Assignment 1 – CS 507 - Theory of Programming Languages – Fall 2020

Due: Thursday October 8, 2020 – 11 PM (on google class)

If any assignment is deemed to be copied from any other student or internet, you may be awarded an F in this course.

Part I – Programming (100 marks)

In this assignment you are required to implement the CYK algorithm for parsing a string. The CYK algorithm requires that the given CFG(Context Free Grammer) be in CNF (Chomsky Normal Form). For this assignment you will be given a **CFG that will already be in CNF. Therefore you do not need to code for grammar conversion.**

Your program will take two strings as arguments. The first string will be the name of the file that contains a CFG in CNF. The file of the name **MUST BE** “grammar.txt”.

The second argument will be the name of the file that will contain one or more strings to be parsed. The file of the name **MUST BE** “strings.txt”.

The program shall read the CFG, and then use the CYK algo to find if the strings to be parsed are in the language or not. The program shall output each string followed by "Part of language" or "Not part of language" depending on whether the string is in the language or not. You can output the answer to the console.

The input file format is as follows.Everything in ***RED*** in the following format is comments

ALPHABET:
(comma and <SPACE> separated alphabets) **;**
VARIABLES:
(comma and <SPACE> separated variables) **;**
PRODUCTIONS:
(comma/<SPACE>/<TAB> separated transition functions) **;**

The productions will be of the following form.
(Variable) -> *Variable1Variable2* | *Variable3Variable4* | *terminal* **....;**
Example.

```
ALPHABET:
a, b ;
VARIABLES:
S, A, B ;
PRODUCTIONS:
S -> AB | BB ;
B -> AB | b ;
A -> a ;
```

Notice

- the <**SPACE**> before and after the "|".
- the **;** at the end of the input.
- that the productions of one variable can span multiple lines. Also in one line you can have multiple productions for the variable. These will be separated by "|"
- the **;** at the end of each line of productions.
- "S" will be the start variable.

Also notice that the actual values start on a separate line after the label. So the symbols of the alphabet will be on next line after the label **ALPHABET:**

You may safely assume that all the productions of one variable will occur together. So the productions for A will not start until the productions for S have ended. So you will not have

```
S -> AB
A -> CD
S -> AS
```

Programming Specifications

1. The code should be in C++ only.
2. Symbols in the ALPHABET and the VARIABLES will be single character long
3. The name of the variables may be A, B, …, Z.
4. The terminals maybe from a, b, …z.

What to Hand In:

- All coding files should be zipped together into a single file, whose name is comprised of your FAST NUCES ID followed by the course number followed by the letters "assignment1", followed by the assignment number. (E.g. L1XXXXXassignment1.zip).

- If there is anything special the I should know about your program, be sure to document it in the documentation file. Anything special I need to do to run your program and understand the results should also be documented.

Part II – (50 marks)

a) Give CFGs for the following. If not given, strings consist of a’s and b’s only.

a. A = {all strings such that number of a’s and number of b’s are equal}
b. B = {$a^i b^j$ | i <= 2j }
c. C = { $a^{2n} b^{3n}$ | n > 0 }
d. D = { $a^i b^j c^k$ | i, j, k ≥ 0, and i = j or j = k }
e. E = {w such that w has an ‘a’ on second last position}
f. F = {w is a palindrome}

b) Convert the following grammars to CNF form:

a) S → ASB A → aAS | a | ε B → SbS | A | bb	b) S → BSB | B | ε B → 00 | ε