简易专家系统演示（动物识别）

一个可以识别7种动物的专家系统，可以根据前提（Q）推导出结论（P），如果只有部分前提，将进行询问然后继续推导。

1.知识库和推理函数

In [1]:

```
# 输入知识库
kbase="""
有毛发 哺乳动物
有奶 哺乳动物
有羽毛 鸟
会飞 下蛋 鸟
吃肉 食肉动物
有犬齿 有爪 眼盯前方 食肉动物
哺乳动物 有蹄 有蹄类动物
哺乳动物 嚼反刍动物 有蹄类动物
哺乳动物 食肉动物 黄褐色 暗斑点 金钱豹
哺乳动物 食肉动物 黄褐色 黑色条纹 虎
有蹄类动物 长脖子 长腿 暗斑点 长颈鹿
有蹄类动物 黑色条纹 斑马
鸟 长脖子 长腿 黑灰二色 不飞 鸵鸟
鸟 会游泳 不飞 黑白二色 企鹅
鸟 善飞 信天翁
哺乳动物 食肉动物 吃老鼠 猫
"""
```

In [2]:

```
# 将产生式规则放入规则库中，if P then Q
Q = []
P = []
fo = kbase.split('\n')
for line in fo:
    line = line.strip('\n')
    if line == '':
        continue
    line = line.split(' ')
    Q.append(line[len(line) - 1])
    del (line[len(line) - 1])
    P.append(line)
```

In [3]:

```
# 输出知识库中的P和Q
print(Q)
print(P)
```

```
['哺乳动物', '哺乳动物', '鸟', '鸟', '食肉动物', '食肉动物', '有蹄类动物', '有蹄类动物', '金钱豹', '虎', '长颈鹿', '斑马', '鸵鸟', '企鹅', '信天翁', '猫']
[['有毛发'], ['有奶'], ['有羽毛'], ['会飞', '下蛋'], ['吃肉'], ['有犬齿', '有爪', '眼盯前方'], ['哺乳动物', '有蹄'], ['哺乳动物', '嚼反刍动物'], ['哺乳动物', '食肉动物', '黄褐色', '暗斑点'], ['哺乳动物', '食肉动物', '黄褐色', '黑色条纹'], ['有蹄类动物', '长脖子', '长腿', '暗斑点'], ['有蹄类动物', '黑色条纹'], ['鸟', '长脖子', '长腿', '黑灰二色', '不飞'], ['鸟', '会游泳', '不飞', '黑白二色'], ['鸟', '善飞'], ['哺乳动物', '食肉动物', '吃老鼠']]
```

In [4]:

```
# 判断list中所有元素是否都在集合set中
def ListInSet(li, se):
    for i in li:
        if i not in se:
            return False
    return True

# 判断list中至少有一个在集合set中
def ListOneInSet(li, se):
    for i in li:
        if i in se:
            return True
    return False

# 推理
def go(DB):
    str = ""
    flag = True
    temp = ""
    for x in P:  # 对于每条产生式规则
        if ListInSet(x, DB):  # 如果所有前提条件都在规则库中
            DB.add(Q[P.index(x)])
            temp = Q[P.index(x)]
            flag = False  # 至少能推出一个结论
            str += "%s --> %s" % (x, Q[P.index(x)])
            return str

    if flag:  # 一个结论都推不出
        print("一个结论都推不出，补充询问")
        for x in P:  # 对于每条产生式
            if ListOneInSet(x, DB):  # 事实是否满足部分前提
                flag1 = False        # 默认提问时否认前提
                for i in x:  # 对于前提中所有元素
                    if i not in DB:  # 对于不满足的那部分
                        btn = input("是否" + i +"? 是请输入1，否则输入0。")
                        if btn == '1':
                            DB.add(i)  # 确定则增加到规则库中
                            flag1 = True     # 肯定前提
                if flag1:  # 如果肯定前提，则重新推导
                    return go(DB)
```

2.输入事实开始推理

输入要动物的事实，如会飞、下蛋、吃肉等。不管推理是否成功，都会给出提示。

In [5]:

```
ask=[]
lines = input("输入动物事实，如“有毛发”，多个事实用“-”分开，如“鸟-长脖子”……")
lines = lines.split('-')  # 分割成组
for i in lines:
    ask.append(i)
DB = set(ask)
go(DB)
```

```
输入动物事实，如“有毛发”，多个事实用“-”分开，如“鸟-长脖子”……鸟-长脖子
一个结论都推不出，补充询问
是否有蹄类动物？是请输入1，否则输入0。0
是否长腿？是请输入1，否则输入0。1
是否暗斑点？是请输入1，否则输入0。0
一个结论都推不出，补充询问
是否有蹄类动物？是请输入1，否则输入0。0
是否暗斑点？是请输入1，否则输入0。0
是否黑灰二色？是请输入1，否则输入0。1
是否不飞？是请输入1，否则输入0。1
```

Out[5]:

```
"['鸟', '长脖子', '长腿', '黑灰二色', '不飞'] --> 鸵鸟"
```

3.其他

下面的代码，可以整理知识库，完成拓扑排序。

In [6]:

```
# 将知识库做拓扑排序，调用方式如kbase=topological(kbase)
def topological(kbase):
    Q = []
    P = []
    ans = ""  # 排序后的结果
    for line in kbase.split('\n'):
        if line == '':
            continue
        line = line.split(' ')
        Q.append(line[len(line) - 1])
        del (line[len(line) - 1])
        P.append(line)

    # 计算入度
    inn = []
    for i in P:
        sum = 0
        for x in i:
            if Q.count(x) > 0:  # 能找到，那么
                sum += Q.count(x)
        inn.append(sum)

    while (1):
        x = 0
        if inn.count(-1) == len(inn):
            break
        for i in inn:
            if i == 0:
                str = ' '.join(P[x])
                # print("%s %s" %(str, Q[x]))
                ans = ans + str + " " + Q[x] + "\n"  # 写入结果
                inn[x] = -1
                # 更新入度
                y = 0
                for j in P:
                    if j.count(Q[x]) == 1:
                        inn[y] -= 1
                    y += 1
            x += 1
    return ans
```

In [ ]:

```
```