CSC 471 – Compiler Construction I

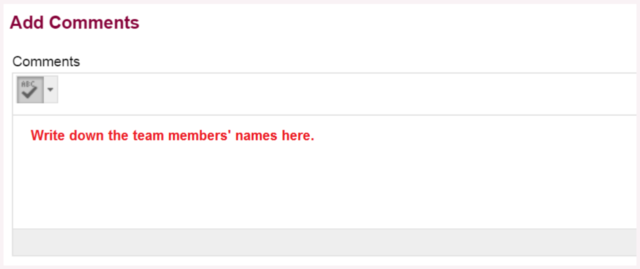
Project 2 (Due date: 5/1)

Dr. Liudong Zuo (lzuo@csudh.edu)

Instructions:

(1) Deadline of the submission is 11:59 PM on 5/1. Late assignments will be accepted within 24 hours after the deadline and 15% points loss penalty will be applied. The submission page on Blackboard will be closed after 11:59 PM on 5/2.

(2) This project can be finished in groups (each group can have up to two students). However, the purpose of letting you finish the project in groups is not to reduce the work load, it is to give you an opportunity to learn from each other and improve your collaboration skills. Try to distribute the workload equally to group members. All students will receive the same grade. One group only needs to upload one submission. Write down group members' names in "Add Comments" area of the submission page on Blackboard shown below:



Write down the team members' names here.

(3) Strictly follow all the requirements, and do not change any. For example, one requirement of this project is to read a text file, then your program must be able to read a text file. If you do not follow the above requirement, such as your program asks user to enter the transition table of a NFA instead, then your program will lose all the points for this part.

(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 Blackboard. 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.

There might be some online resources for part of our project. It is OK for you to go over these resources and refer to them in your project (write down the reference in your code). However, you need to fully understand them first and then implement the concept of the resources using your own words. The difference between your implementation and online resources might be significant. The changes like changing names of the variables, removing blanking lines or changing positions of the code blocks will be regarded as ”no changes". If the changes are not significant, there is a high chance that your behavior will be regarded as plagiarism and cheating, and the consequences will be severe. Note that I am very familiar with online resources, so be very careful if you refer to any online resource.

****

Project Problem: Simplification of Context-Free Grammars

We learned three context-free grammar (CFG) simplification techniques in our classes: removing ε-rules, unit-rules, and useless rules. In this project, read a CFG from a txt file, simplify it by removing ε-rules and useless rules, and print out the simplified equivalent CFG. No need to care or remove unit-rules.

For example, given the following CFG in a txt file (0 denotes empty string and "-" denotes arrow head "→"):

```
S-aA|aBB
A-aaA|0
B-bB|bbC
C-B
```

After processing, your program should print out the following simplified equivalent CFG:

```
S-aA|a
A-aaA|aa
```

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) S-AaBaCbD
A-0|a
B-0|b
C-0|c
D-0|d

For the above CFG, the print out should be

```
S-aab|Aaab|aBab|AaBab|aaCb|AaaCb|aBaCb|AaBaCb|aabD|AaabD|aBabD|AaBabD|aaCbD|AaaCbD|aBaCbD|AaBaCbD
A-a
B-b
C-c
D-d
```

(2) S-AaB|aaB
A-0
B-baA|0

For the above CFG, the print out should be

```
S-aB|aaB|a|aa
B-ba
```

(3) S-ASA|aB
A-B|S
B-b|0

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

```
S-ASA|aB|a|SA|AS
```

A-B|S
B-b

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

END