LING 570 Hw2
Due date: 11:45pm on Oct 13

All the files are under **~/dropbox/10-11/570/hw2/.**

Note: In this and all future assignments, "write a program, foo.sh" means "write the source code in one of the languages: Perl/Python/C/C++/C#/Java/Ruby/shell-script. Include both source code and shell script (foo.sh) in your submission. If you use C, C++, C#, or Java, you need to compile the code, and include the binary file in your submission as well." When you submit your homework, you need to tar your directory and submit only the tar file. Remember to include a note file, hw2.[txt|pdf], as a part of the tar file for your submission.

Q1 (8 points): Learn the Carmel package. We will use the package for hw2 and hw3. The whole package is stored under 570/hw2/graehl/.

(1) (4 free points) Read the tutorial under hw2/graehl/carmel/doc/, and play with the examples under hw2/graehl/carmel/sample/.
   a. The command "carmel" is under hw2/graehl/carmel/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) (4 points) Under the hw2/graehl/sample/ directory, run the following commands:
   carmel -k 1 fsa7 wfst1
   cat wfst1_test | carmel -k 1 -sli wfst1

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

Q2 (12 points): Manually create FSAs for the following regular languages. The FSAs should be in Carmel format. Store the fsa files under **hw2_dir/q2/.** Run your fsa_acceptor.sh with those FSAs and hw2/examples/ex as input file, save the output files in ex.fsa1, ex.fsa2, ex.fsa3, and ex.fsa4, respectively under **hw2_dir/q2/.**

- fsa1 for {a*b*}
- fsa2 for {a^+b*}
- fsa3 for { a*b^+}
- fsa4 for { a^+ b^+}

Q3 (20 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 (60 points): Build **fsa_acceptor2.sh** WITHOUT using Carmel.

- 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.

- Note that the input FSA could be an NFA. Your code could either convert the NFA into a DFA first, or your code could follow multiple paths for an input string and check whether any of the paths ends at a final state.

- In your note file, briefly explain what data structure you use to store the input FSA and how your code handles NFA.

- Run fsa_acceptor2.sh with the fsa_input files created in Q2 and store the output files under **hw2_dir/q4/**.

The submission dir hw2_dir/ should include:

1. hw2.[txt|pdf], which includes the answers to Q1 and Q4.

2. The source and shell scripts for Q3 and Q4: fsa_acceptor.sh and fsa_accept2.sh and any scripts called by them.

3. The subdirectory q2/, which includes fsa1, fsa2, fsa3, and fsa4.

4. The subdirectory q3/ and q4/, which each includes ex.fsa[1-4].