300CIG007
Computabilidad y Lenguajes Formales:
Teoría de la Computabilidad:
Máquinas de Turing

Pontificia Universidad Javeriana Cali
Ingenieria de Sistemas y Computación
Prof. Gloria Inés Alvarez V.

Máquina de Turing

- Propuesta por **Alan Turing** en1936.
- Similar a un autómata finito, pero con una cantidad de memoria ilimitada y sin restricciones.
- Los problemas que no pueden ser resueltos por una máquina de Turing están por fuera de los límites teóricos de la computación.

Alan Mathison Turing – fuente: http://en.wikipedia.org/wiki/Alan_turing

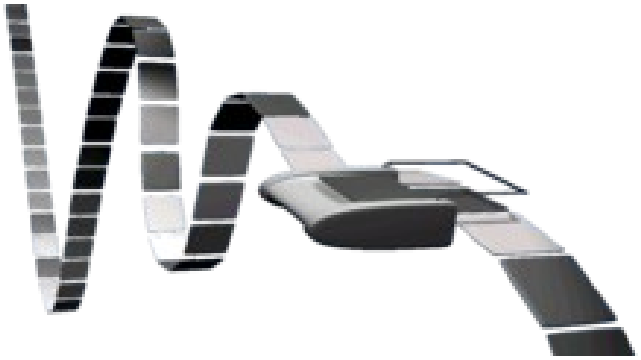


Diagrama Artístico de una Máquina de Turing – fuente: http://es.wikipedia.org/wiki/M%C3%A1quina_de_Turing

Máquina de Turing

- Tiene una cinta infinita (memoria ilimitada).
- Una cabeza que permite leer, escribir, y moverse (libremente, hacia la izquierda o derecha) en la cinta.
- Inicialmente la cinta tiene solamente la cadena de entrada, seguida por el símbolo especial ***blank (□)*.**
- Las salidas: Aceptación y Rechazo se obtienen cuando la máquina llega a un(os) estado(s) designado(s) como Aceptación o como Rechazo.
- Si la máquina no llega a un estado de aceptación o rechazo, continuará computando, no se detiene.

Máquina de Turing: Def. Formal

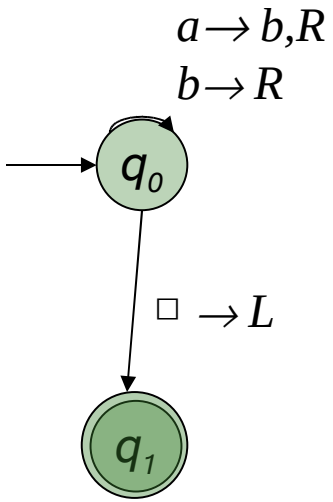
Una Máquina de Turing es una 7-tupla

$(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ *donde:*

- Q es un conjunto finito de Estados
- Σ el Alfabeto de entrada es un conjunto finito, que NO incluye el símbolo especial *blank* ($\square$).
- Γ el Alfabeto de la cinta, $B \in \Gamma$ y $\Sigma \subseteq \Gamma$
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$ es una función de transición.
- $q_0 \in Q$ es el Estado Inicial
- $q_{accept} \in Q$ es el Estado de Aceptación
- $q_{reject} \in Q$ es el Estado de Rechazo ($q_{accept} \neq q_{reject}$)

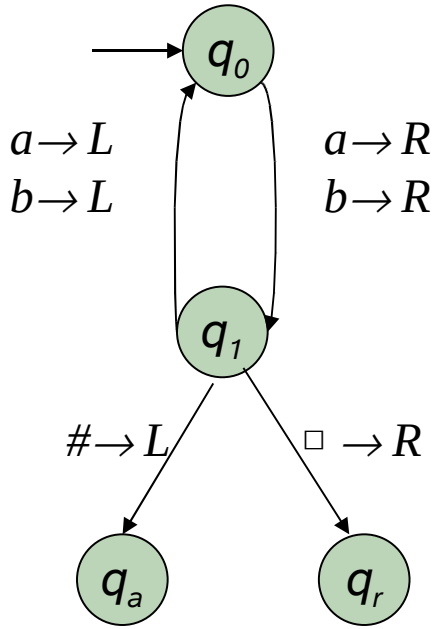
Máquina de Turing: Ejemplo

- $Q = \{ q_0, q_1 \}$
- $\Sigma = \{ a, b \}$
- $\Gamma = \{ a, b, \square \}$
- $\delta(q_0, a) = (q_0, b, R)$
- $\delta(q_0, b) = (q_0, b, R)$
- $\delta(q_0, \square) = (q_1, \square, L)$
- q_0 *(inicio)*
- q_1 *(aceptación)*



Máquina de Turing: Ejemplo

- $Q = \{ q_0, q_1 \}$
- $\Sigma = \{ a, b \}$
- $\Gamma = \{ a, b, \#, \square \}$
- $\delta(q_0, a) = (q_1, b, R)$
- $\delta(q_0, b) = (q_1, b, R)$
- $\delta(q_1, a) = (q_1, a, L)$
- $\delta(q_1, b) = (q_1, b, L)$
- $\delta(q_1, \#) = (q_a, \#, L)$
- $\delta(q_1, \square) = (q_r, \square, R)$
- q_0 *(inicio)*
- q_a *(aceptación)*
- q_r *(rechazo)*



Máquina de Turing: Ejemplo

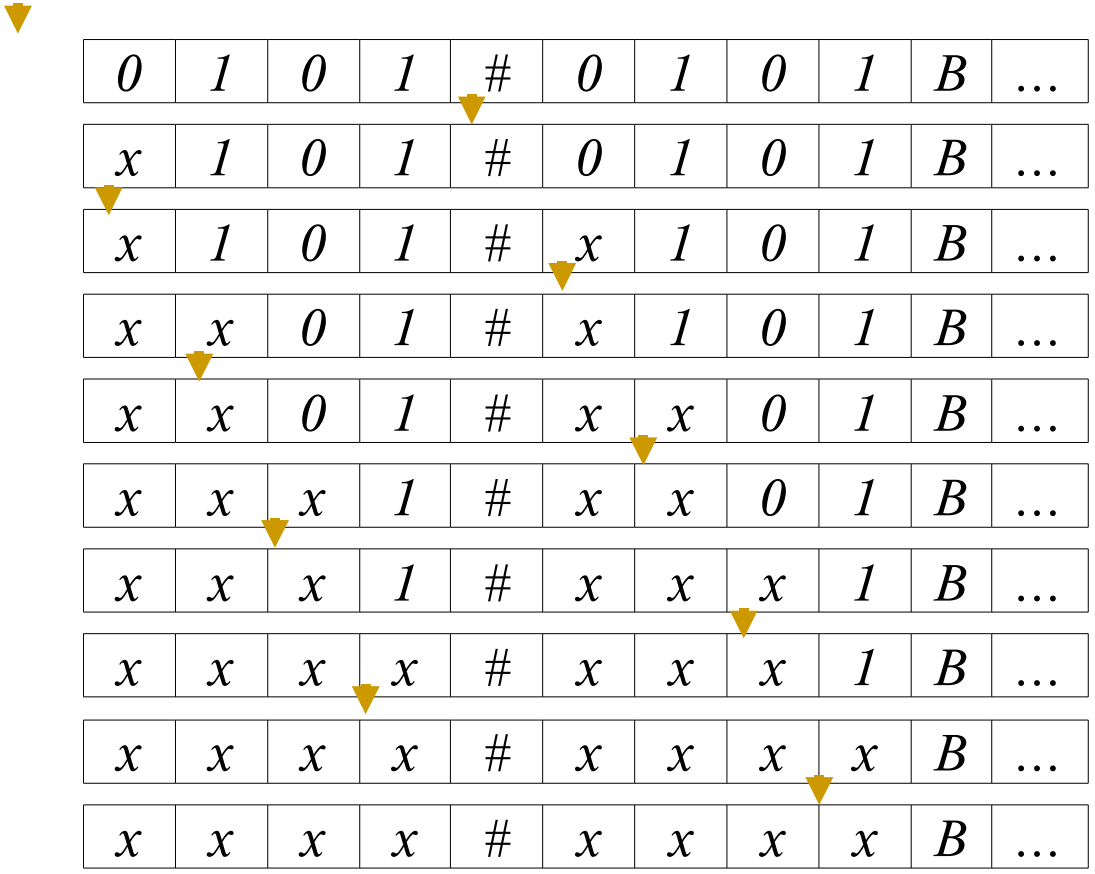
MT que verifica si una cadena pertenece al lenguaje

$$B = \{ w\#w \mid w \in \{ 0, 1\}^* \}$$

- Algoritmo:
  - Haga zig-zag en la cinta, para verificar que en las posiciones correspondientes a cada lado del símbolo #, aparece el mismo símbolo (*0* o *1*). Si esto no se cumple *Rechace*.
  - Cuando todos los símbolos a la izquierda del símbolo # han sido verificados, revise los símbolos que no hayan sido revisados en el lado derecho, si hay algún símbolo no revisado en el lado derecho, *Rechace*, de lo contrario *Acepte*.

Máquina de Turing: Ejemplo

MT que verifica si una cadena pertenece al lenguaje $B = \{ w\#w \mid w \in \{0, 1\}^* \}$



Máquina de Turing: Ejemplo

MT que verifica si una cadena pertenece al lenguaje

$$B = \{ w\#w \mid w \in \{0, 1\}^* \}$$

MT = $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

- $Q = \{ q_0, q_1, q_2, q_4, q_5, q_6, q_7, q_8, q_9, q_{10} \}$
- $\Sigma = \{ 0, 1, \# \}$
- $\Gamma = \{ blank, 0, 1, \#, x \}$
- q_0 es el Estado Inicial
- q_{10} es el Estado de Aceptación
- q_9 es el Estado de Rechazo

Máquina de Turing: Ejemplo

MT que verifica si una cadena pertenece al lenguaje

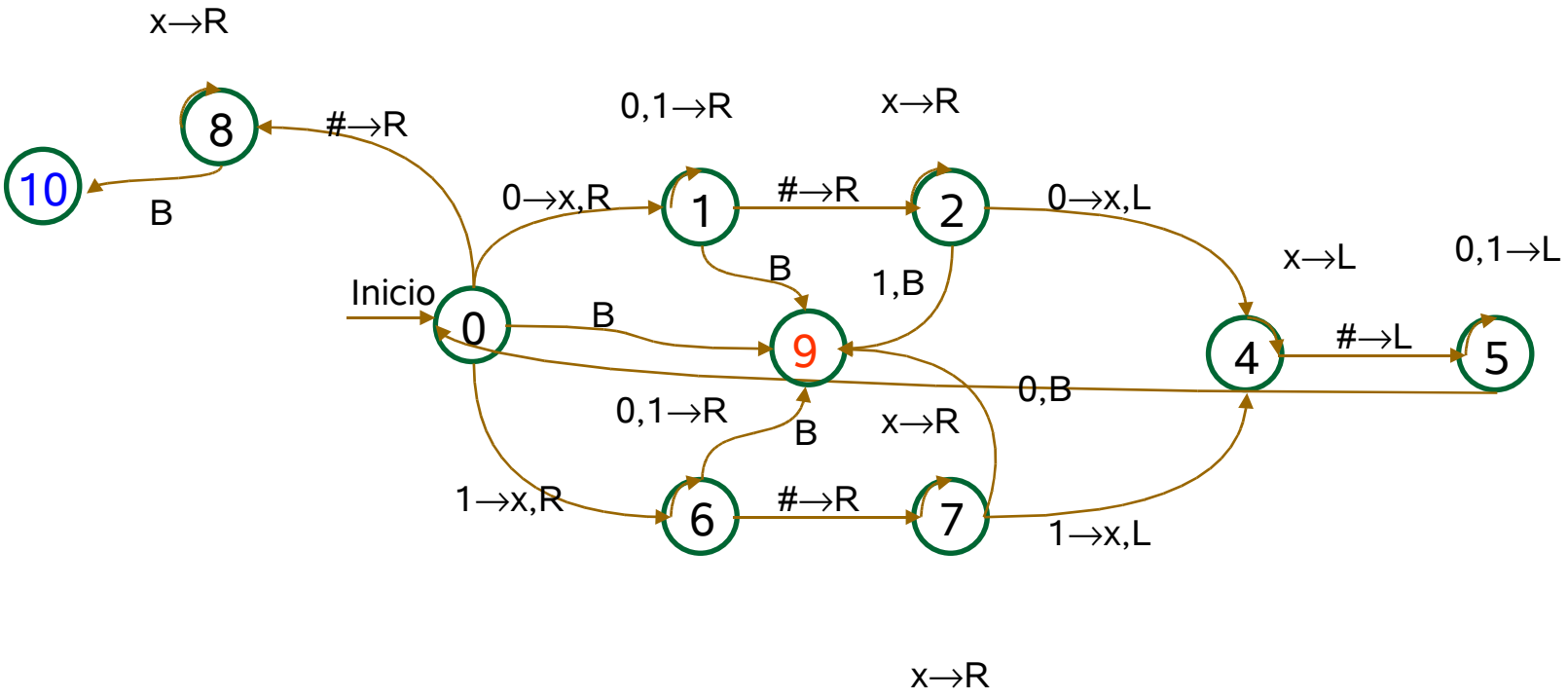
$$B = \{ w\#w \mid w \in \{0, 1\}^* \}$$

$\delta(q_0, 0) \rightarrow (q_1, x, R)$	$\delta(q_2, 1) \rightarrow (q_9, 1, R)$	$\delta(q_6, \#) \rightarrow (q_7, \#, R)$
$\delta(q_0, 1) \rightarrow (q_6, x, R)$	$\delta(q_2, B) \rightarrow (q_9, 1, R)$	$\delta(q_7, x) \rightarrow (q_7, x, R)$
$\delta(q_0, B) \rightarrow (q_9, B, R)$	$\delta(q_4, x) \rightarrow (q_4, x, L)$	$\delta(q_7, 1) \rightarrow (q_4, x, L)$
$\delta(q_0, \#) \rightarrow (q_8, \#, R)$	$\delta(q_4, \#) \rightarrow (q_5, \#, L)$	$\delta(q_7, 0) \rightarrow (q_9, 0, R)$
$\delta(q_1, 0) \rightarrow (q_1, 0, R)$	$\delta(q_5, 0) \rightarrow (q_5, 0, L)$	$\delta(q_7, B) \rightarrow (q_9, B, R)$
$\delta(q_1, 1) \rightarrow (q_1, 1, R)$	$\delta(q_5, 1) \rightarrow (q_5, 1, L)$	$\delta(q_8, x) \rightarrow (q_8, x, R)$
$\delta(q_1, \#) \rightarrow (q_2, \#, R)$	$\delta(q_5, x) \rightarrow (q_0, x, R)$	$\delta(q_8, 0) \rightarrow (q_9, 0, R)$
$\delta(q_1, B) \rightarrow (q_9, B, R)$	$\delta(q_6, 0) \rightarrow (q_6, 0, R)$	$\delta(q_8, 1) \rightarrow (q_9, 1, R)$
$\delta(q_2, x) \rightarrow (q_2, x, R)$	$\delta(q_6, 1) \rightarrow (q_6, 1, R)$	$\delta(q_8, B) \rightarrow (q_{10}, B, R)$
$\delta(q_2, 0) \rightarrow (q_4, x, L)$	$\delta(q_6, B) \rightarrow (q_9, 1, R)$	

Máquina de Turing: Ejemplo

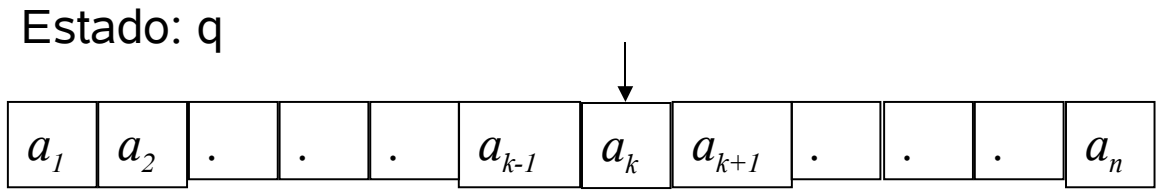
MT que verifica si una cadena pertenece al lenguaje

$$B = \{ w\#w \mid w \in \{ 0, 1\}^* \}$$



Máquina de Turing: Computación

- Cuando una Máquina de Turing realiza un cómputo, ocurren los siguientes cambios:
  - Se actualiza el contenido de la cinta de entrada
  - Se actualiza la posición de la cabeza de lectura/escritura
  - Se actualiza el estado actual.
- Una **configuración** de una Máquina de Turing muestra esos tres elementos.



Configuración: $a_1 a_2 \ldots a_{k-1} q a_k a_{k+1} \ldots a_n$

Máquina de Turing: Computación

- Decimos que una configuración C_1 lleva a otra configuración C_2, si la máquina puede ir de C_1 a C_2 en un solo paso:
  - Si $a, b, c \in \Gamma$, $u, v \in \Gamma^*$, y los estados q_i y q_j:
  - Si uaq_ibv y uq_jacv son dos configuraciones, decimos que uaq_ibv lleva a uq_jacv, si hay una transición $\delta(q_i, b) = (q_j, c, L)$
  - Si uaq_ibv y $uacq_jv$ son dos configuraciones, decimos que uaq_ibv lleva a $uacq_jv$, si hay una transición $\delta(q_i, b) = (q_j, c, R)$
- La configuración de inicio de una máquina M con una entrada w, es la configuración q_0w, la configuración de aceptación es q_{accept}, y la configuración de rechazo es q_{reject}, estas últimas son configuraciones de parada.

Máquina de Turing: Computación

- Una máquina M acepta una entrada w si existe una secuencia de configuraciones $C_1, C_2, \ldots, C_k$, tal que:
  - C_1 es la configuración de inicio de la máquina
  - Cada C_i lleva a C_{i+1} y
  - C_k es una configuración de aceptación
- El conjunto de cadenas que M acepta es el **Lenguaje de *M***, y se denota $L(M)$

$$L(M) = \{ w \in \Sigma+ : q_0 w \vdash^* x_1 q_f x_2,\ q_f \in F,\ x_1, x_2 \in \Gamma^* \}$$

Máquina de Turing: Definiciones

- Un lenguaje es **Reconocible** (Turing-Recognizable) si alguna Máquina de Turing lo acepta.

- El conjunto de lenguajes que son aceptados por una Máquina de Turing, son los **Lenguajes Recursivamente Enumerables** (conjuntos recursivamente enumerables)

  - Cuando se inicia una máquina de Turing con una entrada, hay tres situaciones posibles: la máquina acepta la entrada, la máquina rechaza la entrada, o la máquina entra en un loop (nunca llega a un estado de aceptación o rechazo), en este caso la cadena no pertenece a $L(M)$.

Máquina de Turing: Definiciones

- **Decider**: Máquinas de Turing que se detienen (llegan a aceptación o rechazo) para toda entrada posible. Es decir, decide el lenguaje.

- Un lenguaje es **Decidible** (Turing-Decidable) si alguna Máquina de Turing lo decide (se detiene para todas las posibles entradas).

- El subconjunto de los Lenguajes Recursivamente Enumerables, que además son decidibles, son los **Lenguajes Recursivos** (Conjuntos Recursivos).

Funciones Computables

- Una Máquina de Turing cálcula el resultado de una función, si inicia con la entrada a la función en su cinta, y se detiene cuando el resultado de aplicar la función sobre esa entrada esta escrito en la cinta.

$$\text{Si } w' = f(w) \text{ entonces } q_o w \vdash^*_M q_f w'$$

Definición:

- Una función $f : \Sigma^* \rightarrow \Sigma^*$ es una **función computable** si existe alguna Máquina de Turing M que para toda entrada $w \in \Sigma^*$, se detiene cuando $f(w)$ esta en la cinta.

Variantes de la Máquina de Turing

- Máquinas de Turing Multi-cinta
  - Cambia la función de transición:

$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{ L, R \}^k$$

Donde k es el número de cintas.

Ejemplo: $\delta(q_i, a_1, a_2, .., a_k) = (q_j, b_1, b_2, \&, b_k, L, L, R, .., R)$

- Máquinas de Turing no Deterministas
  - Cambia la función de transición:

$$\delta: Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{ L, R \})$$

La computación de una máquina no determinista, es un árbol cuyas ramas corresponden a diferentes posibilidades para la máquina. Si alguna de las ramas llega a un estado de aceptación, la cadena de entrada es aceptada.

Variantes de la Máquina de Turing

- Enumeradores
  - Una máquina que tiene conectado un dispositivo de salida, una impresora donde puede imprimir cadenas.
  - La cadena de entrada es vacía, la cinta esta inicialmente en blanco.
  - La máquina enumera un lenguaje, imprime las cadenas que pertenecen al lenguaje.

Variantes de la Máquina de Turing

- Las variantes de la máquina de Turing tienen mayor poder de computo que la Máquina de Turing estándar?

  NO!

- Las variantes tienen el mismo poder, reconocen el mismo conjunto de lenguajes.

- Esta invarianza muestra la **Robustez** de la definición de la máquina.

Variantes de la Máquina de Turing

Toda Máquina de Turing Multi-cinta tiene una Máquina de Turing de una cinta equivalente

- La computación de máquina multi-cinta M se puede simular con una máquina de una sola cinta S.
- Si M tienen k cintas, S puede simular la k cintas, guardando su información en una cinta, usando un delimitador (ej. #) para separar los contenidos de las diferentes cintas, y teniendo una marca que le permita llevar control de las posiciones de la cabeza en cada "cinta" (ej. $\hat{a}$, puede denotar el símbolo a con la cabeza de lectura posicionada sobre él)
- Inicialmente la cinta para computar la entrada $w = w_1 w_2 \ldots w_n$

$$\# \hat{w}_1 w_2 \ldots w_n \# \wedge \# \wedge \# \ldots \# \wedge \#$$

Variantes de la Máquina de Turing

Toda Máquina de Turing No Determinista tiene una Máquina de Turing Determinista equivalente

- La computación de máquina no determinista N se puede simular con una máquina determinísta D de tres cintas .

- La primera cinta de D contiene la cadena de entrada, la segunda cinta tiene una copia de la cinta que se produce en algún momento de la simulación, y la tercera cinta lleva el control de la ubicación de la rama del árbol que se está computando

- Un nodo del árbol se representa con una cadena que pertenece a Σ_b*, donde Σ_b* = { 1, 2, .., b}, donde b la mayor cantidad de elecciones que tiene la máquina en una transición dada.
  Ejemplo: un nodo del árbol se representa como 13214 que indica que siguió el camino por el primer hijo de la raíz, de aquí por el tercer hijo…

Variantes de la Máquina de Turing

Toda Máquina de Turing No Determinista tiene una Máquina de Turing Determinista equivalente

- Para la simulación se debe realizar un recorrido primero en amplitud.
- Un recorrido primero en profundidad puede llevar a que la máquina quede en un recorrido infinito.

Variantes de la Máquina de Turing

Un lenguaje es Reconocible (Turing-recognizable) si y solo si existe algún enumerador que lo enumera

Demostración:

- Si un enumerador E enumera el lenguaje A, entonces es posible construir una TM M que reconoce A:
  - M = *" para la entrada w:*
    1. *Ejecute E. Cada vez que E genera una salida, compare la cadena de salida con w*
    2. *Si w alguna vez aparece en las salidas de E, acepte."*

Variantes de la Máquina de Turing

Un lenguaje es Reconocible (Turing-recognizable) si y solo si existe algún enumerador que lo enumera

Demostración:

- Si una una TM M reconoce el lenguaje A, entonces es posible construir un enumerador E para A:
  - Sea $s_1, s_2, s_3, \ldots$ La lista de todas las posibles cadenas de Σ^*
  - *E = " Para i = 1,2,3, …*
    - *Ejecute M para i pasos de cada entrada* $s_1, s_2, \ldots, s_i$
    - *Si M acepta, imprima el* s_j *correspondiente"*

La Tesis Church - Turing

- En 1936, Alonzo Church usó un sistema notacional, el λ-Calculo para hacer una definición de **Algoritmo**.

- También, en 1936, Alan Turing usó su Máquina, para hacer una definición de **Algoritmo.**

Estas dos definiciones son equivalentes

Y de ello se deriva la **intuición** de que cualquier computo que puede ser realizado por medios mecánicos puede ser realizado por una máquina de Turing

Todo lo que podemos hacer ahora con computadores, y lo que podamos hacer con ellos en el futuro corresponde con aquello que se puede realizar con una máquina de Turing?

La Tesis Church - Turing

- No ha sido probado, pero hay muchos argumentos a favor:
  - Formalismos como: el λ-Calculo, Post-systems, General Recursive Functions, y Modelos de máquinas como La RAM, son equivalentes en capacidad de computo a la Máquina de Turing
  - Todo lo que podemos hacer hoy con computadores digitales se puede realizar con una máquina de Turing
  - Nadie ha encontrado un problema solucionable por un algoritmo, que no sea solucionable por una Máquina de Turing

Definición de Algoritmo

- Informal: Un algoritmo (procedimiento, receta9 es un conjunto de instrucciones simples, para llevar a cabo una tarea

- Definición con Máquinas de Turing:

Un algoritmo para una función $f: D \rightarrow R$ es una Máquina de Turing que para cualquier entrada $d \in D$, eventualmente se detiene con la respuesta correcta $f(d) \in R$ en su cinta, es decir

Para todo $d \in D$, $q_o d \vdash^*_M q_f f(d)$, $q_f \in F$

"an algorithm is a computational process defined by a Turing machine."(Gurevich 2000:3)