LING 570: Hw3
Due date: 11pm on Oct 21
Total points: 105

Goal: Become familiar with formal language and formal grammar:

- Q1-Q4: To prove some languages are regular
- Q5-Q6: To prove some languages are not regular
- Q7: To create FSA when given a regular grammar.
- The total point is 105: 100 plus 5 bonus points.

All the example files mentioned below are under **hw3/examples**/.

Q1 (5 points): Use regular language's definition (see slide #7 in day05-regex.pdf) to prove that any **finite** set of strings is a regular language.

Q2 (10 points): Given a language L, the **complement** of L is the set of all the strings that are not in L but with the same alphabet. For instance, if the alphabet is {a, b}, and L = {a*}, then the complement of L is the set of all the strings of the form (a | b)* that are not of the form a*. In other words, the complement of L is the set of all the strings that contain at least one b.

Prove that regular language is closed under complementation; that is, for any regular language L, the complement of L is also regular.

To prove this, you can assume that regular language and FSA are equivalent. That is, given any regular language L, you can assume that there exists an FSA that accepts L, and vice versa. Now to prove that the complement of L is regular, you need to explain how you can construct an FSA that accepts the complement of L, given an FSA for L.

Q3 (5 points): Prove that regular language is closed under intersection; that is, if L1 and L2 are regular languages, so is the intersection of L1 and L2. Prove it by the definition of regular language. Hint: you can assume that regular language is closed under union and complementation.

Q4 (10 points): Same as Q3, but prove it by constructing an FSA. That is, let's assume that FSA1 is an FSA for accepting L1, and FSA2 is an FSA for L2. Describe how you can build an FSA that accepts the intersection of L1 and L2.

Q5 (10 free points) Self-study pumping lemma for regular languages. Here are some links on that topic (They should be sufficient for hw3, but free feel to find more on your own if you prefer):

- https://en.wikipedia.org/wiki/Pumping_lemma_for_regular_languages
- https://www.youtube.com/watch?v=N6bPCbXueHU&ab_channel=hhp3

Q6 (15 points) Use pumping lemma for regular languages to prove that the following languages are not regular:

a) L1 = { wv | w and v are strings of the form (a | b)*, and v is the reverse of w}, wv is the concatenation of w and v. For instance, if w is aab, v would be baa; wv would be aabbaa and it belongs to L1.

b) L2 = { waaaw | w is a string of the form (a | b)*}, waaaw is the string w, followed by three a's and then w again. For instance, if w is aab, waaaw would be aabaaaabb and it belongs to L2.

Q7 (25 points): Write a script, **reg_to_fsa.sh**, that reads a right-regular grammar from the input and output an FSA (either NFA or DFA is fine) in the carmel format.

- The format is: cat input_file | ./reg_to_fsa.sh > output_file

- Each line in the input_file is a production rule, with one of the following two formats. An example file is in **hw3/examples/reg_ex1**.
  - A t B (for the rule "A => t B"): t is a terminal symbol
  - A t (for the rule "A => t"): t is a terminal symbol or an empty string written as *e*.
  - **The symbol on the left-hand side of the first line is the start symbol of the grammar.**

- The output_file is an FSA (DFA or NFA) in the carmel format.

- In addition to submitting the code, **in the readme file, explain how you create an FSA = (Σ, Q, I, F, δ), given a right-regular grammar G = (N, Σ, P, S). That is, what should each element of the FSA be, based on the elements in G.**

Your submission should include:

- The readme.(pdf | txt) file should include your answers to Q1-Q6, the description in Q7, and anything you want us to read.

- Hw.tar.gz that includes all the files specified in submit-file-list (all for Q7).