斑马问题与八皇后问题

助教

浙江大学计算机学院

斑马问题

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）。根据以下提示，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？

1. 英国人住在红色的房子里
2. 西班牙人养了一条狗
3. 日本人是一个油漆工
4. 意大利人喜欢喝茶
5. 挪威人住在左边的第一个房子里
6. 绿房子在白房子的右边
7. 摄影师养了一只蜗牛
8. 外交官住在黄房子里
9. 中间那个房子的人喜欢喝牛奶
10. 喜欢喝咖啡的人住在绿房子里
11. 挪威人住在蓝色的房子旁边
12. 小提琴家喜欢喝橘子汁
13. 养狐狸的人所住的房子与医生的房子相邻
14. 养马的人所住的房子与外交官的房子相邻

八皇后

八皇后问题：如何能在 8*8 的国际象棋棋盘上放置八个皇后，使得任何一个皇后都无法直接吃掉其他的皇后？为了到达此目的，任两个皇后都不能处于同一条横行、纵行或斜线上。

```
  0 1 2 3 4 5 6 7
0 x - - - x - - -
1 - x - - x - - x
2 - - x - x - x -
3 - - - x x x - -
4 x x x x o x x x
5 - - - x x x - -
6 - - x - x - x -
7 - x - - x - - x
```

[3]: True

kanren简介

[kanren] (https://github.com/logpy/logpy) 是 Python 的一个逻辑编程包。

```
from kanren import run, eq, membero, var, conde        # kanren一个描述性Python逻辑编程系统
from kanren.core import lall                           # lall包用于定义规则
```

2.1.1 等价关系表达式

在该Cell块中，我们将介绍等价关系表达式eq(x, y),其意即为变量x等价于变量y.

```
# 等价关系格式一: eq(var(), value) / eq(var(), var())
x = var()                          # 变量声明，kanren的推理基于变量var进行
z = var()
run(0, x, eq(x, z), eq(z, 3))      # 规则求解器，kanren的推理通过run函数进行
                                   # 格式要求为: run(n, var(), rules,[rules, ...])
                                   # 求解指定规则下符合的变量结果
```

```
(3,)
```

```
# 等价关系格式二: (eq, var(), value) / (eq, var(), var())
x = var()
z = var()
run(0, x, (eq, x, z), (eq, z, 3))
```

```
(3,)
```

kanren简介

2.1.2 成员关系表达式

```
# 属于关系格式 membero(var(), list / tuple)
x = var()
run(0, x, membero(x, (1, 2, 3)),  # x is a member of (1, 2, 3) #x是 (1,2,3) 的成员之一
          membero(x, (2, 3, 4)))  # x is a member of (2, 3, 4) #x是 (2,3,4) 的成员之一
```

```
(2, 3)
```

2.1.3 逻辑和/或的目标构造函数

```
# 逻辑和关系格式 conde((rules, rules))
x = var()
run(0, x, conde((membero(x, (1, 2, 3)), membero(x, (2, 3, 4)))))
```

```
(2, 3)
```

```
# 逻辑或关系格式 conde([rules], [rules]))
x = var()
run(0, x, conde([membero(x, (1, 2, 3))], [membero(x, (2, 3, 4))]))
```

```
(1, 2, 3, 4)
```

kanren简介

2.1.4 定义规则集合

```
# 调用lall包定义规则集合，lall(rules, [rules, ...])
x = var()
z = var()
rules = lall(
    eq(x, z),
    eq(z, 3)
)
run(0, x, rules)
```

```
(3,)
```

```
# 调用lall包定义规则集合，lall(rules, [rules, ...])
x = var()
z = var()
rules = lall(
    (eq, x, z),
    (eq, z, 3)
)
run(0, x, rules)
```

```
(3,)
```

Thanks