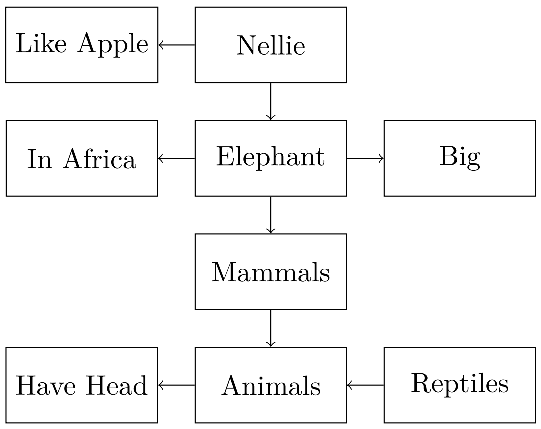
人工智能作业

骆天奇
2016254060407

2019 年 11 月 6 日

1 semantic networks



2 汉诺塔结果

4层汉诺塔:

```
[1, 1, 1, 1]
[2, 1, 1, 1]
[2, 3, 1, 1]
[3, 3, 1, 1]
[3, 3, 2, 1]
[1, 3, 2, 1]
[1, 2, 2, 1]
[2, 2, 2, 1]
[2, 2, 2, 3]
[3, 2, 2, 3]
[3, 1, 2, 3]
[1, 1, 2, 3]
[1, 1, 3, 3]
[2, 1, 3, 3]
[2, 3, 3, 3]
[3, 3, 3, 3]
```

5层汉诺塔:

```
[1, 1, 1, 1, 1]
[3, 1, 1, 1, 1]
[3, 2, 1, 1, 1]
[2, 2, 1, 1, 1]
[2, 2, 3, 1, 1]
[1, 2, 3, 1, 1]
[1, 3, 3, 1, 1]
[3, 3, 3, 1, 1]
[3, 3, 3, 2, 1]
[2, 3, 3, 2, 1]
[2, 1, 3, 2, 1]
[1, 1, 3, 2, 1]
[1, 1, 2, 2, 1]
[3, 1, 2, 2, 1]
[3, 2, 2, 2, 1]
[2, 2, 2, 2, 1]
[2, 2, 2, 2, 3]
[1, 2, 2, 2, 3]
[1, 3, 2, 2, 3]
[3, 3, 2, 2, 3]
[3, 3, 1, 2, 3]
[2, 3, 1, 2, 3]
[2, 1, 1, 2, 3]
[1, 1, 1, 2, 3]
[1, 1, 1, 3, 3]
[3, 1, 1, 3, 3]
[3, 2, 1, 3, 3]
[2, 2, 1, 3, 3]
[2, 2, 3, 3, 3]
[1, 2, 3, 3, 3]
[1, 3, 3, 3, 3]
[3, 3, 3, 3, 3]
```

6层汉诺塔:

```
[1, 1, 1, 1, 1, 1]
[2, 1, 1, 1, 1, 1]
[2, 3, 1, 1, 1, 1]
[3, 3, 1, 1, 1, 1]
[3, 3, 2, 1, 1, 1]
[1, 3, 2, 1, 1, 1]
[1, 2, 2, 1, 1, 1]
[2, 2, 2, 1, 1, 1]
[2, 2, 2, 3, 1, 1]
[3, 2, 2, 3, 1, 1]
[3, 1, 2, 3, 1, 1]
[1, 1, 2, 3, 1, 1]
[1, 1, 3, 3, 1, 1]
[2, 1, 3, 3, 1, 1]
[2, 3, 3, 3, 1, 1]
[3, 3, 3, 3, 1, 1]
[3, 3, 3, 3, 2, 1]
[1, 3, 3, 3, 2, 1]
[1, 2, 3, 3, 2, 1]
[2, 2, 3, 3, 2, 1]
[2, 2, 1, 3, 2, 1]
[3, 2, 1, 3, 2, 1]
[3, 1, 1, 3, 2, 1]
[1, 1, 1, 3, 2, 1]
[1, 1, 1, 2, 2, 1]
[2, 1, 1, 2, 2, 1]
[2, 3, 1, 2, 2, 1]
[3, 3, 1, 2, 2, 1]
[3, 3, 2, 2, 2, 1]
[1, 3, 2, 2, 2, 1]
[1, 2, 2, 2, 2, 1]
[2, 2, 2, 2, 2, 1]
[2, 2, 2, 2, 2, 3]
[3, 2, 2, 2, 2, 3]
[3, 1, 2, 2, 2, 3]
[1, 1, 2, 2, 2, 3]
[1, 1, 3, 2, 2, 3]
[2, 1, 3, 2, 2, 3]
[2, 3, 3, 2, 2, 3]
[3, 3, 3, 2, 2, 3]
[3, 3, 3, 1, 2, 3]
[1, 3, 3, 1, 2, 3]
[1, 2, 3, 1, 2, 3]
[2, 2, 3, 1, 2, 3]
[2, 2, 1, 1, 2, 3]
[3, 2, 1, 1, 2, 3]
[3, 1, 1, 1, 2, 3]
[1, 1, 1, 1, 2, 3]
[1, 1, 1, 1, 3, 3]
[2, 1, 1, 1, 3, 3]
[2, 3, 1, 1, 3, 3]
[3, 3, 1, 1, 3, 3]
[3, 3, 2, 1, 3, 3]
[1, 3, 2, 1, 3, 3]
[1, 2, 2, 1, 3, 3]
[2, 2, 2, 1, 3, 3]
[2, 2, 2, 3, 3, 3]
[3, 2, 2, 3, 3, 3]
[3, 1, 2, 3, 3, 3]
[1, 1, 2, 3, 3, 3]
[1, 1, 3, 3, 3, 3]
[2, 1, 3, 3, 3, 3]
[2, 3, 3, 3, 3, 3]
[3, 3, 3, 3, 3, 3]
```

7层汉诺塔:

```
[1, 1, 1, 1, 1, 1, 1]
[3, 1, 1, 1, 1, 1, 1]
[3, 2, 1, 1, 1, 1, 1]
[2, 2, 1, 1, 1, 1, 1]
[2, 2, 3, 1, 1, 1, 1]
[1, 2, 3, 1, 1, 1, 1]
[1, 3, 3, 1, 1, 1, 1]
[3, 3, 3, 1, 1, 1, 1]
[3, 3, 3, 2, 1, 1, 1]
[2, 3, 3, 2, 1, 1, 1]
[2, 1, 3, 2, 1, 1, 1]
[1, 1, 3, 2, 1, 1, 1]
[1, 1, 2, 2, 1, 1, 1]
[3, 1, 2, 2, 1, 1, 1]
[3, 2, 2, 2, 1, 1, 1]
[2, 2, 2, 2, 1, 1, 1]
[2, 2, 2, 2, 3, 1, 1]
[1, 2, 2, 2, 3, 1, 1]
[1, 3, 2, 2, 3, 1, 1]
```

```
[3, 3, 2, 2, 3, 1, 1]
[3, 3, 1, 2, 3, 1, 1]
[2, 3, 1, 2, 3, 1, 1]
[2, 1, 1, 2, 3, 1, 1]
[1, 1, 1, 2, 3, 1, 1]
[1, 1, 1, 3, 3, 1, 1]
[3, 1, 1, 3, 3, 1, 1]
[3, 2, 1, 3, 3, 1, 1]
[2, 2, 1, 3, 3, 1, 1]
[2, 2, 3, 3, 3, 1, 1]
[1, 2, 3, 3, 3, 1, 1]
[1, 3, 3, 3, 3, 1, 1]
[3, 3, 3, 3, 3, 1, 1]
[3, 3, 3, 3, 3, 2, 1]
[2, 3, 3, 3, 3, 2, 1]
[2, 1, 3, 3, 3, 2, 1]
[1, 1, 3, 3, 3, 2, 1]
[1, 1, 2, 3, 3, 2, 1]
[3, 1, 2, 3, 3, 2, 1]
[3, 2, 2, 3, 3, 2, 1]
[2, 2, 2, 3, 3, 2, 1]
[2, 2, 2, 1, 3, 2, 1]
[1, 2, 2, 1, 3, 2, 1]
[1, 3, 2, 1, 3, 2, 1]
[3, 3, 2, 1, 3, 2, 1]
[3, 3, 1, 1, 3, 2, 1]
[2, 3, 1, 1, 3, 2, 1]
[2, 1, 1, 1, 3, 2, 1]
[1, 1, 1, 1, 3, 2, 1]
[1, 1, 1, 1, 2, 2, 1]
[3, 1, 1, 1, 2, 2, 1]
[3, 2, 1, 1, 2, 2, 1]
[2, 2, 1, 1, 2, 2, 1]
[2, 2, 3, 1, 2, 2, 1]
[1, 2, 3, 1, 2, 2, 1]
[1, 3, 3, 1, 2, 2, 1]
[3, 3, 3, 1, 2, 2, 1]
[3, 3, 3, 2, 2, 2, 1]
[2, 3, 3, 2, 2, 2, 1]
[2, 1, 3, 2, 2, 2, 1]
[1, 1, 3, 2, 2, 2, 1]
[1, 1, 2, 2, 2, 2, 1]
[3, 1, 2, 2, 2, 2, 1]
[3, 2, 2, 2, 2, 2, 1]
[2, 2, 2, 2, 2, 2, 1]
[2, 2, 2, 2, 2, 2, 3]
[1, 2, 2, 2, 2, 2, 3]
[1, 3, 2, 2, 2, 2, 3]
[3, 3, 2, 2, 2, 2, 3]
[3, 3, 1, 2, 2, 2, 3]
[2, 3, 1, 2, 2, 2, 3]
[2, 1, 1, 2, 2, 2, 3]
[1, 1, 1, 2, 2, 2, 3]
[1, 1, 1, 3, 2, 2, 3]
[3, 1, 1, 3, 2, 2, 3]
[3, 2, 1, 3, 2, 2, 3]
[2, 2, 1, 3, 2, 2, 3]
[2, 2, 3, 3, 2, 2, 3]
[1, 2, 3, 3, 2, 2, 3]
[1, 3, 3, 3, 2, 2, 3]
[3, 3, 3, 3, 2, 2, 3]
[3, 3, 3, 3, 1, 2, 3]
[2, 3, 3, 3, 1, 2, 3]
[2, 1, 3, 3, 1, 2, 3]
[1, 1, 3, 3, 1, 2, 3]
[1, 1, 2, 3, 1, 2, 3]
[3, 1, 2, 3, 1, 2, 3]
[3, 2, 2, 3, 1, 2, 3]
[2, 2, 2, 3, 1, 2, 3]
[2, 2, 2, 1, 1, 2, 3]
[1, 2, 2, 1, 1, 2, 3]
[1, 3, 2, 1, 1, 2, 3]
[3, 3, 2, 1, 1, 2, 3]
[3, 3, 1, 1, 1, 2, 3]
[2, 3, 1, 1, 1, 2, 3]
[2, 1, 1, 1, 1, 2, 3]
[1, 1, 1, 1, 1, 2, 3]
[1, 1, 1, 1, 1, 3, 3]
[3, 1, 1, 1, 1, 3, 3]
[3, 2, 1, 1, 1, 3, 3]
[2, 2, 1, 1, 1, 3, 3]
[2, 2, 3, 1, 1, 3, 3]
[1, 2, 3, 1, 1, 3, 3]
[1, 3, 3, 1, 1, 3, 3]
```

```
[3, 3, 3, 1, 1, 3, 3]
[3, 3, 3, 2, 1, 3, 3]
[2, 3, 3, 2, 1, 3, 3]
[2, 1, 3, 2, 1, 3, 3]
[1, 1, 3, 2, 1, 3, 3]
[1, 1, 2, 2, 1, 3, 3]
[3, 1, 2, 2, 1, 3, 3]
[3, 2, 2, 2, 1, 3, 3]
[2, 2, 2, 2, 1, 3, 3]
[2, 2, 2, 2, 3, 3, 3]
[1, 2, 2, 2, 3, 3, 3]
[1, 3, 2, 2, 3, 3, 3]
[3, 3, 2, 2, 3, 3, 3]
[3, 3, 1, 2, 3, 3, 3]
[2, 3, 1, 2, 3, 3, 3]
[2, 1, 1, 2, 3, 3, 3]
[1, 1, 1, 2, 3, 3, 3]
[1, 1, 1, 3, 3, 3, 3]
[3, 1, 1, 3, 3, 3, 3]
[3, 2, 1, 3, 3, 3, 3]
[2, 2, 1, 3, 3, 3, 3]
[2, 2, 3, 3, 3, 3, 3]
[1, 2, 3, 3, 3, 3, 3]
[1, 3, 3, 3, 3, 3, 3]
[3, 3, 3, 3, 3, 3, 3]
```

3 汉诺塔代码

```
class hannuota:
    def __init__(self, size):
        self.size = size
        self.str=[]
        for i in range(size):
            self.str.append('1')
        print(str(size)+'层汉诺塔:')
        self.show()
    #  显示当前结果
    def show(self):
        num_str=[]
        for char in self.str:
            num_str.append(int(char))
        print(num_str)
    # 移动 @move_range from @source to @target
    def move(self, move_range='default', source='defaul
        if move_range == 'default':
            move_range = range(self.size)
        if source == 'default':
            source = 1
        if target == 'default':
            target = 3
        # 计算中间位置
        mid = 6 - source - target
        # 原子操作
        if move_range[0] == move_range[-1] :
            index=move_range[0]
            self.str[index]=str(target)
            self.show()
            return
        # 分离上下层
        up = move_range[:-1]
        down = move_range[-1]
        down = range(down,down+1)
        # 上层移动到中间位置
        self.move(up, source, mid)
        # 下层移动到目标位置
        self.move(down, source, target)
        # 上层移动到目标位置
        self.move(up, mid, target)
for i in range(4,8):
    #  构建i层汉诺塔
    han = hannuota(i)
    # 开始移动
    han.move()
```