

## Práctica. Primera Fase

### Desarrollo de un analizador léxico para un lenguaje sencillo

Considera un lenguaje de programación sencillo en el que los programas están formados por: (i) una sección de *declaraciones*, y (ii) una sección de *instrucciones*. Ambas secciones están separadas por **&&**. La sección de declaraciones, por su parte, está compuesta por una o más declaraciones, separadas por *puntos y coma*. Cada declaración consta de: (i) un *nombre de tipo*, (ii) un *nombre de variable*. Los nombres de tipo pueden ser **num** o **bool**. Por su parte, los nombres de variable comienzan necesariamente por una letra, seguida de una secuencia de cero o más letras, dígitos, o subrayado (**\_**). Por su parte, la sección de instrucciones consta de una o más instrucciones separadas por *puntos y coma*. El lenguaje únicamente considera instrucciones de *asignación*. Dichas instrucciones constan de una *variable*, seguida de **=**, seguida de una expresión. Las expresiones básicas consideradas son números reales con y sin signo, **true** y **false**. Los números comienzan, opcionalmente, con un signo **+** o **-**. Seguidamente debe aparecer una secuencia de 1 o más dígitos cualesquiera (no se prohíbe la aparición de ceros no significativos a la izquierda). A continuación, opcionalmente, puede aparecer una parte decimal (un **.**, seguido de una secuencia de 1 o más dígitos cualesquiera –no se prohíbe la aparición de ceros no significativos a la derecha). Por último, y también opcionalmente, puede aparecer una parte exponencial (**e** o **E**, seguida, opcionalmente, de un signo **-** o **+**, y, a continuación una secuencia de 1 o más dígitos cualesquiera). Los operadores que pueden utilizarse en las expresiones son los operadores aritméticos binarios usuales (**+**, **-**, **\*** y **/**), el *menos* unario (**-**), los operadores lógicos **and**, **or** y **not** y los operadores relacionales (**<**, **>**, **<=**, **>=**, **=**, **!=**). También es posible utilizar paréntesis para alterar las precedencias y asociatividades de los operadores.

#### Ejemplo de programa

```
num peso;  
bool pesado  
&&  
peso = (45.0 * 12e-56) / -002.00;  
pesado = (peso > 10.00) or (peso / 002 <= +04)
```

La memoria final de la práctica (a entregar al final del cuatrimestre) deberá contener un apartado **fase 1** con los siguientes sub-aptados relativos a esta primera fase:

- (1) Enumeración de las clases léxicas del lenguaje. Para cada clase debe incluirse, además, una descripción informal, en lenguaje natural.
- (2) Una especificación formal del léxico del lenguaje mediante definiciones regulares.
- (3) Diseño de un analizador léxico para el lenguaje mediante un diagrama de transiciones

Así mismo, en la entrega final deberá incluirse, en relación con esta primera fase:

- (1) Una implementación manual, en Java, del analizador léxico.
- (2) Una implementación alternativa de dicho analizador léxico, utilizando JLex.