****
TAREA DE UN ALUMNO, EN EL RANGO: LO SUPERA; TAREA DE 10

Algoritmo Thompson

Resultados obtenidos del programa:

```
Ingrese su Expresion Regular - ((a.b)|a)

El automata finito no deterministico tiene las
siguientes transiciones :

q0 a q1 : Simbolo - E
q1 a q2 : Simbolo - a
q2 a q3 : Simbolo - E
q3 a q4 : Simbolo - b
q4 a q7 : Simbolo - E
q0 a q5 : Simbolo - E
q5 a q6 : Simbolo - a
q6 a q7 : Simbolo - E
```

```
Ingrese su Expresion Regular - a*

El automata finito no deterministico tiene las
siguientes transiciones :

q0 a q1 : Simbolo - E
q1 a q2 : Simbolo - a
q2 a q3 : Simbolo - E
q2 a q1 : Simbolo - E
q0 a q3 : Simbolo - E
```

```
Ingrese su Expresion Regular - (a|b*)


El automata finito no deterministico tiene las
siguientes transiciones :


q0 a q1 : Simbolo - E
q1 a q2 : Simbolo - a
q2 a q7 : Simbolo - E
q0 a q3 : Simbolo - E
q3 a q4 : Simbolo - E
q4 a q5 : Simbolo - b
q5 a q6 : Simbolo - E
q5 a q4 : Simbolo - E
q3 a q6 : Simbolo - E
q6 a q7 : Simbolo - E
```

Algoritmo Subconjuntos

Resultados obtenidos del programa:

```
Numero de estados en Automata No determinista : 5

Estado inicial: 0
Numero de estados finales : 1

Estados finales :
4

Numero de simbolos : 2

Que simbolos son :
 a b

Ingrese transiciones : (-1 para parar)
mov(0,a) : -1
mov(0,b) : -1
mov(0,e) : 1
mov(0,e) : -1
mov(1,a) : 2
mov(1,a) : 3
mov(1,a) : -1
mov(1,b) : -1
mov(1,e) : -1
mov(2,a) : -1
mov(2,b) : 3
mov(2,b) : -1
mov(2,e) : 1
mov(2,e) : -1
mov(3,a) : 4
mov(3,a) : -1
mov(3,b) : -1
mov(3,e) : 2
mov(3,e) : 4
mov(3,e) : -1
mov(4,a) : -1
mov(4,a) : -1
mov(4,b) : -1
mov(4,e) : -1

         Tabla de transicion Automata Deterministico    a   b
     ---------
A    01  B    C
*B   1234     B    B
C        C    C
```

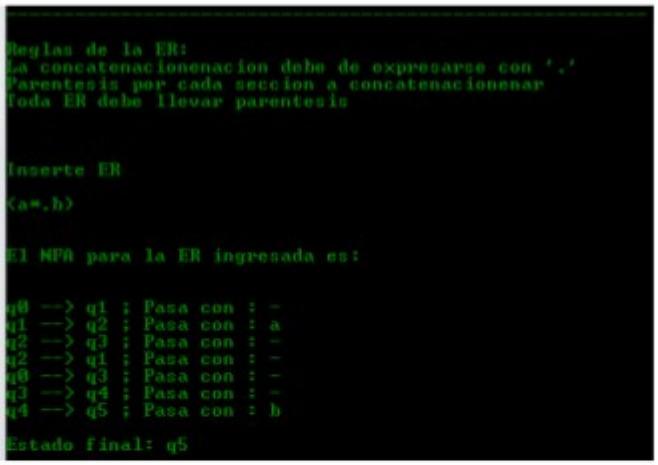
****

ESTE ALUMNO LO LOGRA

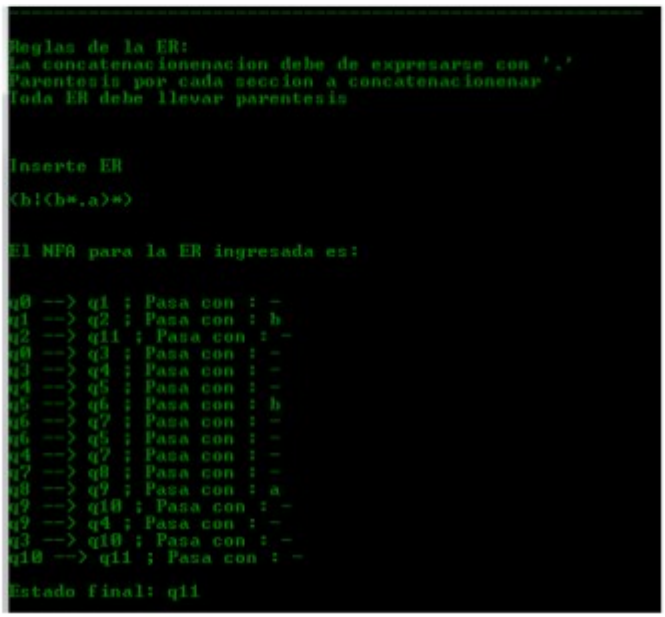
Compiladores

1 Dada la ER, hallar AFN

1.a) a*b

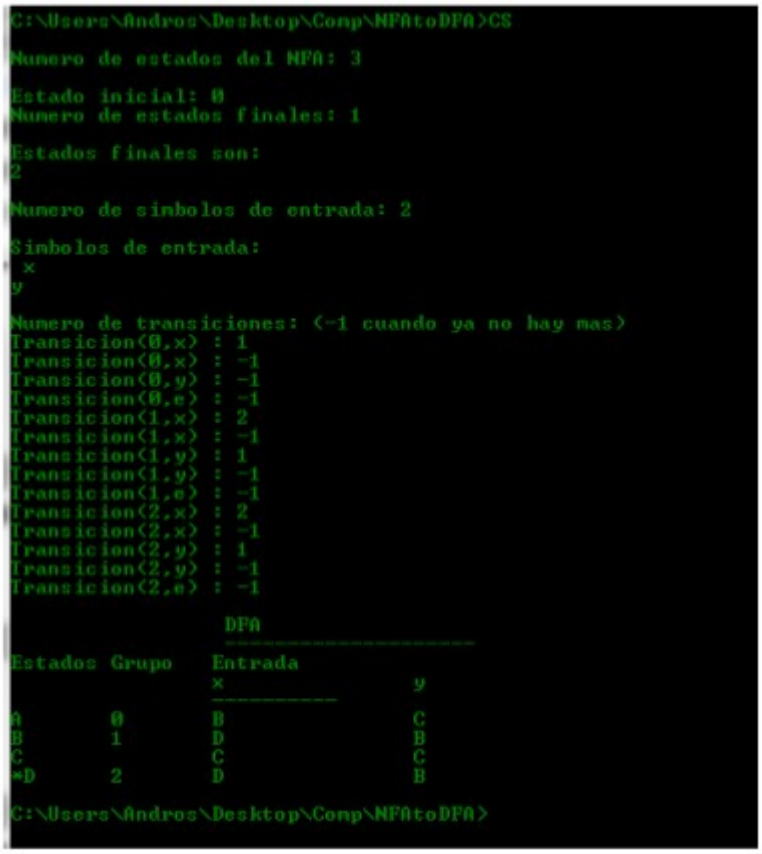


1.b) b|(b*a)*



2 Dado AFN hallar AFD

2.a)



```
C:\Users\Andros\Desktop\Comp\NFAtoDFA>CS

Numero de estados del NFA: 3

Estado inicial: 0
Numero de estados finales: 1

Estados finales son:
2

Numero de sinbolos de entrada: 2

Sinbolos de entrada:
 x
y

Numero de transiciones: (-1 cuando ya no hay mas)
Transicion(0,x) : 1
Transicion(0,x) : -1
Transicion(0,y) : -1
Transicion(0,e) : -1
Transicion(1,x) : 2
Transicion(1,x) : -1
Transicion(1,y) : 1
Transicion(1,y) : -1
Transicion(1,e) : -1
Transicion(2,x) : 2
Transicion(2,x) : -1
Transicion(2,y) : 1
Transicion(2,y) : -1
Transicion(2,e) : -1

                DFA
                ___________________________

Estados Grupo   Entrada
                x           y
                ___________

A       0       B           C
B       1       D           B
C               C           C
*D      2       D           B

C:\Users\Andros\Desktop\Comp\NFAtoDFA>
```

2.b)

```
C:\Users\Andros\Desktop\Comp\NFAtoDFA>CS

Numero de estados del NFA: 4

Estado inicial: 0
Numero de estados finales: 1

Estados finales son:
2

Numero de simbolos de entrada: 2

Simbolos de entrada:
x
y

Numero de transiciones: (-1 cuando ya no hay mas)
Transicion(0,x) : 1
Transicion(0,x) : -1
Transicion(0,y) : -1
Transicion(0,e) : -1
Transicion(1,x) : 2
Transicion(1,x) : -1
Transicion(1,y) : 3
Transicion(1,y) : -1
Transicion(1,e) : -1
Transicion(2,x) : 2
Transicion(2,x) : -1
Transicion(2,y) : 3
Transicion(2,y) : -1
Transicion(2,e) : -1
Transicion(3,x) : 2
Transicion(3,x) : -1
Transicion(3,y) : -1
Transicion(3,e) : -1

                DFA
                ---------------------
Estados Grupo   Entrada
                x           y
                ----------
A       0       B           C
B       1       D           E
C               C           C
*D      2       D           E
E       3       D           C

C:\Users\Andros\Desktop\Comp\NFAtoDFA>
```

****
TAREA DE ALUMNO QUE PARCIALMENTE (1/3) LO LOGRA

```
package thomson;

import java.util.ArrayList;
import java.util.HashSet;


public class Automata {

    private Estado inicial;
    private final ArrayList<Estado> aceptacion;
    private final ArrayList<Estado> estados;
    private HashSet alfabeto;
    private String lenguajeR;

    public Automata()
    {
        this.estados = new ArrayList();
        this.aceptacion = new ArrayList();
        this.alfabeto = new HashSet();

    }


    public Estado getEstadoInicial() {
        return inicial;
    }

    public void setEstadoInicial(Estado inicial) {
        this.inicial = inicial;
    }

    public ArrayList<Estado> getEstadosAceptacion() {
        return aceptacion;
    }

    public void addEstadosAceptacion(Estado fin) {
        this.aceptacion.add(fin);
    }

    public ArrayList<Estado> getEstados() {
        return estados;
    }

    public Estado getEstados(int index){
```

```
    return estados.get(index);
}

public void addEstados(Estado estado) {
    this.estados.add(estado);
}


public HashSet getAlfabeto() {
    return alfabeto;
}

public void createAlfabeto(String regex) {
    for (Character ch: regex.toCharArray()){

        if (ch != '|' && ch != '.' && ch != '*' && ch != AutomataMain.EPSILON_CHAR)
            this.alfabeto.add(Character.toString(ch));
    }
}

public void setAlfabeto(HashSet alfabeto){
    this.alfabeto=alfabeto;
}



public Estado getInicial() {
    return inicial;
}

public void setInicial(Estado inicial) {
    this.inicial = inicial;
}


@Override
public String toString(){
    String res = new String();
    res += "Alfabeto " + this.alfabeto+"\r\n";
    res += "Estado inicial " + this.inicial +"\r\n";
    res += "Conjutos de estados de aceptacion " + this.aceptacion +"\r\n";
    res += "Conjunto de Estados " + this.estados.toString()+"\r\n";
    res += "Conjunto de transiciones ";
    for (int i =0 ; i<this.estados.size();i++){
        Estado est = estados.get(i);
        res += est.getTransiciones()+"-";
    }
    res += "\r\n";
    res += "Lenguaje r: " +this.lenguajeR + "\r\n";
```

```
    return res;
}
```