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

```
PS E:\TY\CD> & e:/TY/CD/venv/Scripts/python.exe "e:/TY/CD/Practical 10/prac_10_LR_Parser.py"
                    Goto Steps
Goto(I0,a):{('C', 'a.C'), ('C', '.aC'), ('C', '.d')} That is I1
Goto(I0,d):{('C', 'd.')} That is I2
Goto(I0,S):{("S'", 'S.')} That is I3
Goto(I0,C):{('C', '.aC'), ('S', 'C.C'), ('C', '.d')} That is I4
Goto(I1,a):{('C', 'a.C'), ('C', '.aC'), ('C', '.d')} That is I1
Goto(I1,d):{('C', 'd.')} That is I2
Goto(I1,C):{('C', 'aC.')} That is I5
Goto(I4,a):{('C', 'a.C'), ('C', '.aC'), ('C', '.d')} That is I1
Goto(I4,d):{('C', 'd.')} That is I2
Goto(I4,C):{('S', 'CC.')} That is I6


                    List of I's
I0: {('C', '.aC'), ('S', '.CC'), ("S'", '.S'), ('C', '.d')}
I1: {('C', 'a.C'), ('C', '.aC'), ('C', '.d')}
I2: {('C', 'd.')}
I3: {("S'", 'S.')}
I4: {('C', '.aC'), ('S', 'C.C'), ('C', '.d')}
I5: {('C', 'aC.')}
I6: {('S', 'CC.')}
['aC', 'CC', 'S', 'd']

                    State Table
         |a          |d          |$          |S          |C
 ------------------------------------------------------------------
 I(0)    |s1         |s2         |           |3          |4
 I(1)    |s1         |s2         |           |           |5
 I(2)    |r3         |r3         |r3         |           |
 I(3)    |           |           |Accept     |           |
 I(4)    |s1         |s2         |           |           |6
 I(5)    |r0         |r0         |r0         |           |
 I(6)    |r1         |r1         |r1         |           |
PS E:\TY\CD>
```