

CSC 371 – Finite Automata

Project 2 (Due date: 11/23)

General 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.

For this project, you may choose to work individually (those who complete the project on their own will receive **up to 10% bonus points** as additional credit) or in groups, with **a maximum of two students per group**. Please note that the purpose of allowing group work is not merely to reduce individual effort, but rather to promote collaboration and mutual learning. Ensure that the workload is shared fairly between group members, as **both members will receive the same grade**. Only one submission is required per group. Be sure to include the names of all group members in the “Add Comments” section of the submission page on Canvas, as illustrated on the right.

- (3) **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.**

Plagiarism and cheating are considered serious academic offenses. You are strongly advised **not to refer to or use any code written by others**. In particular, referencing code from the following websites—or any similar online sources—is strictly prohibited. Be aware that students in previous semesters have received a grade of 0 for submitting copied code.

- <https://github.com/aliyazdi75/Simplifying-CFG/tree/master/src>
- <https://www.chegg.com/homework-help/questions-and-answers/help-java-code-leave-notes-q88425273>

100 Possible Points

 Add Comment

Implementation Instructions:

- (1) **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.
- (2) **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.
- (3) You may use any programming language; however, Java is preferred. **After completing your implementation, thoroughly test your program by running it multiple times to verify its correctness.** Capture screenshots of at least three separate executions showing both the input and output (sample inputs and outputs are provided in the problem description below). Place these screenshots in your project folder, then export the entire project as a single .zip file and upload it to Canvas. It is also recommended that you create and test with additional input files of your own. Your submission must consist of **one and only one** zip file.

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

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 “ \rightarrow ”):

```
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 CFG, 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*****