T.C.
DOKUZ EYLUL UNIVERSTY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

2022 – 2023
SPRING SEMESTER

CME 3202
CONCEPTS OF PROGRAMMING LANGUAGES

ASSIGNMENT 1:
LEFT AND RIGHT MOST DERIVATION

DUE DATE:
23:55 – 28.04.2023

In this assignment, you are asked to write a Python program for left and right most derivation. For this purpose, your program should take three inputs. First is “ll.txt” which contains LL(1) parsing table for left most derivation. Second is “lr.txt” which contains LR(1) parsing table for right most derivation. Third is “input.txt” which contains input strings (could be more than 2) you should process with their corresponding tables. You should use the following variables to store file names.

```
FILE_LL = “ll.txt”
FILE_LR = “lr.txt”
FILE_INPUT = “input.txt”
# These variables contains names of the files you must read. Their default value should be the values given here, not different or personalized. In addition, you should NOT use global addressing for these files (e.g. C:\user\student\Desktop\input.txt). You should only use their names so that they are searched on local folder only.
```

Example files for ll.txt, lr.txt and input.txt are also given. You can examine them and change them to test your software. Your program will be graded by using different parsing tables and inputs, so make sure that your program works for all parsing tables and inputs, not just one.

Assuming your program takes the given example files as input, you are expected to generate an output on console (not in a file) like given on the page below (the text given inside parenthesis is for information only, you are not required to print it out but you can if you wish):

```
Read LL(1) parsing table from file ll.txt.
Read LR(1) parsing table from file lr.txt.
Read input strings from file input.txt.

Processing input string id+id*id$ for LL(1) parsing table.

NO | STACK  | INPUT     | ACTION
1  | $      | id+id*id$ | E->TA
2  | $AT    | id+id*id$ | T->FB
3  | $ABF   | id+id*id$ | F->id
4  | $ABid  | id+id*id$ | Match and remove id
5  | $AB    |   +id*id$ | B->ε
6  | $A     |   +id*id$ | A->+TA
7  | $AT+   |   +id*id$ | Match and remove +
8  | $AT    |    id*id$ | T->FB
9  | $ABF   |    id*id$ | F->id
10 | $ABid  |    id*id$ | Match and remove id
11 | $AB    |      *id$ | B->*FB
12 | $ABF*  |      *id$ | Match and remove *
13 | $ABF   |       id$ | F->id
14 | $ABid  |       id$ | Match and remove id
15 | $AB    |         $ | B->ε
16 | $A     |         $ | A->ε
17 | $      |         $ | ACCEPTED

Processing input string acd$ for LR(1) parsing table.

NO | STATE STACK | READ | INPUT | ACTION
1  | 1           | a    | acd$  | Shift to state 3
2  | 1 3         | c    | acd$  | Shift to state 6
3  | 1 3 6       | d    | acd$  | Shift to state 5
4  | 1 3 6 5     | $    | acd$  | Reverse B->d
5  | 1 3 6       | B    | acB$  | Shift to state 7
6  | 1 3 6 7     | $    | acB$  | Reverse B->cB
7  | 1 3         | B    |  aB$  | Shift to state 4
8  | 1 3 4       | $    |  aB$  | Reverse S->aB
9  | 1           | S    |   S$  | Shift to state 2
10 | 1 2         | $    |   S$  | ACCEPTED

Processing input string +id*id$ for LL(1) parsing table.

NO | STACK | INPUT   | ACTION
1  | $     | +id*id$ | E->TA
2  | $AT   | +id*id$ | REJECTED (T does not have an action/step for +)

Processing input string cd$ for LR(1) parsing table.

NO | STATE STACK | READ | INPUT | ACTION
1  | 1           | c    | cd$   | REJECTED (State 1 does not have an action/step for c)
```

You should try to make your output as orderly as possible but do not spend days or weeks just for small aesthetic improvements to the console output.

Correct execution and output will be very important for your grade, if your program does not work correctly, your grade will be very low. In addition, your program should NOT import any libraries or external packages, adding these and using alternative and easier methods and algorithms for this assignment, will make your grade zero. If you really believe that this assignment is not possible to do without external libraries, discuss it with me on class forums or lab sessions.

As you may have seen, there is a special character at your input files called Epsilon (ϵ). Make sure your program detects this character correctly. Uppercase and lowercase version of this character is shown in the links below.

https://www.compart.com/en/unicode/U+03B5
https://www.compart.com/en/unicode/U+03F5

Your input files uses spaces to make it more readable and table like, however this is not a requirement. You should eliminate empty spaces when you read these files because input strings “E->TA” and “E -> T A” are functionally equivalent in this context and you should consider them as the same too.

You are allowed to do this assignments as groups of two or alone. However, 3 or more students per group will not be accepted.

You are only required to upload a single python file. Please do NOT upload given input text files or any other file. If you do, you grade will be reduced accordingly. Only one of the students of groups are required to make an upload.

The required naming format of uploaded python file are given below. Please do NOT zip or rar your upload, only a single python file (py) is enough. If you are working within a two student group, you should write student numbers and names in ascending order. An example of both one and two student uploads are given below.

FORMAT:
<student_number_1>_<student_name_1>.py
<student_number_1>_<student_name_1>_<student_number_2>_<student_name_2>.py

EXAMPLE:
2023510123_fatih_dicle.py
2023510122_ali_cuvitoglu_2023510123_fatih_dicle.py

As you can see, you should also not use any Turkish characters in naming of your files. Additionally, you should not have any empty spaces in file name, please replace them with underscore (_) You should also try to use lowercase for student names. If you do not use this format, we may not be able to understand who is your group partner and therefore that person could be considered as not made an upload.

Your uploaded source codes will be checked for cheating and plagiarism. If cheating is detected, your entire assignment will be graded zero. If you or other students copy your code from an online source rather than writing it yourself, it will be considered as cheating as well.

Make sure that you upload your correct assignment. If you accidentally upload another assignment (from another class for example), it will be considered as not turned in and it will be graded as zero. Worst of all, you will only realize it after grades are published and it will be too late to fix it.

The upload of this assignment will be opened for section 1 alone, meaning section 2 students should also make their upload there.

Your assignments will be evaluated and graded on a Linux computer. Therefore make sure you do not write a code that will not work on Linux as well. Test your code on a Linux virtual machine as well.

If you have any questions or problems regarding this lab paper, you can ask about it in our lab sessions. If you wish, you can also ask it in class forums or assignment page comments. If you have problems with posting on class forums, please inform us. If you send an email and if your question is answered, please share this information with other students to prevent asking of the same question again and again.

Your assignment will be open for upload until 23:55, 30.04.2023. This is done to allow students who may experience extreme problems (no Internet or electricity, computer crash or failure, etc.) and miss the deadline as a result. This extension will allow them to upload. If you made a late upload, please explain the reason(s) within assignment python file as a comment at the top of the file.

If you are still unable to upload, send us an email informing your situation and at the same time, try everything you can to make your assignment upload. This does not mean your grade will not be reduced if you upload late however, so don’t be late.

Lastly, please do not forget to click “Submit” button after you upload your assignment files. If you do not, even though your files are uploaded to Sakai, you are labeled as “No Submission” and ignored when we try to download your assignments, making your uploaded files invisible to us, leading us to assume you did not make an assignment submission.

ASSIGNMENT CHECKLIST (INCLUDING BUT NOT LIMITED TO):

You can use the following checklist to make your assignment and upload better, however the requirements may not be limited to given list below, so make sure to read the assignment for fully understanding assignment requirements.

Make sure to use correct file name format given above.
Make sure not to use Turkish characters in file name.
Make sure you are uploading correct file to correct assignment upload.
Make sure to click submit after upload.
Make sure to explain your reasons if you make a late upload.
Make sure to send an email with your code if you are unable to make an upload.

Make sure you do not use any additional Python libraries or modules.
Make sure your code works on a Linux PC, not just Windows or Mac.
Make sure your code works for different inputs in addition to given examples.
Make sure the requested global variables are used and assigned correctly.
Make sure your code correctly handles Epsilon character.
Make sure your code correctly handles spaces present in input files.

Make sure your code works and outputs correctly to console.
Make sure your console output is orderly enough.

GOOD LUCK TO YOU ALL!