

程序报告

一、问题重述

利用python进行逻辑编程，尝试自主撰写逻辑规则解决斑马问题。

斑马问题概述：题干给出一系列的零碎条件，从这些条件中推理出拥有斑马的人。

二、设计思路

利用python的kanren逻辑编程系统进行对题目给出的条件的形式化翻译。

三、代码内容

```
1  # kanren一个描述性Python逻辑编程系统
2  from kanren import run, eq, membero, var, conde
3  # lall包用于定义规则
4  from kanren.core import lall
5  import time
6
7
8  def left(q, p, list):
9      return membero((q,p), zip(list, list[1:]))
10 def _next(q, p, list):
11     return conde([left(q, p, list)], [left(p, q, list)])
12
13
14 class Agent:
15     """
16     推理智能体.
17     """
18
19     def __init__(self):
20         """
21         智能体初始化.
22         """
23         # 单个unit变量指代一座房子的信息(国家, 工作, 饮料, 宠物, 颜色)
24         # 例如('英国人', '油漆工', '茶', '狗', '红色')即为正确格式, 但不是本
25         题答案
26         # 请基于给定的逻辑提示求解五条正确的答案
27         self.units = var()
28
29         # 用lall包定义逻辑规则
```

```

29         self.rules_zebraproblem = None
30         # 存储结果
31         self.solutions = None
32
33     def define_rules(self):
34         """
35         定义逻辑规则.
36         """
37
38         self.rules_zebraproblem = lall(
39             # 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
40             # (国家，工作，饮料，宠物，颜色)
41             # 各个unit房子又包含五个成员属性：(国家，工作，饮料，宠物，颜色)
42             (eq, (var(), var(), var(), var(), var()), self.units),
43
44             #0.有人养斑马，有人喜欢喝矿泉水
45             (membero, (var(), var(), var(), '斑马', var()),
self.units),
46             (membero, (var(), var(), '矿泉水', var(), var()),
self.units),
47
48             #1.英国人住在红色的房子里
49             (membero, ('英国人', var(), var(), var(), '红色'),
self.units),
50
51             #2.西班牙人养了一条狗
52             (membero, ('西班牙人', var(), var(), '狗', var()),
self.units),
53
54             #3.日本人是一个油漆工
55             (membero, ('日本人', '油漆工', var(), var(), var()),
self.units),
56
57             #4.意大利人喜欢喝茶
58             (membero, ('意大利人', var(), '茶', var(), var()),
self.units),
59
60             #5.挪威人住在左边的第一个房子里
61             (eq, (('挪威人', var(), var(), var(), var()), var(),
var(), var(), var()), self.units),
62
63             #6.绿房子在白房子的右边
64             (left, (var(), var(), var(), var(), '白色'), (var(), var(),
var(), var(), '绿色'), self.units),
65
66             #7.摄影师养了一只蜗牛

```

```

67         (membero, (var(), '摄影师', var(), '蜗牛', var()),
self.units),
68
69         #8.外交官住在黄房子里
70         (membero, (var(), '外交官', var(), var(), '黄色'),
self.units),
71
72         #9.中间那个房子的人喜欢喝牛奶
73         (eq, (var(), var(), (var(), var(), '牛奶', var(), var()),
var(), var()), self.units),
74
75         #10.喜欢喝咖啡的人住在绿房子里
76         (membero, (var(), var(), '咖啡', var(), '绿色'),
self.units),
77
78         #11.挪威人住在蓝色的房子旁边
79         (_next, ('挪威人', var(), var(), var(), var()), (var(),
var(), var(), var(), '蓝色'), self.units),
80
81         #12.小提琴家喜欢喝橘子汁
82         (membero, (var(), '小提琴家', '橘子汁', var(), var()),
self.units),
83
84         #13.养狐狸的人所住的房子与医生的房子相邻
85         (_next, (var(), var(), var(), '狐狸', var()), (var(), '医
生', var(), var(), var()), self.units),
86
87         #14.养马的人所住的房子与外交官的房子相邻
88         (_next, (var(), var(), var(), '马', var()), (var(), '外交
官', var(), var(), var()), self.units),
89
90     )
91
92     def solve(self):
93         """
94         规则求解器
95         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.
96         """
97
98         self.define_rules()
99         self.solutions = run(0, self.units, self.rules_zebraproblem)
100
101         return self.solutions
102
103     #实例
104     a=Agent()
105     b=a.solve()
106     for i in range(5):
107         print(b[0][i])

```

四、实验结果

运行输出结果如下

1	('挪威人', '外交官', '矿泉水', '狐狸', '黄色')
2	('意大利人', '医生', '茶', '马', '蓝色')
3	('英国人', '摄影师', '牛奶', '蜗牛', '红色')
4	('西班牙人', '小提琴家', '橘子汁', '狗', '白色')
5	('日本人', '油漆工', '咖啡', '斑马', '绿色')

五、总结

是否达到目标预期：是

可能改进的方向：自己定义逻辑语法，提高性能

实验过程中遇到的困难：对于python语言和kanren包的不熟悉