|  |  |
| --- | --- |
| Name: Varun Khadayate | Subject: Compiler Design |
| Roll No: A016 | Date of Submission: 12th October 2021 |

# Aim

Implement operator precedence parser/ LR Parser.

# Lab Assignment Program

Implement operator precedence parser/ LR Parser.

## Code

gram = {

    "S":["CC"],

    "C":["aC","d"]

}

start = "S"

terms = ["a","d","$"]

non\_terms = []

for i in gram:

    non\_terms.append(i)

gram["S'"]= [start]

new\_row = {}

for i in terms+non\_terms:

    new\_row[i]=""

non\_terms += ["S'"]

stateTable = []

def closure(term, I):

    if term in non\_terms:

        for i in gram[term]:

            I+=[(term,"."+i)]

    I = list(set(I))

    for i in I:

        if "." != i[1][-1] and i[1][i[1].index(".")+1] in non\_terms and i[1][i[1].index(".")+1] != term:

            I += closure(i[1][i[1].index(".")+1], [])

    return I

Is = []

Is+=set(closure("S'", []))

print("\t\t\tGoto Steps")

countI = 0

omegaList = [set(Is)]

while countI<len(omegaList):

    newrow = dict(new\_row)

    vars\_in\_I = []

    Is = omegaList[countI]

    countI+=1

    for i in Is:

        if i[1][-1]!=".":

            ind = i[1].index(".")

            vars\_in\_I+=[i[1][ind+1]]

    vars\_in\_I = list(set(vars\_in\_I))

    for i in vars\_in\_I:

        In = []

        for j in Is:

            if "."+i in j[1]:

                rep = j[1].replace("."+i,i+".")

                In+=[(j[0],rep)]

        if (In[0][1][-1]!="."):

            temp = set(closure(i,In))

            if temp not in omegaList:

                omegaList.append(temp)

            if i in non\_terms:

                newrow[i] = str(omegaList.index(temp))

            else:

                newrow[i] = "s"+str(omegaList.index(temp))

            print(f'Goto(I{countI-1},{i}):{temp} That is I{omegaList.index(temp)}')

        else:

            temp = set(In)

            if temp not in omegaList:

                omegaList.append(temp)

            if i in non\_terms:

                newrow[i] = str(omegaList.index(temp))

            else:

                newrow[i] = "s"+str(omegaList.index(temp))

            print(f'Goto(I{countI-1},{i}):{temp} That is I{omegaList.index(temp)}')

    stateTable.append(newrow)

print("\n\n\t\t\tList of I's")

for i in omegaList:

    print(f'I{omegaList.index(i)}: {i}')

I0 = []

for i in list(omegaList[0]):

    I0 += [i[1].replace(".","")]

print(I0)

for i in omegaList:

    for j in i:

        if "." in j[1][-1]:

            if j[1][-2]=="S":

                stateTable[omegaList.index(i)]["$"] = "Accept"

                break

            for k in terms:

                stateTable[omegaList.index(i)][k] = "r"+str(I0.index(j[1].replace(".","")))

print("\n\t\t\tState Table")

print(f'{" ": <9}',end="")

for i in new\_row:

    print(f'|{i: <11}',end="")

print(f'\n{"-":-<66}')

for i in stateTable:

    print(f'{"I("+str(stateTable.index(i))+")": <9}',end="")

    for j in i:

        print(f'|{i[j]: <10}',end=" ")

    print()

## Output

Text

Description automatically generated