

Proyecto: Analizador Léxico

Importante:

Antes de leer este documento, por favor lea el documento "Aplicaciones de los AFD"

Objetivo:

Elaborar un programa de computador con las siguientes características:

- Entrada: un archivo de texto, con un programa fuente en un lenguaje inventado por Ustedes. El programa podrá tener errores de sintaxis.
- Salida: un listado de tokens del programa de entrada.

Ejemplo de Entrada

Un archivo de texto con las siguientes líneas:

```
programa suma1N
comienzo
  n := @1;
  suma := ;
  repetir
    n := n + 1;
    suma := suma + n;
  hasta n = 100;
```

```
    imprimir suma;
fin
```

Ejemplo de salida

La salida correspondiente al anterior ejemplo es la siguiente:

Token	Tipo
programa	identificador
suma1N	identificador
comienzo	identificador
n	identificador
:=	operador de Asignación
@	Error
1	Entero
;	Punto y coma
suma	identificador
:=	operador de asignación
;	Punto y coma
repetir	identificador
...	etcétera.

Note que el único error que debe detectar su programa es el de que el tipo de un token es inválido. Los demás errores, como variable no declarada, falta punto y coma, falta la expresión en la instrucción de asignación, no están al alcance de este proyecto y se tratarán el próximo semestre en la asignatura compiladores.

Requerimientos

- El proyecto se podrá realizar en el lenguaje de computador de su preferencia: C++, Java, Builder etc.
- Deberán inventar un lenguaje de computador con lo básico:
 - Instrucciones de asignación
 - Estructura de decisión (si ... entonces)
 - Bucle con test al comienzo

- Bucle con test al final
- Bucle predeterminado (para)
- Instrucciones de salida por pantalla
- Instrucción de entrada por teclado.

Puede ser un lenguaje multipropósito o uno especializado para, por ejemplo, sitios web o programar un robot.

- El programa DEBE incluir la función *siguienteToken*, similar basada en un AFD. Recordemos que esta función retorna un token y su tipo cada vez que se ejecuta. Ejecutándola varias veces, se extraen todos los tokens.

Al ser *siguienteToken* una función basada en un AFD, cada ruta del AFD debe reflejarse como un si ... entonces y cada bucle del AFD debe reflejarse como un bucle mientras o de otro tipo.

- Deberá entregarse la gramática BNF del lenguaje que inventaron. Este tema se verá más adelante.
- Debe elaborarse un instalador (setup.exe) para el programa, elaborado con algún software para creación de instaladores. Este instalador debe copiar en el disco duro tanto el código fuente como el código compilado.
- Debe entregarse el dibujo del AFD en formato de computador. Por ejemplo en un archivo .JPG. Por claridad se recomienda omitir estados de rechazo, de donde ya no pueda escapar el AFD y las transiciones correspondientes, como en el ejemplo acerca de este tema.

Evaluación

En la evaluación se tendrá en cuenta:

- La corrección del software: Que el programa produzca los resultados pedidos en forma correcta.
- La usabilidad del software: Que el programa sea sencillo y cómodo de usar, incluso para alguien que lo utilice por

primera vez. La presentación de la pantalla puede ser sencilla, pero clara y ordenada.

- La facilidad de compresión y de modificación del software: Que, gracias a estándares de programación usados y comentarios, el programa sea fácil de entender y modificar, incluso para programadores que no lo realizaron.
- El Aporte de cada uno de los estudiantes al proyecto.

Entrega

Miércoles 5 de noviembre (Semana 15).

Se descontará una unidad por cada día de retraso.

De ser necesario se podrá ampliar el plazo al curso en general. No se ampliará el plazo a grupos particulares de estudiantes. El plazo siempre se mantendrá igual para todos los grupos de proyecto.