

Práctica 1B

Introducción al manejo de *Flex*

Tareas

1. Lee la introducción de esta práctica y realiza los ejercicios 1 a 5 propuestos.
2. Elabora la memoria de la práctica y entrégala junto con los ficheros fuente según el Procedimiento de Entrega de Prácticas explicado en la Introducción a las Prácticas de la Asignatura. La fecha tope de entrega será hasta el día anterior al comienzo de la Práctica 2.

Nota: El incumplimiento de las normas de entrega se reflejará en la calificación de la práctica.

Introducción

El objetivo principal de esta práctica de la asignatura es familiarizarse con la herramienta de creación de analizadores léxicos *Flex* y **aprender aspectos básicos sobre teoría de lenguajes y expresiones regulares**. Para ello, se propone la creación con dicha herramienta de una serie de pequeños procesadores de texto.

El cuadro 1.1 muestra algunas correspondencias entre la notación empleada en clase de teoría y la que se utiliza en *Flex* para trabajar con Expresiones Regulares (ERs).

| Operación | ERs Teoría | ERs Flex |
|------------------------|-----------------------|------------------------------|
| Concatena ERs | \cdot ó {vacío} | {vacío} (yuxtaposición) |
| Unión de ERs | + | |
| Cerradura de Kleene | * | * |
| Cerradura Positiva | + | + |
| Paréntesis | () | () |
| Símbolos a,b,c,d,0,1,2 | {a, b, c, d, 0, 1, 2} | [abcd012] ó [a - d0 - 2] |
| Cadena vacía ó a | $\epsilon + a$ | a? |
| Cadena vacía | ϵ | Sin equivalencia (ver * ó ?) |
| Conjunto vacío | \emptyset | Sin equivalencia |

Cuadro 1.1: Correspondencia entre notaciones

Ejercicio 1

Escribe un programa con *Flex* que elimine los *caracteres blancos* al final de cada línea del texto de entrada y reduzca a un solo *carácter blanco* cualquier grupo de éstos en el interior de cada línea. Se entiende por *carácter blanco* el espacio en blanco y el carácter de tabulación. Ejemplo:

Entrada:

```
Este es    un  texto <EOL>
con      varios blancos <EOL>
<EOF>
```

Salida:

```
Este es un texto <EOL>
con varios blancos <EOL>
<EOF>
```

Ejercicio 2

Elabora un programa en *Flex* que copie el archivo de entrada en uno de salida, sustituyendo todo número entero positivo múltiplo de 7 por el carácter '*'.

- Construye una ER para leer números en decimal, y después en la acción asociada en Flex comprueba si éstos son múltiplos de 7.
- ¿Se puede construir una ER que reconozca los números decimales múltiplos de 7? ¿Cómo sería la expresión?

Ejercicio 3

Construye un programa en *Flex* que cuente el número de palabras del texto de entrada que contengan alguna vez el carácter 'w'. ¿Cómo sería la ER? ¿Y cómo sería la ER

si queremos identificar las palabras que contengan un solo carácter 'w'? Completa el programa anterior para reconocer ambas ERs.

Ejercicio 4

Escribe un programa en *Flex* que para una secuencia de entrada dada, realice lo siguiente para cada uno de estos casos:

- Informar del número de caracteres, palabras y líneas de texto.
- Reconocer los números reales, calcular e informar de su media y su máximo.
- Reconocer los números enteros e indicar cuántos de ellos están comprendidos en el rango entre cero y 10.000.

Nota: La principal dificultad de este ejercicio está en saber tratar los números como palabras compuestas por caracteres numéricos. Además, hay que tener en cuenta que un carácter del fichero de entrada en *Flex* sólo se puede emparejar con un solo patrón (ER) como máximo.

Ejercicio 5

Elabora un programa con *Flex* que, a partir de un fichero de texto que se pasa a la entrada, verifique que siempre que se abre un paréntesis o un corchete se cierre adecuadamente. Si detecta algún error, se informará de la línea donde se produce dicho error. La dificultad de este ejercicio radica en saber responder a las siguientes cuestiones:

- ¿Es regular el lenguaje de las palabras sobre el alfabeto $\{(,)\}$ que representa todos los pares de paréntesis correctamente anidados? ¿Puedes demostrarlo?
- Si no fuera regular, ¿Se puede hacer algo?