

***Reporte de práctica : transductor complemento a doses***

***Nombre: Edgar Celestino Larios Amezcua***

***Materia: Teoría de la computación.***

***Programa para simular una maquina de Turing (MT) que transforma una cadena binaria en longitud arbitraria en su complemento a doses. Donde leerá una cadena ingresada por el usuario y mostrará el contenido de cinta paso a paso hasta que la maquina pare reescribiendo la misma cinta.***

***Para realizar el complemento a doses es necesario hacer el complemento a unos a la cadena binaria w= b1,b2,...,b3 ingresada por el usuario ya que deberá cambiar sus valores de la misma cadena w=b1=0 w’=b’1=1 y si w=b1=0 entonces w’=b’=1 un ejemplo es si w = 10010110 entonces w ′= 01101001 , mostrando en pantalla el proceso de cómo va cambiando la cinta con el cabezal. Esta máquina se llamara Mcomp.***

***Ya que se realice el complemento a unos se realiza el complemento a doses que consiste en sumarle el valor 1 a la cadena w’ , donde también utiliza un elemento adicional ‘’c’’ llamado acarreo, que indica con un 1 si hay desbordamiento o 0 para el caso contrario.*** ***Por ejemplo, si w = 10010110 entonces***

***w ′ = w ′ + 1 = 01101010 y c = 0. Por otro lado, si w = 00000000 entonces w ′ = w ′ + 1 = 11111111 + 1 00000000 con c = 1. Esta máquina se llamara Minc.***

***Su transductor entonces será M= Mcomp Minc.***

***Se usara el carácter espacio (‘ ‘) para representar los blancos de la cadena, también se usara el carácter punto (‘ . ’) para representar la posición del cabezal en la cadena.***

***Para llevar a cabo estos procedimientos se hizo el uso de maquinas como R, L, Rx, Lx y A.***

***Se represento a cinta por medio de un mapeo de la forma std::map<int,char> cinta;***

***Para de esta manera relacionar algún valor tipo int) con el contenido de la casilla correspondiente (algún valor tipo char). Donde también se lleva el registro de los blancos de la cadena w que son las casillas visitadas mas a la izquierda y mas a la derecha.***

***Para operar sobre la cinta se implementaron tres funciones básicas:***

***char write(char c):***

***Esta función escribe el carácter c en la casilla sobre la cual se encuentra***

***posicionado actualmente el cabezal de lectura. Además devuelve como resultado el***

***mismo carácter c.***

***char write(char c){***

***return cinta[cle]= c;***

***}***

***char read():***

***Esta función lee la casilla sobre la cual se encuentra posicionado actualmente***

***el cabezal de lectura.***

***char read(){***

***return cinta[cle];***

***}***

***bool move(char a, char b, int d):***

***Esta función implementa una operación de transición: lee la letra a de la posición actual del cabezal (función read), escribe la letra b sobre la misma casilla (función write) y desplaza el cabezal una posición a la derecha/ izquierda (actualización –incremento ó decremento– del índice de posición i), según lo señale la variable d. Además, deberá retornar el valor true o false para informar si la operación de transición tuvo éxito o no, respectivamente.***

***bool move(char a, char b, int d){***

***if(read()==a){***

***write(b);***

***if(d==1){***

***cle++;***

***}***

***else if(d==0){***

***cle--;***

***}***

***return true;***

***}***

***else{***

***return false;***

***}***

***}***

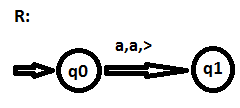
***R:***

***tiene la función de mover el CLE (cabezal) una posición a la derecha, reescribiendo la celda actual***

***void R(){***

***move(read(), read(), 1);***

***}***

******

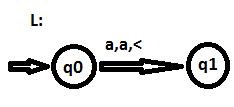
***L:***

***tiene la función de mover el CLE (cabezal) una posición a la izquierda, reescribiendo la celda actual.***

***void L(){***

***move(read(),read(),0);***

***}***

******

***Rx:***

***busca el primer símbolo x que se localice a la derecha de la posición actual.***

***void Rx(char a){***

***if(read()=='\0'){***

***cinta[cle]=' ';***

***}***

***while(read()!= ' '){***

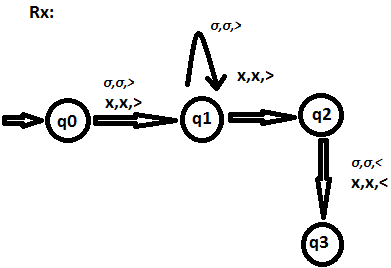
***move(read(), read(), 1);***

***if(read()=='\0'){***

***cinta[cle]=' ';***

***}***

***} }***

******

***Lx:***

***busca el primer símbolo x que se localice a la izquierda de la posición actual.***

***void Lx(char a){***

***if(read()=='\0'){***

***cinta[cle]=' ';***

***}***

***while((read()=='1')||(read()=='0')){***

***move(read(), read(), 0);***

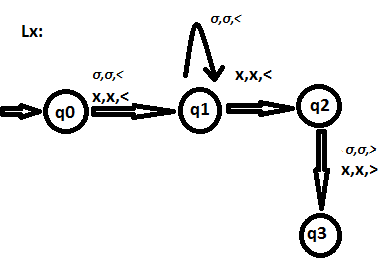
***if(read()=='\0'){***

***cinta[cle]=' ';***

***}***

***}***

***}***

******

***A:***

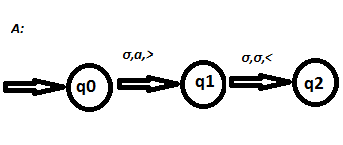
***escribe un símbolo en la posición actual sin desplazar el CLE.***

***void A(char a){***

***move(read(), a, 1);***

***move(read(), read(), 0);***

***}***

******

***Mcomp:***

***Esta máquina realiza los cambios en la cinta para complemento a unos donde primero se busca la posición más a la izquierda para encontrar el blanco y guardar un carácter espacio (‘ ‘) con Lx y después incrementar el CLE con la maquina R una posición a la derecha y mientras haya un carácter diferente de espacio (‘ ’) si encuentra un uno lo cambie por cero y si es cero lo cambie por un uno y avance la posición del CLE de nuevo por cada carácter que cambie de la cadena w.***

***char MComp(){***

***Lx('\0');***

***leerCinta();***

***R();***

***while(read()!= ' '){***

***if(read()== '1'){***

***A('0');***

***leerCinta();***

***R();***

***}***

***if(read() == '0'){***

***A('1');***

***leerCinta();***

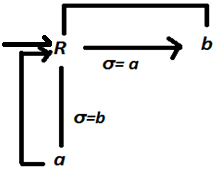
***R();***

***}***

***}***

***leerCinta();***

***}***

******

***MInc:***

***Se avanza una posición a la izquierda y si encuentra un uno lo cambie por cero y entre a otra condición que si vuelve a ser todo el tiempo el valor de la cadena uno, al aumentar CLE guardara un acarreo de valor 1, pero si después de encontrar un uno encuentra un cero o simplemente entrar con un cero, lo cambiara por valor uno y los demás valores de la cadena w se dejaran igual, teniendo un acarreo de valor cero.***

***void MInc(){***

***leerCinta();***

***L();***

***if(read()=='1'){***

***A('0');***

***leerCinta();***

***L();***

***if(read()=='1'){***

***while(read()== '1'){***

***A('0');***

***leerCinta();***

***L();***

***}***

***}***

***if(read()== ' '){***

***leerCinta();***

***L();***

***A('1');***

***leerCinta();***

***}***

***}***

***if(read()== '0'){***

***A('1');***

***leerCinta();***

***L();***

***while((read()=='0') || (read()== '1')){***

***if(read()== '1'){***

***A('1');***

***leerCinta();***

***L();***

***}***

***if(read()== '0'){***

***A('0');***

***leerCinta();***

***L();***

***}***

***}***

***if(read()== ' '){***

***leerCinta();***

***L();***

***A('0');***

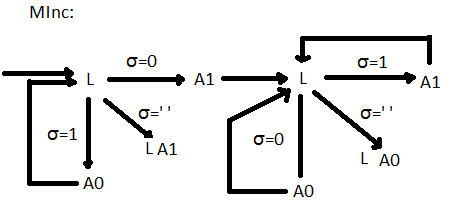
***leerCinta();***

***}***

***}***

***cout<<endl;***

***}***

******

***leerCinta:***

***se encarga de mostrar los valores almacenados de la cinta y la posición del cabezal. Es utilizada para mostrar el procedimiento de la máquina de complemento a unos (MCom) y la máquina de complemento a doses (MInc).***

***void leerCinta(){***

***int j=-1;***

***if((cinta[j-1]=='1')||(cinta[j-1]=='0')){***

***cout<<"."<<cinta[j-1];***

***}***

***if((read()==' ')&&(cle==-1)){***

***cout<<"."<<cinta[j++];***

***}***

***else{***

***cout<<cinta[j++];***

***}***

***while((cinta[j] == '0')||(cinta[j] == '1')){***

***if(j == cle){***

***cout<<"."<<cinta[j++];***

***}***

***if(j != cle){***

***cout<<cinta[j++];***

***}***

***}***

***if((read()==' ')&&(cle>0)){***

***cout<<"."<<cinta[j++];***

***}***

***cout<<endl;***

***}***

***insertandoCadena:***

***esta funciona guarda la cadena que es ingresada por el usuario.***

***void insertandoCadena( char \*z){***

***while(z[cle] != '\0'){***

***write(z[cle]);***

***R();***

***}***

***Rx('\0');***

***cout<<endl;***

***cout<<endl;***

***cle=0;***

***}***

***M:***

***Tiene la función de hacer que se ejecuten las maquinas de complemento a unos y la de complemento a doses.***

***void M(){***

***insertandoCadena(cadena);***

***cout<<"complemento a unos: ";***

***cout<<endl;***

***MComp();***

***cout<<endl;***

***cout<<"complemento a 2's ";***

***cout<<endl;***

***MInc();***

***cout<<endl;***

***cout<<"la operacion de transicion tuvo exito"<<endl;}***