

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* de nombre *ej1.l* que elimine todos los saltos de línea repetidos más de 2 veces y los espacios en blanco y tabuladores al principio de línea.

Ejemplo:

```

Entrada:
Este es un texto
<EOL><EOL><EOL><EOL>

con 2 saltos de linea <EOL>
<EOF>

```

```

Salida:
Este es un texto <EOL><EOL>

con dos saltos de linea <EOL>
<EOF>

```

Ejercicio 2

Elabora un programa en *Flex* de nombre *ej2.l* que copie el archivo de entrada en uno de salida, sustituyendo todo número entero positivo múltiplo de 4 por el carácter '*'.

Este ejercicio tiene al menos dos enfoques:

- Construir una ER para leer números en decimal, y después en la acción asociada en Flex comprobar si éstos son múltiplos de 4.
- Construir una ER que denote el lenguaje de las cadenas sobre el alfabeto {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} que en decimal sean múltiplos de 4 para luego buscarlas en Flex.

Ejercicio 3

Construye un programa en *Flex* de nombre *ej3.l* que cuente el número de palabras del texto de entrada que contengan exactamente dos mayúsculas. ¿Cómo sería la ER? ¿Y cómo sería la ER si queremos identificar las palabras que contengan más de dos mayúsculas? Completa el programa anterior para reconocer ambas ERs a la vez.

Ejercicio 4

Escribe un programa en *Flex* de nombre *ej4.l* que para una secuencia de entrada dada, realice lo siguiente para cada uno de estos casos:

- Informar del número de caracteres, palabras, líneas de texto y párrafos.
- Reconocer los números enteros, calcular e informar de su suma.
- Reconocer los números reales e indicar cuántos de ellos están comprendidos en el rango entre uno y diez.

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* de nombre *ej5.l* que, a partir de un fichero de texto que se pasa a la entrada, verifique que siempre que se abre un corchete, se cierre. 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:

- ¿Puedes escribir un programa *Flex* con una sola expresión regular y menos de 3 líneas de código?
- ¿Te imaginas por qué?