

## UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA Facultad de Ingeniería



Ingeniería en Ciencias de la Computación

## TEORÍA DE LA COMPUTACIÓN **Programa AFD**

Trabajo de: ADRIAN A. GONZÁLEZ DOMÍNGUEZ [359834] Asesor: MARIO ANDRES CUEVAS GUTIERREZ

leoria de la Computación Lunes 16/Septiembre/1024 Programa AFD Adrian Conzález Domínguez #include <stdio.h> int automaton (void); int choose\_column (int alphabet\_length, char alphabet [], char c); int is\_final(int finals\_length, int finals I] int state); char result [100] int result\_index; HILE \* input Sile; int man() ( I I takes) along this inpat file stding while (! feof (input\_Sile)) { result\_index=0; 0= state In if (automoton ()) { result Tresult\_index='\0'; printf ("Cadena válida: /.s/n", resalt); Jelse { 3 prints ("Cadena invalida \n"); result [restalt unlex++]=c return O; Sinds\_length, Sinds, states seturn is finally state trades state

// Automota A // Automata B int automation (void) { int automaton(void) { Int states [][3]={ int states [][3]={ 10, 1, 33, 110 {2,1,4},10 20,2,33,1/1 11, 1, 43, 111 £2,2,33,1/2 {2,3,4},112 £3,3,37, 1/3 {2,3,43,113 {4, 4, 43 114 int finals []={2}; int finals length =1; int finals []={1,3}; char alphabet []-{a', b'}; int finals\_length= 2; 10 mom In int alphabet length=2; char alphabet[]={a, b'}; int state = D; int alphabet\_length=2, char ci int state = 0; De xibing Alicen whilel char cg ) (Onotomodus) } ! feof (input\_f, le) & & Cc= getc(input\_f, le)) != 'In' while (1 = when Albertal) was Seof (input\_file) & & (c=getc(inpat\_file)) = 'In' result [result\_index ++]=c; int column=choose\_column[ result\_index++]=c; alphabet\_length alphabet, e int column=choose\_column ( alphabet\_length, alphabet, c state = states [state][column]; state=states[state][column]; return 15\_final (
finals\_length, finals, state
). return is\_finall
finals\_length, finals, state

```
int choose_column (int alphabet_length, char alphabet[], char c){
  int i=0;
  while (i < alphabet_length) {
  if (alphabet[i]==c) return i;
return i;
int is final (int finals_length, int finals[], int state) {
  inf i=0;
  whileli finals_ length){
    if(finals[i]==state) return 1;
   return O;
```

## Para el automata A

```
int automaton(void)
{
    int states[][3] = {
        {0, 1, 3}, // 0
        {0, 2, 3}, // 1
        {2, 2, 3}, // 2
        {3, 3, 3}, // 3
    };
    int finals[] = {2};
    int finals_length = 1;
```

```
MINGW64:/c/Users/thead/Do ×
thead@adrigondo MINGW64 ~/Documents/UACH/Seventh Semester/Theory Of Computation/code $ gcc TOC.September_10_2024.c -o TOC.September_10_2024 && ./TOC.September_10_2024 bb
Cadena valida bb
abb
Cadena valida abb
Cadena valida bbb
Cadena invalida
aaab
Cadena invalida
aaaaabbaaaaaaa
Cadena valida aaaaabbaaaaaa
bababababa
Cadena invalida
babababba
Cadena valida babababba
```

## Para el automata B

```
int automaton(void)
{
    int states[][3] = {
        {2, 1, 4}, // 0
        {1, 1, 4}, // 1
        {2, 3, 4}, // 2
        {2, 3, 4}, // 3
        {4, 4, 4}, // 4
    };
    int finals[] = {1,3};
    int finals_length = 2;
```

```
×
 MINGW64:/c/Users/thead/Do X
thead@adrigondo MINGW64 ~/Documents/UACH/Seventh Semester/Theory Of Computation/code
$ gcc TOC.September_10_2024.c -o TOC.September_10_2024 && ./TOC.September_10_2024
Cadena valida b
ab
Cadena valida ab
Cadena valida ba
aaaaa
Cadena invalida
aaaba
Cadena invalida
ababab
Cadena valida ababab
bab
Cadena valida bab
baaaaaa
Cadena valida baaaaaa
```