

Practical 4

Niketkumar Kothari

18bce134

```
*C:\Users\kotha\Desktop\leftFinal.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
pract.c leftFinal.py
1  """
2  A->ABd|Aa|a
3  B->Be|b
4
5  A->ABd|Aa|a
6  B->BAe|b
7
8  S->E+E|E*E|a
9
10 S->(L)|a
11 L->L,S|S
12 """
13
14 terminals = []
15 nonterminals = []
16 production_count = int(input("Enter the number of productions : "))
17 productions = []
18 print("Enter the productions : ")
19 for k in range(production_count):
20     input_prod = input()
21     for i in input_prod:
22         if(i.isupper() and i not in nonterminals):
23             nonterminals.append(i)
24         elif(not i.isupper() and i not in ['|','-','>','*']) and i not in terminals):
25             terminals.append(i)
26     productions.append(input_prod)
27
28 # enable the dict to hold lists
29 production_dict = {}
30 for nt in nonterminals:
31     production_dict[nt] = []
32
33 # split the productions into parts to simplify parsing
34 for production in productions:
35     nonterminal_to_production = production.split(">")
36     expanded = nonterminal_to_production[1].split("|") # assumption : single char terminals
37     for ex in expanded:
38         production_dict[nonterminal_to_production[0]].append(ex)
39     print(production_dict)
40
41 grammar_visited = {},[]
42 for k in production_dict.keys():
43     while True:
length: 1,972 lines: 77 Ln: 48 Col: 5 Sel: 0|0 Windows (CR LF) UTF-8 INS
Python file
```

```
*C:\Users\kotha\Desktop\leftFinal.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2

34 # split the productions into parts to simplify parsing
35 for production in productions:
36     nonterminal_to_production = production.split("=>")
37     expanded = nonterminal_to_production[1].split("|") # assumption : single char terminals
38     for ex in expanded:
39         production_dict[nonterminal_to_production[0]].append(ex)
40     print(production_dict)
41
42 grammar, visited = {}, {}
43 for k in production_dict.keys():
44     while True:
45         temp = []
46
47         for v in production_dict[k]:
48             if v[0] in visited:
49                 temp += [x+v[1:] for x in production_dict[v[0]]]
50             else:
51                 temp.append(v)
52
53         if all([False if x[0] in visited else True for x in temp]):
54             production_dict[k] = temp
55             break
56         production_dict[k] = temp
57
58 present = False
59 for v in production_dict[k]:
60     if v[0] == k:
61         print("Left recursion present in production of {}".format(k))
62         present = True
63         break
64
65 if present:
66     temp, grammar[k+""] = [], {}
67     for v in production_dict[k]:
68         if v[0] == k:
69             grammar[k+""] += v[1:] + k + ""
70         else:
71             temp.append(v+k+"")
72     grammar[k+""] += "1" + append("1")
73     production_dict[k] = temp
74     visited.append(k)
75
76 print("\nGrammar after elimination of recursion : ")
77 for k, v in (**production_dict, **grammar).items(): print(k, " -> ", " | ".join(v))

Python file length: 1,972 lines: 77 Ln: 48 Col: 5 Sel: 0 | 0 Windows (CR LF) UTF-8 INS
Type here to search
```

```
C:\Users\kotha\Desktop\leftFinal.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2

1 A->ABd|Aa|a
2 B->Bae|b
3
4 A->ABd|Aa|a
5 B->Bae|b
6
7 B->B*E|E*B|a
8
9 S->(L) | a
10 L->L, S | S
11
12
13
14
15 terminals = []
16 nonterminals = ['A', 'B']
17 production_count = 2
18 productions = {'A': ['ABd', 'Aa', 'a'], 'B': ['Bae', 'b']}
19 print("Enter the left recursion present in production of A")
20 for k in range(2):
21     input_prod = input("Enter the left recursion present in production of {}".format(k))
22
23     for i in input_prod:
24         if i.isalpha():
25             nonterminals.append(i)
26         elif i == '(':
27             terminals.append(i)
28
29 productions[k] = C:\Users\kotha\Desktop>
30
31 # enable the dictionary
32 production_dict = {}
33 for nt in nonterminals:
34     production_dict[nt] = []
35
36 # split the productions into parts to simplify parsing
37 for production in productions:
38     nonterminal_to_production = production.split("=>")
39     expanded = nonterminal_to_production[1].split("|") # assumption : single char terminals
40     for ex in expanded:
41         production_dict[nonterminal_to_production[0]].append(ex)
42     print(production_dict)
43
44 grammar, visited = {}, {}
45 for k in production_dict.keys():

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kotha\Desktop>leftFinal.py
C:\Users\kotha\Desktop>leftFinal.py
Enter the number of productions : 2
Enter the productions :
nonterminals = ['A', 'B']
production_count = 2
productions = {'A': ['ABd', 'Aa', 'a'], 'B': ['Bae', 'b']}
print("Enter the left recursion present in production of A")
for k in range(2):
    input_prod = input("Enter the left recursion present in production of {}".format(k))

    for i in input_prod:
        if i.isalpha():
            nonterminals.append(i)
        elif i == '(':
            terminals.append(i)

    productions[k] = C:\Users\kotha\Desktop>

# enable the dictionary
production_dict = {}
for nt in nonterminals:
    production_dict[nt] = []

# split the productions into parts to simplify parsing
for production in productions:
    nonterminal_to_production = production.split("=>")
    expanded = nonterminal_to_production[1].split("|") # assumption : single char terminals
    for ex in expanded:
        production_dict[nonterminal_to_production[0]].append(ex)
    print(production_dict)

grammar, visited = {}, {}
for k in production_dict.keys():

Python file length: 1,974 lines: 78 Ln: 3 Col: 1 Sel: 7 | 1 Windows (CR LF) UTF-8 INS
Type here to search
```