Autómata Finito No Determinístico (NFA)
Matemáticas Computacionales
ITESM CSF

Estudiantes:
Karol José Gutiérrez Suárez A01024536
Mauricio Peón García A01024162

Funcionamiento del código

Lectura de datos

Dentro del main lo primero que se hace es extraer los datos del archivo txt proporcionado. Para ello es necesario conocer el nombre del archivo y en este caso se eligió nombrarlo como **"nfa.txt"**. Se declara la variable **in** como **ifstream in ("nfa.txt")**, lo que nos permite extraer datos del archivo.
Todos los datos se almacenan en variables globales de tipo **set** y **tuple**. El set **states** almacena los strings con los nombres de todos los estados, el set **alphabet** almacena los strings con todos los caracteres del alfabeto, el set **finals** almacena todos los strings correspondientes a los nombres de los estados finales.
Se utilizan dos strings auxiliares **line** y **entrada** que nos permiten ir agregando elementos al set.
Por ejemplo, el siguiente código lee un string llamado line, lo convierte a un dato de tipo stringstream y con él es posible obtener subcadenas separadas por comas. Estas subcadenas se insertan en el set que le corresponda. En este ejemplo, se insertan al set que contiene los estados.

```
getline(in,line);
        stringstream ss(line);
        while( ss.good() )
        {
            getline( ss, entrada, ',' );//separa los strings usando la coma
            states.insert(entrada);
        }
```

Asimismo, se crea un vector que contiene tuplas de tres strings para indicar las reglas de las transiciones. Dicho vector es llamado **rules** y la forma de añadirle datos es análoga a como se hace con los sets.
Por último, se cierra el ifstream llamado in usando el comando **in.close()**, pues ya hemos terminado de cargar todos los datos.

Función recursiva
Para verificar si el string es una cadena válida en el NFA se usa una función recursiva llamada Solver. Dicha función se declara así:

```
void solver(int pos, string current, set<tuple<int, string>> visited, string path)
```

Sus parámetros son los siguientes:

- pos: posición sobre la cadena s, que es en la que estamos trabajando. La cadena s es una variable global
- current: string con el estado actual
- visited: conjunto que se utiliza para detectar ciclos. Almacena el estado y la posición en un momento dado y más adelante se usa para saber si hemos dado un ciclo de transiciones épsilon.
- path: es un string que va describiendo los pasos a seguir para llegar desde el estado inicial hasta el estado actual

El mecanismo de la función consiste en imprimir el string path una vez que se ha validado que la cadena es aceptada. Para saber esto es necesario que se haya pasado por todos los caracteres de la cadena y que el estado actual pertenezca a un estado final.
En caso de que la cadena no haya sido validada, la función se llama recursivamente para todas las transiciones que parten del estado actual. Hacemos hincapié que se debe verificar que no estemos en un ciclo que pudiera volverse infinito y consideramos el caso en el que la transición sea de tipo épsilon (en tal caso, la variable pos conserva su valor y en caso contrario la variable pos se pasa como pos+1 en la llamada recursiva).

Main
En esta parte se imprime un menú para el programa donde se describen las instrucciones de uso. Más adelante hay un ciclo while donde se piden las cadenas que el usuario desee validar y el programa termina hasta que el usuario introduce la cadena ".".

Uso del programa

Dentro de la misma carpeta donde se encuentre el archivo C++, es necesario colocar el archivo de texto "nfa.txt" el cual debe contener la descripción del NFA en el siguiente formato:

- Primera línea: estados iniciales separados por comas, no debe haber espacios
- Segunda línea: caracteres del alfabeto separados por comas, sin espacios
- Tercera línea: estado inicial
- Cuarta línea: estados finales separados por comas, sin espacios
- Lineas 5 en adelante: cada línea contiene las transiciones en el formato **q0,a:q1** lo cual significa que hay una transición usando el símbolo **a** que lleva de **q0** a **q1**. Para describir las transiciones épsilon se utiliza el caracter **&**. No debe haber espacios.

Se ejecuta el programa usando un compilador de C++, preferentemente GNU GCC.
Una vez ejecutado el programa se abre una ventana donde se pide al usuario ingresar su cadena para que el programa determine si es una cadena aceptada por el NFA.

El usuario presiona enter y se despliega alguno de dos mensajes:

- Si la cadena es aceptada, el programa imprime el texto
  "yourString" is a valid string and the resulting transitions are:
  p0 ---s1--> p1---s2--> p2 … ---sk--> pk

El programa imprime todas las soluciones no primitivas que encuentre, es decir, aquellas que no se puedan obtener al añadir ciclos con transiciones épsilon a otra solución ya existente

- Si la cadena no es aceptada, el programa imprime el texto
  Invalid string.

Para finalizar el programa, se introduce la cadena ".".