

PRÁCTICA 2: 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:

- Proporcionar el código fuente en su totalidad, incluyendo todos los archivos y librerías indispensables para su compilación y ejecución.
- Incluir un archivo README que contenga la información relevante para la ejecución y revisión de la práctica.
- Incluir las definiciones de las Máquinas de Turing ($M = (Q, \Sigma, \Gamma, s, b, F, \delta)$) diseñadas para resolver los dos problemas propuestos.
 - Incluir la ruta y nombre de los archivos en el README.

Fecha límite de entrega: 24 de octubre de 2025.

Notas de implementación

Posibles variaciones de la Máquina de Turing a implementar: (seleccionar una opción en cada fila). Las opciones seleccionadas deben indicarse en el README.

- 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.

Los elementos de la Máquina de Turing 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^1

q1 # estado inicial

b # símbolo blanco

q2 q3 # conjunto F

q1 a₁ q2 a₂ m # función de transición²: $\delta(q1, a_1) = (q2, a_2, m)$

... # cada una de las transiciones en una línea distinta

¹ Los símbolos contenidos en Γ estarán formados por un único carácter.

² La estructura de las transiciones se modificará, si es necesario, para adaptarse a 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. Para cada cadena de entrada, la salida del programa debe incluir:

- Si la MT se ha parado en un estado de aceptación o no.
- El contenido de la cinta, indicando la posición en la que se encuentra la cabeza de L/E. Por ejemplo, "ab[c]d" si la cabeza de L/E apunta al símbolo "c".

Restricciones de código:

- Usar los lenguajes C++ o Java.
- 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. En una MT multicinta inicialmente la cadena de entrada estará únicamente en la primera cinta. Al finalizar la ejecución, el resultado debe estar también en la primera cinta.

Máquinas de Turing a diseñar:

1. MT que reconozca el lenguaje $L = \{ a^n b^m \mid m > n, n > 0 \}$
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 'a', seguido del número de símbolos 'b' separados por un símbolo blanco. El número se codificará como $n = 1^{n+1}$.

La cabeza de L/E debe quedar al principio del resultado:

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

Cinta final (con el resultado): •11111•111111•

Si la cadena está compuesta por símbolos del mismo tipo, se reflejará con un "1" la ausencia del otro símbolo:

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

Cinta al final (con el resultado): •111•1•

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

Cinta al final (con el resultado): •1•111•