



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**

TEORÍA COMPUTACIONAL

2CM4

PROFESOR: LUZ MARÍA SÁNCHEZ GARCÍA

PRÁCTICA 4 GRAMÁTICAS LIBRES DE CONTEXTO (GLC)

INTEGRANTES:

VÁZQUEZ MORENO MARCOS OSWALDO

2016601777

QUINTANA RUÍZ AJITZI RICARDO

2017631261



FECHA DE ENTREGA: 16 DE ABRIL DE 2018

INTRODUCCIÓN

En la siguiente práctica se pretende realizar un programa en lenguaje de programación C, la cual se basa en el la derivación de un lenguaje libre de contexto, también conocido como (LLC) o también llamados “Gramática en la forma Backus-Nour” (usado para describir lenguajes de programación. Se elegirá un LLC y se generarán cadenas derivadas del lenguaje de acuerdo hasta cuál número de repetición decida el usuario.

PLANTEAMIENTO DEL PROBLEMA

Implementar el algoritmo de codificación de un programa el cual acepte un número entero del 0 al 300 en el cual se mostrarán las cadenas derivadas del lenguaje que en este caso es **{a,b}**:

DISEÑO DE LA SOLUCIÓN

Una vez con la problemática planteada se necesitaba de poder darle sentido a la solución y es por eso por lo que se decidió crear un programa desarrollado en lenguaje C utilizando unas librerías y unas palabras reservadas capaz de admitir una expresión regular y llevarla a su validación.

Se llevó a cabo mediante una definición de estructuras con 5 cadenas, 3 con un solo carácter y 2 para guardar concatenaciones a la cual se le llamó *cadena*, teniendo posteriormente funciones las cuales una concatena, dos pegan tanto a **a** y **b** entonces se toma a la cadena como un arreglo y justamente en donde sea el contador igual al número dado por el usuario y así hasta llegar a la repetición del número de derivación de la cadena solicitada.

IMPLEMENTACIÓN DE LA SOLUCIÓN

```
1  /*
2  Ajitzi Ricardo Quintana Ruiz  2017631261
3  Vázquez Moreno Marcos Oswaldo Vázquez 2016601777
4  Práctica 4: Grámaticas Libres de contexto
5  2cmd
6  12/04/18
7  */
8
9  //S->aSb|Îµ
10
11 #include <stdio.h>
12 #include <string.h>
13 #include <windows.h>
14
15 typedef struct{
16     char a[2]; //solo contener el caracter a
17     char b[2]; //solo contiene el caracter b
18     char S[2]; //solo contiene el caracter S
19     char aa[300];
20     char bb[300];
21 }cadenas; //declaro una estructura con 5 cadenas, 3 con
22           //un solo caracter y 2 para guardar concatenaciones y la llamo "cadenas"
23
24 cadenas lenguaje = {
25     "a",
26     "b",
27     "S",
28     "", //va contener la concatenacion de a's y b's sin la S
29     ""
30 }; //mi primera estructura "cadenas" se llama lenguaje y contiene los 3 caracteres
31    // de mi LLC y los otros dos vacios
32
33 cadenas simbolo = {
34     "a",
35     "b",
36     "S"
```

```

33 cadenas simbolo ={
34     "a",
35     "b",
36     "S",
37     "",//va a contener las a's concatenadas
38     "",//va a contener las b's concatenadas
39 };//mi segunda estructura "cadenas" se llama simbolo y contiene los 3 caracteres
40 // de mi LLC y los otros dos vacios
41
42 void Concatena (cadenas *len, int n); //declaracion de las funciones
43 void Pasos(cadenas lenguaje, int n);
44 void PegaA (cadenas *sim, int j);
45 void PegaB (cadenas *sim, int j);
46 void Limpiar (cadenas *sim);
47
48 int main (void)
49 {
50     int n;//declaro n para guardar el numero de a's y b's
51     printf("Ingresa el numero de a's y b's\n");//pido el valor de n
52     scanf("%d",&n);//guardo el numero del usuario
53     system("cls");//limpio la pantalla
54     printf("%c\n",156);//imprimo Epsilon
55     Pasos(lenguaje,n);//llamo a mi funcion pasos y le mando una copia de mi
56     //estructura lenguaje y el numero n
57
58     return 0;
59 }
60
61 void Pasos(cadenas lenguaje, int n)
62 {
63     int i=1;//mi contador que empieza en 1
64     char res[500]="";//mi cadena que va a guardar los pasos y los va ir imprimiendo
65     char coma[]=",";//mi cadena coma que contiene una coma
66
67     while(i<n)
68     {
69         PegaA(&simbolo,i);//llamo a mi funcion que pega las a's desde 1 hasta i
70         //pero le mando la dir de simbolo*/
71         PegaB(&simbolo,i);//llamo a mi funcion que pega las b's desde 1 hasta i
72         //pero le mando la dir de simbolo*/
73         strcat(simbolo.aa,lenguaje.S);//concateno las a's con la cadena S
74         strcat(simbolo.aa,simbolo.bb);//concateno las a's con la S y las b's
75         strcat(res,simbolo.aa);//concateno todo en resultado
76         strcat(res,coma);//le concateno la coma
77         printf("%s\n",res);//imprimo mi cadena que contiene todo "res"
78         strcpy(simbolo.aa,"");//vacio donde se guardan las a's
79         strcpy(simbolo.bb,"");//vacio donde se guardan las b's
80
81         i++;//incremento i
82     }
83     Concatena(&lenguaje,n);//llamo a la funcion que me da el ultimo resultado
84     //de n a's y n b's sin la S en la mitad*/
85     strcat(res,lenguaje.aa);//concateno el ultimo valor de res y el resultado
86     //de mi funcion concatena*/
87     printf("%s\n",res);//imprimo res
88 }
89
90
91 void Concatena (cadenas *len, int n)//concatena cambia los valores de lenguajes
92 {
93     for (int i = 0; i < n; ++i)
94     {
95         strcat(len->aa,len->a);//concatena en aa el n a's
96     }
97     for (int i = 0; i < n; ++i)
98     {
99         strcat(len->aa,len->b);//le concatena a las n a's n b's
100     }
101 }
102

```

```

83 Concatena(&lenguaje,n);/*Llamo a la funcion que me da el ultimo resultado
84 de n a's y n b's sin la S en la mitad*/
85 strcat(res,lenguaje.aa);/*concateno el ultimo valor de res y el resultado
86 de mi funcion concatena*/
87 printf("%s\n",res);/*imprimo res
88
89 }
90
91 void Concatena (cadenas *len, int n)//concatena cambia los valores de lenguajes
92 {
93     for (int i = 0; i < n; ++i)
94     {
95         strcat(len->aa,len->a);/*concatena en aa el n a's
96     }
97     for (int i = 0; i < n; ++i)
98     {
99         strcat(len->aa,len->b);/*Le concatena a las n a's n b's
100     }
101 }
102
103 void PegaA (cadenas *sim, int j)//manipula los valores de simbolo
104 {
105     for (int i = 0; i < j; ++i)
106     {
107         strcat(sim->aa,sim->a);/*guarda en aa desde una a hasta i a's
108     }
109 }
110
111 void PegaB (cadenas *sim, int j)
112 {
113     for (int i = 0; i < j; ++i)
114     {
115         strcat(sim->bb,sim->b);/*guarda en bb desde una a hasta i b's
116     }
117 }

```

FUNCIONAMIENTO

C:\WINDOWS\system32\cmd.exe

Ingresa el numero de a's y b's

C:\WINDOWS\system32\cmd.exe

Ingresa el numero de a's y b's

0

£

Quieres ingresar otro numero?

1.-Si 2.-No

C:\WINDOWS\system32\cmd.exe

Ingresa el numero de a's y b's

1

£

ab

Quieres ingresar otro numero?

1.-Si 2.-No

C:\WINDOWS\system32\cmd.exe

Ingresa el numero de a's y b's

3

£

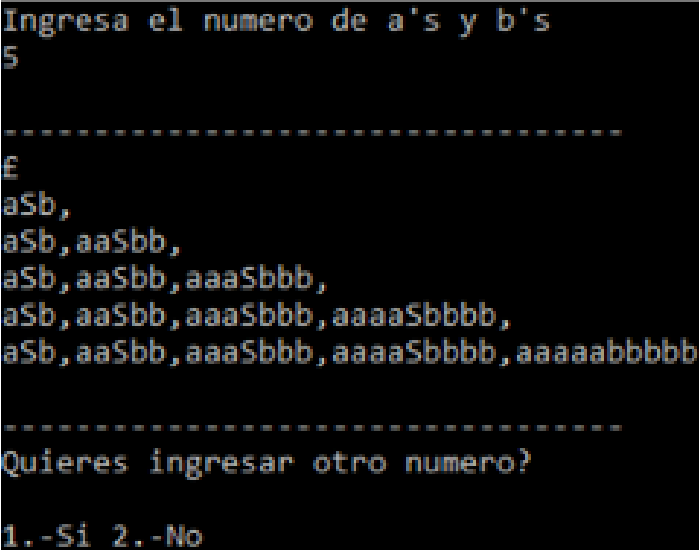
aSb,

aSb,aaSbb,

aSb,aaSbb,aaabbb

Quieres ingresar otro numero?

1.-Si 2.-No



nunca jamás están contentos con el desempeño mostrado, por fortuna, contamos con que nuestra profesora es totalmente clara y no tiene molestias para explicarnos algo que no estamos entendiendo.

Finalmente, el hecho de dejar las prácticas en pareja se nos hace bastante adecuado ya que dos cabezas piensan mejor que una y qué mejor que a la hora de programar los dos compañeros o alumnos se estén aportando ideas.