

Programa 7 - Gramatica No Ambigua

Leon Tejeda 2CM5

Enero 2021

PLANTEAMIENTO DEL PROBLEMA:

Elaborar un programa para procesar la gramática ambigua:

$B \rightarrow (RB \mid e \mid R \mid) \mid (RR$

Adicionalmente, el programa debe de contar con las siguientes características:

1. La cadena puede ingresar por el usuario o automáticamente.
2. Mandar a un archivo y en pantalla la evaluación de la gramática imprimiendo el símbolo que está evaluando, indicando que producción se aplico y la cadena evaluada.
4. La longitud máxima será manejar cadenas de 1,000 caracteres.

DESARROLLO:

Se realizo un menu para que el usuario decida si quiere generar una cadena aleatorio o si desea ingresarla manualmente.

Ya una vez con la cadena bien definida se empezara derivando la lentra B, sera una proceso de LL(1)

Una vez cumpliendo con una condicional, se asignara la produccion correspondiente y se añadira en otra cadena auxiliar

Tambien se pondran condicionales para determinar que simbolo se desea expandir

Se mostrara en pantalla el procedimiento de evaluacion, y de igual manera se guardara en una archivo de texto

CODIGO:

```
1 import random
2
3 def archivo(simbolo, produccion, cadena_evaluada, cadena_resultante
  , estado, indice):
4
5     if estado == 'e':
6         if indice == 0:
7             ##Abre el archivo por primera vez
8             archivo = open("Programa7_Evaluacion.txt", "w")
9             cadena = "Simbolo que se esta evaluando: " + simbolo +
"\tProduccion aplicada: " + produccion + "\nCadena Evaluada: "
+ cadena_evaluada + "\t\t\tCadena Resultante: " +
cadena_resultante + "\n"
10             archivo.write(cadena)
11             print(cadena)
12
13         else:
14             ##Segunda o mas veces que abre el archivo
15             archivo = open("Programa7_Evaluacion.txt", "a")
16             cadena = "Simbolo que se esta evaluando: " + simbolo +
"\tProduccion aplicada: " + produccion + "\nCadena Evaluada: "
+ cadena_evaluada + "\t\t\tCadena Resultante: " +
cadena_resultante + "\n"
17             archivo.write(cadena)
18             print(cadena)
19
20     elif estado == 'a':
21         archivo = open("Programa7_Evaluacion.txt", "a")
22         cadena = "Simbolo que se esta evaluando: " + simbolo + "\
\tProduccion aplicada: " + produccion + "\nCadena Evaluada: " +
"\t\t\tCadena Resultante: " + cadena_resultante + "\n"
23         archivo.write(cadena)
24         print(cadena)
25
26         archivo.write("--Cadena Aceptada--")
27         print("--Cadena Aceptada--")
28
29     else:
30         if indice == 0:
31             ##Abre el archivo por primera vez
32             archivo = open("Programa7_Evaluacion.txt", "w")
33             cadena = "Simbolo que se esta evaluando: " + simbolo +
"\tProduccion aplicada: " + produccion + "\nCadena Evaluada: "
+ cadena_evaluada + "\t\t\tCadena Resultante: " +
cadena_resultante + "\n"
34             archivo.write(cadena)
35             print(cadena)
36
37             archivo.write("--Cadena Rechazada--")
38             print("--Cadena Rechazada--")
39
40         else:
41             archivo = open("Programa7_Evaluacion.txt", "a")
42             archivo.write("--Cadena Rechazada--")
43             print("--Cadena Rechazada--")
44
45
46
47
48     archivo.close()
49
```

```

50 def proceso (cadena):
51     expandir = 'B'
52     cadena_aux = expandir
53
54     if cadena[0] != ')':
55         for i in range (0, len(cadena)):
56             nueva_cadena= "e"
57             simbolo = cadena[i]
58
59             if expandir == 'B' and simbolo == '(':
60                 eleccion = "(RB"
61                 cadena_aux = cadena_aux.replace("B", eleccion)
62
63             elif expandir == 'R' and simbolo == ')':
64                 eleccion = ")"
65                 cadena_aux = cadena_aux.replace("R", eleccion)
66
67             elif expandir == 'R' and simbolo == '(':
68                 eleccion = "(RR"
69                 cadena_aux = cadena_aux.replace("R", eleccion)
70
71             for j in range (0, len(cadena_aux)):
72                 if cadena_aux[j] == 'R':
73                     expandir = 'R'
74                     break
75
76                 elif cadena_aux[j] == 'B':
77                     expandir = 'B'
78                     break
79
80             for k in range(i, len(cadena)):
81                 if nueva_cadena == 'e':
82                     nueva_cadena = cadena[k]
83
84             else:
85                 nueva_cadena = nueva_cadena + cadena[k]
86
87             archivo(simbolo, eleccion, nueva_cadena, cadena_aux, 'e
88             ',i)
89
90             if i == len(cadena)-1:
91                 if cadena_aux == cadena + 'B':
92                     archivo(simbolo, eleccion, nueva_cadena, cadena
93                     , 'a', i)
94                     return
95
96             archivo(simbolo, eleccion, nueva_cadena, cadena_aux, 'r', i
97             )
98
99         else:
100             archivo(')', 'NULL', cadena, 'NULL', 'r', 0)
101
102     ciclo = True
103
104     while ciclo == True:
105         print ("Seleccione una Opcion (Se tomara en cuenta que se desea
106         expandir B desde el inicio)")
107         print ("1. Ingresar la cadena manualmente")
108         print ("2. Generarla aleatoriamente")
109
110         opcion = int(input(""))

```

```

108     if opcion == 1:
109         print("Por favor ingrese la cadena, EJ: (()())")
110         cadena = input("")
111         proceso(cadena)
112
113
114     elif opcion == 2:
115         print("Modo aleatorio")
116         n = random.randint(1, 10)
117         cadena = 'e'
118
119         for i in range(0, n*2):
120             digito = random.randint(0, 1)
121
122             if digito == 0:
123                 digito = '('
124
125             else:
126                 digito = ')'
127
128             if cadena == 'e':
129                 cadena = digito
130
131             else:
132                 cadena = cadena + digito
133
134         proceso(cadena)
135
136     else:
137         print("Ingrese una opcion valida")
138
139     print("Desea volver a correr el programa?")
140     print("Presione (y) si lo desea volver a correr, de lo contrario presione cualquier tecla")
141     repetir = input("")
142
143     if repetir != 'y' and repetir != 'Y':
144         ciclo = False

```

FUNCIONAMIENTO:

Menu del programa

```
Seleccione una Opcion (Se tomara en cuenta que se desea expandir B desde el inicio)
1. Ingresar la cadena manualmente
2. Generarla aleatoriamente
█
```

Cadena = (()()- Modo Manual

```
Seleccione una Opcion (Se tomara en cuenta que se desea expandir B desde el inicio)
1. Ingresar la cadena manualmente
2. Generarla aleatoriamente
1
Por favor ingrese la cadena, EJ: (()())
(())()
Simbolo que se esta evaluando: ( Produccion aplicada: (RB
Cadena Evaluada: (())() Cadena Resultante: (RB
Simbolo que se esta evaluando: ( Produccion aplicada: (RR
Cadena Evaluada: (())() Cadena Resultante: ((RRB
Simbolo que se esta evaluando: ) Produccion aplicada: )
Cadena Evaluada: (())() Cadena Resultante: (())B
Simbolo que se esta evaluando: ) Produccion aplicada: )
Cadena Evaluada: (())() Cadena Resultante: (())B
Simbolo que se esta evaluando: ( Produccion aplicada: (RB
Cadena Evaluada: () Cadena Resultante: (())(RB
Simbolo que se esta evaluando: ) Produccion aplicada: )
Cadena Evaluada: ) Cadena Resultante: (())()B
Simbolo que se esta evaluando: ) Produccion aplicada: )
Cadena Evaluada: ) Cadena Resultante: (())()
--Cadena Aceptada--
¿Desea volver a correr el programa?
Presione (y) si lo desea volver a correr, de lo contrario presione cualquier tecla
█
```

Cadena =)(((())()())((- Modo Aleatorio

```
Seleccione una Opcion (Se tomara en cuenta que se desea expandir B desde el inicio)
1. Ingresar la cadena manualmente
2. Generarla aleatoriamente
2
Modo aleatorio
Simbolo que se esta evaluando: ) Produccion aplicada: NULL
Cadena Evaluada: )(((())()())(( Cadena Resultante: NULL
--Cadena Rechazada--
¿Desea volver a correr el programa?
Presione (y) si lo desea volver a correr, de lo contrario presione cualquier tecla
█
```