**DOKUZ EYLUL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CME 3203 THEORY OF COMPUTATION**

**REPORT**

**HOMEWORK**

**Context Free Grammar**

**by**

**Rasim Efe Sazcı - 2016510122**

**Rinat Zhulfayev - 2020510130**

**IZMIR**

**10.01.2022**

**DESCRIPTION**

The main task of the assignment is to determine whether a given string belongs to the any given context free grammar and show derivation path.

To handle this task, we decided to use parsing method. Our Parsing Tree method looks through all possible paths and can give either all possible derivation paths for string or just the first correct, there is a variable “findAllPaths” in the main class for this choice.

During completing the assignment, we encountered a small problem. If alphabet of the given grammar contains “empty(#)”, then Parse Tree method didn’t work correctly since empty was the reason of infinite loop. The solution to this problem was transform context free grammar to the Chebyshev Normal Form.

**PSEUDOCODE**

grammar= file

String start=user.input.start

String endstr = user.input.end

if(grammar.contains(empty))

{

grammar= convertToChebucev(grammar)

}

ParseTree root= new ParseTree(startstate , grammar)

inLanguage(ParseTree root, endstr, Boolean allpaths)

{

if(allpaths or flag)

{

for(every character in root)

{

for(every rule for every character)

{

root.addChild(Derivation(root));

}

}

if(Children.frequency = endstr.frequency)

{

recursivelyPrintParents(child)

}

if(Children.frequency< endstr.frequency)

{

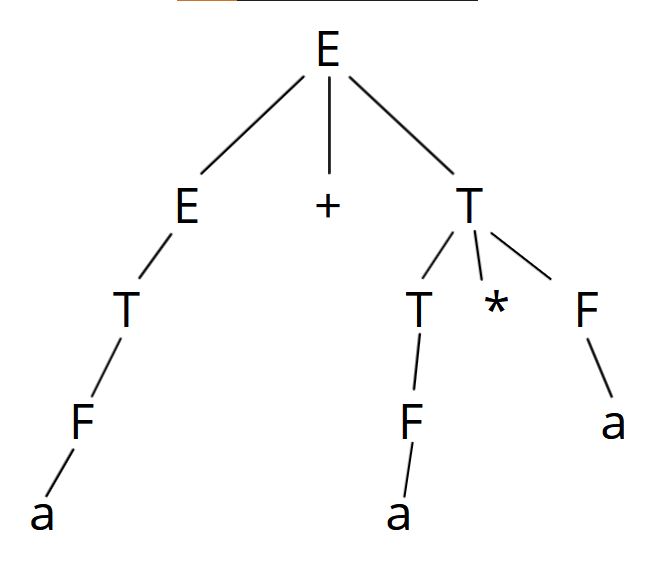
inLanguage(child, endstr, allpaths)

}

}

**PARSING TREE AND DERIVATION**

String:a+a\*a



SCREENSHOOTS

