**COMPUTING FIRST AND FOLLOW**

Code

import sys

sys.setrecursionlimit(60)

def first(string):

    first\_ = set()

    if string in nont:

        alternatives = productions\_dict[string]

        for alternative in alternatives:

         first\_2 = first(alternative)

         first\_ = first\_ |first\_2

    elif string in terminals:

        first\_ = {string}

    elif string=='' or string=='e':

        first\_ = {'e'}

    else:

        first\_2 = first(string[0])

        if 'e' in first\_2:

         i = 1

         while 'e' in first\_2:

            first\_ = first\_ | (first\_2 - {'e'})

            if string[i:] in terminals:

             first\_ = first\_ | {string[i:]}

             break

            elif string[i:] == '':

             first\_ = first\_ | {'e'}

             break

            first\_2 = first(string[i:])

            first\_ = first\_ | first\_2 - {'e'}

            i += 1

        else:

            first\_ = first\_ | first\_2

    return first\_

def follow(nT):

    follow\_ = set()

    prods = productions\_dict.items()

    if nT==s:

        follow\_ = follow\_ | {'$'}

    for nt,rhs in prods:

        for alt in rhs:

            for char in alt:

                if char==nT:

                    following\_str = alt[alt.index(char) + 1:]

                    if following\_str=='':

                        if nt==nT:

                            continue

                        else:

                            follow\_ = follow\_ | follow(nt)

                    else:

                        follow\_2 = first(following\_str)

                        if 'e' in follow\_2:

                            follow\_ = follow\_ | follow\_2-{'e'}

                            follow\_ = follow\_ | follow(nt)

                        else:

                            follow\_ = follow\_ | follow\_2

    return follow\_

terminals=['+','-','\*','(',')','a','b','c']

no\_of\_productions = int(input("Enter no of productions: "))

productions = []

print("Enter the productions:")

for \_ in range(no\_of\_productions):

    productions.append(input())

productions\_dict = {}

nont=[]

for nT in productions:

    if nT[0].isupper():

        nont.append(nT[0])

        productions\_dict[nT[0]] = []

s=productions[0][0]

#print(s)

print("Non Terminals:", nont)

print("productions\_dict",productions\_dict)

for production in productions:

        nonterm\_to\_prod = production.split("->")

        alternatives = nonterm\_to\_prod[1].split("/")

        for alternative in alternatives:

            productions\_dict[nonterm\_to\_prod[0]].append(alternative)

print()

print("productions\_dict",productions\_dict)

FIRST = {}

FOLLOW={}

for non\_terminal in nont:

    FIRST[non\_terminal] = set()

for non\_terminal in nont:

    FOLLOW[non\_terminal] = set()

for non\_terminal in nont:

    FIRST[non\_terminal] = FIRST[non\_terminal] | first(non\_terminal)

print()

print("FIRST: ",FIRST)

FOLLOW[s] = FOLLOW[s] | {'$'}

for non\_terminal in nont:

 FOLLOW[non\_terminal] = FOLLOW[non\_terminal] | follow(non\_terminal)

print()

print("FOLLOW: ",FOLLOW)

Output:

