Compiler Design

Exp–5 First And Follow

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

Branch:- CSE-SE

Date:- 22/02/22

AIM: To implement First and Follow

CODE:

#First and Follow Pos

n=int(input("Enter Number of productions "))

Sym=[]

V=[]

T=[]

P=[]

for i in range(n):

    prod=input("Enter production of the form V->(V U T) ")

    v=prod[0]

    if v not in V:

        V.append(v)

        P.append([])

    if(v not in Sym):

        Sym.append(v)

    ind=V.index(v)

    for j in range(3,len(prod)):

        if(prod[j] not in Sym and prod[j]!='|' and prod[j]!='ε'):

            Sym.append(prod[j])

    lst=prod[3:].split('|')

    for z in lst:

        P[ind].append(z)

for j in Sym:

    if j not in V:

        T.append(j)

First=[]

for i in range(len(Sym)):

    First.append([])

Sym=[]

Sym.extend(T)

Sym.extend(V)

eps=[]

t=len(T)

for i in range(t):

    First[i].append(Sym[i])

for j in range(t,len(Sym)):

    for x in P[j-t]:

        if(x[0] in T):

            First[j].append(x[0])

    if('ε' in P[j-t]):

        First[j].append('ε')

        eps.append(V[j-t])

change=1

while(change!=0):

    change=0

    for j in range(t,len(Sym)):

        if(Sym[j] not in eps):

            for p in P[j-t]:

                flag=0

                for k in p:

                    if(k not in eps):

                        flag=1

                        break

                if(flag==0):

                    First[j].append('ε')

                    eps.append(V[j-t])

                    change+=1

                    break

change=1

while(change!=0):

    change=0

    for j in range(t,len(Sym)):

         for p in P[j-t]:

           if(p!='ε'):

            k=0

            idx=Sym.index(p[k])

            for z in First[idx]:

                if(z not in First[j] and z!='ε'):

                    First[j].append(z)

                    change+=1

            while('ε' in First[idx] and k<len(p)):

                k+=1

                idx=Sym.index(p[k])

                for z in First[idx]:

                    if (z not in First[j] and z!='ε'):

                        First[j].append(z)

                        change+=1

print("Symbol   First-Pos")

for i in range(len(V)):

    print(V[i],"      ",First[t+i])

Follow=[]

for i in range(len(V)):

    Follow.append([])

Follow[0].append('$')

for i in P:

    for j in i:

        for k in range(len(j)-1):

            var=j[k]

            if(var in V):

                idx=V.index(var)

                l=k+1

                while(l<len(j)):

                    if(j[l] in T):

                        Follow[idx].append(j[l])

                        break

                    else:

                        ind=Sym.index(j[l])

                        for x in First[ind]:

                            if(x not in Follow[idx] and x!='ε'):

                                Follow[idx].append(x)

                        if(j[l] not in eps):

                            break

                    l+=1

change=1

while(change!=0):

    change=0

    for i in range(len(P)):

        for j in P[i]:

            l=len(j)-1

            if(j[l] in V):

                ind=Sym.index(j[l])-t

                for x in Follow[i]:

                    if(x not in Follow[ind] and x!='ε'):

                        Follow[ind].append(x)

                        change+=1

                l-=1

                while(l>=0):

                    if(j[l+1] in eps):

                        ind=Sym.index(j[l])-t

                        for x in Follow[i]:

                            if(x not in Follow[ind] and x!='ε'):

                                Follow[ind].append(x)

                                change+=1

                        l-=1

                    else:

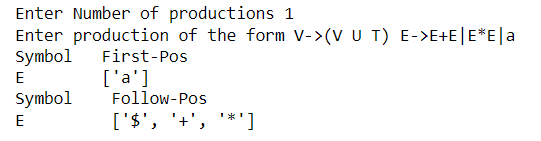
                        break

print("Symbol    Follow-Pos")

for i in range(len(Follow)):

    print(V[i],"       ",Follow[i])

OUTPUT:



RESULT:

The code is successfully implemented in Python language.