题目一：常见的正则表达式

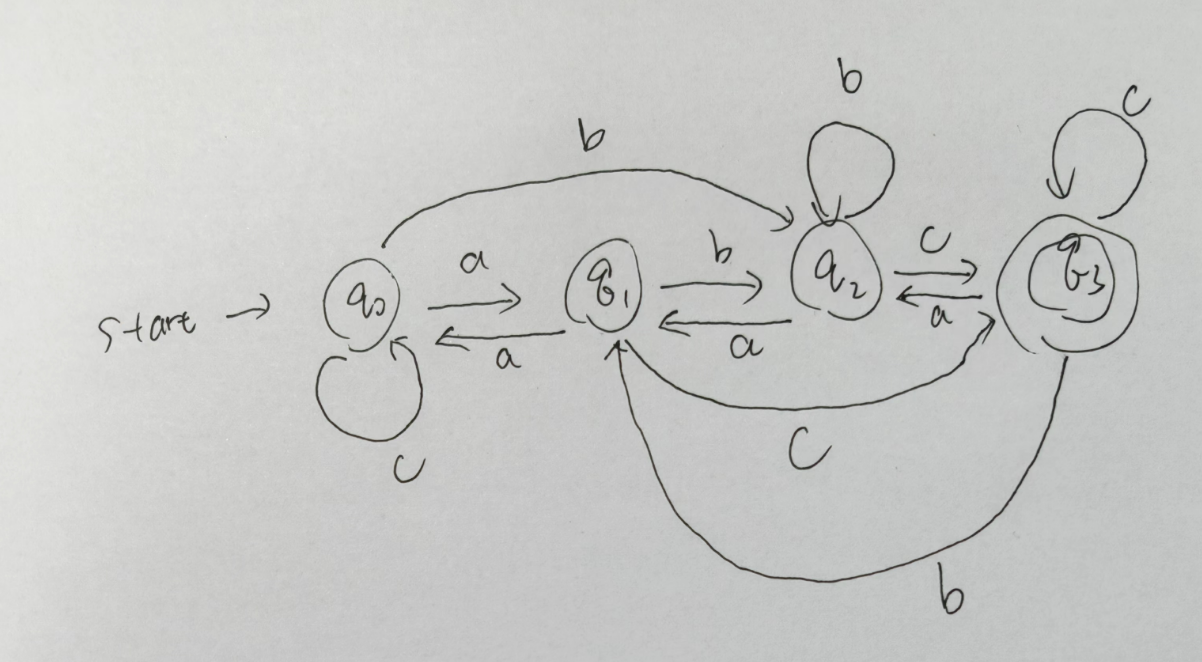
题目：  
请列举并解释以下几种常见的正则表达式模式：  
1. 匹配163或126电子邮件地址的正则表达式，其中@前面可以是数字或大小写字母。  
2. 匹配至少一个数字的正则表达式。  
3. 匹配以字母开头且长度为5的单词。

# 参考答案：

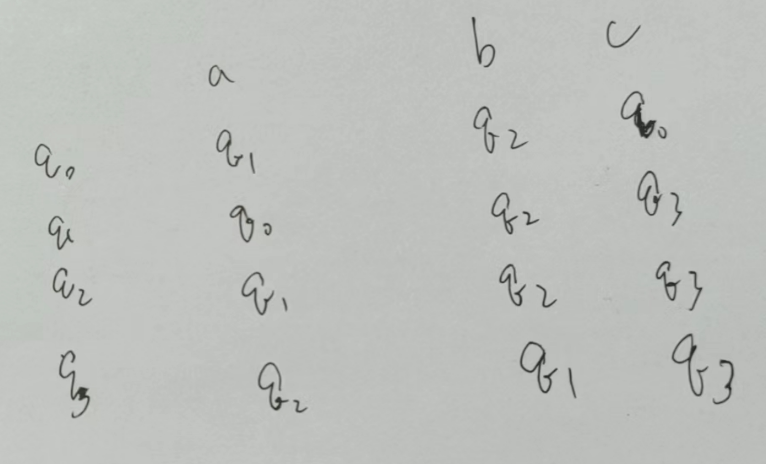
1. 匹配电子邮件地址的正则表达式：  
   /^[a-zA-Z1-9\_\-]+@[163,126].com$/

/^[a-zA-Z1-9\_\-]+@(163|126).com$/  
解释：匹配一个标准的电子邮件地址，包括字母、数字和特殊字符（如"-"或"\_"），@符号后接域名，域名部分由若干个字符组成。  
  
2. 匹配至少一个数字的正则表达式：  
/\d+/  
解释：\d表示任意数字，+表示至少出现一次。  
  
3. 匹配以字母开头且长度为5的单词：  
/^[a-zA-Z]{5}$/  
解释：^[a-zA-Z]表示以字母开头，{5}表示总共五个字符。

题目二：DFA图和其定义

题目：  
给出以下DFA的图，请写出该DFA的定义（五元组）：  


# 参考答案：

DFA定义五元组为：  
M = (Q, Σ, δ, q\_0, F)  
1. 状态集 Q = {q\_0, q\_1, q\_2, q\_3}  
2. 输入字母表 Σ = {a, b, c}  
3. 转移函数 δ 矩阵定义如下：  
  
4. 初始状态 q\_0  
5. 接受状态集 F = {q\_3}

题目三：下推自动机和DFA的区别

题目：  
请简要说明下推自动机（PDA）与确定性有限自动机（DFA）之间的主要区别，并举例说明。

# 参考答案：

1. \*\*内存结构\*\*：  
- \*\*DFA\*\* 具有有限的状态集，没有额外的存储结构。  
- \*\*PDA\*\* 具有一个栈（无限深度），允许其在计算过程中存储和回溯信息。  
  
2. \*\*语言识别能力\*\*：  
- \*\*DFA\*\* 只能识别正则语言。  
- \*\*PDA\*\* 能识别更复杂的上下文无关语言（如括号匹配问题）。  
  
3. \*\*应用举例\*\*：  
- \*\*DFA\*\* 可以用于简单的模式匹配，如词法分析中的标识符识别。  
- \*\*PDA\*\* 用于解析上下文无关语法，如编译器中的语法分析器。

题目四：N-gram分析与概率计算

题目：  
给定以下完整语料库：  
1. "I love natural language processing"  
2. "language processing is fun"  
3. "natural language understanding is important"

请：

1.列举语料库中所有的2-gram和3-gram。

2.计算其中2-gram P(is | processing)和3-gram P(processing | natural language)的概率。

# 参考答案：

# ① 请列举语料库中所有的2-gram和3-gram

## 2-gram:

I love  
love natural  
natural language  
language processing  
processing is  
is fun  
natural language  
language understanding  
understanding is  
is important

## 3-gram:

I love natural  
love natural language  
natural language processing  
language processing is  
processing is fun  
natural language understanding  
language understanding is  
understanding is important

# ② 计算其中2-gram P(is | processing)和3-gram P(processing | natural language)的概率

## 2-gram的概率:

P(is | processing) = 1/2 = 0.5

## 3-gram的概率:

P(processing | natural language) = 1/2 = 0.5

题目五：Good-Turing平滑概率计算

题目：  
在某个语料库中，单词的出现次数如下：  
出现1次的单词有100个  
出现2次的单词有60个  
出现3次的单词有30个  
  
请：  
1. 计算从未见过的单词的概率。  
2. 计算出现1次、2次单词的平滑概率。

# 参考答案：

## 1. 计算从未见过的单词的概率：

根据Good-Turing公式：  
P(unseen) = N1 / N\_total  
其中，N1是出现1次的单词的数量，N\_total是语料库中的总单词数。根据单词出现次数计算总单词数为：  
N\_total = 1 × 100 + 2 × 60 + 3 × 30 = 310  
  
因此，从未见过单词的概率为：  
P(unseen) = 100 / 310 ≈ 0.323

## 2. 计算出现1次、2次单词的平滑概率：

使用Good-Turing平滑公式，平滑后的概率计算如下：  
P\*(r) = ((r+1) × N\_{r+1}) / (N\_r × N\_total)  
其中，N\_r 是出现 r 次的单词的数量，N\_{r+1} 是出现 r+1 次的单词的数量，N\_total 是语料库中的总单词数（310）。  
  
P(1次) = ((1+1) × 60) / (100 × 310) = 120 / 31000 ≈ 0.387  
P(2次) = ((2+1) × 30) / (60 × 310) = 90 / 18600 ≈ 0.161

题目六：Zero问题与Unseen问题

题目：  
请解释在N-gram模型中，“Zero概率问题”和“Unseen问题”的区别，并简要说明各自的解决方法。

# 参考答案：

## Zero概率问题：

定义：在训练数据集中，某些N-gram组合从未出现过，导致在模型中被分配为0概率。这意味着这些组合在测试数据中一旦出现，模型将会认为这种情况不可能发生，直接导致该句子或片段的概率为0。  
  
解决方法：为了解决这个问题，可以使用平滑技术。例如：  
1. 加1平滑（Laplace平滑）：通过给所有可能的N-gram组合都加上一个小的正数（如1），避免出现0概率。  
2. Good-Turning平滑，Witten-Bell平滑，Kneser-Ney平滑：一种更为先进的平滑技术，它不仅考虑单词的出现频率，还考虑上下文中不同词的多样性。

## Unseen问题：

定义：Unseen问题是指在测试数据中出现了训练集中没有见过的词或N-gram组合。这种情况通常出现在词汇表不完整或训练数据量不足的情况下，导致模型无法处理这些未见过的词。  
  
解决方法：  
1. <UNK>处理：在训练数据中，将所有出现次数较少或不在词汇表中的词替换为<UNK>，并为<UNK>训练出一个概率分布。在测试阶段，任何未知词都会用<UNK>的概率进行处理。  
2. 开放词汇任务（Open Vocabulary Task）：这种任务允许出现词汇表外的词，通过对未知词进行建模来应对Unseen问题。

## 区别：

1. Zero概率问题发生在训练集中未见过的N-gram组合，在模型中被直接赋值为0概率。  
2. Unseen问题涉及测试集中出现的词或N-gram组合在训练集中从未见过，模型无法对其进行处理。