LING 570: Hw5
Due date: 11pm on Nov 4
Total points: 100

As usual, the example files are stored under **~/dropbox/21-22/570/hw5/examples/.**

Q1 (25 points): Are the following true or false?

- If true, please provide a proof:
  - You can assume that regular languages and FSAs are equivalent, so are regular relations and FSTs.
  - For instance, if you want to prove a relation R is regular, you just need to show that you can build an FST that transduces x into y for every pair (x, y) in R.
- If false, provide a counterexample:
  - for instance, if (a) is false, you need to find a regular relation R and shows the corresponding L is not regular.

(a) Let R be a relation. Let L = {x | there exists y, such that (x, y) ∈ R}. If R is a regular relation, then L is a regular language.

(b) Let L1 and L2 be two languages. Let R be the cross product of L1 and L2. That is, R = {(x, y) | x ∈ L1 and y ∈ L2}. If L1 and L2 are regular languages, then R is a regular relation.

Q2 (20 points) Write **expand_fst.sh**, which builds an expanded FST given a lexicon and morphotactic rules expressed by an FSA.

- The command line: **expand_fst.sh** lexicon morph_rules output_fst

- lexicon and morph_rules are input files; output_fst is the output file.

- The lexicon file has the format "word classLabel":
  - word and classLabel are separated by whitespace
  - word and classLabel can be any string that does not contain whitespace.
  - Your code should ignore any blank lines in a lexicon file.
  - An example file is **examples/lexicon_ex**.

- The morph_rules file is an FSA (in the Carmel format) that encodes the morphotactic rules; that is, the input symbols in the FSA are class labels (e.g., regular_verb_stem). An example is **examples/morph_rules_ex**, which represents an FSA that is equivalent to the one on Slide #23 of day06_morph.pdf.

- The output_fst file is the expanded FST (in the Carmel format), where an arc in the morph_rule FSA is replaced by multiple paths and each path corresponds to a word in the lexicon that belongs to that category:

    - Ex: An irreg_verb_stem arc will be replaced by multiple paths, one of them will correspond to the word "cut".
    - In addition to the states on those paths, if you need, feel free to add more states in order to output class labels.
    - Note that the input symbol in the expanded FST should be a single character (e.g., "c") or an empty string ε. It should not be a word (e.g., "cut"). The output symbol should be a single character, an empty string, or a class label in the lexicon.

- In the readme file, briefly explain how the FST produced by expand_fst.sh differs from the one produced by expand_fsa.sh in Hw4. For instance, with the same lexicon and morph_rules, does the output_fst produced by the former have the same number of states as the output_fsa produced by the latter? Do the FST in the former have the same number of arcs as the FSA in the latter?

Q3 (20 points): Write **morph_acceptor.sh**, which checks whether the input words are accepted by the FST created in Q2. Your code can call Carmel.

- The command line: **morph_acceptor.sh** fst_file word_list output_file
- fst_file and word_list are input files; output_file is the output file.
- "word_list" is a list of words, one word per line (e.g., **examples/wordlist_ex**)
- "fst_file" is an FST (in the Carmel format).
- "output_file" has the format "word => answer" for each word in the word_list:
    - If the word is accepted by the morph acceptor, "answer" has the format "morph1/label1 morph2/label2 …"
    - If the word is NOT accepted by the morph acceptor, "answer" is simply the string "*NONE*".
    - An example of "output_file" is **examples/q3_result_ex**.
    - If there are more than one path for the input, you can just pick any path and use that path to produce the corresponding "answer".
- Note: The example files (e.g., examples/q3_result_ex) are meant to show the format of the files. They are not meant to serve as the gold standard.

Q4 (10 points) Run the following commands and submit the two output files:

./expand_fst.sh lexicon_ex morph_rules_ex q4_expand_fst

./morph_acceptor.sh q4_expand_fst wordlist_ex q4_result

The submission should include:

- The readme.[txt | pdf] that includes answers to Q2 (see the last bullet of Q2 which is in red).

- hw.tar.gz that includes all the files in submit-file-list.