



MANUAL TÉCNICO BASILISK

Eduardo Ixen

INDICE

INDICE	1
CLASE main	2
Función main():	2
Función cargarArchivo():	2
Funcion manejoAFD():	2
Funcion AP():	3
Funcion quitTerminal(pila, listaTK):	3
Funcion quitNT():	3
Funcion cambioNT():	3
Funcion mostrarTrancion(pila, trancion):	3
Funcion diagrama():	4
Función sumNivel():	4
CLASE CargaArchivo	4
Función cargarArchivo(ruta_archivo):	4
CLASE Pila()	4
CLASE Reporte	5
Función crearHtml():	5

CLASE main

Función main():

Esta función se encarga de presentar y controlar el menú de opciones al usuario, dependiendo de la opción seleccionada se llama a diferentes funciones.

Las opciones que proporciona son:

- Cargar script: Esta opción llama a la clase cargararchivo() para cargar un archivo a memoria.
- Manejo AFD: Esta opción llama a la función ManejoAFD() y se encarga de escanear el código en busca de errores léxicos, también verifica si hay algún archivo cargado previamente de lo contrario muestra un mensaje de error.
- Pila interactiva: Esta opción se encarga de verificar que el contenido del archivo cargado a memoria este escrito en el orden correcto. Primero se verifica que haya un archivo cargado previamente de lo contrario se muestra un mensaje de error, luego se llama a la función manejoAFD() para generar una lista de tokens válidos y por último se llama a la función AP() que se encarga de verificar el orden.
- Diagrama: Esta opción genera un diagrama a partir de un script cargado a memoria, primero se verifica que se haya cargado un archivo de lo contrario se muestra un mensaje de error, luego se llama a la función manejoAFD() que genera una lista de tokens válidos y por último se llama a la función diagrama() que se encarga de crear una imagen a partir de la lista de tokens válidos.
- Salir: Esta opción termina la ejecución del programa.

Función cargarArchivo():

Esta función le solicita al usuario ingresar la ruta del archivo a cargar, enviándoselo como parámetro a la clase CargaArchivo.

Función manejoAFD():

Esta función utiliza un autómata finito determinístico para analizar el contenido del archivo cargado a memoria, recorre el archivo carácter por carácter concatenando hasta encontrar una palabra reservada o identificador que pertenezcan al lenguaje definido. Al finalizar el escaneo genera dos listas uno de reportes con tokens válidos y otro con tokens inválidos que son mandados como parámetros a la clase Reporte() para crear los reportes en HTML.

Función AP():

Esta función hace uso de una gramática definida por el proyecto que indica el orden en el que debería estar escrito el código contenido en el archivo cargado a memoria. Para verificar este orden se utiliza un autómata de pila que necesita de tres componentes para su funcionamiento (una entrada, una pila y una transición) se verifica que lo que este en la cima de la pila es un terminal o un no terminal, si es terminal se elimina de la pila y se inserta la producción de ese terminal con la función quitTerminal(), si es no terminal se elimina la entrada y también se elimina de la pila con la función quitNT().

Este proceso se repita hasta vaciar la pila o hasta vaciar la lista de entrada, si todo es correcto se imprime un mensaje indicando que la entrada es válida y se genera un reporte en HTML, de lo contrario se muestra un mensaje indicando que la entrada es invalida.

Función quitTerminal(pila, listaTK):

Esta función recorre todas las producciones de los no terminales definidas en la gramática del proyecto hasta encontrar una donde la entrada coincida con lo que está en la cima de la pila, al encontrarlo se elimina de la lista y de la pila.

Función quitNT():

Esta función analiza cual es el no terminal que se está recibiendo y busca entre todas las producciones definidas por la gramática una que coincida con el no terminal recibido y la entrada recibida, al encontrarla se elimina el no terminal de la pila y se inserta la producción encontrada en orden inverso.

Función cambioNT():

Esta función elimina e inserta elementos dentro de una pila, diferenciando si se insertará una lista o una cadena simple, también llama a la función mostrarTrancicion() para mostrar el proceso en consola.

Función mostrarTrancicion(pila, trancicion):

Esta función recibe como parámetro una pila y una transición para mostrar el proceso de eliminación e inserción en la pila. Se utiliza la librería tabulate para presentar la información en forma de tabla.

También genera una lista con todas las transiciones mostradas para crear un archivo HTML con una copia de lo mostrado en consola.

Función diagrama():

Esta función hace uso de la lista generada por la función manejoAFD() y crea un diagrama utilizando la librería graphviz. Recorre token por token encontrando el nivel en el que se encuentra cada definición, función o declaración, asignándoles padres para construir una secuencia.

Función sumNivel():

Esta función crear una lista de niveles que es utilizada por la función diagrama() para crear la imagen de secuencia del código.

CLASE CargaArchivo

Función cargarArchivo(ruta_archivo):

Esta función verifica si la ruta ingresada por el usuario existe de ser así el archivo se abre y se lee todo el contenido almacenándolo en una variable a la que puede ser accedida por todo el proyecto, de no encontrarse se muestra un mensaje de error.

CLASE Pila()

Esta clase contiene varios métodos que facilitan la interacción con los elementos de la pila, son los siguientes:

- isEmpty(): Retorna true o false dependiendo si la pila se encuentra vacía o no.
- push(item): Inserta el ítem recibido como parámetro en la cima de la pila.
- Pop(): Elimina el último elemento insertado en la pila.
- obtenerUltimoAgregado(): Devuelve el ultimo elemento agregado a la pila sin eliminarlo.
- getLength(): Devuelve el tamaño (cantidad de elementos) de la pila.
- getItems(): Devuelve todos los elementos contenidos en la pila.
- Extend(): inserta una lista a la pila de forma inversa (El último se inserta primero y así sucesivamente).

CLASE Reporte

Función crearHtml():

Esta función crea un archivo HTML a partir de una lista recibida como parámetro utilizando la librería tabulate que genera una tabla html a partir de una lista.

Al obtener la cadena html se escribe en un archivo con extensión .html y se le pregunta al usuario si se desea abrir o simplemente guardar. Los reportes generados se almacena en la carpeta “Reportes” del proyecto.