

**DEPARTMENT OF COMPUTER ENGINEERING**

## CSL804 Computational Lab II

**Eighth Semester, 2021-2022 (Even Semester)**

**Name of Student :** Saurav Kumar

## Roll No. 23

**Division :** BE – CMPN

**Day/ Session :** Monday/Afternoon

**Venue :** SLRTCE Lab 305

## Experiment No. 7

**Title of Experiment :** To perform top down and bottom up parsing using CGF for English language.

## Date of Conduction :

**Date of Submission :**

|  |  |  |
| --- | --- | --- |
| **Particulars Max. Marks Marks Obtained** | | |
| Preparedness and Efforts(PE) | **3** |  |
| Knowledge of tools(KT) | **3** |  |
| Debugging and results(DR) | **3** |  |
| Documentation(DN) | **3** |  |
| Punctuality & Lab Ethics(PL) | **3** |  |
| **Total** | **15** |  |

**Grades – Meet Expectations (3 Marks), Moderate Expectations (2 Marks), Below Expectations (1 Mark)**

**Checked and Verified by Name of Faculty :** Prof. Neelam Kulkarni

## Signature :

**Date :**

EXPERIMENT NO: 7

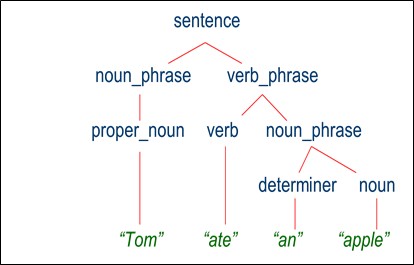
PARSING USING CGF

**AIM:** To perform top down and bottom up parsing using CGF for English language.

**SOFTWARE:** Python

# THEORY:

Parsing, syntax analysis, or syntactic analysis is the process of analyzing a string of symbols, either in natural language, computer languages or data structures, conforming to the rules of a formal grammar. The term parsing comes from Latin pars (operations), meaning part (of speech).



# IMPLEMENTATION:

RECURSIVE DESCENT PARSING CODE

print("Recursive Desent Parsing for following grammar\n") print("E->TE'\nE'->+TE'/@\nT->FT'\nT'->\*FT'/@\nF->(E)/i\n") print("Enter the string want to be checked:\n”)

global s

s = list(input()) global i

i = 0

def match(a):

global s global i if(i>=len(s)):

return False elif(s[i]==a):

i+=1

return True else:

return False def F():

if(match("(")): if(E()):

if(match(")")): return True

else:

return False

else:

return False elif(match("i")):

return True else:

return False def Tx():

if(match("\*")):

if(F()):

if(Tx()):

return True else:

return False

else:

return False

else:

return True def T():

if(F()):

if(Tx()):

return True else:

return False

else:

return False def Ex():

if(match("+")):

if(T()):

if(Ex()):

return True else:

return False

else:

return False

else:

return True def E():

if(T()):

if(Ex()):

return True else:

return False

else:

return False if(E()):

if(i==len(s)):

print("String is accepted") else:

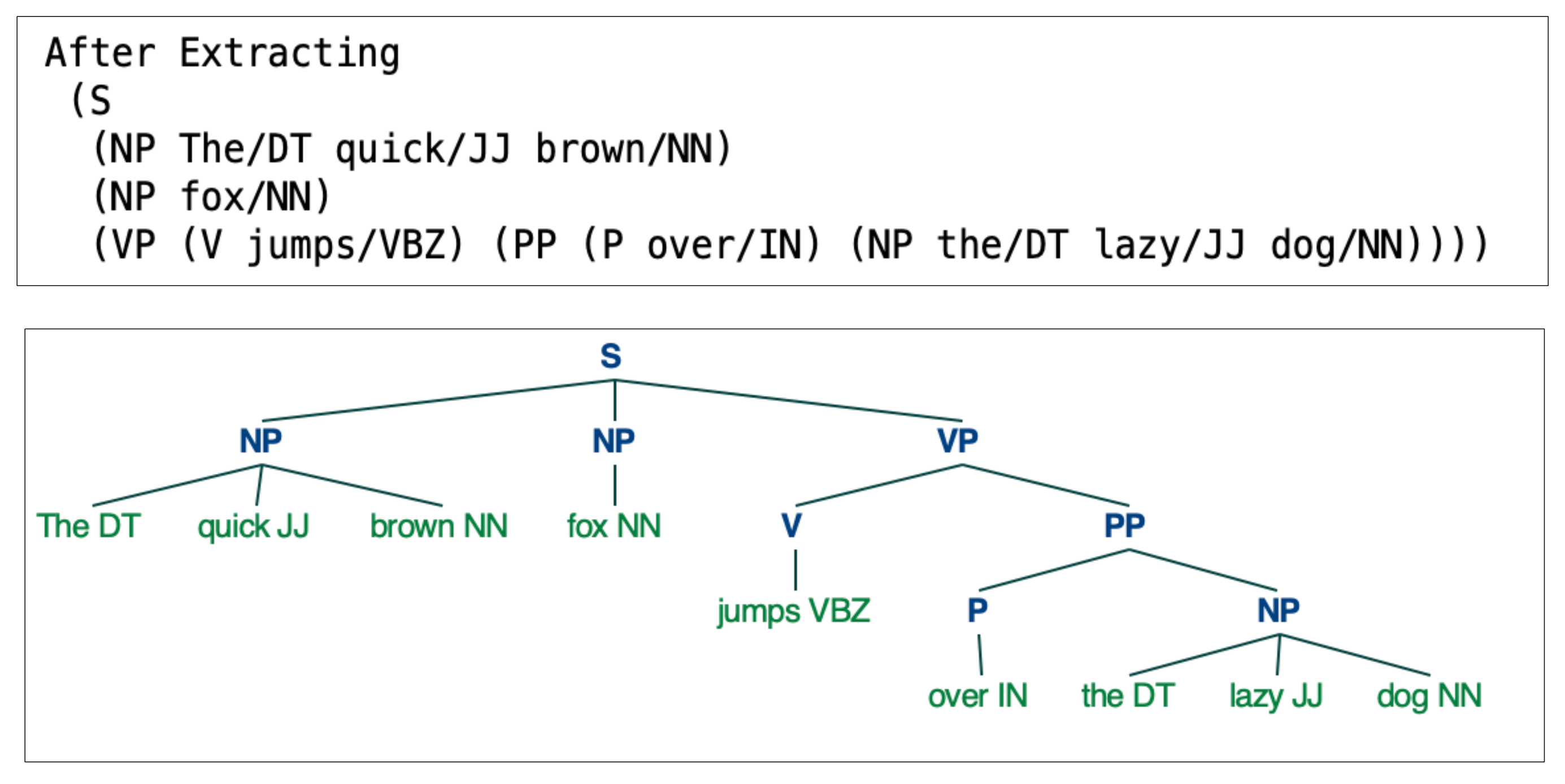
print("String is not accepted")

else:

print("string is not accepted")

OUTPUT

Text

Description automatically generated

# CONCLUSION:

Thus we have studied and performed top down and bottom up parsing using CGF for English language.