

Ejercicios sesión 1 de laboratorio PLG

Albert Rubio

7 de febrero de 2020

Ejercicio 1: *Instalar JLex.* En el Campus Virtual de la asignatura en el apartado de “Código ejemplo y herramientas” encontraréis un enlace para obtener *JLex* y otro para obtener “Código ejemplo sobre analizadores léxicos”.

Instalad JLex en vuestra cuenta usando el archivo “.jar”. Podéis utilizar el entorno para desarrollo de código Java que prefiráis. Deberéis definir los caminos necesarios en el CLASSPATH.

Para procesar el archivo `ejemplo.1` que encontraréis en la carpeta con el código para analizadores léxicos con JLex, tenéis que ejecutar:

```
$ java JLex.Main ejemplo.1
```

Esto generará un archivo `ejemplo.1.java` cuyo contenido debería coincidir con el archivo `AnalizadorLexicoTiny.java` del mismo directorio (si no es el caso, borrado este último y reemplazadlo por `ejemplo.1.java` renombrado). Compilad el código java generado y ya podéis usar el analizador. En la versión del `Main` que se adjunta, se lee de la entrada estándar (podéis modificarlo para que use un fichero que se pasa por parámetro). Probad con la entrada

```
evalua
166.386 * euros + 1.66386 * (centimos1 + centimos2)
donde
euros = 567,
centimos1 = 456,
centimos2 = 10
```

Una opción es guardarla en un archivo `entrada.txt` y ejecutar

```
$ javac alex.Main < alex/entrada.txt
```

desde el directorio superior.

Ejercicio 3: *Variaciones.* Realizad pequeños cambios en la especificación JLex y el código asociado. Por ejemplo, podemos ampliar la representación de números para admitir números en binario y hexadecimal con la notación `0b01101101` para binarios y `0x1faCd5` para hexadecimales.

Ejercicio 3: *Patrones.* Suponemos que nos pueden entrar palabras (formadas solo por letras mayúsculas y minúsculas), patrones (formadas por letras mayúsculas y minúsculas y un carácter ‘*’) y cualquier otra cadena de caracteres (sin separadores). Dada una

entrada, debemos extraer la palabras, los patrones y el resto de cadenas como unidades léxicas distintas (PALABRA, PATRON y OTRO) y para cada patrón indicar el número de palabras que le encajan: palabras que coinciden con el patrón reemplazando el '*' por una cadena de caracteres. Así, con la entrada

```
hola
ho*a
laz+dfg
ho*
ho*+sd
pista
+fdsghf
hoja
a*a
hoyo
ala
a
```

la salida debería ser

```
[clase:PALABRA,filas:1,col:0,lexema:hola]
[clase:PATRON,filas:2,col:0,lexema:ho*a]
[clase:OTRO,filas:5,col:0,lexema:laz+dfg]
[clase:PATRON,filas:3,col:0,lexema:ho*]
[clase:OTRO,filas:6,col:0,lexema:ho*+sd]
[clase:PALABRA,filas:7,col:0,lexema:pista]
[clase:OTRO,filas:8,col:0,lexema:+fdsghf]
[clase:PALABRA,filas:9,col:0,lexema:hoja]
[clase:PATRON,filas:4,col:0,lexema:a*a]
[clase:PALABRA,filas:10,col:0,lexema:hoyo]
[clase:PALABRA,filas:11,col:0,lexema:ala]
[clase:PALABRA,filas:12,col:0,lexema:a]
[clase:EOF,filas:13,col:0]
PALABRAS:
hola
pista
hoja
hoyo
ala
a
PATRONES:
ho*a: 2
ho*: 3
a*a: 1
```