(),var()),
                                (var(),a,b,var(),var()),
                                (var(),var(),var(),a,b),
                                (var(),var(),var(),a,b)))

def right(a,b,self_units):
    return left(b,a,self_units)

def next(a,b,self_units):
    return conde([left(a,b,self_units)],[right(a,b,self_units)])          [eq((var(), x, y, var(), var()),
```

```
units)],
                    [eq((var(), var(), x, y, var()), units)],
                    [eq((var(), var(), var(), x, y), units)])
def near(x,y,units):
    return conde([left(x,y,units)],[left(y,x,units)])
==========================================================================
#逻辑规则补充
            (membero,(var(), var(), var(), '斑马', var()), self.units),
            (membero,(var(), var(), '矿泉水', var(), var()), self.units),
            # 以上两条规则是必要的，因为 14 条提示中 bucunzai1 对’斑马’和’矿泉水’的
约束。
            #1.
            (membero,('英国人', var(), var(), var(), '红色'), self.units),
            #2.
            (membero,('西班牙人',var(),var(),'狗',var()),self.units),
            #3.
            (membero,('日本人', '油漆工', var(), var(), var()), self.units),
            #4.
            (membero,('意大利人', var(), '茶', var(), var()), self.units),
            #5.
            (eq, (('挪威人', var(), var(), var(), var()),var(),var(),var(),var()), self.units),
            #6.
            (right,(var(), var(), var(), var(), '绿色'),(var(), var(), var(), var(), '白色'), self.units),
            #7.
            (membero,(var(), '摄像师', var(), '蜗牛', var()), self.units),
            #8.
            (membero,(var(), '外交官', var(), var(), '黄色'), self.units),
            #9.
            (eq,(var(),var(),(var(), var(),'牛奶', var(), var()),var(),var()), self.units),
            #10.
            (membero,(var(), var(), '咖啡', var(), '绿色'), self.units),
            #11.
            (next,(var(), var(), var(), var(), '蓝色'),('挪威人', var(), var(), var(), var()), self.units),
            #12.
            (membero,(var(), '小提琴家', '橘子汁', var(), var()), self.units),
            #13.
            (next,(var(), var(), var(), '狐狸', var()),(var(), '医生', var(), var(), var()), self.units),
            #14.
            (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', '摄像师', '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')

五、总结

能够根据题目要求正确的执行，并输出正确结果，框架较为合理。本实验为人工智能的第一次实验，使我初步接触到完成人工智能项目的相关知识，并对课上学到的逻辑规则及其处理有了更深的理解。

在优化性能方面，可以采用更改规则顺序，将约束性强的条件放在前面；将多个 membero 约束规则合并；添加唯一性约束减少分支；扩展框架以支持并行运行等方式进行性能优化，减少程序的运行花费。