CS 164, Spring 2009 **CS 164: Homework #2** **P. N. Hilfinger**

Due: Tuesday, 2 February 2010 at 2400

Please submit this homework electronically. See the on the class web page. You'll find templates for some solutions in the directory `~cs164/hw/hw2` and in the repository `$STAFFREPOS/hw2`. Place answers to any questions that don't require programs in a file called **hw2.txt**.

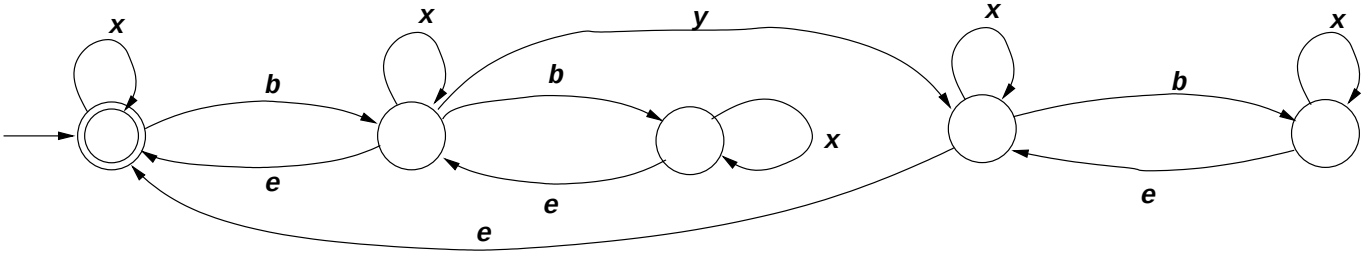
Problems 1 and 2 call for the use of the **fsasim** program, which is a Python program that is available from your instructional account (you can download it from `~cs164/bin`; at home, you'll have to change the first line to wherever you have Python installed, or run fsasim using the 'python' command). There is a manual for **fsasim** available from the CS164 home page.

1. Find the simplest NFA you can for problem 2c in HW#1. Turn in a file called **2c.nfa** containing your answer in **fsasim** format.

2. Find the simplest DFA you can for problem 2c in HW#1. xTurn in a file called **2c.dfa** containing your solution in **fsasim** form (see above).

3. Suppose that I have two NFAs, M_1 and M_2, recognizing the languages L_1 and L_2, respectively. Give a general algorithm for constructing an NFA that recognizes the language $L_1 - L_2$, which is the set of all strings in L_1 that are not in L_2. You don't need to write a program, just the give the algorithm in sufficient detail that a reasonable programmer could fill in the details.

4. Give the simplest description you can of the language described by the DFA below:



5. The authorization file we use for Subversion, which tells which users may access which directories, contains entries that look like this:

```
[/DIR]
login1 = rw
login2 = r
login3 = rw
* =
```

where *login*i are ordinary Unix login names. (This entry means that *login1* and *login3* have read-write (rw) access to the directory *REPOSITORY*/*DIR*, *login2* has read-only (r) access, and everyone else (denoted '`*`') has no access. As shown, these are all on separate lines, and otherwise whitespace may be used freely. The end of the list of authorized users for a given directory is marked by the next directory (`[...]`) line, or the end of file. Write a program using full Python regular expressions to search such a file for cases where there are (at least) two entries with the same *DIR* and remove all but the last entry, using the template file in cs164/hw/hw1/cleanup.py. You will probably want to consult the entry in the Python library reference documentation for the `re` module.

6. Fill in `count.py` (from the hw2 staff skeleton) with a Python program that reads text from the standard input, keeping track of the number of times each word in the text appears, and then prints out the ten most frequently occurring words that are at least 4 letters long, one per line, in order of decreasing frequency. A "word" here is a contiguous substring of letters; anything other than a letter delimits a word. Ignore case (and print words in lower case).