Q1:  
Write a regular expression for the set of strings over {a,b}∗ which have an even number of a's followed by an odd number of b's. For example, it should accept:

* b
* bbb
* aab

(aa)\*(b(bb)\*)  
(aa)∗(aa)^\*(aa)∗ ：確保有偶數個 aaa ，包括根本沒有 aaa 的可能性。  
b(bb)∗b(bb)^\*b(bb)∗ ：確保 bbb 's 為奇數，從 1 bbb 開始，並通過添加成對的 bbb '來允許任何奇數擴展。  
  
Q2:   
Describe in words the pairs of strings that the following FST accepts.  Give three examples of pairs of strings that it accepts. (Note: the start state is q0, and the accept state is q0).

一張含有 寫生, 圓形, 圖表, 圖畫 的圖片

自動產生的描述

This FST accept a pair of input, which b replaced with a and then a replace with b. Since it only accpet a string at a node. It only accept even pair of (ba). The start state is also accept state, which means it also accept empty string.  
e.g.:  
1.input:{b,a}, optput:{a,b}  
2.input:{b,a,b,a}, output:{a,b,a,b}  
3.input:{},output:{}  
  
Q3:  
1.What is the formal definition of a language?  
2.What is the linguistic definition of a grammar?

Formal Definition of a Language: A language is a set of strings of symbols that may be constrained by certain rules. In formal language theory, these symbols are taken from a finite set known as the alphabet. A language is any subset of the set of all possible strings over its alphabet.

Linguistic Definition of a Grammar: A grammar is a set of structural rules governing the composition of clauses, phrases, and words in any given natural language. However, in the context of formal language theory, a grammar is a formal system specifying a set of strings in a formal language, which consists of production rules that describe how to form strings from the language's alphabet that are valid according to the language's syntax.

Q4:

What is morphology?  Give two examples of morphological phenomena in some language.

Morphology: study of words and how they are formed  
Inflection: walking  
Compounding: baseball

Q5:

Consider the following lowercased, tokenized sentence, with <START> and <STOP> symbols:  
<START> we also know there are known unknowns ; there are things we know we don't know . <STOP>  
Using a bigram language model with probabilities estimated from the above sentence, compute the probability of the following sentence:  
<START> we know . <STOP>  
Show your work. You can leave your final answer as fractions multiplied together. You do not need to multiply numbers to get a final probability.

P(we|<START>)\*P(know|we)\*P(.|know)\*P(<STOP>|.)  
"<START> we" occurs 1 time. "<START>" occurs 1 time.  
"we know" occurs 1 times. "we" occurs 3 times.  
"know ." occurs 1 time. "know" occurs 3 times.  
". <STOP>" occurs 1 time. "." occurs 1 time.  
1/1\*1/3\*1/3\*1/1=1/9

Q6:

1. In the type vs token distinction, what are types?  What are tokens?

 **Types**: Types refer to the unique words in a text or corpus. EX: "The cat sat on the mat," each distinct word (like "the," "cat," "sat," "on," "mat") counts as a type, so there are five types in this example.  
**類型**：類型是指文本或語料庫中的唯一單詞。例如，在句子 「The cat sat on the mat」 中，每個不同的單詞（如 」the“、”cat“、”sat“、”on“、”mat“）都算作一個類型，因此此示例中有五種類型。

 **Tokens**: Tokens are all instances of words in a text, counting every occurrence. In the sentence "The cat sat on the mat," there are six tokens (since "the" appears twice). So, tokens represent the actual number of words, including repetitions.**標記**：標記是文本中單詞的所有實例，對每次出現進行計數。在句子 「The cat sat on the mat」 中，有六個標記（因為 “the” 出現了兩次）。因此，標記表示實際的單詞數，包括重複次數。

1. What is Zipf's law?

Zipf's Law is an empirical observation about the frequency distribution of words in natural language. According to Zipf's Law, if we rank words in a large corpus by frequency, the frequency of a word is inversely proportional to its rank. Mathematically, it states that the frequency of the word at rank rrr is roughly proportional to 1r\frac{1}{r}r1​. This means that the most common word will appear twice as often as the second most common word, three times as often as the third, and so on. This results in a few words (like "the," "is," "a") being extremely common, while the vast majority of words are rare.齊普夫定律是對自然語言中單詞頻率分佈的實證觀察。根據齊普夫定律，如果我們按頻率對大型語料庫中的單詞進行排名，則單詞的頻率與其排名成反比。在數學上，它指出單詞 at rank rrr 的頻率大致與 1r\frac{1}{r}r1​ 成正比。這意味著最常用的單詞的出現頻率是第二常用單詞的兩倍，是第三個最常見單詞的三倍，依此類推。這導致少數單詞（如 “the”、“is”、“a”）非常常見，而絕大多數單詞很少見。

1. Describe a problem that Zipf's law causes when building a language model.  Describe a method that can help fix this problem.

Zipf's Law means that while a small number of words occur very frequently, a large number of words appear very infrequently, possibly only once (known as "hapax legomena"). This long tail of rare words can create challenges for language models, especially when predicting or understanding less frequent words. Rare words lack sufficient data for reliable probability estimation, making it difficult for the model to generalize well. This can lead to poor performance in tasks involving rare or domain-specific vocabulary.齊普夫定律意味著，雖然少量單詞出現非常頻繁，但大量單詞出現的頻率非常低，可能只出現一次（稱為“hapax legomena”）。這種長長的生僻詞可能會給語言模型帶來挑戰，尤其是在預測或理解不太常見的單詞時。稀有詞缺乏足夠的數據來進行可靠的概率估計，這使得模型難以很好地泛化。這可能會導致在涉及稀有詞彙或特定於域的詞彙的任務中性能不佳。

**Solution：**

One effective method to address this problem is **smoothing**. Smoothing techniques, such as **Laplace smoothing** or **Good-Turing smoothing**, adjust the probability distribution to account for unseen or rare words, assigning them a small probability rather than zero. Another solution is **subword tokenization** methods like **Byte-Pair Encoding (BPE)** or **WordPiece**, which break down rare words into smaller, more common subunits, allowing the model to better handle rare words and improve generalization without relying solely on whole-word frequencies. This approach is especially useful in modern language models where subwords (morphemes) help reduce sparsity caused by rare word occurrences.解決此問題的一種有效方法是**平滑**。平滑技術（如**拉普拉斯平滑**或 **Good-Turing 平滑）**調整概率分佈以考慮看不見或稀有的單詞，為它們分配一個小概率而不是零。另一種解決方案是位元組**對編碼 （BPE）** 或 **WordPiece** 等**子詞分詞化**方法，它將稀有詞分解為更小、更常見的子單元，使模型能夠更好地處理稀有詞並提高泛化能力，而無需僅依賴全詞頻率。這種方法在現代語言模型中特別有用，其中子詞（語素）有助於減少由罕見單詞出現引起的稀疏性。