

Experiment - 5 WAP to remove left recursion from a grammar

Name: Shaikh Shadab

Rollno : 17DCO74

Class : TE.CO

Batch : B3

#Source Code

```
__author__ = 'Shadab Shaikh'
__title__ = 'Finding & resolving left recursion from a grammar'
__date__ = '26-02-2019'
__version__ = '1.0'

print('Author      : ' + __author__)
print('Title       : ' + __title__)
print('Date        : ' + __date__)
print('Version     : ' + __version__)

grammararr=[]      #stores the grammar maintaining the index
alpharest=[]       #stores the alpha and rest values
beta=[]            #stores beta
inputs=""
newprod=""         #to display new production
while(inputs!='no'):
    grammar = input("\nEnter the grammar left should be variable following with ->
format eg: S->a\n")
    grammar=grammar.replace(" ", "")      #replacing whitespaces with none
    if(grammar[0].islower()):
        grammar[0].upper()               #making left most as variable
    grammararr.append(grammar)             #storing into list
    inputs = input("\nPress no to stop writing productions or write anything to continue")
                                           #asking for the continuity of grammar

def findalpharest(grammararr,k):
    """function to find alpha and rest values"""
    if(grammararr[k][0]==grammararr[k][3]):
        alpharest.append(grammararr[k][0]+grammararr[k][4:])
        k+=1
    if(k<len(grammararr)):
        findalpharest(grammararr,k)

findalpharest(grammararr,0)                #calling findalpharest function

def findbeta(grammararr,alpharest,l,m):
```

```

"""function to find beta for corresponding alpha values"""
if(grammararr[l][0]==alpharest[m][0]):
    if(grammararr[l][0]!=grammararr[l][3]):
        beta.append(grammararr[l][0]+grammararr[l][3])
l+=1
if(l<len(grammararr)):
    findbeta(grammararr,alpharest,l,m)
else:
    l=0
    m+=1
    if(m<len(alpharest)):
        findbeta(grammararr,alpharest,l,m)

findbeta(grammararr,alpharest,0,0) #calling findbeta function

def formnewprod(grammararr,alpharest,beta,i,j):
    #function to print new production
    print(alpharest[i][0]+"->" +beta[j][1]+alpharest[i][0]+""")
    print(alpharest[i][0]+"""+ "-">" +alpharest[i][1]+alpharest[i][0]+""")
    print(alpharest[i][0]+"""+ "#")
    i+=1
    j+=1
    if(i<len(alpharest) and j<len(beta)):
        formnewprod(grammararr,alpharest,beta,i,j)

formnewprod(grammararr,alpharest,beta,0,0) #calling formnewprod function

#print(*alpharest, sep='\n')
#print(*beta, sep='\n')

```

#Sample input

1. E->E+T/T
2. S->(L)/x
L-Ls/s

1. Output for the 1st sample input

```
aikt@aiktcl1:/media/aikt/D2C5-103D/Academics/[Study Material]/Degree/CBG6 REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursion$ python3 leftrec.py
File Edit View Search Terminal Help
aikt@aiktcl1:/media/aikt/D2C5-103D/Academics/[Study Material]/Degree/CBG6 REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursion$ python3 leftrec.py
Author      : Shadab Shaikh
Title       : Finding & resolving left recursion from a grammar
Date        : 26-02-2019
Version     : 1.0
Press no to stop writing productions or write anything to continue*)
Enter the grammar left should be variable following with -> format eg: S->a
E->E+T
def findalpharest(grammararr,n):
Press no to stop writing productions or write anything to continuuees
Enter the grammar left should be variable following with -> format eg: S->a
E->T
def k(len(grammararr)):
Press no to stop writing productions or write anything to continueno
E->TE'
E'->E'
E#
aikt@aiktcl1:/media/aikt/D2C5-103D/Academics/[Study Material]/Degree/CBG6 REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursion$
18         if grammararr[l][0] == alpharest[m][0]:
19             (grammararr[l][0], grammararr[l][1]) =
20                 beta.append(grammararr[l][0], grammararr[l][1:])
21         else:
22             beta.append(grammararr[l][0], grammararr[l][1:])
23     l+=1
24     if (l==len(grammararr)):
25         findbeta(grammararr,alpharest,l,m)
26     else:
27         l=0
28         m=-1
29         if (m==len(alpharest)):
30             findbeta(grammararr,alpharest,l,m)
31
32 findbeta(grammararr,alpharest,0,0) #calling findbeta function
33
34 def forenewprod(grammararr,alpharest,beta,i,j):
35     #for new prod
```

2. Output for the 2nd sample input

```
aikt@aiakt1:/media/aikt/D2C5-103D/Academics/Study Material/Degree/CBG6 REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5 - Left Recursion$ python3 leftrec.py
File Edit View Search Terminal Help
aikt@aiakt1:/media/aikt/D2C5-103D/Academics/[Study Material]/Degree/CBG6 REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5
- Left Recursion$ python3 leftrec.py
Author      : Shadab Shaikh
Title       : Finding & resolving left recursion from a grammar
Date        : 26-02-2019
Version     : 1.0
Press no to stop writing productions or write anything to continue"
Waiting for the completion of grammar

Enter the grammar left should be variable following with -> format eg: S->a
S->(L)
Press no to stop writing productions or write anything to continueyes
Enter the grammar left should be variable following with -> format eg: S->a
S->x
Press no to stop writing productions or write anything to continueyes
Enter the grammar left should be variable following with -> format eg: S->a
L->LS
Press no to stop writing productions or write anything to continueyes
Enter the grammar left should be variable following with -> format eg: S->a
L->S
Press no to stop writing productions or write anything to continuenuo
L->SL'
L'->SL'
L#
aikt@aiakt1:/media/aikt/D2C5-103D/Academics/[Study Material]/Degree/CBG6 REV 2016 2017-2020 MUMBAI UNIVERSITY/TE/Sem 6/Labs/CSL602 System Software Lab 01/Experiment 5
- Left Recursion$
10 m=len(alpharest):
11     findbeta(grammararr,alpharest,l,m)
12
13 findbeta(grammararr,alpharest,0,0) = calling findbeta function
14
15 def fornewprod(grammararr,alpharest,beta,i,j):
16     # new production rule
17     newprod=""
18     newprod+=alpha[i:j]
19     newprod+=beta
20     newprod+=alpharest[j:m]
21     newprod+="\n"
22     grammararr.append(newprod)
23     return newprod
24
25 # main function
26 def main():
27     # taking input from user
28     grammararr=[]
29     alpharest=""
30     # taking input from user
31     while True:
32         s=input("Enter the grammar left should be variable following with -> format eg: S->a\n")
33         if s=="":
34             continue
35         if s=="x":
36             break
37         if s=="y":
38             break
39         if s=="n":
40             break
41         if s=="o":
42             break
43         if s=="u":
44             break
45         if s=="l":
46             break
47         if s=="s":
48             break
49         if s=="l'":
50             break
51         if s=="l#":
52             break
53         if s=="l->SL'":
54             break
55         if s=="L->LS":
56             break
57         if s=="S->(L)":
58             break
59         if s=="S->a":
60             break
61         if s=="S->x":
62             break
63         if s=="S->y":
64             break
65         if s=="S->z":
66             break
67         if s=="S->w":
68             break
69         if s=="S->v":
70             break
71         if s=="S->u":
72             break
73         if s=="S->t":
74             break
75         if s=="S->r":
76             break
77         if s=="S->q":
78             break
79         if s=="S->p":
80             break
81         if s=="S->o":
82             break
83         if s=="S->n":
84             break
85         if s=="S->m":
86             break
87         if s=="S->l":
88             break
89         if s=="S->k":
90             break
91         if s=="S->j":
92             break
93         if s=="S->i":
94             break
95         if s=="S->h":
96             break
97         if s=="S->g":
98             break
99         if s=="S->f":
100            break
101         if s=="S->e":
102            break
103         if s=="S->d":
104            break
105         if s=="S->c":
106            break
107         if s=="S->b":
108            break
109         if s=="S->a":
110            break
111         if s=="S->z":
112            break
113         if s=="S->y":
114            break
115         if s=="S->x":
116            break
117         if s=="S->w":
118            break
119         if s=="S->v":
120            break
121         if s=="S->u":
122            break
123         if s=="S->t":
124            break
125         if s=="S->r":
126            break
127         if s=="S->q":
128            break
129         if s=="S->p":
130            break
131         if s=="S->o":
132            break
133         if s=="S->n":
134            break
135         if s=="S->m":
136            break
137         if s=="S->l":
138            break
139         if s=="S->k":
140            break
141         if s=="S->j":
142            break
143         if s=="S->i":
144            break
145         if s=="S->h":
146            break
147         if s=="S->g":
148            break
149         if s=="S->f":
150            break
151         if s=="S->e":
152            break
153         if s=="S->d":
154            break
155         if s=="S->c":
156            break
157         if s=="S->b":
158            break
159         if s=="S->a":
160            break
161         if s=="S->z":
162            break
163         if s=="S->y":
164            break
165         if s=="S->x":
166            break
167         if s=="S->w":
168            break
169         if s=="S->v":
170            break
171         if s=="S->u":
172            break
173         if s=="S->t":
174            break
175         if s=="S->r":
176            break
177         if s=="S->q":
178            break
179         if s=="S->p":
180            break
181         if s=="S->o":
182            break
183         if s=="S->n":
184            break
185         if s=="S->m":
186            break
187         if s=="S->l":
188            break
189         if s=="S->k":
190            break
191         if s=="S->j":
192            break
193         if s=="S->i":
194            break
195         if s=="S->h":
196            break
197         if s=="S->g":
198            break
199         if s=="S->f":
200            break
201         if s=="S->e":
202            break
203         if s=="S->d":
204            break
205         if s=="S->c":
206            break
207         if s=="S->b":
208            break
209         if s=="S->a":
210            break
211         if s=="S->z":
212            break
213         if s=="S->y":
214            break
215         if s=="S->x":
216            break
217         if s=="S->w":
218            break
219         if s=="S->v":
220            break
221         if s=="S->u":
222            break
223         if s=="S->t":
224            break
225         if s=="S->r":
226            break
227         if s=="S->q":
228            break
229         if s=="S->p":
230            break
231         if s=="S->o":
232            break
233         if s=="S->n":
234            break
235         if s=="S->m":
236            break
237         if s=="S->l":
238            break
239         if s=="S->k":
240            break
241         if s=="S->j":
242            break
243         if s=="S->i":
244            break
245         if s=="S->h":
246            break
247         if s=="S->g":
248            break
249         if s=="S->f":
250            break
251         if s=="S->e":
252            break
253         if s=="S->d":
254            break
255         if s=="S->c":
256            break
257         if s=="S->b":
258            break
259         if s=="S->a":
260            break
261         if s=="S->z":
262            break
263         if s=="S->y":
264            break
265         if s=="S->x":
266            break
267         if s=="S->w":
268            break
269         if s=="S->v":
270            break
271         if s=="S->u":
272            break
273         if s=="S->t":
274            break
275         if s=="S->r":
276            break
277         if s=="S->q":
278            break
279         if s=="S->p":
280            break
281         if s=="S->o":
282            break
283         if s=="S->n":
284            break
285         if s=="S->m":
286            break
287         if s=="S->l":
288            break
289         if s=="S->k":
290            break
291         if s=="S->j":
292            break
293         if s=="S->i":
294            break
295         if s=="S->h":
296            break
297         if s=="S->g":
298            break
299         if s=="S->f":
300            break
301         if s=="S->e":
302            break
303         if s=="S->d":
304            break
305         if s=="S->c":
306            break
307         if s=="S->b":
308            break
309         if s=="S->a":
310            break
311         if s=="S->z":
312            break
313         if s=="S->y":
314            break
315         if s=="S->x":
316            break
317         if s=="S->w":
318            break
319         if s=="S->v":
320            break
321         if s=="S->u":
322            break
323         if s=="S->t":
324            break
325         if s=="S->r":
326            break
327         if s=="S->q":
328            break
329         if s=="S->p":
330            break
331         if s=="S->o":
332            break
333         if s=="S->n":
334            break
335         if s=="S->m":
336            break
337         if s=="S->l":
338            break
339         if s=="S->k":
340            break
341         if s=="S->j":
342            break
343         if s=="S->i":
344            break
345         if s=="S->h":
346            break
347         if s=="S->g":
348            break
349         if s=="S->f":
350            break
351         if s=="S->e":
352            break
353         if s=="S->d":
354            break
355         if s=="S->c":
356            break
357         if s=="S->b":
358            break
359         if s=="S->a":
360            break
361         if s=="S->z":
362            break
363         if s=="S->y":
364            break
365         if s=="S->x":
366            break
367         if s=="S->w":
368            break
369         if s=="S->v":
370            break
371         if s=="S->u":
372            break
373         if s=="S->t":
374            break
375         if s=="S->r":
376            break
377         if s=="S->q":
378            break
379         if s=="S->p":
380            break
381         if s=="S->o":
382            break
383         if s=="S->n":
384            break
385         if s=="S->m":
386            break
387         if s=="S->l":
388            break
389         if s=="S->k":
390            break
391         if s=="S->j":
```