Prácticas de Programación Resueltas: Máquinas Secuenciales

1. *Realizar un programa que al serle introducido un autómata* $A=(S_1,S_2,E,\delta,\lambda)$*, una palabra* $x\in F_{S_1}$ *y el estado de partida e devuelva la tabla de computación correspondiente. Los cardinales de los conjuntos* S_1,S_2 *y* E *no podrán superar el valor de 25 y la longitud de la palabra x será menor o igual que 25. Los elementos de los conjuntos* S_1 *y* S_2 *se representarán por un único carácter. Los elementos de* E *estarán representados por números naturales o 0. Los datos de entrada serán: los elementos de* S_1,S_2*, el número de estados de* E*, las funciones* δ *y* λ*, la longitud de la palabra* x*, las letras de la palabra* x *y el estado e.*

Solución. La idea que se puede seguir para realizar este programa es la siguiente:

- Introducir los datos que se necesitan: Los cardinales de los conjuntos S_1,S_2, los elementos de los conjuntos S_1,S_2, el cardinal de E, el primer valor de E, la función δ y la función λ. Además, deberemos comprobar que ambas funciones están bien definidas y que en cada conjuntos S_1 y S_2 no hay elementos repetidos. Esto último lo haremos con una subrutina auxiliar llamada *pertenece.*
- Pedir la longitud de la palabra x y sus letras.
- Construir la sucesión de $\delta(e,s_{11})$, $\delta(\delta(e,s_{11}),s_{12})$,… suponiendo que $x=s_{11}s_{12}$….
- Construir la sucesión de $\lambda(e,s_{11})$, $\lambda(\delta(e,s_{11}),s_{12})$,… suponiendo que $x=s_{11}s_{12}$….
- Escribir los datos en pantalla.

Por ejemplo, un programa escrito en C que implementa estas ideas es:

```
#include <stdio.h>
int pertenece(char elemento, char conjunto[],int cardinal)
{
   /* Devuelve un 0 si el elemento no pertenece al conjunto y un numero que indica el indice
   del elemento en el conjunto si este pertenece*/
   int i;
   for(i=1;i<=cardinal; i++)
   {
      if(elemento==conjunto[i]) return(i);
   }
   return(0);
}

void escribir_caracteres(char conjunto[], int cardinal)
{
   /* Sirve para escribir un vector de caracteres dejando un espacio entre dos caracteres*/
   int i;
   for(i=1;i<=cardinal;i++) printf("%c ",conjunto[i]);
```

```
   printf("\n");
}


void escribir_enteros(int conjunto[],int cardinal);
{
   /* Sirve para escribir un vector de numeros de hasta dos cifras. Si el numero
    tiene un unica cifra deja un espacio en blanco detras de ella*/
    int i;
    for(i=1;i<=cardinal;i++)
    {
       if(conjunto[i]<10&&conjunto[i]>-1) printf("%d ",conjunto[i]);
       else printf("%d",conjunto[i]);
    }
}

void escribir_delta(int delta[][25], int nestados, int nentradas, int estado_inicial, char
entradas[])
{
   int i,j;
   printf("\n\n La funcion delta viene dada por:\n   ");
   for(i=1;i<=nentradas;i++)
   {
      printf("%c  ",entradas[i]);
   }
   for(i=0;i<nestados;i++)
   {
      if (estado_inicial+i<10) printf("\n%d  ",estado_inicial+i);
      else printf("\n%d ",estado_inicial+i);
      for(j=1;j<=nentradas;j++)
      {
         if (delta[estado_inicial+i][j]<10) printf("\n%d  ",delta[estado_inicial+i][j]);
         else printf("\n%d ",delta[estado_inicial+i][j]);
      }
   }
}

void escribir_lambda(char lambda[][25], int nestados, int nentradas, int estado_inicial, char
entradas[])
{
   int i,j;
   printf("\n\n La funcion lambda viene dada por:\n   ");
   for(i=1;i<=nentradas;i++)
   {
      printf("%c ",entradas[i]);
   }
   for(i=0;i<nestados;i++)
   {
      if (estado_inicial+i<10) printf("\n%d  ",estado_inicial+i);
      else printf("\n%d ",estado_inicial+i);
      for(j=1;j<=nentradas;j++)
      {
```

```
            printf("\n%c ",lambda[estado_inicial+i][j]);
        }
    }
}

void delta_tabla(int indpalabra[],int lpalabra,int estado_inicial,int suc_delta[], int delta[][25])
{
   /* Esta funcion construye la sucesion de las deltas en la tabla de computacion con la palabra
x
   partiendo del estado estado_inicial.*/
   int i;
   suc_delta[1]=estado_inicial;
   for(i=2;i<=lpalabra+1;i++)
   {
      suc_delta[i]=delta[suc_delta[i-1]][indpalabra[i-1]];
   }
}


void lambda_tabla(int indpalabra[],int lpalabra,char suc_lambda[],int suc_delta[], char
lambda[][25])
{
   /* Esta funcion construye la sucesion de las lambdas en la tabla de computacion con la
palabra x
   partiendo del estado suc_delta[1]*/
   int i;
   for(i=1;i<=lpalabra;i++)
   {
      suc_lambda[i]=lambda[suc_delta[i]][indpalabra[i]];
   }
}


void main (void)
{
   /* Programa principal que construye la tabla de computacion de un automata*/
   int nentradas, nsalidas, nestados, lpalabra, estados[100],i,j,delta[100][25],
   int indpalabra[36], suc_delta[37],estado_inicial, esta,estado_partida;
   char entradas[25],salidas[25], palabra[36], suc_lambda[37], lambda[100][25],resp;
   printf("\n Este programa construye la tabla de computacion para una palabra X");
   printf(" partiendo del");
   printf("\n  estado e segun un automata dado. ");
   printf("\n El conjunto de entradas admite hasta 24 elementos");
   printf("\n Los conjuntos de  salidas y estados pueden tener como");
   printf(" mucho 99 elementos."),
   printf(" La longitud de la palabra no puede sobrepasar 35 caracteres. Los elementos");
   printf(" de los conjuntos de entrada y salida solo pueden tener un unico caracter");
   do{
   printf("\n\n Dame el cardinal del conjunto de entradas: ");
   scanf("%d",&nentradas);
   if(nentradas>24||nentradas<1) printf("\n Cardinal no valido. Introduzca de nuevo");
   }while (nentradas>24||nentradas<1);
```

```
  do{
  printf("\n Dame el cardinal del conjunto de salidas: ");
  scanf("%d",&nsalidas);
  if(nsalidas>24||nsalidas<1) printf("\n Cardinal no valido. Introduzca de nuevo");
  }while (nsalidas>24||nsalidas<1);
  do{
  printf("\n Dame el cardinal del conjunto de estados: ");
  scanf("%d",&nestados);
  if(nestados>24||nestados<1) printf("\n Cardinal no valido. Introduzca de nuevo");
  }while (nestados>24||nestados<1);
  printf("\n Introduce los elementos del conjunto de entradas. Recuerda: los elementos solo
pueden tener un caracter");
  for(i=1;i<=nentradas;i++)
  {
    do{
    printf("\n Elemento %d-esimo: ",i);
    scanf("%c",&entradas[i]);
    esta=pertenece(entradas[i], entradas,i-1);
    if(esta!=0)
    {
      printf("\n Elemento no valido. Coincide con uno de los anteriores.");
      printf("\n Introduzca de nuevo.");
    }
    }while(esta!=0);
  }
  printf("\n Introduce los elementos del conjunto de salidas. Recuerda: los elementos solo
pueden tener un caracter");
  for(i=1;i<=nsalidas;i++)
  {
    do{
    printf("\n Elemento %d-esimo: ",i);
    scanf("%c",&salidas[i]);
    esta=pertenece(salidas[i], salidas,i-1);
    if(esta!=0)
    {
      printf("\n Elemento no valido. Coincide con uno de los anteriores.");
      printf("\n Introduzca de nuevo");
    }
    }while(esta!=0);
  }
  printf("\n Los estados se numeran consecutivamente. Por favor introduce el indice del");
  printf("\n primer estado (debe ser 0 o 1)");
  scanf("%d",&estado_inicial);
  printf("\n Introduce la funcion delta");
  for(i=0;i<=nestados;i++)
  {
    for(j=1;j<=nentradas;j++)
    {
      do{
      printf("\n delta[%d][%c]= ",estado_inicial+i,entradas[j]);
      scanf("%d",&delta[estado_inicial+i][j]);
```

```
        if(delta[estado_inicial+i][j]>nestados+estado_inicial-1) printf("\n Estado no valido. Introduzca de nuevo");
        }while(delta[i][j]>nestados+estado_inicial-1);
      }
    }
    printf("\n Introduce la funcion lambda");
    for(i=0;i<=nestados;i++)
    {
      for(j=1;j<=nentradas;j++)
      {
        do{
        printf("\n lambda[%d][%c]= ",estado_inicial+i,entradas[j]);
        scanf("%c",&lambda[estado_inicial+i][j]);
        esta=pertenece(lambda[estado_inicial+i][j], salidas,nsalidas)
        if(esta==0) printf("\n Elemento no valido. No pertenece al conjunto de salidas. Introduzca de nuevo");
        }while(esta==0);
      }
    }
    do{
      escribir_delta(delta, nestados, nentradas, estado_inicial, entradas);
      printf("\n Deseas modificar algun valor de delta (s/n)?: ");
      scanf("%c",&resp);
      scanf("%c",&resp);
      if(resp=='s'||resp=='S')
      {
        printf("\n Indice del estado: ");
        scanf("%d",&i);
        printf("\n Indice de entradas: ");
        scanf("%d",&j);
        do{
        printf("\n Nuevo valor de delta[%d][%c]: ",i,entradas[j]);
        scanf("%d",&delta[i][j]);
        if(delta[estado_inicial+i][j]>nestados+estado_inicial-1) printf("\n Estado no valido. Introduzca de nuevo");
        }while(delta[i][j]>nestados+estado_inicial-1);
      }
    }while(resp=='s'||resp=='S');
    do{
      escribir_lambda(lambda, nestados, nentradas, estado_inicial, entradas);
      printf("\n Deseas modificar algun valor de lambda (s/n)?: ");
      scanf("%c",&resp);
      scanf("%c",&resp);
      if(resp=='s'||resp=='S')
      {
        printf("\n Indice del estado: ");
        scanf("%d",&i);
        printf("\n Indice de entradas: ");
        scanf("%d",&j);
        do{
        printf("\n Nuevo valor de lambda[%d][%c]: ",i,entradas[j]);
        scanf("%c",&lambda[i][j]);
```

```
          esta=pertenece(lambda[i][j], salidas,nsalidas)
          if(esta==0) printf("\n Valor no valido. Introduzca de nuevo");
          }while(esta==0);
       }
   }while(resp=='s'||resp=='S');
   do{
   do{
       printf("\n\n Introduce la longitud de la palabra x");
       scanf("%d",&lpalabra);
       if(lpalabra>35||lpalabra<1) printf("\n Longitud no valida. Introduzca de nuevo");
   }while(lpalabra>35||lpalabra<1);
   printf("\n Introduzca las letras de la palabra");
   for(i=1;i<=lpalabra;i++)
   {
      do{
      printf("\n x[%d]=";i);
      scanf("%c",&palabra[i]);
      esta=pertenece(palabra[i], entradas,nentradas);
      if(esta!=0) indpalabra[i]=esta;
      else printf("\n Letra no valida. Introduzca de nuevo");
      }while(esta==0);
   }
   do{
   do{
       printf("\n Introduce el estado de partida");
       scanf("%d",&estado_partida);
       if(estado_partida>estado_inicial+nestados-1||estado_partida<estado_inicial)    printf("\n
Estado no valido. Introduzca de nuevo");
   }while(estado_partida>estado_inicial+nestados-1||estado_partida<estado_inicial);
   delta_tabla(indpalabra,lpalabra,estado_partida,suc_delta,delta);
   lambda_tabla(indpalabra,lpalabra,suc_lambda,suc_delta,lambda);
   printf("\n La tabla de computacion viene dada por:\n\n");
   escribir_caracteres(palabra, lpalabra);
   escribir_enteros(suc_delta,lpalabra+1);
   escribir_caracteres(suc_lambda, lpalabra);
   printf("\n Quiere cambiar de estado manteniendo la palabra?(s/n)");
   scanf("%c",&resp);
   scanf("%c",&resp);
   }while(resp=='s'||resp=='S');
   printf("\n Quiere cambiar de palabra?(s/n)");
   scanf("%c",&resp);
   scanf("%c",&resp);
   }while(resp=='s'||resp=='S');
   printf("\n Fin de programa.");
   printf("\n Pulsa una tecla para finalizar");
   scanf("%c",&resp);
   scanf("%c",&resp);
}
```