

## 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

La ejecución de la práctica será revisada en la sesión de entrega en el laboratorio.

La entrega se realizará a través del aula virtual:

- Es necesario proporcionar el código fuente en su totalidad, incluyendo todos los archivos y librerías indispensables para su compilación y ejecución.
- Se deberá incluir un archivo README que contenga la información relevante para la compilación, ejecución y revisión de la práctica.

Fecha límite de entrega: 10 de octubre de 2025.

### 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. En el archivo README se debe indicar cuál es el tipo realizado.

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
a <sub>1</sub> a <sub>2</sub> a <sub>3</sub> ...	# conjunto $\Sigma^{(*)1}$
A <sub>1</sub> A <sub>2</sub> A <sub>3</sub> ...	# conjunto $\Gamma^{(*)1}$
q1	# estado inicial
A <sub>1</sub>	# símbolo inicial de la pila
q2 q3	# conjunto F (solamente en el caso de APf)
q1 a <sup>(*)2</sup> A <sub>1</sub> q2 A <sup>(*)3</sup>	# función de transición: $(q2, A) \in \delta(q1, a, A_1)$
...	# cada una de las transiciones en una línea distinta

(\*)1 Los símbolos contenidos en  $\Sigma$  y en  $\Gamma$  estarán formados por un único carácter.

(\*)2 a es un símbolo de  $(\Sigma \cup \epsilon)$ . En el fichero el símbolo  $\epsilon$  se representará por un punto (.)

(<sup>\*3</sup>)  $A$  puede ser  $\varepsilon$  (que se escribirá también en el fichero mediante un punto) o estar formado por uno o más símbolos de  $\Gamma$ , que se escribirán sin espacios en blanco. Por ejemplo:  $A = A_1A_1A_1$

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

El programa debe verificar que la información proporcionada en el fichero cumple con las restricciones de la definición formal de un autómata, por ejemplo:  $s \in Q$ .

Una vez leída la definición del autómata, debe ser posible comprobar varias cadenas de entrada e indicar si pertenecen al lenguaje reconocido por el autómata. Las diferentes cadenas de entrada podrán ser introducidas por teclado o por fichero (no es necesario permitir los dos métodos).

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

Los lenguajes de programación permitidos son C++ o Java.

Se debe utilizar un diseño orientado a objetos.

De forma adicional, el programa podrá tener la opción de ejecutar en *modo traza*, donde se mostrará la información del autómata después de aplicar cada transición (estado, cadena, pila y posibles transiciones a aplicar). La traza puede mostrarse por pantalla o escribirse a un fichero.