

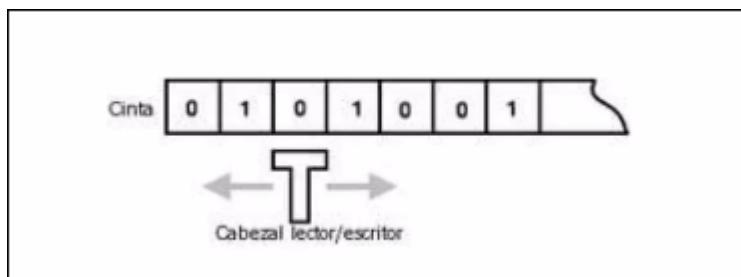
6. Máquinas de Turing

Copyright (c) 2025 Adrián Quiroga Linares Lectura y referencia permitidas; reutilización y plagio prohibidos

6.1 Concepto Fundamental

Una **Máquina de Turing (MT)** es un modelo matemático de computación que consiste en un autómata con una capacidad de memoria ilimitada en forma de una cinta infinita.

A diferencia de los autómatas finitos o de pila, la MT puede mover su cabezal de lectura/escritura tanto a la izquierda como a la derecha, y puede modificar los símbolos de la cinta.



1. La Cinta (The Tape)

Es la **memoria** de la máquina.

- **Qué es:** Es una tira de papel dividida en casillas (celdas) individuales.
- **Características:** En la teoría, es **infinita** (puedes seguir escribiendo a la derecha o izquierda para siempre).
- **Función:** En cada casilla se guarda un único símbolo del alfabeto (una 'a', un '1', un espacio en blanco...).
- A veces se menciona la "cinta seminfinita" , que es una cinta que tiene un muro a la izquierda pero es infinita a la derecha.

2. La Cabeza (The Head)

Es el **lector/escritor**. Es la parte "activa" de la máquina.

- **Qué es:** Un dispositivo que se sitúa sobre una casilla concreta de la cinta en cada momento.
- **Función:**
 1. **Lee** qué símbolo hay en esa casilla.
 2. **Escribe** un símbolo nuevo (borrando el anterior).
 3. **Se mueve** una casilla a la izquierda (*I*), a la derecha (*D*), o se queda quieta, según las instrucciones del estado actual .

- **Analogía:** Es como tu ojo y tu mano con un lápiz, mirando solo un cuadradito de papel a la vez.

3. La Pista (The Track)

Este es el concepto más abstracto y el que suele confundir. Una pista **no es una cinta nueva**, es una **división horizontal** dentro de una misma cinta.

- **Qué es:** Imagina que coges la cinta (que es una fila de casillas) y divides cada casilla horizontalmente en varios pisos o niveles. Cada nivel es una "pista".
- **Para qué sirve:** Permite que la máquina lea o guarde **varios datos a la vez en la misma posición física**.
- **Ejemplo visual:**
 - **Cinta normal (1 pista):** En la casilla 5 hay una "A".
 - **Cinta con 2 pistas:** En la casilla 5, en el *piso de arriba* (pista 1) hay una "A" y en el *piso de abajo* (pista 2) hay un "1".

6.2 Definición

Una MT se define formalmente como $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

- **Q (Conjunto de estados):** Los estados finitos internos de la máquina.
- **Σ (Alfabeto de entrada):** Los símbolos que forman la cadena inicial. Es un subconjunto de Γ y **no** incluye el blanco (B).
- **Γ (Alfabeto de la cinta):** Todos los símbolos que pueden aparecer en la cinta. Incluye a Σ y al símbolo blanco.
- **δ (Función de transición):** El "cerebro" de la máquina. Define qué hacer en cada paso.
- **q_0 (Estado inicial):** Donde comienza el cómputo ($q_0 \in Q$).
- **B (Símbolo Blanco):** Representa una celda vacía en la cinta infinita. Pertenece a Γ pero no a Σ .
- **F (Estados finales):** Conjunto de estados de aceptación.

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$$

L(M) significa "**El Lenguaje de la Máquina M**". En términos sencillos: es la lista VIP de palabras que la máquina acepta. La fórmula formal es:

$$L(M) = \{w \in \Sigma^* \mid q_0 w \vdash^* \alpha_1 q_f \alpha_2, \text{ con } q_f \in F\}$$

1. $w \in \Sigma^*$: Tomas una palabra w cualquiera formada por letras del alfabeto (Σ).
2. $q_0 w$ (**Configuración Inicial**): La máquina empieza en el estado inicial (q_0) con la cabeza lectora al principio de la palabra w .
3. \vdash^* (**Derivación**): Este símbolo significa "después de muchos pasos de cálculo...". Es decir, la máquina lee, escribe, se mueve y cambia de estado muchas veces.
4. $\alpha_1 q_f \alpha_2$ (**Configuración Final**):

- $q_f \in F$: Lo más importante. La máquina ha llegado a un estado q_f que pertenece al conjunto de estados Finales (F) .
- α_1 y α_2 : Representan "cualquier cosa" que haya quedado escrita en la cinta a la izquierda y a la derecha de la cabeza. A diferencia del ALA, aquí estas cadenas pueden ser enormes, porque la cinta es infinita.

6.3 Funcionamiento y Transiciones

La función de transición se define como $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$. Esto significa que, dado un estado actual y un símbolo leído en la cinta, la máquina realiza tres acciones simultáneas:

1. **Escribe** un nuevo símbolo en la celda actual.
2. **Cambia** a un nuevo estado.
3. **Mueve** la cabeza una posición a la Izquierda (I/L) o a la Derecha (D/R).

Ejemplo de lectura: Si $\delta(q_1, a) = (q_5, b, R)$, significa: "Si estoy en el estado q_1 y leo una 'a', escribo una 'b', paso al estado q_5 y me muevo a la derecha".

6.4 Aceptación y Parada

Una MT opera paso a paso hasta que ocurre una de estas situaciones:

Criterio de Aceptación:

- La MT acepta una cadena w si, partiendo de la configuración inicial, llega a un **estado final** $q_f \in F$.
- **Importante:** A diferencia de los autómatas finitos, **no es necesario leer toda la entrada** para aceptar; basta con alcanzar un estado final.
- Se asume que la máquina se detiene al llegar a un estado final (no hay transiciones definidas desde ahí).

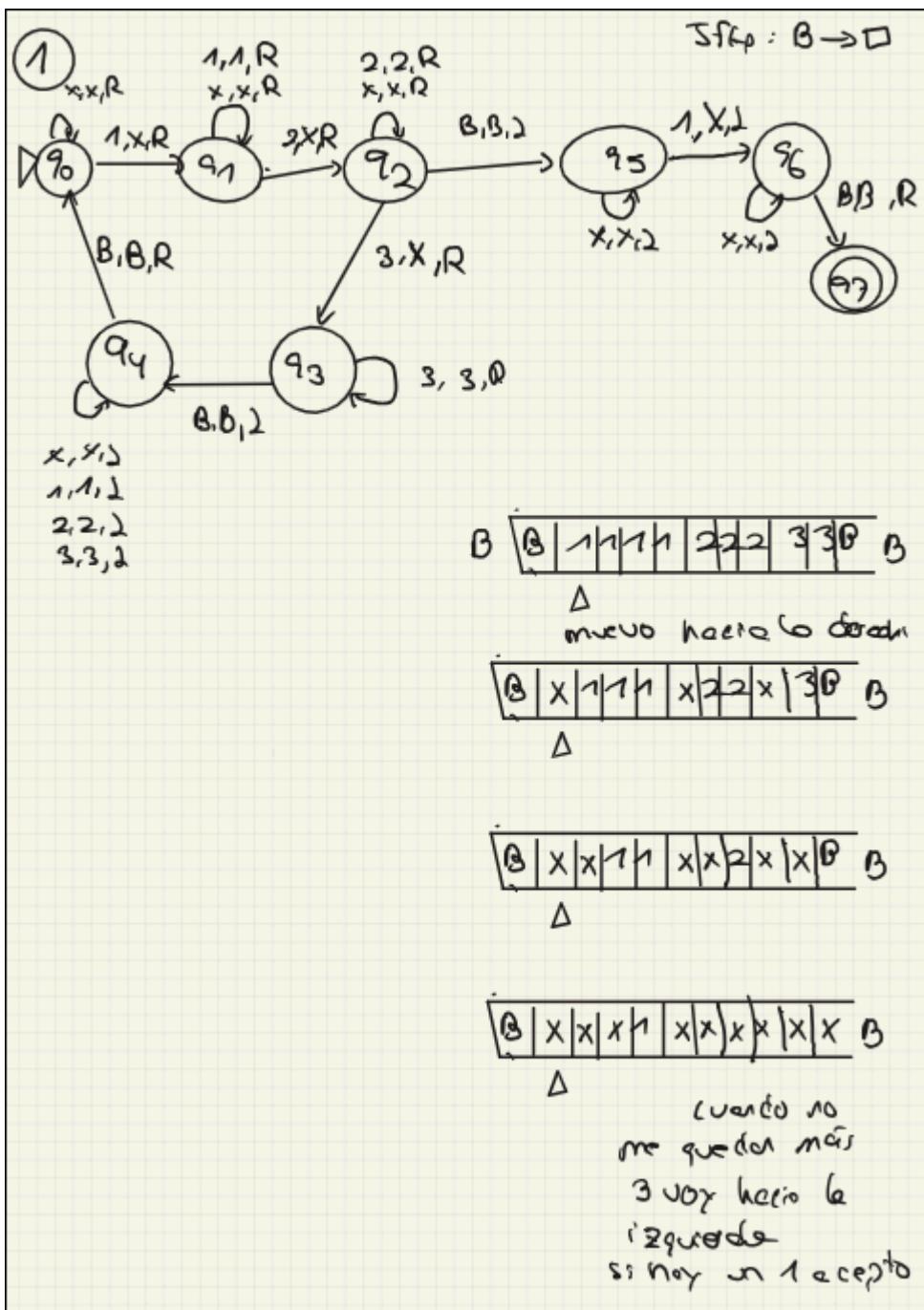
Rechazo y Bucles:

Si la cadena no pertenece al lenguaje ($w \notin L(M)$), pueden pasar dos cosas:

1. La máquina se para en un estado **no final** (porque no hay transición definida).
2. La máquina entra en un **bucle infinito** y nunca se para.

Nota

Si el lenguaje es...	¿Qué hace con las cadenas que NO pertenecen?
Recursivo	Siempre PARA (en estado de rechazo). Jamás hace bucle.
Rec. Enumerable	Puede parar (rechazo) O entrar en BUCLE INFINITO.



6.5 Protips

Normalmente te recomiendo que dibujes la situación inicial y vayas pensando que necesita hacer tu máquina para llegar al final y después esa idea o razonamiento lo transcribas en forma de autómata.

1. **El "Ping-Pong" (Zig-Zag):** Casi todos los ejercicios se resuelven yendo a buscar algo a la derecha, marcándolo, y volviendo a la izquierda a buscar el siguiente. No intentes hacerlo todo en una pasada.
- **Tachar, no borrar:** No "elimines" símbolos (dejando huecos en blanco B) a menos que quieras mover toda la cadena. Lo normal es TACHAR (usar el símbolo auxiliar x de tu alfabeto Γ').
 - **Ejemplo:** Para contar un '1', cámbialo por una 'x'. Así sabes que ya lo has contado.

- **La Cinta es tu Memoria:** No tienes variables como int contador = 0. Tu contador es lo que hay escrito en la cinta. Si quieras saber si el número es par o impar, tienes que recorrerlo y comprobarlo físicamente.
- **Los Blancos (*B*) son Muros:** Úsalos para saber dónde empieza y termina tu dato. Si te sales del *B*, estás en el vacío.

6.6 Variaciones de la Máquina de Turing

6.6.1 Equivalencia General

El punto más importante es que **todas** estas variantes son equivalentes a la MT Estándar . Esto significa que pueden resolver los mismos problemas (computabilidad), aunque la eficiencia (complejidad) varíe.

6.6.2 Variantes de Movimiento y Estructura

Estas variantes cambian cómo se mueve la cabeza o la forma del espacio de trabajo.

- **MT con opción de No-Movimiento:**
 - **Capacidad:** La cabeza puede quedarse estática (*E*) además de moverse a Izquierda (*I*) o Derecha (*D*) .
 - **Simulación:** Una MT estándar simula el "no movimiento" haciendo dos movimientos: uno a la derecha y luego uno inmediatamente a la izquierda (*D, I*) .
- **MT con Cinta Semi-infinita:**
 - **Estructura:** La cinta tiene un tope a la izquierda, es infinita solo hacia la derecha .
 - **Simulación:** Se usa una MT estándar con **2 pistas** (imaginando que doblamos la cinta infinita por la mitad):
 - Pista superior: Parte derecha de la cinta.
 - Pista inferior: Parte izquierda (en orden inverso) .
 - Se usa un marcador especial (#) para saber cuándo "dar la vuelta" en el extremo .
- **MT Multidimensional (Ej. Bidimensional):**
 - **Estructura:** Cinta infinita en más de una dimensión (plano). La cabeza se mueve en 4 direcciones: *I, D, AR* (arriba), *AB* (abajo) .
 - **Simulación:** Se usa una cinta con **2 pistas**:
 1. Contenido de la celda.
 2. Dirección/Coordenadas asociadas para saber dónde está cada dato .

6.6.3 Variantes de Cintas Múltiples

Estas variantes añaden más cintas o pistas para facilitar el trabajo.

- **MT Multicinta:**

- **Estructura:** Tiene n cintas independientes, cada una con su cabeza .
- **Simulación (¡Muy importante para test!):** Para simular n cintas en una sola, se necesitan $2n$ **pistas** .
 - Pistas impares: Guardan el contenido de las cintas.
 - Pistas pares: Marcan la posición de las cabezas .
- **MT con Cinta de Entrada:**
 - **Estructura:** Una cinta de solo lectura (entrada) + una cinta de trabajo .
 - **Simulación:** Se usan **4 pistas**: valores de entrada, posición cabeza entrada, contenido cinta trabajo, posición cabeza trabajo .

6.6.4 Determinismo y Universalidad

- **MT No Determinista (MTN):**
 - **Funcionamiento:** La función de transición δ devuelve un **conjunto** de posibles movimientos, no uno solo. Puede "replicarse" para seguir varios caminos .
 - **Simulación:** Se puede simular con una MT estándar usando cintas con múltiples pistas para gestionar las réplicas (una pista para contenido, otra para el estado de cada réplica) .
- **Máquina de Turing Universal (MTU):**
 - **Concepto:** Es una máquina "reprogramable" que recibe como entrada la descripción de *otra* máquina M y una cadena w , y simula a M ejecutando w .
 - **Estructura:** Típicamente se define con **3 cintas** :
 1. Cinta con la descripción de la máquina a simular (M).
 2. Cinta con el contenido/entrada (w).
 3. Cinta con el estado interno de M .

Tipo de MT	¿Cómo se simula con MT Estándar?	Dato clave para el test
Multicinta (n cintas)	Usa $2n$ pistas	Una pista para símbolo, otra para la cabeza.
Semi-infinita	Usa 2 pistas	Dobla la cinta (pista superior/inferior).
Bidimensional	Usa 2 pistas	Una para contenido, otra para dirección/coords.
Universal (MTU)	Usa 3 cintas	1. Descripción, 2. Contenido, 3. Estado.
No Determinista	Equivalente a Determinista	No es más potente, solo (teóricamente) más rápida.

6.8 Tesis de Church-Turing

Esta tesis establece una equivalencia entre "algoritmo" y "Máquina de Turing".

- Afirma que cualquier problema que pueda ser resuelto por un algoritmo (computable) puede ser resuelto por una MT.
- No se ha encontrado ningún modelo de computación más potente que la MT.