CSC 371 – Finite Automata

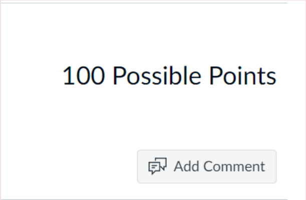
Project 1 (Problem 2/2, Due date: 10/22)

Dr. Liudong Zuo (lzuo@csudh.edu)

Instructions:

(1) Deadline of the submission is 11:59 PM on 10/22. Late assignments will be accepted within 24 hours after the deadline and 15% points loss penalty will be applied. The submission page on Canvas will be closed after 11:59 PM on 10/23, and no further assignments will be accepted after that time point.

(2) For this project, you have the option to work in groups, with each group limited to a maximum of two students. However, it is important to note that the intention behind allowing group work is not solely to lighten individual workloads. Instead, it aims to provide you with an opportunity to enhance your collaboration skills and learn from one another. Make sure that the project's workload is distributed equally among group members. Both group members will receive the same grade. Therefore, only one submission needs to be uploaded by each group. Write down the names of all group members in the designated "Add Comments" section of the submission page on Canvas, as shown on the right.



You do have the choice to work on this problem on your own without forming groups (those who choose to do so will receive up to 10% bonus points).

(3) Adhere to all the specified requirements without making any modifications. Please ensure strict compliance throughout the project. For instance, one specific requirement is to read a text file. Consequently, your program must be capable of successfully reading a text file. Deviating from this requirement, such as prompting the user to input, will result in the loss of all points allocated for this particular component.

(4) You can use any programming language, however, Java is preferred. After you finish, run your program several times to check the correctness. Make screenshots of at least three executions including input and output (sample input and output can be found in the problem description below), put the screenshots in the project folder, and then export the entire project as a zip file and upload that single zip file to Canvas. I highly recommend that you design some other testing files yourself. You submission should be one and only one zip file.

If you do not follow the above instructions, at least 10% points loss penalty will be applied.

(5) All work turned in be the students' own work. Plagiarism and cheating will not be tolerated. Please refer to our syllabus for more information about plagiarism and cheating.

I consider plagiarism and cheating to be serious offenses.

Other Instructions:

(1) You may implement your solution in any programming language of your choice; however, the use of the data structure `Set` is strictly prohibited in any part of your program.

(2) Your solution must be implemented using recursion. Submissions that employ alternative approaches, such as depth-first search (DFS) or any non-recursive method, will receive zero points. Some students may still struggle with understanding recursion; however, I have already presented and explained the corresponding pseudocode in our classes.

(3) Your program must be capable of processing and testing multiple `.txt` files in a single run. Please refer to the `Main.java` file in the attached zip archive as a template for implementing this functionality.

****

Problem 2: Convert NFAs to the equivalent DFAs

Read part of the transition table of a NFA from a txt file, suppose this table only contains two columns: the first column lists out the states, while the second column lists out what each state jumps to upon the input ε. For example, given the following transition table in a txt file:

1,{2,3}

2,empty

3,{4}

4,empty

We know state 1 jumps to states {2,3} upon the input of ε, while state 2 does not have an outgoing arrowhead with ε on, namely it jumps to ∅ upon the input of ε (in the txt file we use "empty" to denote empty set ∅).

Read part of the transition table of a NFA from a txt file, and then print out $E(q)$ for each q in the set of states of the NFA. For example, for the above part of the transition table of a NFA, the print out should be

E(1) = {1,2,3,4}

E(2) = {2}

E(3) = {3,4}

E(4) = {4}

Note that your program must be able to work for any input, not only just for the above given example.

The following are a few more examples you can use to test your program (I highly recommend that you come up with more examples to make sure the correctness of your program):

(1) 1,{3}

2,empty

3,empty

For the above part of the transition table of a NFA, the print out should be

E(1) = {1,3}
E(2) = {2}
E(3) = {3}

(2) 1,{3}
2,{1}
3,empty

For the above part of the transition table of a NFA, the print out should be

E(1) = {1,3}
E(2) = {1,2,3}
E(3) = {3}

(3) 1,{2,3}
2,empty
3,{4,5}
4,empty
5,empty

For the above part of the transition table of a NFA, the print out should be

E(1) = {1,2,3,4,5}
E(2) = {2}
E(3) = {3,4,5}
E(4) = {4}
E(5) = {5}

END