

Práctica 2

Un analizador léxico para “miLenguaje”

1. Objetivos

1. Establecer e identificar cuáles son los tokens fundamentales para un lenguaje imperativo sencillo.
2. Diseñar un analizador léxico para dicho lenguaje.
3. Manejar la herramienta Flex, un generador de analizadores léxicos.

2. Contenidos

En esta práctica se propone el desarrollo de un analizador léxico para el lenguaje definido en la práctica 1. Este lenguaje, denominado “miLenguaje”, será manejado durante el resto de prácticas.

“miLenguaje”, como ya se ha dicho en la práctica 1, tiene como características principales las que se enumeran a continuación. Si bien el lenguaje no es muy rico, sus elementos son suficientes para trabajar con los conceptos fundamentales del diseño de un traductor.

La práctica pide:

1. Definir los tokens para “miLenguaje”. Esto se puede llevar a cabo mediante la definición de constantes o bien mediante el uso de un tipo enumerado. Tener presente que, cuando se implemente el traductor completo, estas definiciones las generará Bison.
2. Construir un analizador léxico para el lenguaje. De momento se trata únicamente de realizar el análisis léxico. Cuando el analizador reconozca

un token, se deberá escribir por la salida *standard* información que permita identificar el token reconocido, así como su localización (de la manera que se muestra más adelante). El nombre del fichero a analizar, el fuente, se debe suministrar como un parámetro en la invocación al analizador.

3. Respecto a los errores léxicos, el analizador construído debe al menos tratar los siguientes errores léxicos:

- Dar un AVISO (warning) o un FATAL (error no recuperable) cuando detecte un error léxico, indicando el tipo de error detectado y la fila y columna en que se ha detectado. Un aviso informa por *stderr* sobre el tipo de error encontrado y aplica una estrategia de recuperación de errores, siguiendo con el análisis. Un error FATAL supone la salida del analizador, además de informar por *stderr* de la naturaleza del error.
- Estrategias de recuperación en el caso de avisos:
 - En el caso de encontrar un carácter extraño, saltarlo, interpretando que se debía tratar de un espacio en blanco.
...#... AVISO! (20, 15) Carácter extraño.
 Sustituyendo por "blanco".
 Lexema: # (En cada caso el que corresponda)
 - En el caso de detectar un salto de línea, un tabulador o el fin de fichero una vez que se haya abierto una cadena, dará el siguiente aviso:
"... AVISO! (20, 15)
 Cadena no terminada. Inserto fin de Cadena.
- En el caso de no existir el fichero fuente, parar el análisis, indicar que dicho fichero no existe.

```
FATAL! Fichero no encontrado.  
Fichero: miFuente.ml (En cada caso el que  
corresponda)
```

3. Resultados

Como resultado de esta práctica hay que entregar el fichero denominado `pract2.tar`. Este fichero se obtendrá a partir del comando “tar” de UNIX (ejecutar `man tar` para tener información sobre él), de manera que la ejecución de la instrucción “tar -xvf pract2.tar” genere un directorio denominado `pract2` con los siguientes ficheros:

- Fichero `AL_miLenguaje_1.1` que contiene el fichero fuente Flex del analizador léxico.
- Fichero `AL_miLenguajeMakefile` que contiene el fichero Make para generar el ejecutable del analizador léxico, de manera que la invocación

```
make -f AL_miLenguajeMakefile
```

genere el ejecutable `AL_miLenguaje` que realiza lo que se indica a continuación. Supongamos que hay un fichero fuente en `miLenguaje` denominado `miFuente.ml`. La invocación

```
AL_miLenguaje miFuente
```

debe escribir en la salida *standard*, en distintas líneas, la información relativa a los tokens reconocidos:

```
(1, 1):      IDENTIFICADOR miPrimerPrograma
             .....
(125, 17):  +
             .....
```

donde (125, 17) indican la fila y la columna del inicio del lexema “+” y lo que sigue es cualquier información que sirva para ver que se ha identificado el token correspondiente.

4. Notas

Como paso previo a la corrección de las prácticas se ejecutará de manera automática un script que comprobará si lo entregado por el alumno como resultado de la práctica es exactamente lo que se pide (ejecución de la descompresión mediante “tar” y comprobación de que se han generado los dos ficheros pedidos). Si esto no es así, la práctica aparecerá como no presentada. En caso de dudas respecto a lo que se tiene que generar, consultar con el profesor.

5. Fecha limite para entrega de resultados

Esta práctica se revisará durante el desarrollo de la tercera sesión de laboratorio.