

## PRÁCTICA 1: Programar un simulador de un autómatas con pila

### Objetivo

El objetivo de la práctica consiste en programar un simulador de un autómatas con pila, realizando un diseño orientado a objetos.

### Entrega

El desarrollo de la práctica será revisado en las sesiones de laboratorio.

Se debe entregar el código fuente en la tarea del aula virtual.

Fecha límite de entrega: 31 de octubre de 2020.

### Notas de implementación

Se puede implementar el autómatas de pila con finalización por vaciado de pila (APv) o por estado final (APf). Únicamente es necesario implementar uno de los dos tipos.

Los elementos del autómatas con pila se introducirán en tiempo de ejecución del programa utilizando un fichero de texto con el siguiente formato:

```
# Comentarios
q1 q2 q3 ...      # conjunto Q
a1 a2 a3 ...      # conjunto Σ
A1 A2 A3 ...      # conjunto Γ
q1                # estado inicial
A1                # símbolo inicial de la pila
q2 q3             # conjunto F
q1 a1 q2 a2 m      # función de transición:  $\delta(q1, a1) = (q2, a2, m)$ 
q1 a(*)1 A1 q2 A(*)2  # función de transición:  $(q2, A) \in \delta(q1, a, A1)$ 
...               # cada una de las transiciones en una línea distinta
```

<sup>(\*)1</sup> a es un símbolo de  $(\Sigma \cup \epsilon)$ . En el fichero el símbolo  $\epsilon$  se representará por un punto (.)

<sup>(\*)2</sup> A puede ser  $\epsilon$  (que se escribirá también en el fichero mediante un punto) o estar formado por uno o más símbolos de  $\Gamma$ , que se escribirán separados por espacios en blanco. Por ejemplo:  $A = A1 A1 A1$

El programa deberá funcionar con los ejemplos descargados del aula virtual.

Se debe verificar que la información proporcionada en el fichero cumpla con las restricciones de la definición formal de un autómeta, por ejemplo:  $s \in Q$ .

Una vez cargada la definición del autómeta, debe ser posible comprobar si varias cadenas de entrada pertenecen al lenguaje reconocido por el autómeta. Las diferentes cadenas de entrada podrán ser introducidas por teclado o por fichero.

Como salida, el programa debe indicar si la palabra pertenece al lenguaje que reconoce el autómeta de pila.

Se puede realizar el código utilizando los lenguajes C++ o Java.

Se debe utilizar un diseño orientado a objetos.

De forma adicional no obligatoria, el programa podrá tener la opción de ejecutar en *modo traza*, donde se mostrará la información del autómeta después de aplicar cada transición.