LING 570 Hw2
Due date: 11pm on Oct 12
Total points: 100

All the example files are under **~/dropbox/17-18/570/hw2/.**

Q1 (6 points): Learn the Carmel package. We will use the package for hw2 and hw3. The whole package is stored under **/NLP_TOOLS/ml_tools/FST/carmel/latest/**

(1) (4 free points) Read the tutorial under doc/, and play with the examples under sample/.
   a. The command "carmel" is under bin/. Make sure that the path is included in $PATH if you want your shell to find the location of that command easily.
   b. Type "carmel" on patas to see what options are available. The most important options are –k, -b, -sli.

(2) (2 points) Under sample/ directory, run the following commands:
   carmel -k 1 fsa7 wfst1
   cat ~/dropbox/17-18/570/hw2/examples/wfst1_test | carmel -k 1 -sli wfst1

   Do they yield the same results? What do these commands do?

Q2 (4 points): Manually create FSAs for the following regular expression. The FSAs should be in Carmel format. Store the fsa files under **q2/.** The FSAs can be NFAs or DFAs.

- fsa1 for a* b*
- fsa2 for $(a \mid b)^+$ c d*
- fsa3 for ($(a \mid b)^+$ c d* | b* d*)
- fsa4 for (a b*)? b a

Q3 (15 points): Use Carmel to build an FSA acceptor, **fsa_acceptor.sh**; that is, fsa_acceptor.sh can call the Carmel command and process Carmel's output if needed.

- The format is: fsa_acceptor.sh fsa_file input_file > output_file

- fsa_file is an FSA in the Carmel format

- Each line in the input file is a string, and each line in the output_file has the format "x => y", where x is the string from the input file, and y is "yes" is x is accepted by the FSA, and "no" otherwise.

- Some example files are under 570/hw2/examples/: let "fsa1" be the fsa_file, "ex" be the input_file. Running the command "fsa_acceptor.sh fsa1 ex > ex.output" should produce an output file with the same format as the file "ex.output" in that directory.

- Run fsa_acceptor.sh with the fsa files created in Q2 and store the output files under **hw2_dir/q3/** by running the following commands, where ex is 570/hw2/examples/ex

  fsa_acceptor.sh q2/fsa1 ex > q3/ex.fsa1
  …

  fsa_acceptor.sh q2/fsa4 ex > q3/ex.fsa4

Q4 (35 points): Build **fsa_acceptor2.sh** WITHOUT using Carmel. Note that the input FSA can be an NFA. To check whether a string is accepted by an NFS, there are two common methods. Choose one method to implement.

(1) Convert the NFA into a DFA first and then run DFA. There are many websites that explain the algorithm, just google "convert NFA to DFA". Some examples are https://www.youtube.com/watch?v=taClnxU-nao, and http://web.cecs.pdx.edu/~harry/compilers/slides/LexicalPart3.pdf.

(2) Use NFA directly: your code will keep track of "current search states". Once you reach the end of the input symbol sequence, check one of the current search states is a final state. Note that you do not need to enumerate all possible paths explicitly. You just need to keep track of multiple current search states. The method is explained in Section 2.2.5 in Jurasfsky & Martin (2nd edition).

For your fsa_acceptor2.sh,

- fsa_acceptor2.sh has the same command line format and functionality as fsa_acceptor.sh.

- The only difference is that fsa_acceptor2.sh CANNOT use Carmel; for example, the code will need to read in the fsa_file, store the FSA in some data structure, and determine whether each line in the input_file is accepted by the FSA.

- In your note file, specify which method you have implemented.
- Run fsa_acceptor2.sh with the fsa_input files created in Q2 and store the output files under **hw2_dir/q4**/.

Q5 (3 points): When you flip a coin, the probability of getting the head is 0.8. Now suppose you flip the coin five times, what is the probability of getting AT LEAST four

heads out of the five flips? Please write down the formula, in addition to provide the probability.

Q6 (4 points): There are two random variables X and Y, and the joint probability P(X,Y) is shown below:

	X=0	X=1
Y=0	0.50	0.25
Y=1	0.10	0.15

a. (1 pt) What is the probability distribution for P(X)? Note that a probability distribution is a function, not a number.

b. (1 pt) What is the probability distribution for P(Y)?

c. (1 pt) What is the probability distribution for P(Y|X)?

d. (1 pt) Are X and Y independent? Why or why not?

Q7 (8 points): There are three coins: c1, c2, and c3. When tossing a coin once, the probabilities of getting a head for c1, c2, and c3 are 0.1, 0.4, and 0.7, respectively. Now suppose that you first pick one of the coins, with the probability 0.2 of being c1, 0.5 of being c2, 0.3 of being c3, and then toss the coin.

a) (4 points) If you toss this selected coin once, what is the probability of getting a head?

b) (4 points) If you toss this selected coin once and get a head, what is the probability that c1 was the coin selected in the first step?

The submission should include:

1. readme.[txt|pdf], which includes the answers to Q1, Q5-Q7 and any note you want the grader to read.

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