LING 570: Hw4
Due date: 11pm on Oct 26

As usual, the example files are stored under **~/dropbox/17-18/570/hw4/examples/.**

Note: In this assignment, "FSM" (finite state machine) means an FSA or an FST. You can decide which one you want to use. **You can use Carmel** (e.g., your code can call Carmel).

Q1 (15 points): Write **expand_fsm1.sh**, which builds an expanded FSM given a lexicon and morphotactic rules expressed by an FSA.

- The command line: **expand_fsm1.sh** lexicon morph_rules output_fsm
- Lexicon and morph_rules are input files; output_fsm is the output file.
- The lexicon file has the format "word classLabel". A sample 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_fsm file is the expanded FSM (in the Carmel format), where an arc in the morph_rule FSA is replaced by multiple paths and each path corresponds to an entry in the lexicon; in other words, the input symbol in the expanded FSM should be a character or an empty string ε, not a word. See slide #19-21 in day06-morph.pdf as an example.

Q2 (20 points): Write **morph_acceptor1.sh**, which checks whether the input words are accepted by the FSM created in Q1.

- The command line: **morph_acceptor1.sh** fsm word_list output_file
- fsm 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**)
- "fsm" is the FSM (in the Carmel format) created in Q1
- "output_file" has the format "word => answer" for each word in the word_list, where "answer" is "yes" if the word is accepted by the morph acceptor, or "no" otherwise (e.g., **examples/q2_result_ex**)

Q3 (30 points): Write **expand_fsm2.sh** and **morph_acceptor2.sh** so that the "output" file produced by morph_acceptor2.sh has the format "word => answer", where "answer" is “morph1/label1 morph2/label2 …” if the word is accepted by the morph acceptor, or “*NONE*” otherwise (e.g., **examples/q3_result_ex**).

- The command line formats of **expand_fsm2.sh** and **morph_acceptor2.sh** are the same as **expand_fsm1.sh** and **morph_acceptor1.sh,** respectively**.**
- In your note file, explain briefly how the fsm produced by expand_fsm1.sh differs from the one produced by expand_fsm2.sh.

Q4 (5 points) Run the following commands and store the results under **q4/**

```
expand_fsm1.sh  lexicon_ex morph_rules_ex  q4/q4_expand_fsm

morph_acceptor1.sh q4/q4_expand_fsm wordlist_ex   q4/q4_result
```

Q5 (5 points) Run the following commands and store the results under **q5/**

```
expand_fsm2.sh  lexicon_ex morph_rules_ex  q5/q5_expand_fsm

morph_acceptor2.sh   q5/q5_expand_fsm   wordlist_ex    q5/q5_result
```

Note: The example files (e.g., q2_result_ex) under examples/ are meant to show the format of the files. They are not meant to serve as the gold standard.

The submission should include:

- The readme.[txt | pdf] that includes answers to Q3.
- hw.tar.gz that includes all the files in submit-file-list