UNIVERSIDAD NACIONAL DE SAN AGUSTIN

ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACION

COMPILADORES

GRUPO A

Laboratorio 02

Docente:
Dr. Yuber Velazco Paredes

Realizado por:
Pinto Medina, Brian

11 de mayo de 2020

1. Objetivo

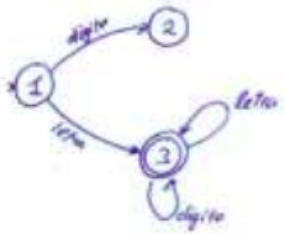
Familiarizarse con la programación de autómatas finitos deterministas utilizando los diagramas de transición o tablas de transición.

2. Desarrollo:

Un identificador puede ser reconocido por un diagrama de transición y/o tabla de transición, así entonces, debe implementar en el lenguaje de programación C o C++ los siguientes programas:

2.1. Implemente el algoritmo por diagramas de transición para el reconocimiento de un identificador.

Solucion 1: Mediante diagramas de transición



Variables
- ***Estado***
- ***símbolo***

```
BEGIN
  Estado:= 1;
  símbolo := primer simbolo de la cadena de entrada
  WHILE símbolo no es fin de cadena DO
     CASE Estado OF
        1: IF símbolo es un digito THEN Estado:= 2
           ELSE IF símbolo es una letra THEN Estado:=3
           ELSE salir a rutina de error
        2: Salir a rutina de error
        3: IF símbolo es letra THEN Estado := 3
           ELSE IF símbolo es digito THEN Estado:=3
           ELSE salir a rutina de error
     ENDCASE
     Leer siguiente simbolo de la cadena
  END WHILE
  IF Estado <> 3 THEN
       Salir a rutina de error
END
```

```cpp
////////////////////////////////////////////////////////////////
//                          EJERCICIO #1                      //
////////////////////////////////////////////////////////////////

#include <iostream>
#include <string>
using namespace std;

bool checkDiagram( string word ) {
    int state = 1;
    for( int i = 0; i < word.size(); ++i ) {
        char symbol = tolower( word[ i ]);
        switch( state ) {
            case 1:
                if( symbol >= 48 && symbol <= 57 )
                    state = 2;
                else if( symbol >= 97 && symbol <= 122 )
                    state = 3;
                else
                    state = 2;
            case 2:
                break;
            case 3:
                if( symbol >= 97 && symbol <= 122 )
                    state = 3;
                else if( symbol >= 48 && symbol <= 57 )
                    state = 3;
                else
                    state = 2;
            default:
                break;
        }
    }

    if( state != 3 )
        return false;
    else
        return true;
}

int main() {
    string word;
```

```
43      printf( "Ingrese la cadena a ser evaluada:\n" );
44      cin >> word;
45      if( checkDiagram( word ))
46          printf( "La cadena es reconocida por el automata.\n" );
47      else
48          printf( "La cadena NO es reconocida por el automata.\n" )
                ;
49      return 0;
50  }
```

```
pimed@horo:~/Documents/ADA$ g++ exercise1.cpp
pimed@horo:~/Documents/ADA$ ./a.out
Ingrese la cadena a ser evaluada:
hola
La cadena es reconocida por el automata.
pimed@horo:~/Documents/ADA$ ./a.out
Ingrese la cadena a ser evaluada:
123
La cadena NO es reconocida por el automata.
pimed@horo:~/Documents/ADA$ ./a.out
Ingrese la cadena a ser evaluada:
Hola123
La cadena es reconocida por el automata.
pimed@horo:~/Documents/ADA$ ./a.out
Ingrese la cadena a ser evaluada:
12Hola
La cadena NO es reconocida por el automata.
pimed@horo:~/Documents/ADA$ ./a.out
Ingrese la cadena a ser evaluada:
+
La cadena NO es reconocida por el automata.
```

2.2. Implemente, el algoritmo por tablas de transición para el reconocimiento de un identificador.

Solucion 2: Mediante Tablas de transición

	digito	letra	FDC
1	2	3	'error'
2	'error'	'error'	'error'
3	3	3	'aceptar'

Donde:

Tabla[Estado,Entrada] es el estado actual proveniente del estado anterior leyendo la entrada.

```
BEGIN
  Estado:=1
  REPEAT
     Leer siguiente simbolo de la cadena de entrada
     CASE simbolo OF
          Digito: Entrada:= 'digito'
          Letra : Entrada:= 'letra'
          Fin de cadena : Entrada := 'FDC'
          Otro simbolo : Salir a rutina de error.
     ENDCASE
     Estado:= Tabla[Estado, Entrada]
     IF Estado:='error'
          Salir a rutina de error
  UNTIL Estado = 'aceptar'
END
```

```
1  ////////////////////////////////////////////////////////////////
2  //                          EJERCICIO #2                      //
3  ////////////////////////////////////////////////////////////////
4
5  /*
6      Para este ejercicio lo primero que se procede a
7      hacer es crear una matriz para que actue como nues_
8      tra tabla de transiciones; una vez hecho esto se
9      procedera a convertir la tabla a una tabla de ente_
10     ros ya que de esta manera sera mas facil trabajar.
11
12     La tabla se convirtio de la siguiente manera:
```

```
13      - Los numeros simplemente se transformaron de string
14        a int.
15      - A los valores "error" y "aceptar" se le dio el va_
16        lor de 3 y 4 respectivamente, ya que esos valores
17        son ajenos al contenido de nuestra tabla y podria
18        evitar errores futuros.
19
20      Una vez hecho esto simplemente se realiza el algorit_
21      mo propuesto en pseudocodigo.
22  */
23
24  #include <iostream>
25  using namespace std;
26
27  bool checkTable( int table[ 3 ][ 3 ], string word ) {
28      int state = 0, input = 0;
29      string pEntrada;
30      char character;
31      enum Controller{ digito, letra, FDC, other };
32      Controller symbol;
33
34      for( int i = 0; i < word.size() + 1; ++i ) {
35          cout << "CARACTER " << word[ i ] << endl;
36          character = tolower( word[ i ]);
37
38          if( character >= 48 && character <= 57 )
39              symbol = digito;
40          else if( character >= 97 && character <= 122 )
41              symbol = letra;
42          else if( character == *( word.end()))
43              symbol = FDC;
44          else
45              symbol = other;
46
47          switch( symbol ) {
48              case digito:
49                  input = 0;
50                  pEntrada = "Digito";
51                  break;
52              case letra:
53                  input = 1;
54                  pEntrada = "Letra";
55                  break;
```

```
56              case FDC:
57                  input = 2;
58                  pEntrada = "FDC";
59                  break;
60              case other:
61                  input = 3;
62                  pEntrada = "Other";
63                  break;
64              default:
65                  break;
66          }
67          cout << "Estado = " << state + 1 << endl;
68          cout << "Entrada = " << pEntrada << endl;
69          state = table[ state ][ input ];
70
71          if( state == 3 || input == 3 )
72              return false;
73          else if( state == 4 )
74              return true;
75          cout << endl;
76      }
77      return false;
78  }
79
80  int main() {
81      string tableBase[ 3 ][ 3 ];
82      string word;
83
84      int tableTrans[ 3 ][ 3 ];
85      int num;
86
87      char control = 'Y';
88
89      // Llenar la tabla:
90      tableBase[ 0 ][ 0 ] = "2";
91      tableBase[ 0 ][ 1 ] = "3";
92      tableBase[ 0 ][ 2 ] = "error";
93      tableBase[ 1 ][ 0 ] = "error";
94      tableBase[ 1 ][ 1 ] = "error";
95      tableBase[ 1 ][ 2 ] = "error";
96      tableBase[ 2 ][ 0 ] = "3";
97      tableBase[ 2 ][ 1 ] = "3";
98      tableBase[ 2 ][ 2 ] = "aceptar";
```

```
99
100     // Imprimir la tabla:
101     cout << "TABLA DE TRANSICIONES" << endl;
102     cout << "-------------------------------------------------"
           << endl;
103     cout << "   |   Digito   |     Letra     |      FDC      |"
           << endl;
104     cout << "-------------------------------------------------"
           << endl;
105     for( int i = 0; i < 3; ++i ) {
106         cout << i + 1 << "  |\t" << "  ";
107         for( int j = 0; j < 3; ++j )
108             cout << tableBase[ i ][ j ] << "\t|\t";
109         cout << endl;
110     }
111     cout << "-------------------------------------------------"
           << endl;
112     cout << endl;
113
114     for( int i = 0; i < 3; ++i ) {
115         for( int j = 0; j < 3; ++j ) {
116             if( tableBase[ i ][ j ] == "error" ) {
117                 num = 3;
118             } else if( tableBase[ i ][ j ] == "aceptar" ) {
119                 num = 4;
120             } else
121                 num = stoi( tableBase[ i ][ j ]) - 1;
122             tableTrans[ i ][ j ] = num;
123         }
124     }
125
126     // Imprimir la tabla:
127     cout << "\nTABLA DE TRANSICIONES" << endl;
128     cout << "-------------------------------------------------"
           << endl;
129     cout << "   |   Digito   |     Letra     |      FDC      |"
           << endl;
130     cout << "-------------------------------------------------"
           << endl;
131     for( int i = 0; i < 3; ++i ) {
132         cout << i << "  |\t" << "  ";
133         for( int j = 0; j < 3; ++j )
134             cout << tableTrans[ i ][ j ] << "\t|\t";
```

```
135         cout << endl;
136     }
137     cout << "-------------------------------------------------"
           << endl;
138     cout << endl;
139
140     cout << "Ingrese la cadena a ser evaluada: " << endl;
141     cin >> word;
142     cout << endl;
143
144     while( control == 'Y' ) {
145         if( checkTable( tableTrans, word ))
146             cout << "\nLa cadena es reconocida por el automata."
                   << endl;
147         else
148             cout << "\nLa cadena NO es reconocida por el automata
                   ." << endl;
149
150         cout << "Desea evaluar otra cadena Y/N..." << endl;
151         cin >> control;
152
153         while( control != 'Y' && control != 'N' ) {
154             cout << "Opcion invalida..." << endl;
155             cout << "Desea evaluar otra cadena Y/N..." << endl;
156             cin >> control;
157             if( control == 'Y' || control == 'N' )
158                 break;
159         }
160
161         if( control == 'N' )
162             break;
163
164         cout << "\nIngrese la cadena a ser evaluada: " << endl;
165         cin >> word;
166         cout << endl;
167     }
168
169     return 0;
170 }
```

```
pimed@horo:~/Documents/ADA$ g++ exercise2.cpp
pimed@horo:~/Documents/ADA$ ./a.out
TABLA DE TRANSICIONES
-------------------------------------------------
   |   Digito   |     Letra     |      FDC      |
-------------------------------------------------
1  |      2     |       3       |       error   |
2  |      error |       error   |       error   |
3  |      3     |       3       |       aceptar |
-------------------------------------------------


TABLA DE TRANSICIONES
-------------------------------------------------
   |   Digito   |     Letra     |      FDC      |
-------------------------------------------------
0  |      1     |       2       |       3       |
1  |      3     |       3       |       3       |
2  |      2     |       2       |       4       |
-------------------------------------------------

Ingrese la cadena a ser evaluada:
Hola

CARACTER H
Estado = 1
Entrada = Letra

CARACTER o
Estado = 3
Entrada = Letra
```

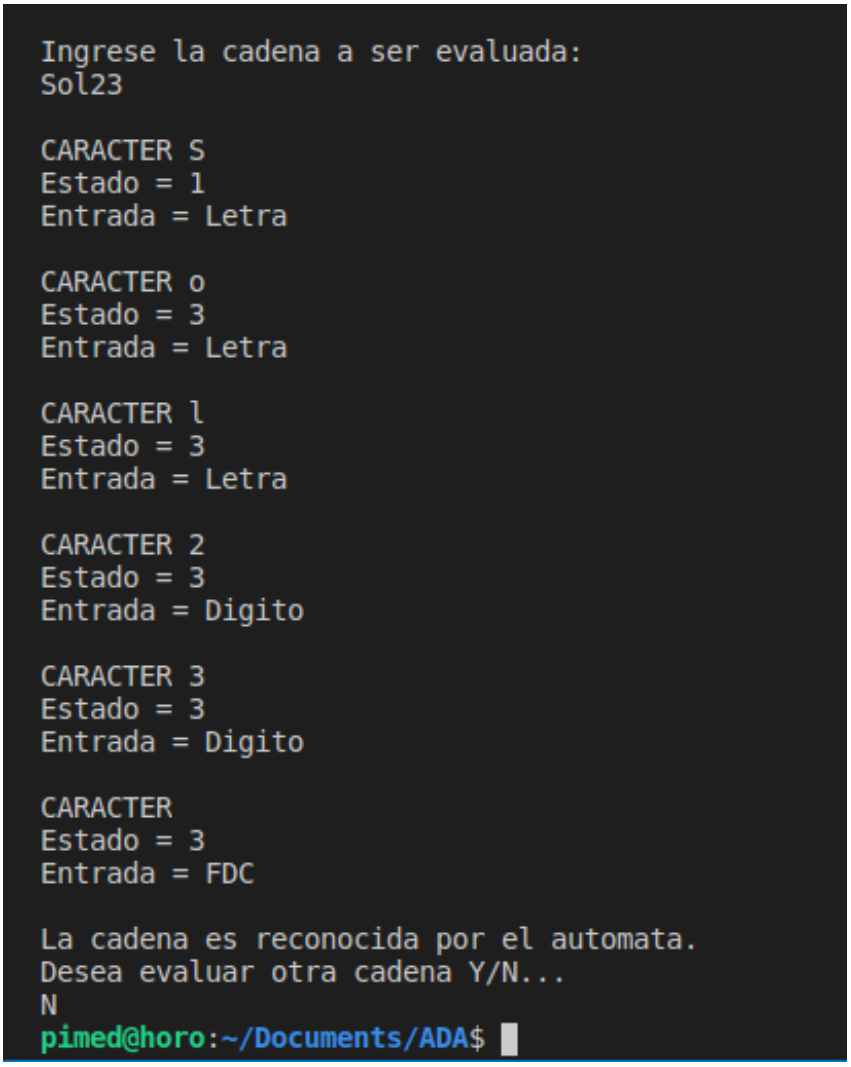
```
CARACTER l
Estado = 3
Entrada = Letra

CARACTER a
Estado = 3
Entrada = Letra

CARACTER
Estado = 3
Entrada = FDC

La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
N
```

```
pimed@horo:~/Documents/ADA$ g++ exercise2.cpp
pimed@horo:~/Documents/ADA$ ./a.out
TABLA DE TRANSICIONES
-------------------------------------------------
   |   Digito   |     Letra     |      FDC      |
-------------------------------------------------
1  |      2     |       3       |       error   |
2  |      error |       error   |       error   |
3  |      3     |       3       |       aceptar |
-------------------------------------------------


TABLA DE TRANSICIONES
-------------------------------------------------
   |   Digito   |     Letra     |      FDC      |
-------------------------------------------------
0  |      1     |       2       |       3       |
1  |      3     |       3       |       3       |
2  |      2     |       2       |       4       |
-------------------------------------------------

Ingrese la cadena a ser evaluada:
Hola

CARACTER H
Estado = 1
Entrada = Letra

CARACTER o
Estado = 3
Entrada = Letra
```

```
CARACTER l
Estado = 3
Entrada = Letra

CARACTER a
Estado = 3
Entrada = Letra

CARACTER
Estado = 3
Entrada = FDC

La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
N
```

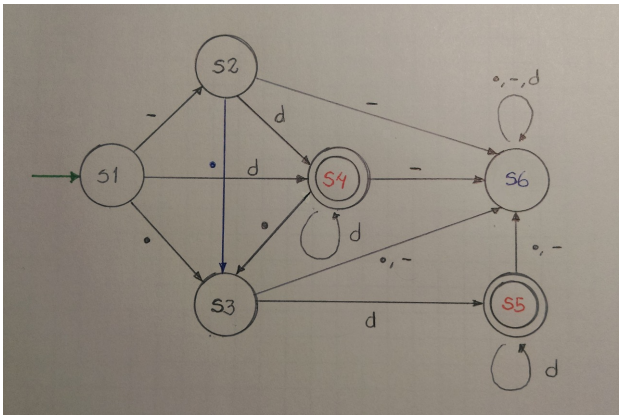
```
pimed@horo:~/Documents/ADA$ g++ exercise2.cpp
pimed@horo:~/Documents/ADA$ ./a.out
TABLA DE TRANSICIONES
-------------------------------------------------
   |   Digito   |     Letra     |      FDC      |
-------------------------------------------------
1  |      2     |       3       |       error   |
2  |      error |       error   |       error   |
3  |      3     |       3       |       aceptar |
-------------------------------------------------


TABLA DE TRANSICIONES
-------------------------------------------------
   |   Digito   |     Letra     |      FDC      |
-------------------------------------------------
0  |      1     |       2       |       3       |
1  |      3     |       3       |       3       |
2  |      2     |       2       |       4       |
-------------------------------------------------

Ingrese la cadena a ser evaluada:
123

CARACTER 1
Estado = 1
Entrada = Digito

CARACTER 2
Estado = 2
Entrada = Digito

La cadena NO es reconocida por el automata.
Desea evaluar otra cadena Y/N...
Y
```

```
Ingrese la cadena a ser evaluada:
+Hola

CARACTER +
Estado = 1
Entrada = Other

La cadena NO es reconocida por el automata.
Desea evaluar otra cadena Y/N...
Y
```


Ingrese la cadena a ser evaluada:
Sol23
CARACTER S
Estado = 1
Entrada = Letra
CARACTER o
Estado = 3
Entrada = Letra
CARACTER l
Estado = 3
Entrada = Letra
CARACTER 2
Estado = 3
Entrada = Digito
CARACTER 3
Estado = 3
Entrada = Digito
CARACTER
Estado = 3
Entrada = FDC
La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
N
pimed@horo:~/Documents/ADA$

2.3. Proponga un diagrama de transición y su correspondiente programa para el reconocimiento de números (Ejemplo: 45,-45, 0.15,-0.15,.67,-.67)

Para realizar el ejercicio primero se realizara el diagrama del automata como AFND, luego se transformara a AFD y se minimizara hallando el automata optimo para nuestro problema.



Estados	Punto	Menos	Digito	FDC
1	3	2	4	error
2	3	error	4	error
3	error	error	5	error
4	3	error	4	aceptar
5	error	error	5	aceptar
6	error	error	error	error

```
1  ////////////////////////////////////////////////////////////////
2  //                          EJERCICIO #3                      //
3  ////////////////////////////////////////////////////////////////
4
5  /*
6      Teniendo en cuenta el ejercicio 2, lo usaremos de plan_
7      tilla para la solucion del ejercicio 3. Basta con cambiar
8      el tamano de las matrices y cambiar unas cuantas condicio_
9      nes y esta hecho.
10 */
```

```
11
12
13  #include <iostream>
14  using namespace std;
15
16  bool checkTable( int table[ 6 ][ 4 ], string word ) {
17      int state = 0, input = 0;
18      string pEntrada;
19      char character;
20      enum Controller{ punto, menos, digito, FDC, other };
21      Controller symbol;
22
23      for( int i = 0; i < word.size() + 1; ++i ) {
24          cout << "CARACTER " << word[ i ] << endl;
25          character = tolower( word[ i ]);
26
27          if( character == '.' )
28              symbol = punto;
29          else if( character == '-' )
30              symbol = menos;
31          else if( character >= 48 && character <= 57 )
32              symbol = digito;
33          else if( character == *( word.end()))
34              symbol = FDC;
35          else
36              symbol = other;
37
38          switch( symbol ) {
39              case punto:
40                  input = 0;
41                  pEntrada = "Punto";
42                  break;
43              case menos:
44                  input = 1;
45                  pEntrada = "Menos";
46                  break;
47              case digito:
48                  input = 2;
49                  pEntrada = "Digito";
50                  break;
51              case FDC:
52                  input = 3;
53                  pEntrada = "FDC";
```

```
                break;
            case other:
                input = 4;
                pEntrada = "Other";
                break;
            default:
                break;
        }
        cout << "Estado = " << state + 1 << endl;
        cout << "Entrada = " << pEntrada << endl;
        state = table[ state ][ input ];

        if( state == 6 || input == 4 )
            return false;
        else if( state == 7 )
            return true;
        cout << endl;
    }
    return false;
}

int main() {
    string tableBase[ 6 ][ 4 ];
    string word;

    int tableTrans[ 6 ][ 4 ];
    int num;

    char control = 'Y';

    // Llenar la tabla:
    tableBase[ 0 ][ 0 ] = "3";
    tableBase[ 0 ][ 1 ] = "2";
    tableBase[ 0 ][ 2 ] = "4";
    tableBase[ 0 ][ 3 ] = "error";
    tableBase[ 1 ][ 0 ] = "3";
    tableBase[ 1 ][ 1 ] = "error";
    tableBase[ 1 ][ 2 ] = "4";
    tableBase[ 1 ][ 3 ] = "error";
    tableBase[ 2 ][ 0 ] = "error";
    tableBase[ 2 ][ 1 ] = "error";
    tableBase[ 2 ][ 2 ] = "5";
    tableBase[ 2 ][ 3 ] = "error";
```

```
97      tableBase[ 3 ][ 0 ] = "3";
98      tableBase[ 3 ][ 1 ] = "error";
99      tableBase[ 3 ][ 2 ] = "4";
100     tableBase[ 3 ][ 3 ] = "aceptar";
101     tableBase[ 4 ][ 0 ] = "error";
102     tableBase[ 4 ][ 1 ] = "error";
103     tableBase[ 4 ][ 2 ] = "5";
104     tableBase[ 4 ][ 3 ] = "aceptar";
105     tableBase[ 5 ][ 0 ] = "error";
106     tableBase[ 5 ][ 1 ] = "error";
107     tableBase[ 5 ][ 2 ] = "error";
108     tableBase[ 5 ][ 3 ] = "error";
109
110     // Imprimir la tabla:
111     cout << "TABLA DE TRANSICIONES" << endl;
112     cout << "-----------------------------------------" <<
           endl;
113     cout << "   |   Punto    |     Menos     |    Digito     |
               FDC      |" << endl;
114     cout << "-----------------------------------------" <<
           endl;
115     for( int i = 0; i < 6; ++i ) {
116         cout << i + 1 << "  |\t" << "  ";
117         for( int j = 0; j < 4; ++j )
118             cout << tableBase[ i ][ j ] << "\t|\t";
119         cout << endl;
120     }
121     cout << "-----------------------------------------" <<
           endl;
122     cout << endl;
123
124     for( int i = 0; i < 6; ++i ) {
125         for( int j = 0; j < 4; ++j ) {
126             if( tableBase[ i ][ j ] == "error" ) {
127                 num = 6;
128             } else if( tableBase[ i ][ j ] == "aceptar" ) {
129                 num = 7;
130             } else
131                 num = stoi( tableBase[ i ][ j ]) - 1;
132             tableTrans[ i ][ j ] = num;
133         }
134     }
135
```

```
136     // Imprimir la tabla:
137     cout << "\nTABLA DE TRANSICIONES" << endl;
138     cout << "-------------------------------------------" <<
           endl;
139     cout << "   |   Punto    |     Menos     |    Digito     |
               FDC      |" << endl;
140     cout << "-------------------------------------------" <<
           endl;
141     for( int i = 0; i < 6; ++i ) {
142         cout << i << "  |\t" << "  ";
143         for( int j = 0; j < 4; ++j )
144             cout << tableTrans[ i ][ j ] << "\t|\t";
145         cout << endl;
146     }
147     cout << "-------------------------------------------" <<
           endl;
148     cout << endl;
149
150     cout << "Ingrese la cadena a ser evaluada: " << endl;
151     cin >> word;
152     cout << endl;
153
154     while( control == 'Y' ) {
155         if( checkTable( tableTrans, word ))
156             cout << "\nLa cadena es reconocida por el automata."
                   << endl;
157         else
158             cout << "\nLa cadena NO es reconocida por el automata
                   ." << endl;
159
160         cout << "Desea evaluar otra cadena Y/N..." << endl;
161         cin >> control;
162
163         while( control != 'Y' && control != 'N' ) {
164             cout << "Opcion invalida..." << endl;
165             cout << "Desea evaluar otra cadena Y/N..." << endl;
166             cin >> control;
167             if( control == 'Y' || control == 'N' )
168                 break;
169         }
170
171         if( control == 'N' )
172             break;
```

```
173
174         cout << "\nIngrese la cadena a ser evaluada: " << endl;
175         cin >> word;
176         cout << endl;
177     }
178
179     return 0;
180 }
```

```
pimed@horo:~/Documents/ADA$ g++ exercise3.cpp
pimed@horo:~/Documents/ADA$ ./a.out
TABLA DE TRANSICIONES
-------------------------------------------------------------------
   |   Punto    |     Menos     |    Digito     |      FDC      |
-------------------------------------------------------------------
1  |      3     |       2       |       4       |       error   |
2  |      3     |       error   |       4       |       error   |
3  |      error |       error   |       5       |       error   |
4  |      3     |       error   |       4       |       aceptar |
5  |      error |       error   |       5       |       aceptar |
6  |      error |       error   |       error   |       error   |
-------------------------------------------------------------------


TABLA DE TRANSICIONES
-------------------------------------------------------------------
   |   Punto    |     Menos     |    Digito     |      FDC      |
-------------------------------------------------------------------
0  |      2     |       1       |       3       |       6       |
1  |      2     |       6       |       3       |       6       |
2  |      6     |       6       |       4       |       6       |
3  |      2     |       6       |       3       |       7       |
4  |      6     |       6       |       4       |       7       |
5  |      6     |       6       |       6       |       6       |
-------------------------------------------------------------------

Ingrese la cadena a ser evaluada:
-..065
```

```
Ingrese la cadena a ser evaluada:
-..065

CARACTER -
Estado = 1
Entrada = Menos

CARACTER .
Estado = 2
Entrada = Punto

CARACTER .
Estado = 3
Entrada = Punto

La cadena NO es reconocida por el automata.
Desea evaluar otra cadena Y/N...
Y
```

```
Ingrese la cadena a ser evaluada:
45

CARACTER 4
Estado = 1
Entrada = Digito

CARACTER 5
Estado = 4
Entrada = Digito

CARACTER
Estado = 4
Entrada = FDC

La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
Y

Ingrese la cadena a ser evaluada:
-45

CARACTER -
Estado = 1
Entrada = Menos

CARACTER 4
Estado = 2
Entrada = Digito
```

```
Ingrese la cadena a ser evaluada:
-45

CARACTER -
Estado = 1
Entrada = Menos

CARACTER 4
Estado = 2
Entrada = Digito

CARACTER 5
Estado = 4
Entrada = Digito

CARACTER
Estado = 4
Entrada = FDC

La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
Y

Ingrese la cadena a ser evaluada:
0.15

CARACTER 0
Estado = 1
Entrada = Digito

CARACTER .
Estado = 4
Entrada = Punto
```

```
CARACTER 1
Estado = 3
Entrada = Digito

CARACTER 5
Estado = 5
Entrada = Digito

CARACTER
Estado = 5
Entrada = FDC
```

```
La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
Y

Ingrese la cadena a ser evaluada:
-0.15

CARACTER -
Estado = 1
Entrada = Menos

CARACTER 0
Estado = 2
Entrada = Digito

CARACTER .
Estado = 4
Entrada = Punto

CARACTER 1
Estado = 3
Entrada = Digito

CARACTER 5
Estado = 5
Entrada = Digito

CARACTER
Estado = 5
Entrada = FDC

La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
Y
```

```
Ingrese la cadena a ser evaluada:
.67

CARACTER .
Estado = 1
Entrada = Punto

CARACTER 6
Estado = 3
Entrada = Digito
```

```
CARACTER 7
Estado = 5
Entrada = Digito

CARACTER
Estado = 5
Entrada = FDC

La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
Y

Ingrese la cadena a ser evaluada:
-.67

CARACTER -
Estado = 1
Entrada = Menos

CARACTER .
Estado = 2
Entrada = Punto

CARACTER 6
Estado = 3
Entrada = Digito
```

```
CARACTER 7
Estado = 5
Entrada = Digito

CARACTER
Estado = 5
Entrada = FDC

La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
N
```

2.4. La solución 2, puede ser automatizada, teniendo como dato de entrada la tabla de transiciones. Defina el formato de la tabla de transición y en un archivo de texto plano represente la respectiva tabla (considere el diagrama del ejercicio C). Finalmente, en base al archivo de texto plano, determine la validez de una entrada.

```
1  ////////////////////////////////////////////////////////////////
2  //                          EJERCICIO #4                      //
3  ////////////////////////////////////////////////////////////////
4
5  /*
6      Habiendo realizado el ejercicio 3, realizaremos algunas
7      variaciones, para comenzar quitaremos el ingreso de datos
8      manual y agregaremos un lector de ficheros, almacenaremos
9      el contenido del fichero en un vector y con este mismo
10     llenaremos nuestra tabla de transiciones, permitiendo
11     reutilizar nuestro codigo del ejercicio anterior.
12 */
13
14 #include <iostream>
15 #include <fstream>
16 #include <vector>
17 using namespace std;
18
19 bool checkTable( int table[ 6 ][ 4 ], string word ) {
20     int state = 0, input = 0;
21     string pEntrada;
22     char character;
23     enum Controller{ punto, menos, digito, FDC, other };
24     Controller symbol;
25
26     for( int i = 0; i < word.size() + 1; ++i ) {
27         cout << "CARACTER " << word[ i ] << endl;
28         character = tolower( word[ i ]);
29
30         if( character == '.' )
31             symbol = punto;
32         else if( character == '-' )
33             symbol = menos;
34         else if( character >= 48 && character <= 57 )
```

```
35              symbol = digito;
36          else if( character == *( word.end()))
37              symbol = FDC;
38          else
39              symbol = other;
40
41          switch( symbol ) {
42              case punto:
43                  input = 0;
44                  pEntrada = "Punto";
45                  break;
46              case menos:
47                  input = 1;
48                  pEntrada = "Menos";
49                  break;
50              case digito:
51                  input = 2;
52                  pEntrada = "Digito";
53                  break;
54              case FDC:
55                  input = 3;
56                  pEntrada = "FDC";
57                  break;
58              case other:
59                  input = 4;
60                  pEntrada = "Other";
61                  break;
62              default:
63                  break;
64          }
65          cout << "Estado = " << state + 1 << endl;
66          cout << "Entrada = " << pEntrada << endl;
67          state = table[ state ][ input ];
68
69          if( state == 6 || input == 4 )
70              return false;
71          else if( state == 7 )
72              return true;
73          cout << endl;
74      }
75      return false;
76  }
77
```

```
78  int main() {
79      string tableBase[ 6 ][ 4 ];
80      string word;
81      vector<string> temp;
82
83      int tableTrans[ 6 ][ 4 ];
84      int num, k = 0;
85
86      char control = 'Y';
87
88      ifstream archive;
89      archive.open( "table.txt", ios::in );
90
91      if( archive.fail()) {
92          cout << "Incapaz de abrir el archivo..." << endl;
93          exit( EXIT_FAILURE );
94      }
95
96      string cadena;
97      while( !archive.eof()) {
98          archive >> cadena;
99          temp.push_back( cadena );
100     }
101
102     cout << "El fichero fue cargado con exito..." << endl;
103
104     archive.close();
105
106     for( int i = 0; i < 6; ++i ) {
107         for( int j = 0; j < 4; ++j ) {
108             tableBase[ i ][ j ] = temp[ k ];
109             ++k;
110         }
111     }
112
113     // Imprimir la tabla:
114     cout << "TABLA DE TRANSICIONES" << endl;
115     cout <<
           "-------------------------------------------------------------------"
            << endl;
116     cout << "   |   Punto    |     Menos     |    Digito     |
                FDC      |" << endl;
```

```
117     cout <<
           "-------------------------------------------------------------------"
            << endl;
118     for( int i = 0; i < 6; ++i ) {
119         cout << i + 1 << "  |\t" << "  ";
120         for( int j = 0; j < 4; ++j )
121             cout << tableBase[ i ][ j ] << "\t|\t";
122         cout << endl;
123     }
124     cout <<
           "-------------------------------------------------------------------"
            << endl;
125     cout << endl;
126
127     for( int i = 0; i < 6; ++i ) {
128         for( int j = 0; j < 4; ++j ) {
129             if( tableBase[ i ][ j ] == "error" ) {
130                 num = 6;
131             } else if( tableBase[ i ][ j ] == "aceptar" ) {
132                 num = 7;
133             } else
134                 num = stoi( tableBase[ i ][ j ]) - 1;
135             tableTrans[ i ][ j ] = num;
136         }
137     }
138
139     // Imprimir la tabla:
140     cout << "\nTABLA DE TRANSICIONES" << endl;
141     cout <<
           "-------------------------------------------------------------------"
            << endl;
142     cout << "   |   Punto    |     Menos     |    Digito     |
               FDC      |" << endl;
143     cout <<
           "-------------------------------------------------------------------"
            << endl;
144     for( int i = 0; i < 6; ++i ) {
145         cout << i << "  |\t" << "  ";
146         for( int j = 0; j < 4; ++j )
147             cout << tableTrans[ i ][ j ] << "\t|\t";
148         cout << endl;
149     }
```

```
150     cout <<
           "-----------------------------------------------------------------"
            << endl;
151     cout << endl;
152
153     cout << "Ingrese la cadena a ser evaluada: " << endl;
154     cin >> word;
155     cout << endl;
156
157     while( control == 'Y' ) {
158         if( checkTable( tableTrans, word ))
159             cout << "\nLa cadena es reconocida por el automata."
                   << endl;
160         else
161             cout << "\nLa cadena NO es reconocida por el automata
                   ." << endl;
162
163         cout << "Desea evaluar otra cadena Y/N..." << endl;
164         cin >> control;
165
166         while( control != 'Y' && control != 'N' ) {
167             cout << "Opcion invalida..." << endl;
168             cout << "Desea evaluar otra cadena Y/N..." << endl;
169             cin >> control;
170             if( control == 'Y' || control == 'N' )
171                 break;
172         }
173
174         if( control == 'N' )
175             break;
176
177         cout << "\nIngrese la cadena a ser evaluada: " << endl;
178         cin >> word;
179         cout << endl;
180     }
181
182     return 0;
183 }
```

```
pimed@horo:~/Documents/ADA$ g++ exercise4.cpp
pimed@horo:~/Documents/ADA$ ./a.out
El fichero fue cargado con exito...
TABLA DE TRANSICIONES
-----------------------------------------------------------------
   |   Punto    |     Menos     |    Digito     |      FDC      |
-----------------------------------------------------------------
1  |      3     |       2       |       4       |       error   |
2  |      3     |       error   |       4       |       error   |
3  |      error |       error   |       5       |       error   |
4  |      3     |       error   |       4       |       aceptar |
5  |      error |       error   |       5       |       aceptar |
6  |      error |       error   |       error   |       error   |
-----------------------------------------------------------------


TABLA DE TRANSICIONES
-----------------------------------------------------------------
   |   Punto    |     Menos     |    Digito     |      FDC      |
-----------------------------------------------------------------
0  |      2     |       1       |       3       |       6       |
1  |      2     |       6       |       3       |       6       |
2  |      6     |       6       |       4       |       6       |
3  |      2     |       6       |       3       |       7       |
4  |      6     |       6       |       4       |       7       |
5  |      6     |       6       |       6       |       6       |
-----------------------------------------------------------------

Ingrese la cadena a ser evaluada:
0.15

CARACTER 0
Estado = 1
Entrada = Digito

CARACTER .
Estado = 4
Entrada = Punto

CARACTER 1
Estado = 3
Entrada = Digito

CARACTER 5
Estado = 5
Entrada = Digito

CARACTER
Estado = 5
Entrada = FDC

La cadena es reconocida por el automata.
Desea evaluar otra cadena Y/N...
N
```