

Clase 1 - TCVP

Resumen

- **Máquina de Turing (MT):** Es un modelo general de computación. Una MT puede modificar un símbolo, cambiar su estado y moverse a la izquierda o a la derecha.
 - **Tesis de Church-Turing:** Todo dispositivo de computación físicamente realizable puede ser simulado por una MT.
 - **Computabilidad:** Estudia qué problemas pueden ser resueltos por algoritmos. Algunos problemas son computables pero no decidibles, lo que significa que una MT puede responder "sí" si la entrada pertenece al lenguaje, pero puede no responder o "loopear" si no pertenece.
 - **Decidibilidad:** Se refiere a problemas para los cuales existe una MT que siempre se detiene, respondiendo "sí" o "no".
 - **Lenguaje Formal:** Un lenguaje con alfabeto Σ es un conjunto de cadenas finitas de símbolos de Σ . Una MT acepta un lenguaje si reconoce las cadenas que pertenecen a él.
 - **Conjuntos y Funciones:** Se revisan conceptos básicos de conjuntos (subconjuntos, intersección, unión, diferencia) y funciones (dominio, codominio).
 - **Grafos:** Un grafo es un par (V, E) donde V es un conjunto de vértices y E es un conjunto de arcos.
 - **Fórmulas booleanas:** Combinación de variables booleanas con operadores lógicos (AND, OR, NOT). Una fórmula es satisfactible si existe una asignación de valores que la hace verdadera.
 - **Modelos equivalentes de MT:** Existen modelos de MT equivalentes al modelo estándar (una cinta), como MT con varias cintas. La MT con múltiples cintas no tiene más poder de computación que una MT con una sola cinta.
 - **Máquina de Turing No Determinista (MTN):** Para un mismo par (estado, símbolo), la máquina puede responder de más de una manera. Una MTN acepta una cadena si al menos una computación la acepta.
-

Máquina de Turing

El funcionamiento de una máquina de Turing se basa en los siguientes componentes:

- Una **unidad de control** que posee un conjunto finito de estados. Entre estos estados se incluyen un estado inicial (q_0), un estado de aceptación (q_A) y un estado de rechazo (q_R).
- Una **función de transición (δ)** que especifica el algoritmo que la máquina ejecuta. Esta función determina el siguiente estado de la unidad de control, el símbolo que se escribe en la cinta y el movimiento del cabezal, basándose en el estado actual y el símbolo leído.
- Una **cinta infinita en ambos extremos**, dividida en celdas, donde se pueden escribir símbolos.
- Un **cabezal** que puede leer y escribir símbolos en la cinta y moverse una celda a la izquierda (L), a la derecha (R) o permanecer en el mismo lugar (S).

En cada momento de la ejecución, la máquina se encuentra en una configuración determinada por su estado actual, el contenido de la cinta y la posición del cabezal. En cada paso, la máquina:

1. Lee el símbolo al que apunta el cabezal en la cinta.
2. Consulta su estado actual y el símbolo leído en la función de transición (δ).
3. Basándose en la definición de δ para esa combinación (estado actual, símbolo leído), la máquina realiza las siguientes acciones:

- Cambia a un nuevo estado.
- Escribe un nuevo símbolo (o el mismo) en la celda actual de la cinta.
- Mueve el cabezal a la izquierda, a la derecha o lo deja en la misma posición. La ejecución continúa hasta que la máquina alcanza un estado de aceptación (q_A) o un estado de rechazo (q_R), en cuyo caso la máquina se detiene.

[!NOTE] Es importante destacar la Tesis de Church-Turing, que postula que cualquier dispositivo computacional físicamente realizable puede ser simulado por una máquina de Turing. Esto subraya la universalidad del modelo de la máquina de Turing como un concepto fundamental en la computación.

Perspectivas

- Como **calculadoras**: a partir de una entrada producen una salida, resolviendo problemas de búsqueda.
- Como **reconocedoras**: aceptan o rechazan entradas, resolviendo problemas de decisión -> la máquina reconoce un lenguaje formado por el conjunto de todas las cadenas que acepta (las instancias que tienen solución). -> A mi me sirve pensarlo como que estas máquinas se centran en **reconocer un lenguaje**.
- Como **generadoras**: se escriben todas las cadenas acorrespondientes a las instancias positivas de un problema en una cinta de salida. El problema de enumeración requiere identificar y presentar todas las soluciones posibles para una instancia dada del problema de satisfactibilidad.

MT de varias cintas

Posee varias cintas, cada una con su propio cabezal de escritura/lectura independiente. La unidad de control sigue siendo única.

Funcionamiento

El funcionamiento de una MT con varias cintas es el siguiente:

1. En un solo paso, la máquina lee el estado corriente de la unidad de control y el símbolo corriente al que apunta cada uno de los K cabezales en sus respectivas cintas.
2. Basándose en esta información, la función de transición (δ) determina:
 - El nuevo estado de la unidad de control.
 - El símbolo que se escribirá en la celda actual de cada una de las K cintas.
 - El movimiento independiente (izquierda, derecha o ninguno) de cada uno de los K cabezales.
 Una configuración de una MT con varias cintas está definida por el estado actual, el contenido de todas las cintas y la posición de cada uno de los cabezales.

Teorema: cualquier MT con $K > 1$ cintas puede ser simulada por una MT con una sola cinta: Las pistas impares almacenan el contenido de las cintas originales, y las pistas pares almacenan la posición del cabezal correspondiente mediante una marca. Sin embargo, esta simulación introduce un retardo en el tiempo de ejecución. El texto indica que simular h pasos de una MT con K cintas puede requerir un tiempo del orden de $O(h^2)$ en una MT con una sola cinta.

Modelos equivalentes de MT

Equivalencia de dos MTs: dos máquinas se consideran equivalentes si aceptan el mismo lenguaje. Se centra en el lenguaje específico que cada máquina reconoce o decide, sin importar los detalles internos de su

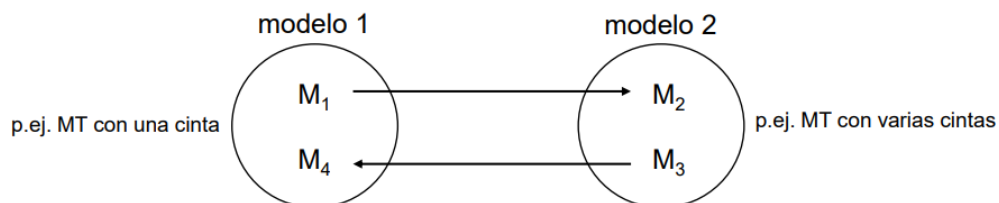
funcionamiento, como el número de estados o la función de transición específica. *Por ejemplo, dos MTs con diferentes conjuntos de estados y funciones de transición podrían ser equivalentes si ambas reconocen el lenguaje.*

Equivalencia de dos modelos de MTs: indica que dada una MT de un modelo, existe una MT equivalente del otro modelo. Cualquier problema que pueda ser resuelto por una MT en un modelo también puede ser resuelto por una MT en el otro modelo y viceversa. La prueba de equivalencia entre dos modelos generalmente implica demostrar que una máquina del primer modelo puede simular cualquier máquina del segundo modelo, y viceversa.

Dos modelos de MT son equivalentes si cualquier función computable por un modelo también puede ser computada por el otro, aunque con una diferencia en la eficiencia (tiempo de ejecución o espacio usado).

Modelos equivalentes de MT

- **Dos MT son equivalentes** si aceptan el mismo lenguaje (es decir, si resuelven el mismo problema).
- **Dos modelos de MT son equivalentes** si dada una MT de un modelo existe una MT equivalente del otro.

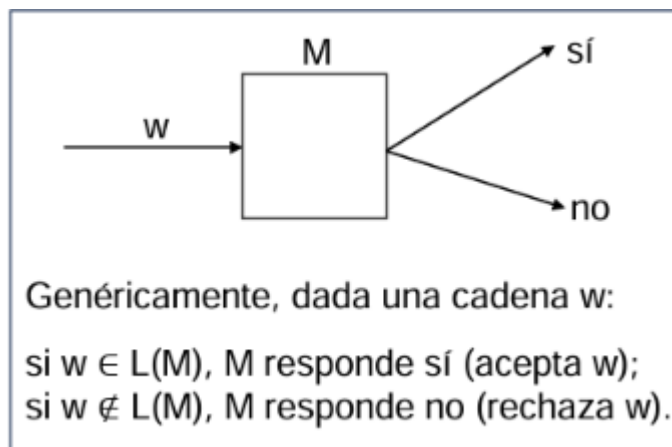


- Ejemplos de modelos de MT equivalentes al modelo de MT con **una cinta**: MT con **varias cintas**, MT con **cintas semi-infinitas** (sólo infinitas a la derecha), MT con **dos cintas y un solo estado**, etc.
- Ejemplos de modelos computacionales equivalentes al modelo de las MT: **máquinas RAM (acceso directo)**, **circuitos booleanos**, **lambda cálculo**, **funciones recursivas**, **gramáticas**, **programas C**, etc.
- **Todo esto refuerza la Tesis de Church-Turing.**

Computabilidad

- Problemas de **búsqueda**: en estos problemas, una MT debe encontrar una solución. -> Hay una solución -> MT la devuelve, -> No hay solución -> MT responde "No". *Ejemplo: Encontrar un camino entre dos vértices en un grafo no dirigido*
- Problemas de **decisión**: en estos problemas, la MT debe determinar si una determinada entrada cumple con una propiedad específica y responder "Si" o "No".
 - La MT solo responde SI o NO aceptando rechazando la entrada. *Ejemplo: Determinar si un grafo tiene un camino del vértice inicial al final. Determinar si una fórmula booleana es satisfacible.*

Serán consideradas los problemas de decisión: las máquinas de Turing sólo van a responder sí o no (aceptar - rechazar)



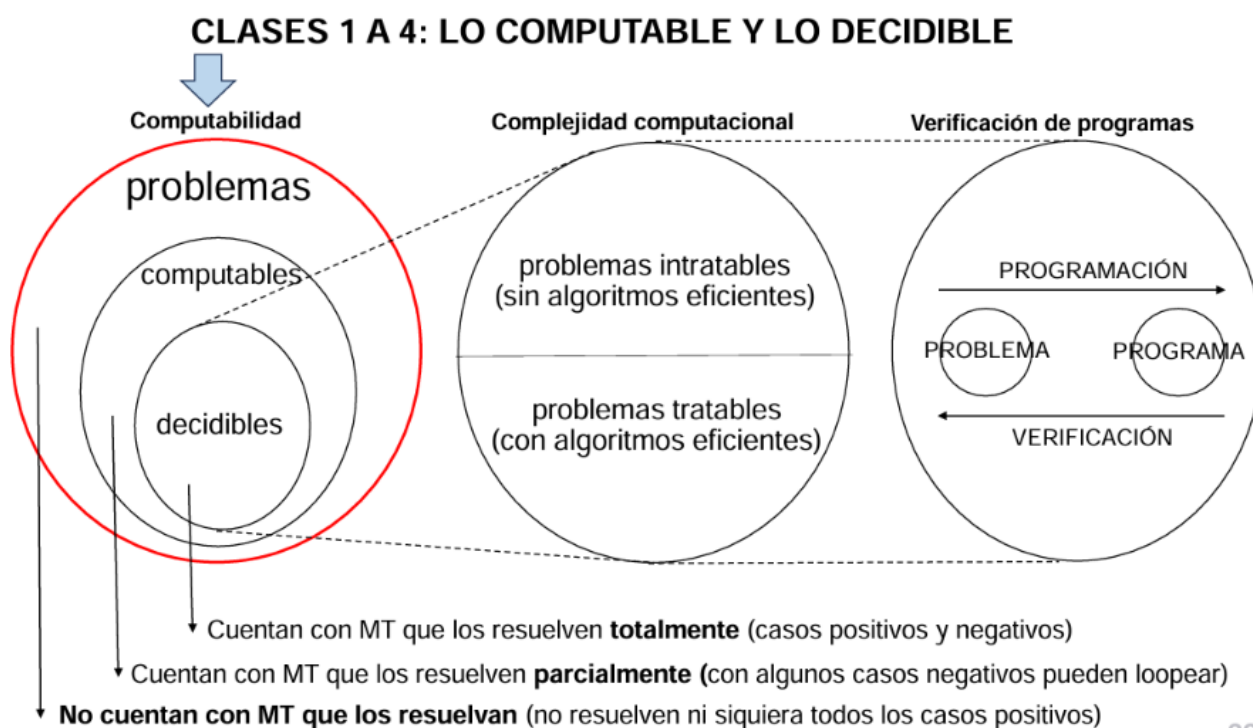
Entonces, $L(M)$ indica el lenguaje que acepta la Máquina de Turing *Ejemplo: $L(M) = \{\varphi_1, \varphi_2, \varphi_3, \dots\}$ donde φ es una fórmula booleana satisfactible*

Se utilizarán las palabras problema y lenguaje de forma indistinta (temporalmente).

- **Problemas computables:**

- **decidibles:** la MT siempre se detiene. Estos problemas se resuelven totalmente, tanto en casos positivos como negativos.
- **no decidibles:** la MT los resuelve parcialmente. No existe una MT que siempre se detenga para estos problemas.
 - Si la entrada pertenece al lenguaje responde "Sí".
 - Si la entrada no pertenece al lenguaje, la MT puede responder "no" o "loopear". *Ejemplo: el problema de decidir si un programa P para (Halting problem)*

- **Problemas no computables:** no tienen MT que los resuelvan (ni siquiera que respondan sí en todos los casos positivos). *Ejemplo: decidir la equivalencia de dos programas P_1 y P_2*

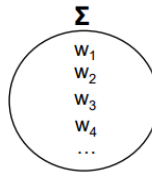


Anexo

Acerca de los lenguajes

- Σ es un **alfabeto** o conjunto de símbolos.

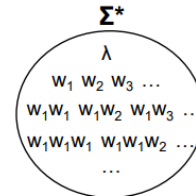
$$\Sigma = \{w_1, w_2, w_3, w_4, \dots\}$$



- Σ^* es el **lenguaje** o conjunto de cadenas de símbolos generado a partir de Σ .

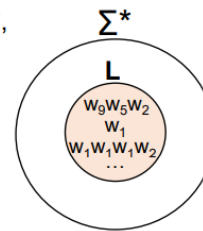
$$\Sigma^* = \{\lambda, w_1, w_2, w_3, \dots, w_1w_1, w_1w_2, w_1w_3, \dots, w_1w_1w_1, w_1w_1w_2, \dots\}$$

Σ^* es **infinito**. Sus cadenas son **finitas**. λ es la **cadena vacía**.



- Todo lenguaje L que consideraremos será un subconjunto de Σ^* , siendo Σ un único alfabeto que tomaremos como **universal**.

Es decir: $L \subseteq \Sigma^*$.



P. ej., L puede ser el lenguaje de las cadenas que representan los grafos con un camino del vértice inicial al vértice final.

- Operaciones** típicas entre lenguajes:

Intersección: $L_1 \cap L_2$

Unión: $L_1 \cup L_2$

Diferencia: $L_1 - L_2$

Complemento: L^c , que con respecto al conjunto universal Σ^* es: $\Sigma^* - L$

Producto (o Concatenación): $L_1 \cdot L_2$ (se concatenan las cadenas de uno y otro).

Repaso conceptos básicos

- Alfabeto** -> conjunto finito de símbolos. $\Sigma = \{a, b, c\}$ es un alfabeto de tres símbolos.
- Lenguaje** -> lenguaje con alfabeto Σ es un conjunto de cadenas finitas de símbolos de Σ . $L = \{aaa, b, abababa, ccb\}$
- Operaciones típicas** -> las mismas que entre conjuntos (resta, unión, disjunción, complemento)
- Grafo** -> par (V, E) tal que V es un conjunto de vértices y E es un conjunto de arcos.
- Una **fórmula booleana** es una fórmula lógica que se obtiene combinando variables booleanas, de valor de verdad verdadero (V) o falso (F), con los operadores lógicos and, or y not (\neg).
- Fórmula booleana satisfactible** -> existe al menos una asignación de valores de verdad que la evalúa verdadera.