



UNIVERSIDAD NACIONAL DE SAN AGUSTIN

ESCUELA PROFESIONAL DE
CIENCIA DE LA COMPUTACIÓN
COMPILADORES

Practica

DATOS

Alumnos :

Johann F. Huaypuna

Huanca

Profesor:

Carlos Atencio Torres

Índice

1. Implementación con C++	2
1.1. Implementar conjunto primeros, siguientes y algoritmo de verificación	2
1.1.1. Código	2
1.1.2. Resultado	9
1.1.3. Github	9

1. Implementación con C++

1.1. Implementar conjunto primeros, siguientes y algoritmo de verificación

Se usó la siguiente gramática.

```
1 E := T A
2 A := + T
3 A := #
4 T := F B
5 B := * F B
6 B := #
7 F := ( E )
8 F := n
```

1.1.1. Código

Código 1: primeros y siguientes

```
1 #include<stdio.h>
2 #include<ctype.h>
3 #include<string.h>
4 #include<iostream>
5 using namespace std;
6
7 int cont, n = 0;
8 char calc_primeros[10][100];
9 char calc_siguiete[10][100];
10 int m = 0;
11 char produccion[10][10];
12 char f[10], primero[10];
13 int k;
14 char ck;
15 int e;
16
17 void siguientePrimero(char, int, int);
18 void siguiente(char c);
19 void buscarPrimero(char, int, int);
```

```
20 void buscarPrimero(char c, int q1, int q2)
21 {
22     int j;
23     if(!(isupper(c))) {
24         primero[n++] = c;
25     }
26     for(j = 0; j < cont; j++)
27     {
28         if(produccion[j][0] == c)
29         {
30             if(produccion[j][2] == '#')
31             {
32                 if(produccion[q1][q2] == '\\0')
33                     primero[n++] = '#';
34                 else if(produccion[q1][q2] != '\\0'
35                     && (q1 != 0 || q2 != 0))
36                 {
37
38                     buscarPrimero(produccion[q1][q2], q1, (q2+1));
39                 }
40                 else
41                     primero[n++] = '#';
42             }
43             else if(!isupper(produccion[j][2]))
44             {
45                 primero[n++] = produccion[j][2];
46             }
47             else
48             {
49                 buscarPrimero(produccion[j][2], j, 3);
50             }
51         }
52     }
53 }
54
55 void siguientePrimero(char c, int c1, int c2)
56 {
```

```
57     int k;
58     if (!(isupper(c)))
59         f[m++] = c;
60     else
61     {
62         int i = 0, j = 1;
63         for(i = 0; i < cont; i++)
64         {
65             if(calc_primer0[i][0] == c)
66                 break;
67         }
68
69         while(calc_primer0[i][j] != '!')
70         {
71             if(calc_primer0[i][j] != '#')
72             {
73                 f[m++] = calc_primer0[i][j];
74             }
75             else
76             {
77                 if(produccion[c1][c2] == '\\0')
78                 {
79                     siguiente(produccion[c1][0]);
80                 }
81                 else
82                 {
83                     siguientePrimero(produccion[c1][c2], c1, c2+1);
84                 }
85             }
86             j++;
87         }
88     }
89 }
90
91
92 void siguiente(char c)
93 {
```

```
94     int i , j ;
95     if(produccion[0][0] == c) {
96         f[m++] = '$';
97     }
98     for(i = 0; i < 10; i++)
99     {
100         for(j = 2; j < 10; j++)
101         {
102             if(produccion[i][j] == c)
103             {
104                 if(produccion[i][j+1] != '\\0')
105                 {
106                     siguientePrimero(produccion[i][j+1], i, (j+2));
107                 }
108
109                 if(produccion[i][j+1]== '\\0' && c!=produccion[i][0])
110                 {
111                     siguiente(produccion[i][0]);
112                 }
113             }
114         }
115     }
116 }
117
118
119 int main(int argc , char **argv)
120 {
121     int jm = 0;
122     int km = 0;
123     int i , opcion;
124     char c , ch;
125     cont = 8;
126     strcpy(produccion[0], "E=TA");
127     strcpy(produccion[1], "A=+TA");
128     strcpy(produccion[2], "A=#");
129     strcpy(produccion[3], "T=FB");
130     strcpy(produccion[4], "B=*FB");
```

```
131     strcpy(produccion[5], "B==");
132     strcpy(produccion[6], "F=(E)");
133     strcpy(produccion[7], "F=n");
134     int kay;
135     char done[cont];
136     int ptr = -1;
137
138
139     for(k = 0; k < cont; k++) {
140         for(kay = 0; kay < 100; kay++) {
141             calc_primer[k][kay] = '!';
142         }
143     }
144     int point1 = 0, point2, auxx;
145     cout<<"PRIMEROS"<<endl;
146     for(k = 0; k < cont; k++)
147     {
148         c = produccion[k][0];
149         point2 = 0;
150         auxx = 0;
151
152         for(kay = 0; kay <= ptr; kay++)
153             if(c == done[kay])
154                 auxx = 1;
155
156         if (auxx == 1)
157             continue;
158
159         buscarPrimer(c, 0, 0);
160         ptr += 1;
161
162         done[ptr] = c;
163         cout<<endl;
164         cout<<c<<" = {";
165         calc_primer[point1][point2++] = c;
166
167         for(i = 0 + jm; i < n; i++) {
```

```
168         int lark = 0, chk = 0;
169
170         for(lark = 0; lark < point2; lark++) {
171
172             if (primero[i] == calc_primero[point1][lark])
173             {
174                 chk = 1;
175                 break;
176             }
177         }
178         if(chk == 0)
179         {
180             printf("%c, ", primero[i]);
181             calc_primero[point1][point2++] = primero[i];
182         }
183     }
184     cout<<"}"<<endl;
185     jm = n;
186     point1++;
187 }
188 cout<<endl;
189 cout<<"_____ "<<endl<<endl;
190
191 char donee[cont];
192 ptr = -1;
193
194 for(k = 0; k < cont; k++) {
195     for(kay = 0; kay < 100; kay++) {
196         calc_siguiente[k][kay] = '!';
197     }
198 }
199 point1 = 0;
200 int land = 0;
201 cout<<"SIGUIENTES"<<endl;
202 for(e = 0; e < cont; e++)
203 {
204     ck = produccion[e][0];
```



```
205         point2 = 0;
206         auxx = 0;
207
208         for(kay = 0; kay <= ptr; kay++)
209             if(ck == donee[kay])
210                 auxx = 1;
211
212         if (auxx == 1)
213             continue;
214         land += 1;
215
216         siguiente(ck);
217         ptr += 1;
218
219         donee[ptr] = ck;
220         cout<<ck<<" = {";
221         calc_siguiente[point1][point2++] = ck;
222
223         for(i = 0 + km; i < m; i++) {
224             int lark = 0, chk = 0;
225             for(lark = 0; lark < point2; lark++)
226             {
227                 if (f[i] == calc_siguiente[point1][lark])
228                 {
229                     chk = 1;
230                     break;
231                 }
232             }
233             if(chk == 0)
234             {
235                 printf("%c, ", f[i]);
236                 calc_siguiente[point1][point2++] = f[i];
237             }
238         }
239         cout<<"}"<<endl<<endl;
240         km = m;
241         point1++;
```

242 }
243 }

1.1.2. Resultado

```
PS C:\Users\frAnz\Documents\Compiladores\p3> g++ p3.cpp
PS C:\Users\frAnz\Documents\Compiladores\p3> .\a.exe
PRIMEROS
E = { (, n, }
A = { +, #, }
T = { (, n, }
B = { *, #, }
F = { (, n, }

SIGUIENTES
E = { $, ), }
A = { $, ), }
T = { +, $, ), }
B = { +, $, ), }
F = { *, +, $, ), }
```

Figura 1: Ejecución Primeros y siguientes con verificación

1.1.3. Github

<https://github.com/johanfranz/Compiladores>