

### PRÁCTICA 3: Programar un simulador de una Máquina de Turing determinista

#### Objetivo

El objetivo de la práctica consiste en programar un simulador de una Máquina de Turing, 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 ejecución y revisión de la práctica.
- Se deberá incluir las definiciones de las Máquinas de Turing ( $M = (Q, \Sigma, \Gamma, s, b, F, \delta)$ ) diseñadas para resolver los dos problemas propuestos.

Fecha límite de entrega: 8 de noviembre de 2024.

#### Notas de implementación

Posibles variaciones de la Máquina de Turing a implementar: (seleccionar una opción en cada fila)

- Máquina de Turing con escritura y movimientos simultáneos o independientes.
- Máquina de Turing donde los únicos movimientos sean izquierda (L) y derecha (R) o que incluya también la posibilidad de no movimiento (S).
- Máquina de Turing con cinta infinita en una única dirección o en ambas direcciones.

En el archivo README se debe indicar cuáles son las opciones seleccionadas.

Los elementos de la Máquina de Turing se introducirán en tiempo de ejecución del programa utilizando un fichero de texto<sup>1</sup> con el siguiente formato:

```
# Comentarios
q1 q2 q3 ...      # conjunto Q
a1 a2 a3 ...      # conjunto  $\Sigma$ 
A1 A2 A3 ...      # conjunto  $\Gamma^2$ 
q1                # estado inicial
b                 # símbolo blanco
q2 q3             # conjunto F
q1 a1 q2 a2 m     # función de transición3:  $\delta(q1, a1) = (q2, a2, m)$ 
...               # cada una de las transiciones en una línea distinta
```

<sup>1</sup> Se puede sustituir por un fichero JSON.

<sup>2</sup> Los símbolos contenidos en  $\Gamma$  estarán formados por un único carácter.

<sup>3</sup> La estructura de las transiciones se modificará según las características de la MT a implementar.

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

Una vez leída la definición de la Máquina de Turing, debe ser posible ejecutarla con diferentes parámetros de entrada. Las cadenas de entrada podrán ser introducidas por teclado o por fichero (no es necesario permitir los dos métodos).

Inicialmente, la cabeza de L/E debe encontrarse en el primer símbolo de los parámetros de entrada.

Como salida, el programa debe indicar si la Máquina de Turing se ha parado en un estado de aceptación y mostrar la cinta.

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

Se debe utilizar un diseño orientado a objetos.

De forma adicional: el programa podrá aceptar Máquinas de Turing Multicintas. En este caso, el número de cintas aparecerá en el fichero de configuración entre F y las transiciones.

### Máquinas de Turing a diseñar:

1. MT que reconozca el lenguaje  $L = \{ a^n b^m \mid m > n, n \geq 0 \}$
2. MT que reciba como parámetro una cadena compuesta por símbolos 'a' y 'b'. La MT debe sustituir la cadena por el número de símbolos 'b', seguido del número de símbolos 'a' separados por un símbolo blanco. El número se codificará como  $n = 1^n$ . La cabeza de L/E se quedará al principio del resultado:

Ejemplo: Cinta inicial (con la cadena de entrada): •abbabaabb•

Cinta final (con el resultado): •11111•1111•

Si no hay símbolos 'b' se escribirá un 0 en sustitución del número.

Si no hay símbolo 'a' no se incluirá ningún dato.

Ejemplos: Cinta al inicio (con la cadena de entrada): •aa•

Cinta al final (con el resultado): •0•11•

Cinta al inicio (con la cadena de entrada): •bb•

Cinta al final (con el resultado): •11•