

**姓名**：胡聿鑫

**学号**：2220192813

**班级**：计算机科学与技术3班

**选课序号**：62

实验报告

# 实验环境

Python 3.8

# 实验目的及内容

## 实验目的

熟悉掌握基于规则系统的表示与推理。

## 实验内容

设计一个动物识别专家系统，规则库至少包含15条规则，可以识别至少7种动物，规则可增加。界面显示要求如下:

1）有若干选择动物特征的选择列表；

2）表现判断动物时，使用了哪些规则;

3）表现数据库的变化；

4）显示规则的调用次序；

5）显示最后的结果，包含动物能识别出来和动物不能识别出来两种情况；

6）至少检查两个例子实现正向推理和反向推理的区别。

# 解题思路及代码

## 解题思路

### 设计数据库与规则库

该专家系统需要事先设计好规则库和事实库，并且要有合理的推理机构负责整个产生式系统的运行。该动物识别专家系统要求至少有15条规则，识别至少7种不同的动物，因此需要事先进行规划，并将动物的特征及种类等写入数据库。

由此首先给出数据库：

["可产奶", "有毛发", "有翅膀", "会飞行", "会下蛋", \

"食肉", "虎牙锋利", "四肢有利爪","眼睛在头前侧", "长有蹄", \

"能快速奔跑","有橙黄色皮肤", "有斑纹", "有黑色条纹", "有长脖子", \

"腿部肌肉发达", "陆地行走", "可下水", "黑白相间", "飞行能力强",\

"哺乳动物", "鸟类", "肉食动物", "蹄类动物", "猎豹", "老虎",\

"长颈鹿", "斑马", "鸵鸟", "企鹅", "海鸥"]

基于数据库再对规则进行设计，得到的规则库如下：

rule1 = [2] # if 动物有毛发2 then 动物是哺乳动物21

rule2 = [1] # if 动物可产奶1 then 动物是哺乳动物21

rule3 = [3] # if 动物有翅膀3 then 动物是鸟类22

rule4 = [4, 5] # if 动物会飞行4 and 会下蛋5 then 动物是鸟类22

rule5 = [6] # if 动物食肉6 then 动物是食肉动物23

rule6 = [7, 8, 9] # if 动物虎牙锋利7 and 四肢有利爪8 and 眼睛在头前侧9 then 动物是肉食动物23

rule7 = [21, 10] # if 动物是哺乳动物21 and 长有蹄10 then 动物是有蹄类动物24

rule8 = [21, 11] # if 动物是哺乳动物21 and 能快速奔跑11 then 动物是有蹄类动物24

rule9 = [21, 23, 12, 13] # if 动物是哺乳动物21 and 是食肉动物23 and 有橙黄色皮肤12 and 有斑纹13 then 动物是猎豹25

rule10 = [21, 23, 12, 14] # if 动物是哺乳动物21 and 是食肉动物23 and 有橙黄色皮肤12 and 有黑色条纹14 then 动物是老虎26

rule11 = [24, 15, 16, 13] # if动物是蹄类动物24 and 有长脖子15 and 腿部肌肉发达16 and 有斑纹13 then 动物是长颈鹿27

rule12 = [24, 14] # if 动物是蹄类动物24 and 有黑色条纹14 then 动物是斑马28

rule13 = [22, 17, 16, 15, 19] # if 动物是鸟类22 and 陆地行走17 and 有长脖子16 and 腿部肌肉发达15 and 黑白相间15 then 动物是鸵鸟29

rule14 = [22, 17, 18, 19] # if 动物是鸟类22 and 陆地行走17 and 能下水18 and 黑白相间19 then 动物是企鹅30

rule15 = [22, 4, 5] # if 动物是鸟类22 and 飞行能力强4 and 会下蛋5 then 动物是海鸥31

需要注意的是，该规则库中的1-8条均为推导动物特征的规则，而9-15条规则则是用于推导动物的种类。

### 设计推理机构

由于要求同时实现正向推理和逆向推理，因此需要设计两套推理方式。

**正向推理**：正向推理需要在选择出某些动物特征后，生成事实库，并且由此推出其他的动物特征，基于所推出的所有特征最终推导出动物的种类。具体的推理步骤为：

1. 判断输入的事实库是否包含rule 1-8的集合，若是则向事实数据库中加入相应规则所对应的事实。
2. 判断输入的事实库是否包含rule 9-15的集合，若是则输出相应规则对应的动物，否则识别失败。

**逆向推理**：与正向推理相反，逆向推理首先要输入某一动物，接着输入相应的事实数据库，然后再根据rule 9-15导出目标动物对应的事实。对此需要循环遍历事实数据库，判断输入的事实是否存在，若存在则继续遍历，否则便需要判断事实数据库中是否有推出该事实的规则，若有则继续遍历直至每个事实均可在事实数据库中找到，否则视为识别失败。

将各个规则推导出的事实进行整合，得到动物种类及其特征的关系图如下：



## 程序完整代码

#动物特征数据库

features = ["可产奶", "有毛发", "有翅膀", "会飞行", "会下蛋", \

"食肉", "虎牙锋利", "四肢有利爪","眼睛在头前侧", "长有蹄", \

"能快速奔跑","有橙黄色皮肤", "有斑纹", "有黑色条纹", "有长脖子", \

"腿部肌肉发达", "陆地行走", "可下水", "黑白相间", "飞行能力强",\

"哺乳动物", "鸟类", "肉食动物", "蹄类动物", "猎豹", "老虎",\

"长颈鹿", "斑马", "鸵鸟", "企鹅", "海鸥"]

#规则库

rule1 = [2] # if 动物有毛发2 then 动物是哺乳动物21

rule2 = [1] # if 动物可产奶1 then 动物是哺乳动物21

rule3 = [3] # if 动物有翅膀3 then 动物是鸟类22

rule4 = [4, 5] # if 动物会飞行4 and 会下蛋5 then 动物是鸟类22

rule5 = [6] # if 动物食肉6 then 动物是肉食动物23

rule6 = [7, 8, 9] # if 动物虎牙锋利7 and 四肢有利爪8 and 眼睛在头前侧9 then 动物是肉食动物23

rule7 = [21, 10] # if 动物是哺乳动物21 and 长有蹄10 then 动物是有蹄类动物24

rule8 = [21, 11] # if 动物是哺乳动物21 and 能快速奔跑11 then 动物是有蹄类动物24

rule9 = [21, 23, 12, 13] # if 动物是哺乳动物21 and 是肉食动物23 and 有橙黄色皮肤12 and 有斑纹13 then 动物是猎豹25

rule10 = [21, 23, 12, 14] # if 动物是哺乳动物21 and 是肉食动物23 and 有橙黄色皮肤12 and 有黑色条纹14 then 动物是老虎26

rule11 = [24, 15, 16, 13] # if动物是蹄类动物24 and 有长脖子15 and 腿部肌肉发达16 and 有斑纹13 then 动物是长颈鹿27

rule12 = [24, 14] # if 动物是蹄类动物24 and 有黑色条纹14 then 动物是斑马28

rule13 = [22, 17, 16, 15, 19] # if 动物是鸟类22 and 陆地行走17 and 有长脖子15 and 腿部肌肉发达16 and 黑白相间19 then 动物是鸵鸟29

rule14 = [22, 17, 18, 19] # if 动物是鸟类22 and 陆地行走17 and 能下水18 and 黑白相间19 then 动物是企鹅30

rule15 = [22, 4, 5] # if 动物是鸟类22 and 飞行能力强20 and 会下蛋5 then 动物是海鸥31

print('首先给出数据库中的动物特征：')

i = 0

while i < 24:

print('%d' %(i+1) +'.'+ features[i]+ ' ', end='')

i = i+1

if i % 8 == 0:

print('\n')

print('该专家系统可识别动物如下：')

while i < 31:

print('%d' %(i+1) +'.'+ features[i]+ ' ', end='')

i = i+1

flag= int(input('\n请选择推理方式\n1:正向推理\n2:反向推理\n'))

if flag==1:

answer = input('\n请选择特征标号，用空格隔开，回车结束输入：')

# 接收到的answer是一个字符串

answer = list(answer.split())

new\_answer = [int(x) for x in answer]

print("事实库：",new\_answer)

print("正向推理过程如下：")

if set(rule1)<=set(new\_answer):

print('rule1：2->21\tif 动物有毛发 then 动物是哺乳动物')

new\_answer.append(21)

if set(rule2)<=set(new\_answer):

print('rule2：1->21\tif 动物可产奶 then 动物是哺乳动物')

new\_answer.append(21)

if set(rule3)<=set(new\_answer):

print('rule3：3->22\tif 动物有翅膀 then 动物是鸟类')

new\_answer.append(22)

if set(rule4)<=set(new\_answer):

print('rule4：4+5->22\tif 动物会飞行 and 会下蛋 then 动物是鸟类')

new\_answer.append(22)

if set(rule5)<=set(new\_answer):

print('rule5：6->23\tif 动物食肉 then 动物是肉食动物')

new\_answer.append(23)

if set(rule6)<=set(new\_answer):

print('rule6：7+8+9->23\tif 动物虎牙锋利 and 有利爪 and 眼睛在头前侧 then 动物是肉食动物')

new\_answer.append(23)

if set(rule7)<=set(new\_answer):

print('rule7：21+10->24\tif 动物是哺乳动物 and 长有蹄 then 动物是蹄类动物')

new\_answer.append(24)

if set(rule8)<=set(new\_answer):

print('rule8：21+11->24\tif 动物是哺乳动物 and 能快速奔跑 then 动物是蹄类动物')

new\_answer.append(24)

print("正向推理结果为：",new\_answer)

if set(rule9)<=set(new\_answer):

print("if 动物是哺乳动物 and 是肉食动物 and 有橙黄色皮肤 and 有斑纹 then 动物是猎豹")

print("结果为：", end=" ")

print(features[24])

elif set(rule10)<=set(new\_answer):

print("if 动物是哺乳动物 and 是肉食动物 and 有橙黄色皮肤 and 有黑色条纹 then 动物是老虎")

print("结果为：", end=" ")

print(features[25])

elif set(rule11)<=set(new\_answer):

print("if动物是蹄类动物 and 有长脖子 and 腿部肌肉发达 and 有斑纹 then 动物是长颈鹿")

print("结果为：", end=" ")

print(features[26])

elif set(rule12)<=set(new\_answer):

print("if 动物是蹄类动物 and 有黑色条纹 then 动物是斑马")

print("结果为：", end=" ")

print(features[27])

elif set(rule13)<=set(new\_answer):

print("if 动物是鸟类 and 陆地行走 and 有长脖子 and 腿部肌肉发达 and 黑白相间 then 动物是鸵鸟")

print("结果为：", end=" ")

print(features[28])

elif set(rule14)<=set(new\_answer):

print("if 动物是鸟类 and 陆地行走 and 会下水 and 黑白相间 then 动物是企鹅")

print("结果为：", end=" ")

print(features[29])

elif set(rule15)<=set(new\_answer):

print("if 动物是鸟类 and 飞行能力强 and 会下单 then 动物是海鸥")

print("结果为：", end=" ")

print(features[30])

else:

print('识别失败！')

elif flag==2:

animal=int(input('\n请选择要识别的动物标号：'))

answer = input('\n请选择相应的动物特征标号，用空格隔开，回车结束输入：')

# 接收到ans的wer是一个字符串

answer = list(answer.split())

new\_answer = [int(x) for x in answer]

print("事实库：", new\_answer)

print("逆向推理过程如下：")

real=[]

if animal==25:

print(features[animal-1],'rule9：25->21,23,12,13 if 动物是哺乳动物 and 是肉食动物 and 有橙黄色皮肤 and 有斑纹 then 动物是猎豹')

real=rule9

elif animal==26:

print(features[animal-1],'rule10：26->21,23,12,114 if 动物是哺乳动物 and 是肉食动物 and 有橙黄色皮肤 and 有黑色条纹 then 动物是老虎')

real=rule10

elif animal==27:

print(features[animal-1],'rule11：27->24, 15, 16, 13 if动物是蹄类动物 and 有长脖子and 腿部肌肉发达 and 有斑纹 then 动物是长颈鹿')

real=rule11

elif animal==28:

print(features[animal-1],'rule12：28->24, 14 if 动物是蹄类动物 and 有黑色条纹 then 动物是斑马')

real=rule12

elif animal==29:

print(features[animal-1],'rule13：29->22, 17, 16, 15, 19 if 动物是鸟类 and 陆地行走 and 有长脖子and 腿部肌肉发达 and 黑白相间 then 动物是鸵鸟')

real=rule13

elif animal==30:

print(features[animal-1],'rule14：30->22, 17, 18, 19 if 动物是鸟类 and 陆地行走 and 能下水 and 黑白相间9 then 动物是企鹅')

real=rule14

elif animal==31:

print(features[animal-1],'rule15：31->22, 4, 5 if 动物是鸟类 and 飞行能力强 and 会下蛋 then 动物是海鸥')

real=rule15

key=0

for i in real:

if i in new\_answer:

print(i)

continue

elif i ==21:

if set(rule1)<=set(new\_answer) :

print("rule1:21->2 if 动物有毛发 then 动物是哺乳动物")

continue

elif set(rule2)<=set(new\_answer):

print("rule2:21->1 if 动物可产奶 then 动物是哺乳动物")

continue

elif i==22:

if set(rule3) <= set(new\_answer):

print("rule3:22->3 if 动物有翅膀 then 动物是鸟类")

continue

elif set(rule4) <= set(new\_answer):

print("rule4:22->4,5 if 动物会飞行 and 会下蛋 then 动物是鸟类")

continue

elif i == 23:

if set(rule5) <= set(new\_answer):

print("rule5:23->6 if 动物食肉 then 动物是肉食动物")

continue

elif set(rule6) <= set(new\_answer):

print("rule6:23->7,8,9 if 动物虎牙锋利 and 有利爪 and 眼睛在头前侧 then 动物是肉食动物")

continue

elif i == 24:

if set(rule7) <= set(new\_answer) :

print("rule7:24->21,10 if 动物是哺乳动物 and 长有蹄 then 动物是蹄类动物")

continue

elif set(rule8) <= set(new\_answer):

print("rule8:24->21,11 if 动物是哺乳动物 and 能快速奔跑 then 动物是蹄类动物")

continue

else:

key=1

if key==0:

print("推导成功！")

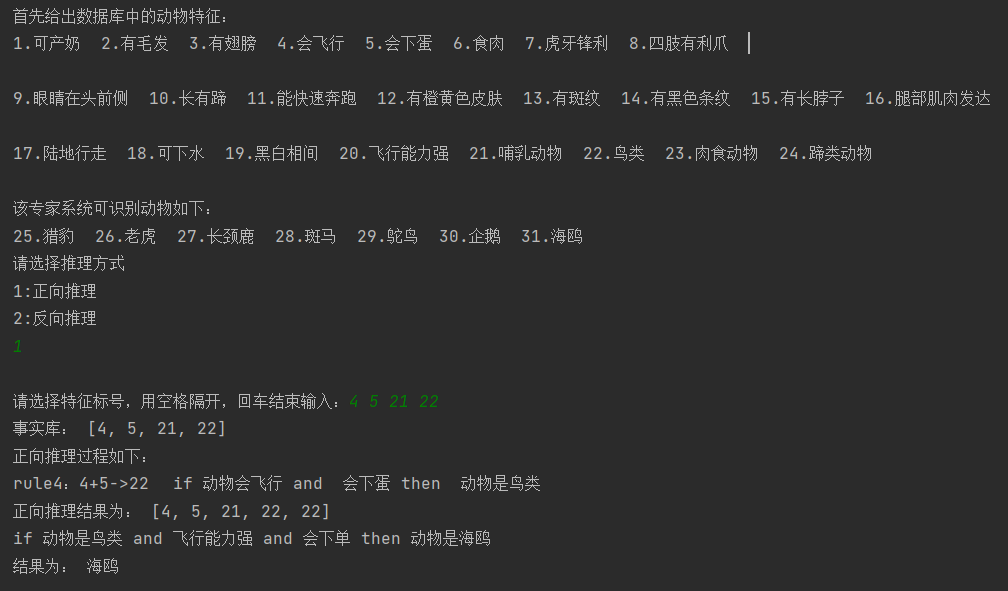
elif key==1:

print("推导失败！")

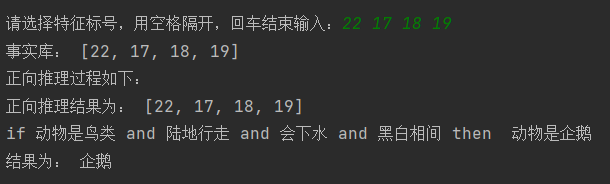
# 运行结果分析

## 正向推理结果

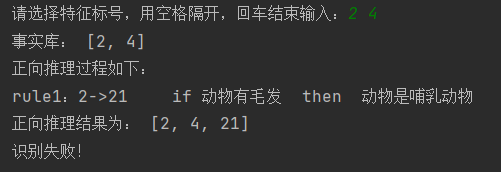
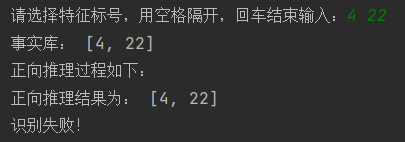
根据提示选择正向推理（1），并给出某一动物的特征标号，例如输入4 5 21 22则推理得到的动物种类为海鸥，结果如下图所示：



若选择特征标号为22 17 18 19，则可以推理出企鹅，结果如下图所示：



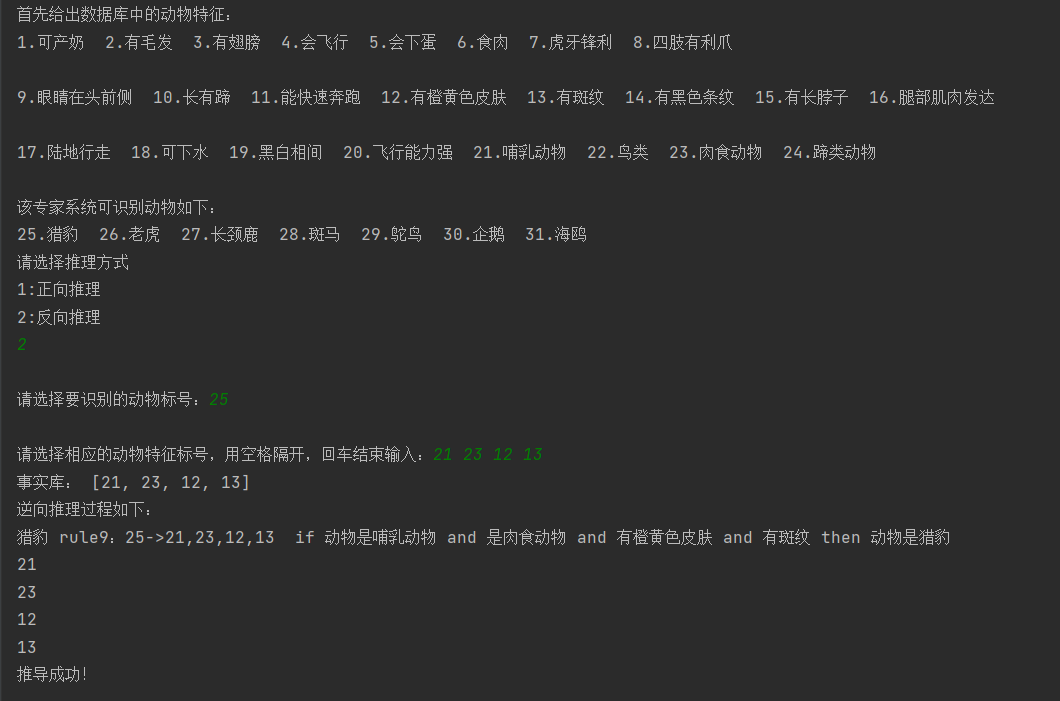
需要注意的是，输入的特征标号需要能够完整且可以唯一地推导出某一动物，特征缺失或是输入有误，均会导致推理失败。

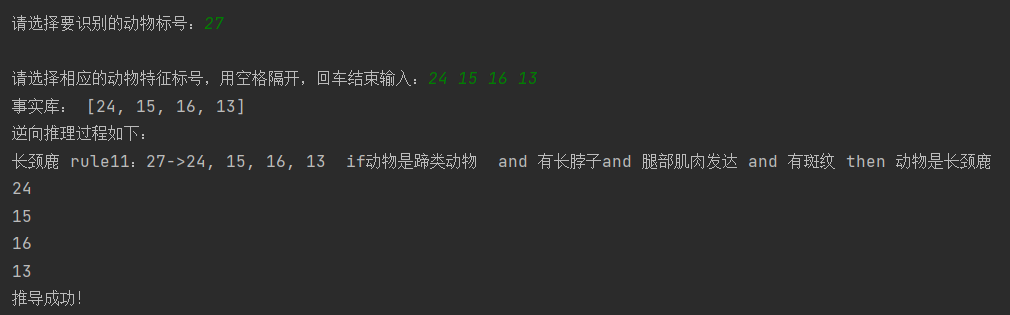
（输入错误示意图） （输入缺失示意图）

## 逆向推理结果

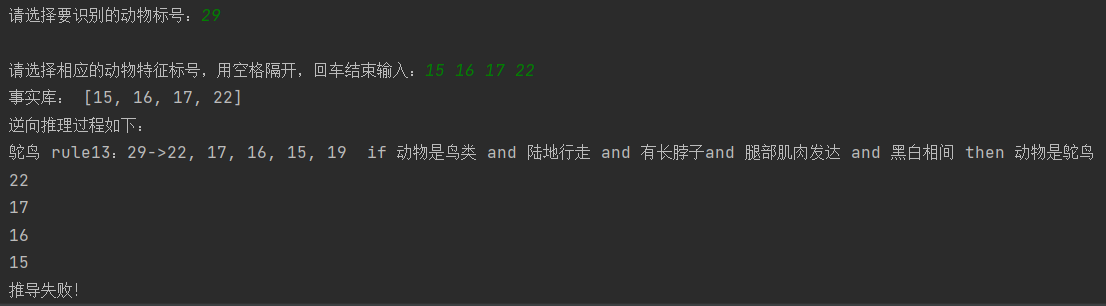
根据提示选择逆向推理（2），并给出目标动物的标号，例如输入25则对应动物种类为猎豹，再输入相应的动物特征标号21 23 12 13，即可显示逆向推理过程，结果如下图所示：



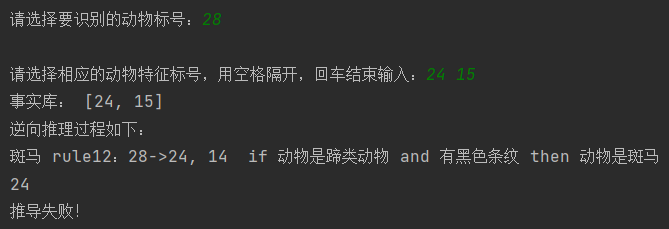
若想得到长颈鹿的相关推导过程，则需要先输入动物种类标号27，再输入其相应的特征标号24 15 16 13，即可得到逆向推理结果，如下图所示：



此时如果输入的动物特征缺失或者输入动物特征与目标动物不匹配，尽管会导致推导失败，但在编写程序时已经设定输入某一动物种类，便显示相应的推导规则，因此可以从规则中看出动物对应的全部特征，用户若不清楚动物对应的特征，可以根据提示得到所有正确的特征标号，再次输入时即可得到正确的推导结果。



（输入特征缺失时结果示意图）



（输入特征与目标不对应时结果示意图）

# 实验总结

## 心得体会

此次实验使得我对于基于规则系统的表示与推理更加熟练，通过编写动物识别专家系统的相关程序，我对于正向推理和反向推理的理解也更为深入。

开始实验时，对于设计特征数据库和规则库无从下手，因而参考了网上大量的资料，借鉴他人的设计思路，并画出了动物与其相应特征的关系图，通过关系图逐条设计规则，最终完成了对于特征综合数据库和规则库的设计。此过程极大地锻炼了我的逻辑思维能力和推理归纳能力，收获颇丰，受益匪浅。

## 实验的不足与改进

1. 实验的规则设计不够完整，部分特征有重复，例如“会飞”和“飞行能力强”之间并没有特别明显的区别。为此在设计规则库时要尽量避免出现界限模糊的规则，要尽量使用明确的词语描述。
2. 由于在设计规则时就已经尽量避免出现冲突的规则，因而程序里没有设计规则冲突的处理，但这并不能百分百保证不出现冲突，要想改进可以在程序中利用FIRST法解决冲突。
3. 输入界面较为死板，每次需要锁定输入位置，并且正向推理时需要记住动物所对应的所有特征或者根据规则库进行输入，不够灵活。可以借助pycharm，实现输入界面的可视化，并在每次输入时给出更多的提示，以便于用户输入。