

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

**MAESTRÍA EN CIENCIAS DE LA ADMINISTRACIÓN
CON ESPECIALIDAD EN SISTEMAS**

TRABAJO FINAL

MÁQUINAS DE TURING

ELABORADO POR :

EQUIPO 5

**BRUNO LÓPEZ TAKEYAS
JAIME DAVID JOHNSTON B.
JAVIER CASTAÑEDA AMBRIZ
MA. GUADALUPE MENDOZA GARCÍA
MIGUEL GUERRERO CRESPO
SERGIO GARZA CARRANZA**

FEBRERO DE 1996

CONTENIDO

Tema	Página
INTRODUCCIÓN	3
REQUERIMIENTOS	6
OBJETIVO	7
DESCRIPCIÓN DEL PROGRAMA	8
LISTADO DEL PROGRAMA	10
CONCLUSIONES	22
BIBLIOGRAFÍA	23

INTRODUCCIÓN

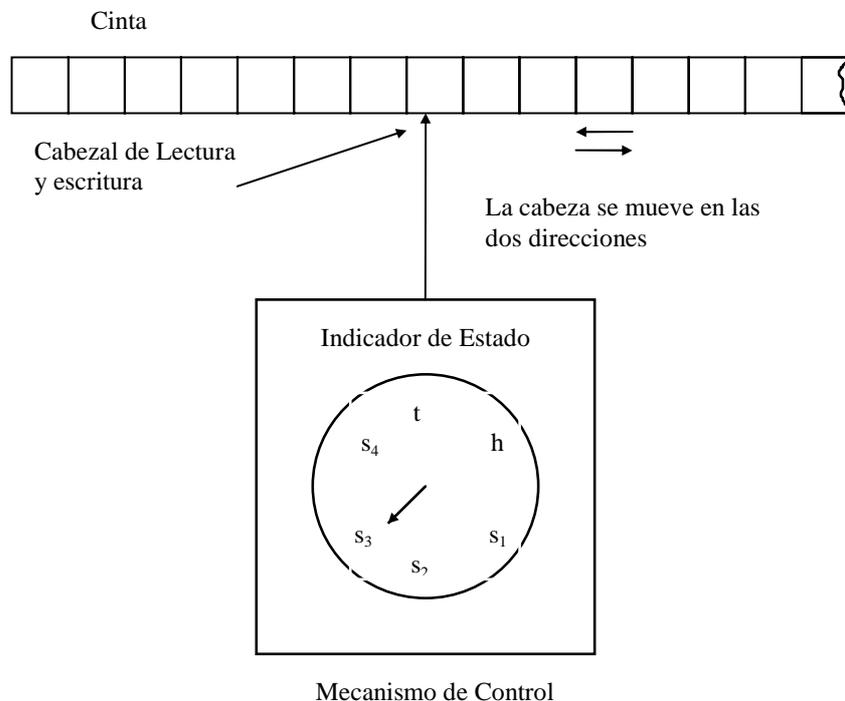
La clase de autómatas que conocemos como *máquinas de Turing* fué propuesta por Alan M. Turing en 1936. La idea básica de Turing fué estudiar los procesos algorítmicos utilizando un modelo computacional

El propósito de las máquinas de Turing fué desarrollar un sistema en el cual fuera posible modelar cualquier proceso que pudiera considerarse como un cálculo.

Propiedades básicas

- Tienen un mecanismo de control que en cualquier momento puede estar en uno de entre un número finito de estados.
- Realiza operaciones de movimiento, que consisten en mover el cabezal de cinta una celda a la derecha ó una celda a la izquierda y pasar a un nuevo estado.
- Realiza operaciones de escritura, que consisten en reemplazar un símbolo en la cinta con otro símbolo y entonces pasar a un nuevo estado.

Representación de una máquina de Turing



Notación

Γ : Símbolos de cinta

S : Conjunto finito de estados

S' : Conjunto finito de estados no finales

δ : Función de transición $\delta : (S' \times \Gamma) \longrightarrow S \times (\Gamma \cup \{LR\})$
donde $LR \notin \Gamma$

Δ : Espacio en blanco

Definición formal de Máquina de Turing

Una Máquina de Turing es una sextupla de la forma $(S, \Sigma, \Gamma, \delta, i, h)$ donde:

S es una colección finita de estados.

Σ es un conjunto finito de símbolos distintos de espacio en blanco, llamado alfabeto.

Γ es un conjunto finito de símbolos, incluidos los de Σ , que se conocen como símbolos de la cinta de la máquina.

δ es la función de transición de la máquina.

i es un elemento de S llamado estado inicial.

h es un elemento de S llamado estado de parada.

Ejemplo de una máquina de Turing

S i, h

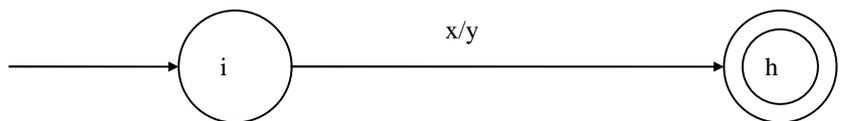
Σ $\{x, y\}$

Γ $\{x, y, \Delta\}$

$\delta(i, x) = (h, y)$

i estado inicial

h estado de parada



La Tesis de Turing nos dice que el poder computacional de una máquina de Turing es tan grande como el de cualquier sistema computacional posible.

Sistema de codificación de máquinas de Turing

Dada una máquina M por representar, nuestro sistema de codificación necesita que acomodemos los estados de M en una lista cuyo primer elemento corresponda al estado inicial y el segundo elemento corresponda al estado de parada. Con base en el orden de la lista, podemos hablar del primer estado de M , del segundo estado de M y, en general, del estado j de M . Establecemos que el estado j de M se representará con una cadena de ceros de longitud j . Así, el estado inicial estará representado por 0, el estado de parada por 00 y el siguiente estado (si existe) por 000.

Luego representamos los símbolos L y R como los símbolos en Σ (los símbolos de cinta de M distintos de espacio en blanco) como cadenas de ceros. Esto se hace acomodando en una lista los símbolos de Σ y representando L con 0, R con 00, el primer símbolo de la lista con 000, el segundo símbolo con 0000 y, en general, el símbolo j con una cadena de ceros de longitud $j+2$.

Si ahora establecemos que el símbolo del espacio en blanco se representará por la cadena vacía, obtenemos un sistema en el cual podemos representar los símbolos L y R , los estados de M y los símbolos de la cinta de M por medio de cadenas de ceros. A su vez esto nos permite representar cualquier transición de M como una cadena de ceros y unos.

A fin de cuentas se puede representar cualquier transición (que debe tener la forma :

$\delta(p,x)=(q,y)$ con la cuádrupla (p,x,q,y) donde

p es el estado actual

x es el símbolo actual

q es el nuevo estado

y es o bien un símbolo de cinta si la transición es de escritura o L o

R si la transición es de movimiento

De esta manera podemos representar toda la transición como cuatro cadenas de ceros separadas por unos.

Por ejemplo;

La cadena 01000100100 representaría la transición $\delta(i,x)=(h,R)$ donde x es el símbolo representado por 000 y h es el estado de parada de la máquina.

REQUERIMIENTOS

Para poder correr este programa debemos contar con los siguientes recursos computacionales:

a) Una computadora personal compatible con IBM con las siguientes características :

Sistema Operativo Msdos v. 3.0 ó mayor

Procesador 80286 ó mayor

Un Megabyte de memoria Ram

Unidad de drive de 3 ½

b) Un regulador de voltage.

c) Una impresora de matriz de puntos compatible con IBM.

d) Procedimiento de ejecución

Encender la computadora.

Insertar el disco flexible en el drive de 3 ½.

Teclear el nombre Turing que es el nombre del programa.

Correr el programa.

Salir del programa.

OBJETIVO

El objetivo de este trabajo es comprender y mostrar el funcionamiento de una máquina de Turing codificada, además de analizar cadenas en dichas máquinas para determinar si son aceptadas o no por la máquina.

Este trabajo puede consultarse como material de apoyo para el entendimiento de el funcionamiento de estas máquinas.

En el mostramos lo que es una máquina de Turing, cuales son sus componentes, cuales son las operaciones que realiza y también de que manera podemos implementarla.

DESCRIPCIÓN DEL PROGRAMA

El programa consta de 6 pantallas, cada una de ellas realiza una función diferente, cabe mencionar que para cada pantalla existen reglas de validación de acuerdo a las reglas que rigen las máquinas de Turing.

La primera pantalla nos muestra la pantalla principal del programa simulador de las Máquinas de Turing. La función de esta pantalla es la de proporcionarnos información y continuar con la ejecución del programa.

La segunda pantalla se utiliza para registrar los símbolos Σ del alfabeto de nuestra máquina, cabe resaltar que no podemos registrar 2 veces el mismo símbolo.

La tercera pantalla se utiliza para registrar los estados S de nuestra máquina, cabe resaltar que no podemos registrar 2 veces el mismo estado.

Primero registramos el estado inicial y el de parada. Los estados que no son el inicial ni el de parada los podemos registrar en cualquier orden.

La cuarta pantalla se utiliza para registrar las transiciones δ de nuestra máquina, cabe resaltar que no podemos registrar 2 veces la misma transición, tampoco podemos realizar una transición que salga del estado de parada.

La quinta pantalla se utiliza para realizar la construcción de la Máquina codificada. La notación utilizada es la siguiente: el indicador de inicio, el separador de transiciones y el indicador de fin son un 1 en color rojo, el estado inicial y el estado final son una cadena de 0 en color verde, el separador de estados y de símbolos son un 1 en color azul, los símbolos son una cadena de 0 de color morado.

La sexta pantalla se utiliza para mostrar la cadena que deseamos analizar en esta máquina. Esta pantalla nos muestra la cadena original a analizar, el cabezal de lectura, la cadena que va resultando, el estado actual, el símbolo actual y la transición que se está ejecutando

El procedimiento es el siguiente:

- **Primero debemos teclear la cadena que deseamos analizar.**
- **Oprimir la barra de espacio para ir analizando la cadena.**
- **La máquina nos despliega un mensaje de aceptación si la cadena llega a un estado de parada, de lo contrario despliega un mensaje de no aceptación.**
- **Para terminar el programa nos pregunta si deseamos analizar otra cadena con esta máquina, si respondemos que si regresamos al primer punto de lo contrario el programa termina la ejecución.**

LISTADO DEL PROGRAMA

```
/* -----*/
/* TEORIA DE AUTOMATAS FINITOS */
/* DRA. ADA MARGARITA ALVAREZ S. */
/* PROGRAMA SIMULADOR DE UNA MAQUINA DE TURING. FEB 1996 */
/* -----*/
/* EQUIPO # 5 */
/* María Guadalupe Mendoza García */
/* Javier Castañeda Ambriz */
/* Jaime David Johnston Barrientos */
/* Sergio Garza Carranza */
/* Miguel Angel Guerrero Crespo */
/* Bruno López Takeyas */
/* -----*/
```

```
#define CODIFICA
/*#define DATOS*/
#define LIMPIAR
#define PORTADA
#define MESSAGE
#define CURSOR
#define MAX 100
```

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include "blt.h"
```

```
#ifdef MESSAGE
void message(char texto[81],int atributo);
#endif MESSAGE
#ifdef PORTADA
void portada(void);
#endif PORTADA
void CAPTURA(void);
int BUSCA(char simbolo,char A[MAX]);
int BUSCA_TRANS(char trans[5],char A[MAX][5]);
void SORTEA(char A[MAX]);
void SORTEA_TRANS(char X[MAX][5],int top);
void OPRIMA(char texto[81],int atributo);
void CONSTRUCCION(int transiciones);
void RECORRIDO(void);
```

```
char ALFABETO[MAX];
char ESTADOS[MAX];
char INICIO,PARADA;
char TRANSICION[MAX][5],otra;
```

```
void main(void)
{
#ifdef DATOS
RECORRIDO();
exit(0);
#endif DATOS
```

```

#ifdef PORTADA
portada();
#endif PORTADA
CAPTURA();
do
{
    RECORRIDO();
    do
    {
        OPRIMA(" " Desea evaluar otra cadena con la misma m quina ? [S/N]           ",27);
    }while(otra!='N' && otra!='S');
    }while(otra=='S');
#ifdef LIMPIAR
textattr(7); window(1,1,80,25); clrscr();
#endif LIMPIAR
return;
}

#ifdef PORTADA
void portada(void)
{
    no_cursor();
    textattr(7);clrscr();
    fondo(BLUE);
    fondo1(CYAN);
    textattr(78);
    ventana(20,2,60,6,RED);
    centerv(2,"TEORIA DE AUTOMATAS FINITOS");
    centerv(3,"Dra. Ada Margarita Alvarez S.");

    ventana(15,8,65,19,BLUE);textattr(27);
    centerv(1," MAQUINAS DE TURING ");
    centerv(3,"Equipo # 5");
    centerv(5,"Ma. Gpe. Mendoza Garcia ");
    centerv(6,"Javier Castañeda Ambriz ");
    centerv(7,"Jaime David Johnston B. ");
    centerv(8,"Sergio Garza Carranza ");
    centerv(9,"Miguel Angel Guerrero C.");
    centerv(10,"Bruno López Takeyas ");

    ventana(13,21,67,24,GREEN);
    textattr(32);
    centerv(2,"<<< Oprima cualquier tecla para continuar >>>");
    getch();
    si_cursor();
    window(1,1,80,25);
    return;
}
#endif PORTADA

void CAPTURA(void)
{
    int columna=10, renglon=1, flag,i;
    char simbolo;
    char transicion[5];
    textattr(BLUE);
    clrscr();
    fondo1(CYAN);
    textattr(30);
    ventana(2,2,78,23,BLUE); textattr(31);
    centerv(1," PROGRAMA SIMULADOR DE UNA MAQUINA DE TURING ");
    window(4,3,76,21);

```

```

for(flag=0;flag<=MAX;flag++) { ALFABETO[flag]='\0'; ESTADOS[flag]='\0'; }
i=0;
do
{
do
{
textattr(27);
message(" Anote los s;mbolos del alfabeto ä < ENTER > = Salir ",30);
gotoxy(columna, renglon);
cprintf(" %c( %3d ) = ",228,i);
simbolo=tolower(getch());
if(simbolo==32)
OPRIMA(" No introduzca el blanco !!! <<< Oprima cualquier tecla para continuar >>> ",206);
}while(simbolo==32);

flag=BUSCA(simbolo,ALFABETO);
if(flag<0)
{
ALFABETO[i]=simbolo;
gotoxy(columna+12, renglon); cprintf("%c",ALFABETO[i]);
renglon++; columna=10;
i++;
}
else
OPRIMA(" S;mbolo repetido !!! <<< Oprima cualquier tecla para continuar >>> ",206);
if(i%19==0)
{
clrscr(); columna=10; renglon=1;
}
}while(ALFABETO[i-1]!='\r');
ALFABETO[i-1]=32;
ALFABETO[i]='L';
ALFABETO[i+1]='R';
ALFABETO[i+2]='\0';
SORTEA(ALFABETO);

/* CAPTURAR LOS ESTADOS DE LA MAQUINA */
renglon=2;
window(1,1,80,25);
textattr(GREEN);
clrscr();
fondo1(GREEN);
ventana(2,2,78,23,BLUE); textattr(31);
centerv(1," PROGRAMA SIMULADOR DE UNA MAQUINA DE TURING ");
ventana(20,10,60,15,GREEN);
textattr(32);
centerv(1," ESTADOS INICIAL Y DE PARADA ");
do
{
message(" Estado inicial de la m quina ",32);
textattr(33);
gotoxy(15,3);
cprintf("Inicial = ",i);
simbolo=toupper(getch());
if((simbolo<65 || simbolo>90) && simbolo!='\r')
OPRIMA(" Estado inv lido !!! <<< Oprima cualquier tecla para continuar >>> ",206);
flag=BUSCA(simbolo,ESTADOS);
if(flag<0)
{
gotoxy(25,3); cprintf("%c",simbolo);
INICIO=simbolo;

```

```

    }
    else OPRIMA(" Estado repetido !!! <<< Oprima cualquier tecla para continuar >>> ",206);
}while(simbolo<65 || simbolo>90 || flag!=-1);

do
{
    message(" Estado de parada de la m quina ",32);
    textattr(33);
    gotoxy(15,4);
    cprintf("Parada = ");
    simbolo=toupper(getch());
    if((simbolo<65 || simbolo>90) && simbolo!='\r')
        OPRIMA(" Estado inv lido !!! <<< Oprima cualquier tecla para continuar >>> ",206);
    if(simbolo==INICIO)
    {
        OPRIMA(" El estado de inicio NO puede ser de parada ",206);
        flag=1;
    }
    else
    {
        flag=BUSCA(simbolo,ESTADOS);
        if(flag<0)
        {
            gotoxy(25,4); cprintf("%c",simbolo);
            PARADA=simbolo;
        }
        else OPRIMA(" Estado repetido !!! <<< Oprima cualquier tecla para continuar >>> ",206);
    }
}while(simbolo<65 || simbolo>90 || flag!=-1);

window(1,1,80,24);
window(4,3,76,21);
textattr(17);
clrscr();
si_cursor();
i=0; columna=10; renglon=1;
do
{
    do
    {
        message(" Estados de la m quina. No incluya el inicio y parada !!! < ENTER > = Salir ",32);
        textattr(27);
        gotoxy(columna,renglon);
        cprintf("S ( %3d ) = ",i);
        simbolo=toupper(getch());
        if((simbolo<65 || simbolo>90) && simbolo!='\r')
            OPRIMA(" Estado inv lido !!! <<< Oprima cualquier tecla para continuar >>> ",206);
    }while((simbolo<65 || simbolo>90) && simbolo!='\r');

    flag=BUSCA(simbolo,ESTADOS);
    if(simbolo==INICIO || simbolo==PARADA) flag=1;
    if(flag<0)
    {
        ESTADOS[i]=simbolo;
        gotoxy(columna+12,renglon); cprintf("%c",ESTADOS[i]);
        renglon++; columna=10;
        i++;
    }
    else OPRIMA(" Estado repetido !!! <<< Oprima cualquier tecla para continuar >>> ",206);
    if(i%19==0)
    {
        clrscr(); columna=10; renglon=1;
    }
}

```

```

    }
}while(ESTADOS[i-1]!='r');
ESTADOS[i-1]='\0';

ESTADOS[strlen(ESTADOS)]=32; ESTADOS[strlen(ESTADOS)+1]='\0';
ESTADOS[strlen(ESTADOS)]=32; ESTADOS[strlen(ESTADOS)+1]='\0';
SORTEA(ESTADOS);
ESTADOS[0]=INICIO;
ESTADOS[1]=PARADA;

/* CAPTURAR LAS TRANSICIONES */
columna=10; renglon=1;
window(1,1,80,25);
textattr(RED);
clrscr();
fondo1(6);
ventana(2,2,78,23,BLUE); textattr(31);
centerv(1," TRANSICIONES DE LA MAQUINA DE TURING ");
window(4,3,76,21); gotoxy(1,1);
i=0;
while(1)
{
    do
    {
        flag=-1;
        message(" Estado origen                                < ENTER > = Salir ",111);
        columna=10; textattr(27);
        gotoxy(columna,renglon); cprintf("è( , ) = ( , )");
        columna+=2;
        gotoxy(columna,renglon); simbolo=toupper(getch());
        if(simbolo==PARADA)
            OPRIMA(" El estado de parada NO puede originar una transición !!!           ",206);
        else
        {
            flag=BUSCA(simbolo,ESTADOS);
            if(flag<0 && simbolo!='r')
                OPRIMA(" Estado inv lido !!!           <<< Oprima cualquier tecla para continuar >>> ",206);
        }
    }while(flag<0 && simbolo!='r');
}
/* *****
/* Aqui validar que el estado capturado NO sea el estado de parada */
/* *****

if(simbolo=='r') break;
transicion[0]=simbolo;
gotoxy(columna,renglon); cprintf("%c",simbolo);

columna+=2;
do
{
    message(" Símbolo actual en la cinta de entrada                < ESPACIO > = Blanco ",111);
    gotoxy(columna,renglon); simbolo=tolower(getch());
    flag=BUSCA(simbolo,ALFABETO);
    if(flag<0)
        OPRIMA(" Símbolo inv lido !!!           <<< Oprima cualquier tecla para continuar >>> ",206);
}while(simbolo=='r' || flag<0);
transicion[1]=simbolo;
if(simbolo==32) simbolo=127;
gotoxy(columna,renglon); cprintf("%c",simbolo);

columna+=6;
do

```

```

{
    message(" Estado destino                                     ",111);
    gotoxy(columna,renglon); simbolo=toupper(getch());
    flag=BUSCA(simbolo,ESTADOS);
    if(flag<0)
        OPRIMA(" Estado inv lido !!!          <<< Oprima cualquier tecla para continuar >>> ",206);
}while(simbolo=='\r' || flag<0);
transicion[2]=simbolo;
gotoxy(columna,renglon); cprintf("%c",simbolo);

columna+=2;
do
{
    message(" Símbolo a escribir en la cinta u operación <F1>=L <F2>=R <ESPACIO>=Blanco
",111);
    gotoxy(columna,renglon); simbolo=tolower(getch());
    if(simbolo==0)
    {
        simbolo=getch();
        if(simbolo==F1) simbolo='L';
        if(simbolo==F2) simbolo='R';
    }
    flag=BUSCA(simbolo,ALFABETO);
    if(flag<0)
        OPRIMA(" Símbolo inv lido !!!          <<< Oprima cualquier tecla para continuar >>> ",206);
}while(simbolo=='\r' || flag<0);
transicion[3]=simbolo; transicion[4]='\0';
if(simbolo==32) simbolo=127;
gotoxy(columna,renglon); cprintf("%c",simbolo);
if(transicion[0]==transicion[2] && transicion[1]==transicion[3])
    OPRIMA(" Transición redundante !!!      <<< Oprima cualquier tecla para continuar >>> ",206);
else
{
    flag=BUSCA_TRANS(transicion,TRANSICION);
    if(flag<0)
    {
        simbolo=transicion[0]; TRANSICION[i][0]=simbolo;
        simbolo=transicion[1]; TRANSICION[i][1]=simbolo;
        simbolo=transicion[2]; TRANSICION[i][2]=simbolo;
        simbolo=transicion[3]; TRANSICION[i][3]=simbolo;
        simbolo='\0'; TRANSICION[i][4]=simbolo;
        i++; renglon++;
        if(i%19==0)
        {
            clrscr(); columna=10; renglon=1;
        }
    }
    else
        if(flag==(MAX+1))
        {
            OPRIMA(" No se permite el NO determinismo << Oprima cualquier tecla para continuar >>
",206);
        }
    else
    {
        OPRIMA(" Transición repetida !!!      <<< Oprima cualquier tecla para continuar >>> ",206);
    }
}
}

SORTEA_TRANS(TRANSICION,i);
renglon++; columna=3;

```

```

    gotoxy(columna, renglon);
#ifdef CODIFICA
    CONSTRUCCION(i);
#endif CODIFICA
    return;
}

int BUSCA(char simbolo, char A[MAX])
{
    int k, flag=-1;
    for(k=0; k<=strlen(A); k++)
    {
        if(simbolo==A[k]) flag=k;
        if(flag!=-1) break;
    }
    return flag;
}

#ifdef MESSAGE
void message(char texto[81], int atributo)
{
    gettextinfo(&info);
    window(1, 1, 80, 25);
    textattr(atributo);
    gotoxy(1, 24); cputs(texto);
    window(info.winleft, info.wintop, info.winright, info.winbottom);
    textattr(info.attribute);
    gotoxy(info.curx, info.cury);
    return;
}
#endif MESSAGE

void SORTEA(char A[MAX])
{
    int i, j;
    char car;
    for(i=0; i<strlen(A)-1; i++)
        for(j=i+1; j<strlen(A); j++)
            if(A[i]>A[j])
            {
                car=A[i];
                A[i]=A[j];
                A[j]=car;
            }
    return;
}

int BUSCA_TRANS(char trans[5], char A[MAX][5])
{
    int k, flag=-1;
    for(k=0; k<=MAX; k++)
    {
        if(strcmp(A[k], trans)==0) flag=k;
        if(strcmp(A[k], "")==0 || flag!=-1) break;
        if(trans[0]==A[k][0] && trans[1]==A[k][1]) flag=MAX+1;
    }
    return flag;
}

void SORTEA_TRANS(char X[MAX][5], int top)
{
    int i, j, flag=0;

```

```

char cade[5];
char INI[MAX][5];
int top_i=0;

for(i=0;i<top;i++)
{
    if(X[i][0]==INICIO)
    {
        strcpy(INI[top_i],X[i]);    top_i++;
        strcpy(X[i]," ");
    }
}

for(i=0;i<top-1;i++)
for(j=i+1;j<top;j++)
if(strcmp(INI[i],INI[j])>0)
{
    strcpy(cade,INI[i]);
    strcpy(INI[i],INI[j]);
    strcpy(INI[j],cade);
}

for(i=0;i<top-1;i++)
for(j=i+1;j<top;j++)
{
    if(strcmp(X[i],X[j])>0)
    {
        strcpy(cade,X[i]);
        strcpy(X[i],X[j]);
        strcpy(X[j],cade);
    }
}

for(i=0;i<top;i++)
strcpy(X[i],INI[i]);
return;
}

void OPRIMA(char texto[81],int atributo)
{
    no_cursor();
    message(texto,atributo);
    otra=toupper(getch());
    si_cursor();
    return;
}

void CONSTRUCCION(int transiciones)
{
    int i,j,k,flag,columna=2,renlon=2;
    char texto[81];
    window(1,1,80,25);
    textattr(5);
    clrscr();
    fondo1(5);
    textattr(30);
    ventana(2,2,78,23,BLUE);
    textattr(30);
    centerv(1," CONSTRUCCION DE LA MAQUINA DE TURING ");
    window(4,3,76,21); gotoxy(columna,renlon);
    OPRIMA(" Indicador de inicio                < ENTER > = Continuar ",28);
    textattr(28);
}

```

```

gotoxy(columna, renglon);
textov("1");
for(i=0;i<transiciones;i++)
{
for(j=0;j<strlen(TRANSICION[i]);j++)
{
texto[0]='\0';
strcpy(texto," Analizando e (");
flag=strlen(texto);
texto[flag]=TRANSICION[i][0];
texto[flag+1]='\0';
strcat(texto," ");
flag=strlen(texto);
if(TRANSICION[i][1]==32) TRANSICION[i][1]=127;
texto[flag]=TRANSICION[i][1];
texto[flag+1]='\0';
strcat(texto," = (");
flag=strlen(texto);
texto[flag]=TRANSICION[i][2];
texto[flag+1]='\0';
strcat(texto," ");
flag=strlen(texto);
if(TRANSICION[i][3]==32) TRANSICION[i][3]=127;
texto[flag]=TRANSICION[i][3];
texto[flag+1]='\0';
strcat(texto,"");
if(j%2==0) strcat(texto," con el estado ");
else
{
if(TRANSICION[i][j]=='R' || TRANSICION[i][j]=='L')
strcat(texto," con la operaciøn ");
else strcat(texto," con el símbolo ");
}
flag=strlen(texto);
texto[flag]=TRANSICION[i][j]; texto[flag+1]='\0';
strcat(texto," ");
if(j%2==0) OPRIMA(texto,26); else OPRIMA(texto,25);

if(j%2==0) flag=BUSCA(TRANSICION[i][j],ESTADOS)+1;
else flag=BUSCA(TRANSICION[i][j],ALFABETO);
if(j%2==0) textattr(26); else textattr(25);
for(k=0;k<flag;k++)
{
textov("0");
}
if(j==3)
{
textattr(28);
OPRIMA("Indicador de cambio de transición",28);
}
else
{
textattr(27);
OPRIMA("Separador",27);
}
textov("1");
}
}
OPRIMA("Indicador de fin de la codificación.",28);
textattr(28);
textov("1");
OPRIMA("<<< OPRIMA ENTER PARA CONTINUAR >>> ",206);

```

```

return;
}

void RECORRIDO(void)
{
char cadena[70];
char edo_actual;
int b,i,k,s,flag,columna,col_ant,fin_cadena=0;
window(1,1,80,25);
textattr(28);
clrscr();
fondo1(12);
gotoxy(1,24);
cputs(" Anote la cadena a evaluar          < ENTER > = Salir  ");
textattr(30);
ventana(2,2,78,23,BLUE);
textattr(30);
centerv(1," EVALUACION DE UNA CADENA CON ESTA MAQUINA DE TURING ");
window(4,3,76,21);

i=0;
gotoxy(2,2);
textattr(27);
do
{
do
{
cadena[i]=tolower(getch());
}while(cadena[i]!='\r' && i==0);
if(cadena[i]!='\r')
{
cadena[i]='\0';
break;
}
cadena[i+1]='\0';
if(cadena[i]==32) cadena[i]=127;
if(cadena[i]!=32) cprintf("%c",cadena[i]); else cprintf("%c",127);
if(i==70)
{
beep(700,20);
OPRIMA(" Cadena demasiado larga !!!          < ENTER > = Continuar ",156);
MESSAGE(" Anote la cadena a evaluar          < ENTER > = Salir  ",28);
clrscr();
i=0;
gotoxy(2,2);
textattr(27);
}
else i++;
}while(i>=70 || cadena[i-1]!='\r');
for(i=strlen(cadena)-1;i>0;i--)
if(cadena[i]==127)
cadena[i]='\0';
else break;
textattr(28);
clrscr();

#ifdef DATOS
INICIO='I';
PARADA='Z';
strcpy(ALFABETO," LRxyz");
strcpy(ESTADOS," IZMNQRT");
strcpy(TRANSICION[0]," I TR");

```

```

strcpy(TRANSICION[1],"IxTR");
strcpy(TRANSICION[2],"IyTR");
strcpy(TRANSICION[3],"M MR");
strcpy(TRANSICION[4],"MxNx");
strcpy(TRANSICION[5],"MyMR");
strcpy(TRANSICION[6],"N Z ");
strcpy(TRANSICION[7],"NxZx");
strcpy(TRANSICION[8],"NyZy");
strcpy(TRANSICION[9],"Q QR");
strcpy(TRANSICION[10],"QxQR");
strcpy(TRANSICION[11],"QyRy");
strcpy(TRANSICION[12],"R Z ");
strcpy(TRANSICION[13],"RxZx");
strcpy(TRANSICION[14],"RyZy");
strcpy(TRANSICION[15],"T Z ");
strcpy(TRANSICION[16],"TxMR");
strcpy(TRANSICION[17],"TyQR");
strcpy(TRANSICION[18],"TyQR");
#endif DATOS

edo_actual=INICIO;
i=0;
columna=2;

textattr(29); gotoxy(2,2); cprintf("Cadena original : ");
gotoxy(2,3);
for(s=0;s<=strlen(cadena);s++)
    if(cadena[s]==32) cprintf("%c",127); else cprintf("%c",cadena[s]);
do
{
    textattr(26);
    gotoxy(columna,4); cprintf("%c",157);
    textattr(28);
    gotoxy(2,5);
    for(s=0;s<=strlen(cadena);s++)
        if(cadena[s]==32) cprintf("%c",127); else cprintf("%c",cadena[s]);
    textattr(25); gotoxy(2,7); cprintf("Estado actual : ");
    textattr(30); cprintf("%c",edo_actual);
    textattr(25); gotoxy(2,9); cprintf("Símbolo actual : ");
    textattr(30);
    if(cadena[i]!='\0' && cadena[i]!=32) cprintf("%c",cadena[i]);
    else if(cadena[i]!='\0') cprintf("%c",127);
    textattr(25); gotoxy(2,11); cprintf("Transición : ");

    if(cadena[i]!='\0' && cadena[i]==127) { flag=BUSCA(32,ALFABETO);}
    else if(cadena[i]!='\0' && cadena[i]!=127) { flag=BUSCA(cadena[i],ALFABETO);}
    else { flag=0; cadena[i]=127;}
    if(flag<0) break;
    for(k=0,flag=-1;k<=MAX;k++)
    {
        if(TRANSICION[k][0]!=edo_actual) continue;
        if(TRANSICION[k][1]==cadena[i])
        {
            flag=1;
            textattr(30);
            cprintf("ë(%c,",TRANSICION[k][0]);
            if(TRANSICION[k][1]==32) cprintf("%c) = (",127); else cprintf("%c) = (",TRANSICION[k][1]);
            cprintf("%c,",TRANSICION[k][2]);
            if(TRANSICION[k][3]==32) cprintf("%c)",127); else cprintf("%c)",TRANSICION[k][3]);
            edo_actual=TRANSICION[k][2];
            col_ant=columna;
            if(TRANSICION[k][3]=='L')

```

```

        { columna--; i--; }
        if(TRANSICION[k][3]=='R')
        { columna++; i++; }
        if(TRANSICION[k][3]!='R' && TRANSICION[k][3]!='L')
        {
            cadena[i]=TRANSICION[k][3];
        }
        OPRIMA(" <<< Oprima cualquier tecla para continuar >>> ",27);
        gotoxy(col_ant,4); cprintf(" ");
        if(i>=strlen(cadena)-1) fin_cadena=1;
        break;
    }
}
if(flag==-1) break;
if(i==strlen(cadena) && edo_actual!=PARADA)
{
/* @@@@  cadena[i]=32;*/
for(b=0;b<=MAX;b++)
    if(TRANSICION[b][0]==edo_actual && TRANSICION[b][1]==127 &&
TRANSICION[b][2]!=edo_actual)
    {
        cadena[i]=127;  cadena[i+1]='\0';
        break;
    }
}
}while(i>=0 && i<strlen(cadena) && edo_actual!=PARADA);

textattr(26); gotoxy(columna,4); cprintf("%c",157);
textattr(28);
gotoxy(2,5);
for(s=0;s<=strlen(cadena);s++)
    if(cadena[s]==32) cprintf("%c",127); else cprintf("%c",cadena[s]);
textattr(25); gotoxy(2,7); cprintf("Estado actual : ");
textattr(30); cprintf("%c",edo_actual);
textattr(25); gotoxy(2,9); cprintf("Símbolo actual : ");
textattr(30); cprintf("%c",cadena[i]);
textattr(25); gotoxy(2,11); cprintf("Transición : ");

if(edo_actual!=PARADA || (i<strlen(cadena)-1 && fin_cadena!=1))
{
    cprintf(" ");
    OPRIMA(" Cadena NO aceptada !!! < ENTER > = Continuar ",206);
}
else
{
    OPRIMA(" Cadena aceptada !!! < ENTER > = Continuar ",206);
}
return;
}

```

CONCLUSIONES

El programa nos muestra el funcionamiento de una máquina de Turing codificada, en la que podemos observar el estado inicial, las transiciones y el estado de parada de una manera clara y sencilla, además de poder analizar cadenas y determinar si son o no aceptadas por la máquina .

Las máquinas de Turing encierran todo el poder de los procesos computacionales, pues son capaces de simular cualquier proceso algorítmico. También encontramos que si una máquina de Turing no puede resolver un problema, entonces ninguna computadora podrá hacerlo simplemente porque no existe un algoritmo para obtener su solución. En otras palabras, las limitaciones que hemos detectado corresponden a los procesos computacionales, no a la tecnología.

Las máquinas de Turing pueden ser simuladas por autómatas, los lenguajes aceptados por estas máquinas son los lenguajes estructurados por frases.

BIBLIOGRAFÍA

Teoría de la computación
J. Glenn Brookshear
Addison-Wesley Iberoamericana

Introducción a la teoría de Automatas, lenguajes y computación
John E. Hopcroft, Jeffrey D. Ullman
Cecsa