Project 1: Instructions

Help

When you're ready to submit your solution, go to the assignments list.

Programming Details:

Materials for Download

- **Java Version**
- **Python Version** **(Please use Python 2.7.5)**

In this assignment, you are expected to fill in programs that convert a regular expression to an ϵ-NFA. The parsing of the regular expression and the NFA data structure have been finished for you. What is left to fill in are the following functions in epsnfa.java: (Note: For Python version, the names of files, methods and variables are exactly the same as Java version)
//unite two ϵ-NFAs, with start state s1 and s2, final state t1 and t2, respectively
//return an array of length 2, where the first element is the start state of the combined NFA. the second being the final state
int[] union(int s1,int t1,int s2,int t2)

//concatenation of two ϵ-NFAs, with start state s1 and s2, final state t1 and t2, respectively
//return an array of length 2, where the first element is the start state of the combined NFA. the second being the final state
int[] concat(int s1,int t1,int s2,int t2)

//Closure of a ϵ-NFA, with start state s and final state t
//return an array of length 2, where the first element is the start state of the closure ϵ-NFA. the second being the final state
int[] clo(int s,int t)

For definitions, you can refer to slides 14-16 of lecture 5 (Regular Expressions). You are expected to implement the functons that appear in diagram form in the slides.
In each case, you should assign new states for the start state and final state of the resulting ϵ-NFA. You can do this via method:
newstate=incCapacity();
To add edges, you can use methods:
addEdge(a,0,b); //add an edge with symbol 0 from a to b
or addEdge(a,1,b); //add an edge with symbol 1 from a to b
or addEdge(a,epssymbol,b); //add an edge with symbol ϵ from a to b
Basically, the whole program, including what you write, will read regular expressions and transform them one by one into an ϵ-NFA. Then the program will check whether the ϵ-NFA you construct is correct.

sample input (sampleRE.in):

1.0.1
0+1
0+1.0.1

Submission Details:

After you fill in the program in epsnfa.java, you can run Submit.java. This program will first run your epsnfa.java on sample input, and only after you get it right on samples, can you submit your programs.
In submission, you will select the programs you want to submit (Select "All of the above"), and it will need your website credentials (note that the password should be the assignment password, which you can find when you click "Programming Assignment" on the website).
Then, the website will compare your output on testRE.in with the solution. If it's correct, you will get full score and see "Correct!" on the screen; otherwise, you should go back and check your filled part to make it correct.
Example of what you will see in the terminal screen when you run Submit.java:

```
==
== [automata-class] Submitting Solutions | Programming Exercise 1
==
**********************
Test on sample input (sampleRE.in).....

You output is correct for sample input. Now start submitting to servers.....
**********************


== Select which part(s) to submit:
== 1) 1-1 [ epsnfa.java ]
== 2) All of the above
==
Enter your choice [1-2]:
2
Login (Email address): xxxxx@yyyy.com
Password: abcdefg

== Connecting to automata-class ...

== [automata-class] Submitted Homework 1 - Part 1 - 1-1
== Correct!
```