

AUTÓMATAS Y LENGUAJES

PRÁCTICA 1

En esta práctica hemos desarrollado una librería en c para poder simular la definición de un autómata finito no determinista sin transiciones λ .

En primer lugar hemos implementado `alfabeto.h` y `alfabeto.c`, cuya estructura `Simbolo` está compuesta por un `char*`, que es el nombre que recibe el símbolo, y un `int` que representa el tamaño.

La estructura `Palabra` tiene otro `char*` con el nombre, un `char**` que contiene símbolos, un `int` con el tamaño y otro con el número de símbolos del alfabeto.

Luego el módulo `estado` tiene una estructura de tipo `Estado` compuesta de un `char*` que tiene el nombre y un `int` que describe el tipo de estado. Luego el módulo `palabra` tiene un `char**` que es la letra y un `int` con el tamaño de la palabra.

Finalmente el módulo `afnd` que es el autómata que tiene una estructura llamada `afnd` que contiene un `char *` representando al nombre, un alfabeto de tipo `Alfabeto*`, un `int` que representa el numero de estados, otro con el número de símbolos y finalmente un `Estado**` donde se almacenan los estados y la cadena actual almacenada como `Palabra*`.

Según abrimos la carpeta principal de nuestra práctica, encontramos el `makefile` que compilaremos para probar nuestros módulos. En la carpeta `lib` tenemos todos los códigos de los diferentes módulos (tanto los ficheros `.h` como los `.c`)

Una vez que hacemos *make all-test* desde la carpeta raíz, se nos generan los ejecutables de los ficheros *nombremodulo-test.c* que están en `/lib`, que con `valgrind -leak-check=full ./pruebas_nombremodulo` lo ejecutamos y comprobamos su funcionamiento.