

## **PRACTICA 4: Eliminador de comentarios**

### **4.1. Objetivos**

- Trabajar con los conceptos de estados y transiciones en un autómata diseñado para realizar una determinada tarea.
- Profundizar en las capacidades de desarrollar programas orientados a objetos en C++.

### **4.2. Rúbrica de evaluación del ejercicio**

Se señalan a continuación los aspectos más relevantes (la lista no es exhaustiva) que el profesorado tendrá en cuenta a la hora de evaluar el trabajo que el alumnado presentará en la sesión de evaluación de la práctica:

- El comportamiento del programa debe ajustarse a lo solicitado en el enunciado.
- Capacidad del programador(a) de introducir cambios en el programa desarrollado.
- Orientación a objetos: el programa ha de seguir ese paradigma de programación. Identifique clases y objetos que permitan modelar adecuadamente el escenario de trabajo que se propone.
- Modularidad: el programa ha de escribirse de modo que las diferentes funcionalidades que se precisen fueran encapsuladas en métodos concretos cuya extensión textual se mantuviera acotada.
- Formato propuesto para la escritura de programas en C++ en esta asignatura. En particular:
  - Todos los ficheros tendrán comentarios de cabecera.
  - El código estará correctamente indentado de acuerdo al estándar definido.
  - Todos los identificadores del código se nominarán de modo que sean significativos y de acuerdo al estándar.

- No se utilizarán aspectos de C que no sean estándar en C++ o que hayan quedado obsoletos (*deprecated*).
- El código no utilizará constantes (de ningún tipo) de forma directa sino que aquellas serán convenientemente definidas y utilizadas a partir de su identificador.
- Estudie y ponga en práctica las 94 recomendaciones de la *C++ Programming Style Guidelines* [1]. Allí donde encuentre discrepancias entre esta guía y la *Google C++ Style Guide* [2] puede usarse cualquiera de ambas alternativas.

Si el alumnado tiene dudas respecto a cualquiera de estos aspectos, debiera acudir al foro de discusiones de la asignatura para plantearlas allí. Se espera que, a través de ese foro, el alumnado intercambie experiencias y conocimientos, ayudándose mutuamente a resolver dichas dudas. También el profesorado de la asignatura intervendrá en las discusiones que pudieran suscitarse, si fuera necesario.

### 4.3. Ejercicio práctico

Desarrollar un programa `unComment.cpp` que lea un fichero de código fuente en C++ (o Java) y elimine todos los comentarios [3] contenidos en el fichero. Esta utilidad sería útil como parte de un compilador de C++, puesto que una de las primeras fases de un compilador consiste en eliminar los comentarios del código fuente, puesto que aquellos son significativos solo para los humanos, pero no para el compilador.

El programa recibirá por línea de comandos el fichero de entrada con el código fuente y el nombre del fichero donde se deberá almacenar el mismo código fuente pero sin comentarios. De esta forma, la ejecución del programa en línea de comandos debería ser similar a lo siguiente:

```
./program inputfile.cpp outputfile.cpp
```

El programa eliminará los comentarios de estilo tanto C como C++ (`/* */` y `//`) comportándose como lo haría un autómata finito cuyo diagrama de transición de estados sería como el que se muestra en la Figura 4.1 o similar a éste.

El programa posee una serie de estados entre los que transita como reacción a los caracteres que va procesando en el fichero de entrada. Partiendo del estado inicial va consumiendo caracteres y solo traslada al fichero de salida aquellos caracteres que no corresponden con los delimitadores ni el contenido de los comentarios, que son eliminados. El fichero de salida ha de ser exactamente igual al de entrada, habiéndose eliminado los comentarios de cualquier tipo.

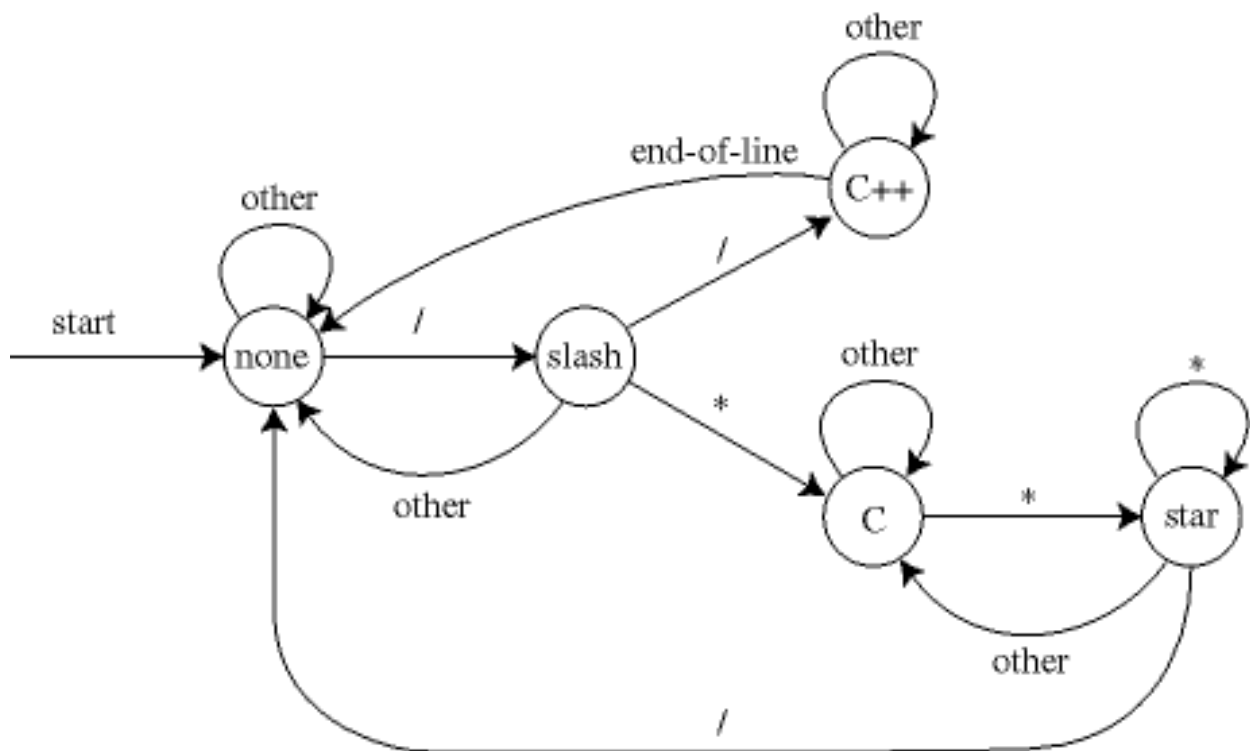


Figura 4.1: Diagrama de transición de estados

En la Figura 4.1 las transiciones etiquetadas como *other* corresponden con la presencia de cualquier carácter diferente de los que propician el resto de transiciones del estado en cuestión. Así por ejemplo, el programa permanece en el estado C ante la lectura de cualquier carácter en la entrada (*other*) excepto el carácter asterisco (\*) que le hace transitar al estado *star*.

Cabe destacar que para llevar a cabo la eliminación de los comentarios **se deberá simular explícitamente el autómata de la Figura anterior**, evitando, por tanto, en esta práctica la utilización de expresiones regulares.

# Bibliografía

- [1] C++ Programming Style Guidelines <https://geosoft.no/development/cppstyle.html>
- [2] Google C++ Style Guide, <https://google.github.io/styleguide/cppguide.html>
- [3] C++ Comments, [https://en.wikibooks.org/wiki/C%2B%2B\\_Programming/Code/Style\\_Conventions/Comments](https://en.wikibooks.org/wiki/C%2B%2B_Programming/Code/Style_Conventions/Comments)