Compiler Design

Exp–8 Computation of Leading & Trailing

Name:- K. DUSHYANT REDDY Reg No.:- RA1911033010029

Branch:- CSE-SE

Date:- 15/03/22

AIM: To implement Computation of Leading and Trailing.

CODE:

#Leading & Trailing

a = ["E=E+T","E=T","T=T\*F","T=F","F=(E)","F=i"]

rules = {}

terms = []

for i in a:

    temp = i.split("=")

    terms.append(temp[0])

    try:

        rules[temp[0]] += [temp[1]]

    except:

        rules[temp[0]] = [temp[1]]

terms = list(set(terms))

print(rules,terms)

*def* leading(*gram*, *rules*, *term*, *start*):

    s = []

    if *gram*[0] not in terms:

        return *gram*[0]

    elif len(*gram*) == 1:

        return [0]

    elif *gram*[1] not in terms and *gram*[-1] is not *start*:

        for i in *rules*[*gram*[-1]]:

            s+= leading(i, *rules*, *gram*[-1], *start*)

            s+= [*gram*[1]]

        return s

*def* trailing(*gram*, *rules*, *term*, *start*):

    s = []

    if *gram*[-1] not in terms:

        return *gram*[-1]

    elif len(*gram*) == 1:

        return [0]

    elif *gram*[-2] not in terms and *gram*[-1] is not *start*:

        for i in *rules*[*gram*[-1]]:

            s+= trailing(i, *rules*, *gram*[-1], *start*)

            s+= [*gram*[-2]]

        return s

leads = {}

trails = {}

for i in terms:

    s = [0]

    for j in rules[i]:

        s+=leading(j,rules,i,i)

    s = set(s)

    s.remove(0)

    leads[i] = s

    s = [0]

    for j in rules[i]:

        s+=trailing(j,rules,i,i)

    s = set(s)

    s.remove(0)

    trails[i] = s

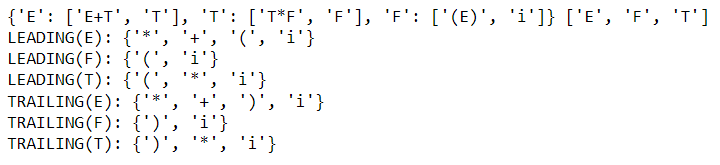
for i in terms:

    print("LEADING("+i+"):",leads[i])

for i in terms:

    print("TRAILING("+i+"):",trails[i])

OUTPUT:



RESULT:

The code is successfully implemented in Python language.