CS6320, Spring 2021
Dr. Mithun Balakrishna
Homework 2
Due Friday, March 12th, 2021 11:59pm

A. Submission Instructions:

- Submit your solutions via eLearning.
- Please submit a single zip file containing **ALL** the relevant homework solution files. The zip filename should follow the pattern "HW#_FirstnameLastname.zip" (Example: HW3_ClaireUnderwood.zip)
    - **Penalty of 5 points** if not followed
- For all non-programming questions:
    - Please include **ALL** the solutions in a **single** PDF/Doc/PS/Image file
    - The filename should follow the pattern "HW#_FirstnameLastname.FileExtension" (Example: HW3_ClaireUnderwood.pdf)
    - **Penalty of 5 points** if not followed
- For programming questions:
    - Write the programming solutions in C/C++, Java, or Python. For using any other programming language, please get prior approval from the TA.
    - Include a Readme file with instructions on how to build and run your programming question solution
        - Instructions should be very simple:
            - python bigram.py input_arguments

              OR

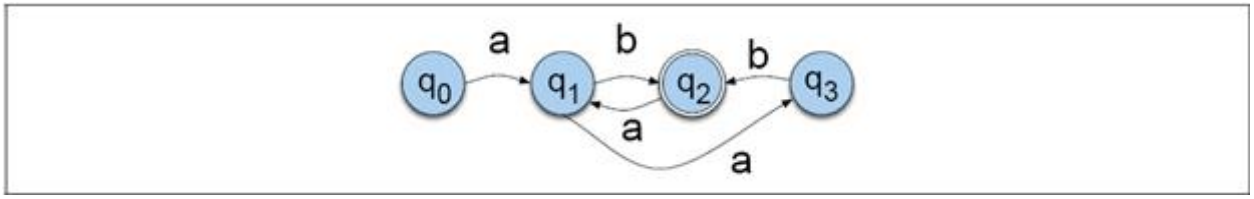
              python bigram.py (if the input arguments are hard coded)
        - Hard coding the input arguments to your program is fine unless the TA cannot run your code directly. Do **NOT** include instructions such as: "Please modify the path in my main function. Then copy the training data in the same folder."
        - Provide your training data together unless the dataset is too large.
        - **Penalty of 10 points** if not followed
    - Submit ALL your source code files
        - Do not write your solutions in the readme file
        - **Penalty of 10 points** if not followed
- Late Submission Penalty:
    - up to 2 hours late — 10% deduction
    - 2 - 4 hours late — 20% deduction
    - 4 - 12 hours late — 35% deduction
    - 12 - 24 hours late — 50% deduction
    - 24 - 48 hours late — 75% deduction
    - more than 48 hours late — 100% deduction (zero credit)

B. Problems:

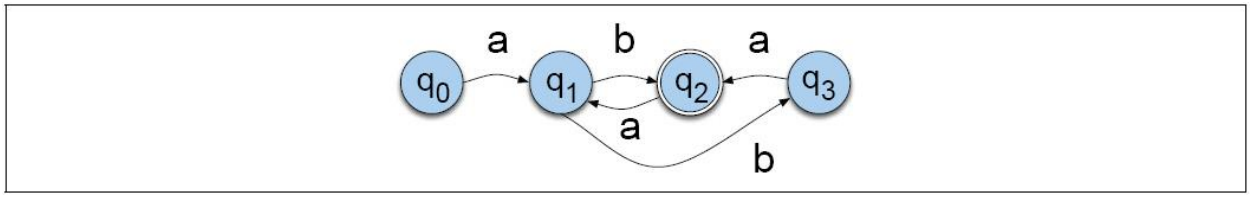
1. NFSA to Regular Expression (20 points)

Please note that **ONLY** operators presented in the Lectures can be used to answer Regex questions in the homeworks and exams.

a. (**10 points**) Write a regular expression for the language accepted by the FSA:



b. (**10 points**) Write a regular expression for the language accepted by the NFSA:



2. Bigram Probabilities – By-Hand (20 points)

a. The following text is being used as a training corpus:

<s> a man a man a man a plan a plan a canal panama panama </s>

Build a word-based bigram model from this training corpus. Show all the bigrams and their frequency (count).

Note: *<s>* is symbol for start-of-sentence and *</s>* is the symbol for end-of-sentence.

b. Using the above trained model, compute the bigram based probability of the following test sentence

<s> plan a panama </s>

under the following three (3) scenarios:

i. No Smoothing
ii. Add-one Smoothing
iii. Good-Turing Discounting based Smoothing
    a. Note: Please the set the Good Turing smoothed count C*_(c)=0 if N_(c+1)=0

Note: Smoothing performed only for the bigram model.

3. Bigram Probabilities – Programmatically (60 points):

a. Write a computer program to compute the bigram model (counts and probabilities) on the given corpus (*NLP6320_POSTaggedTrainingSet.txt)* provided as Addendum to this homework on eLearning) under the following three (3) scenarios:

iv. No Smoothing
v. Add-one Smoothing
vi. Good-Turing Discounting based Smoothing
    a. Note: Please the set the Good Turing smoothed count C*_(c)=0 if N_(c+1)=0

Note: Smoothing performed only for the bigram model.

b. Write a computer program to compute the bigram based probability of any input test sentence using the above trained model(s)

Note:

a. Your program should accept one of the above specified smoothing "type" as an input argument.
b. Smoothing performed only for the bigram model.

Other Instructions:

1. Use each line (ending with newline character) in the corpus as a single text sentence.

2. Use whitespace (i.e. space, tab, and newline) to tokenize each text sentence words/tokens that are required for the unigram and bigram model.

3. Use the WORD_POS pattern to extract the actual word (i.e. the WORD part in the WORD_POS pattern) from the tokenized word.

   For example, in the tokenized word “Brainpower_NNP”, “Brainpower” is the WORD and “NNP” is the POS. Use “Brainpower” as the actual word/token for the unigram and bigram creation.

4. The bigrams should be created ONLY within each text sentence and computed for the entire corpus.

5. Convert all words/tokens to lowercase. Do NOT perform any other type of word/token normalization (i.e. stem, lemmatize, etc.).

6. Creation and matching of unigrams and bigrams should be exact and case-insensitive.

7. Please do not submit the unigram and bigram models (counts and probabilities) with the homework solution submission. The TA can run your program to produce and check the unigram and bigram models.