# Máquinas de Turing.

## Autómatas y Lenguajes Formales 2017-2

Facultad de Ciencias
Universidad Nacional Autónoma de México.

# Introducción

Hasta el momento hemos estudiado dos modelos de computación: autómatas finitos y autómatas de pila.

En ambos casos, la máquina puede ejecutar únicamente algoritmos específicos. Por ejemplo, en el caso de un autómata finito, se ejecutan algoritmos en los cuales hay un límite muy marcado de la cantidad de información que se puede recordar; por otro lado, en los autómatas de pila se ejecutan algoritmos para los cuales es suficiente la información almacenada en una pila.

# Introducción

Sin embargo, tanto los autómatas finitos como los de pila tienen limitaciones que no permiten aceptar diversos lenguajes. Por ejemplo, un autómata finito no acepta

$$Pal = \{wcw^r \mid x \in \{a, b\}^*\}$$

Un autómata de pila sí acepta $Pal$, pero no acepta

$$ABC = \{a^n b^n c^n \mid n \in \mathbb{N}\} \qquad L = \{wcw \mid w \in \{a, b\}^*\}$$

Notamos que un autómata de pila con dos pilas acepta $ABC$ y que un autómata con una cola en vez de pila aceptaría a $L$. Ambas opciones proporcionan mayor poder de cómputo y podrían considerarse candidatas a un modelo de computación de propósito general.

El *Entscheidungsproblem* es el problema de decisión para el cálculo de primer orden: ¿Hay un procedimiento efectivo (algoritmo) el cual, dado un conjunto de axiomas y una proposición, decida si la proposición puede ser demostrada a partir de los axiomas?

En 1936, Alan Turing y Alonzo Church demostraron de manera independiente que no existe un algoritmo para resolver el problema de desición.

# Introducción

En el artículo *On Computable Numbers, with an Application to the Entscheidungsproblem*, se introdujo la *máquina de Turing*, una entidad matemática abstracta que formaliza el concepto de algoritmo. En este artículo, Turing reduce el problema de decisión al problema de la detención para máquinas de Turing.

Turing demostró la existencia de un problema irresoluble: ¿es posible construir un algoritmo que, dada una máquina de Turing cualquiera $M1$, nos diga si esa máquina se detendrá después de leer una cadena $w$, o bien seguirá funcionando indefinidamente?

# Introducción

El concepto de máquina de Turing es tan general y potente que es posible construir una máquina que sea capaz de simular el comportamiento de otra máquina de Turing cualquiera (*Máquina universal de Turing*).

Para el problema de la detención, Turing supuso que se puede resolver el problema:

- Entonces sería posible construir una máquina $M2$ que lo resuelva.

# Introducción

El concepto de máquina de Turing es tan general y potente que es posible construir una máquina que sea capaz de simular el comportamiento de otra máquina de Turing cualquiera (*Máquina universal de Turing*).

Para el problema de la detención, Turing supuso que se puede resolver el problema:

- Entonces sería posible construir una máquina $M2$ que lo resuelva.
- $M2$ simulará a $M1$ al leer la cadena $w$.

# Introducción

El concepto de máquina de Turing es tan general y potente que es posible construir una máquina que sea capaz de simular el comportamiento de otra máquina de Turing cualquiera (*Máquina universal de Turing*).

Para el problema de la detención, Turing supuso que se puede resolver el problema:

- Entonces sería posible construir una máquina $M2$ que lo resuelva.
- $M2$ simulará a $M1$ al leer la cadena $w$.
- Además, se puede configurar $M2$ de tal forma que, si $M1$ se detiene entonces $M2$ seguirá funcionando indefinidamente; por el contrario, si $M1$ no se detiene, $M2$ debe detenerse.

# Introducción

El concepto de máquina de Turing es tan general y potente que es posible construir una máquina que sea capaz de simular el comportamiento de otra máquina de Turing cualquiera (*Máquina universal de Turing*).

Para el problema de la detención, Turing supuso que se puede resolver el problema:

- Entonces sería posible construir una máquina $M2$ que lo resuelva.
- $M2$ simulará a $M1$ al leer la cadena $w$.
- Además, se puede configurar $M2$ de tal forma que, si $M1$ se detiene entonces $M2$ seguirá funcionando indefinidamente; por el contrario, si $M1$ no se detiene, $M2$ debe detenerse.
- Como $M2$ simula el comportamiento de cualquier MT, en particular puede simular su propia descripción. Lo cual nos lleva a una contradicción.

# Máquinas de Turing

Una **máquina de Turing** consiste de una cinta infinita dividida en casillas, cada una de las cuales contiene un símbolo de algún alfabeto de entrada, o bien, el símbolo de espacio en blanco. Sobre esta cinta actúa un dispositivo que transita por diferentes estados y que, en cada instante lee un símbolo de la casilla en la que está situado.Dependiendo del símbolo que ha leído y del estado en el que se encuentra, se realizan las siguientes acciones:

- El dispositivo pasa a un nuevo estado.

# Máquinas de Turing

Una **máquina de Turing** consiste de una cinta infinita dividida en casillas, cada una de las cuales contiene un símbolo de algún alfabeto de entrada, o bien, el símbolo de espacio en blanco. Sobre esta cinta actúa un dispositivo que transita por diferentes estados y que, en cada instante lee un símbolo de la casilla en la que está situado.Dependiendo del símbolo que ha leído y del estado en el que se encuentra, se realizan las siguientes acciones:

- El dispositivo pasa a un nuevo estado.
- Escribe un símbolo en la casilla en la que acaba de leer.

# Máquinas de Turing

Una **máquina de Turing** consiste de una cinta infinita dividida en casillas, cada una de las cuales contiene un símbolo de algún alfabeto de entrada, o bien, el símbolo de espacio en blanco. Sobre esta cinta actúa un dispositivo que transita por diferentes estados y que, en cada instante lee un símbolo de la casilla en la que está situado.Dependiendo del símbolo que ha leído y del estado en el que se encuentra, se realizan las siguientes acciones:

- El dispositivo pasa a un nuevo estado.
- Escribe un símbolo en la casilla en la que acaba de leer.
- Se desplaza una casilla hacia la izquierda, hacia la derecha, o bien la máquina se detiene.

# Máquinas de Turing

## Definición

Una **máquina de Turing** es una séptupla

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F \rangle$$

donde:

# Máquinas de Turing

## Definición

Una **máquina de Turing** es una séptupla

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F \rangle$$

donde:

- $Q \neq \emptyset$ es un conjunto finito de estados.

## Máquinas de Turing

### Definición

Una **máquina de Turing** es una séptupla

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F \rangle$$

donde:

- $Q \neq \emptyset$ es un conjunto finito de estados.
- $\Sigma$ es el alfabeto de entrada.

# Máquinas de Turing

## Definición

Una **máquina de Turing** es una séptupla

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \textvisiblespace, F \rangle$$

donde:

- $Q \neq \emptyset$ es un conjunto finito de estados.
- $\Sigma$ es el alfabeto de entrada.
- $\Gamma$ es el alfabeto de la cinta. Además $\Sigma \subseteq \Gamma$.

# Máquinas de Turing

## Definición

Una **máquina de Turing** es una séptupla

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \llcorner, F \rangle$$

donde:

- $Q \neq \emptyset$ es un conjunto finito de estados.
- $\Sigma$ es el alfabeto de entrada.
- $\Gamma$ es el alfabeto de la cinta. Además $\Sigma \subseteq \Gamma$.
- $\delta : Q \, x \, \Gamma \to Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$, es la función de transición.

# Máquinas de Turing

## Definición

Una **máquina de Turing** es una séptupla

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \text{␣}, F \rangle$$

donde:

- $Q \neq \emptyset$ es un conjunto finito de estados.
- $\Sigma$ es el alfabeto de entrada.
- $\Gamma$ es el alfabeto de la cinta. Además $\Sigma \subseteq \Gamma$.
- $\delta : Q \, x \, \Gamma \to Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$, es la función de transición.
- $q_0 \in Q$ es el estado inicial.

# Máquinas de Turing

## Definición

Una **máquina de Turing** es una séptupla

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \textvisiblespace, F \rangle$$

donde:

- $Q \neq \emptyset$ es un conjunto finito de estados.
- $\Sigma$ es el alfabeto de entrada.
- $\Gamma$ es el alfabeto de la cinta. Además $\Sigma \subseteq \Gamma$.
- $\delta : Q \, x \, \Gamma \rightarrow Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$, es la función de transición.
- $q_0 \in Q$ es el estado inicial.
- $\textvisiblespace \in \Gamma$ es el símbolo para espacio en blanco. Además $\textvisiblespace \notin \Sigma$.

# Máquinas de Turing

## Definición

Una **máquina de Turing** es una séptupla

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F \rangle$$

donde:

- $Q \neq \emptyset$ es un conjunto finito de estados.
- $\Sigma$ es el alfabeto de entrada.
- $\Gamma$ es el alfabeto de la cinta. Además $\Sigma \subseteq \Gamma$.
- $\delta : Q \, x \, \Gamma \to Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$, es la función de transición.
- $q_0 \in Q$ es el estado inicial.
- $\sqcup \in \Gamma$ es el símbolo para espacio en blanco. Además $\sqcup \notin \Sigma$.
- $F \subseteq Q$ es el conjunto de estados finales, el cual puede ser vacío.

# Máquinas de Turing

La función de transición está determinada por:

$$\delta : Q \, x \, \Gamma \rightarrow Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$$

Es decir, $\delta(q, a) = (p, b, D)$ se interpreta como:

- El estado actual es $q$ y el símbolo que se lee de la cinta es $a$.

# Máquinas de Turing

La función de transición está determinada por:

$$\delta : Q \, x \, \Gamma \rightarrow Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$$

Es decir, $\delta(q, a) = (p, b, D)$ se interpreta como:

- El estado actual es $q$ y el símbolo que se lee de la cinta es $a$.
- La transición es hacia el estado $p$.

# Máquinas de Turing

La función de transición está determinada por:

$$\delta : Q \, x \, \Gamma \rightarrow Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$$

Es decir, $\delta(q, a) = (p, b, D)$ se interpreta como:

- El estado actual es $q$ y el símbolo que se lee de la cinta es $a$.
- La transición es hacia el estado $p$.
- Se escribe el símbolo $b$ en la misma casilla donde se leyó el símbolo $a$.

# Máquinas de Turing

La función de transición está determinada por:

$$\delta : Q \, x \, \Gamma \rightarrow Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$$

Es decir, $\delta(q, a) = (p, b, D)$ se interpreta como:

- El estado actual es $q$ y el símbolo que se lee de la cinta es $a$.
- La transición es hacia el estado $p$.
- Se escribe el símbolo $b$ en la misma casilla donde se leyó el símbolo $a$.
- El dispositivo de lectura/escritura (cabeza) se mueve una celda en la dirección dada por $D \in \{\leftarrow, \rightarrow\}$.

# Máquinas de Turing

La función de transición está determinada por:

$$\delta : Q \, x \, \Gamma \rightarrow Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$$

Es decir, $\delta(q, a) = (p, b, D)$ se interpreta como:

- El estado actual es $q$ y el símbolo que se lee de la cinta es $a$.
- La transición es hacia el estado $p$.
- Se escribe el símbolo $b$ en la misma casilla donde se leyó el símbolo $a$.
- El dispositivo de lectura/escritura (cabeza) se mueve una celda en la dirección dada por $D \in \{\leftarrow, \rightarrow\}$.

# Máquinas de Turing

La función de transición está determinada por:

$$\delta : Q \, x \, \Gamma \rightarrow Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\}$$

Es decir, $\delta(q, a) = (p, b, D)$ se interpreta como:

- El estado actual es $q$ y el símbolo que se lee de la cinta es $a$.
- La transición es hacia el estado $p$.
- Se escribe el símbolo $b$ en la misma casilla donde se leyó el símbolo $a$.
- El dispositivo de lectura/escritura (cabeza) se mueve una celda en la dirección dada por $D \in \{\leftarrow, \rightarrow\}$. Este movimiento del dispositivo se realiza únicamente después de leer $a$ y escribir $b$ en la cinta.

# Máquinas de Turing

## Definición

Una **configuración o descripción instantánea** es de la forma

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

y está determinada por:

# Máquinas de Turing

## Definición

Una **configuración o descripción instantánea** es de la forma

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

y está determinada por:

- El estado actual de la cabeza de lectura/escritura.

# Máquinas de Turing

## Definición

Una **configuración o descripción instantánea** es de la forma

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

y está determinada por:

- El estado actual de la cabeza de lectura/escritura.
- El contenido de la cinta.

# Máquinas de Turing

## Definición

Una **configuración o descripción instantánea** es de la forma

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

y está determinada por:

- El estado actual de la cabeza de lectura/escritura.
- El contenido de la cinta.
- La posición de la cabeza de lectura/escritura.

## Máquinas de Turing

### Definición

Un **cómputo o paso de computación** es el cambio de una descripción instantánea a otra mediante una transición dada por $\delta$:

$w_1 \mathbf{q} a w_2 \vdash w_1 b \mathbf{p} w_2$ si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

$w_1 c \mathbf{q} a w_2 \vdash w_1 \mathbf{p} c b w_2$ si y sólo si $\delta(q, a) = (p, b, \leftarrow)$

Nota:
$w_1 \mathbf{q} a \vdash w_1 b \mathbf{p} \llcorner\!\lrcorner$ si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

$\mathbf{q} a w_2 \vdash \mathbf{p} \llcorner\!\lrcorner b w_2$ si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

# Máquinas de Turing

## Definición

El lenguaje de aceptación se define como todas aquellas cadenas de entrada con las cuales la máquina se detiene en un estado final:

$$L(M) = \{w \in \Sigma^* \mid q_0 w \vdash^* w_1 q_f w2,\ q_f \in F\}$$

Nota:

# Máquinas de Turing

## Definición

El lenguaje de aceptación se define como todas aquellas cadenas de entrada con las cuales la máquina se detiene en un estado final:

$$L(M) = \{w \in \Sigma^* \mid q_0 w \vdash^* w_1 q_f w2,\ q_f \in F\}$$

Nota:

- A diferencia de los autómatas, se acepta una cadena en el momento en el que la máquina se detiene en un estado final.

# Máquinas de Turing

## Definición

El lenguaje de aceptación se define como todas aquellas cadenas de entrada con las cuales la máquina se detiene en un estado final:

$$L(M) = \{w \in \Sigma^* \mid q_0 w \vdash^* w_1 q_f w2,\ q_f \in F\}$$

Nota:

- A diferencia de los autómatas, se acepta una cadena en el momento en el que la máquina se detiene en un estado final.
- No es necesario consumir toda la cadena.

# Máquinas de Turing

## Definición

El lenguaje de aceptación se define como todas aquellas cadenas de entrada con las cuales la máquina se detiene en un estado final:

$$L(M) = \{w \in \Sigma^* \mid q_0 w \vdash^* w_1 q_f w2,\ q_f \in F\}$$

Nota:

- A diferencia de los autómatas, se acepta una cadena en el momento en el que la máquina se detiene en un estado final.
- No es necesario consumir toda la cadena.
- Únicamente en caso de no haber estados finales, se acepta una cadena en el momento en que la máquina se detiene.

# Máquinas de Turing

Ejemplo: Una máquina de Turing que acepta el lenguaje

$$L = \{w | w \in \{0,1\}^*, w \text{ tiene un número par de ceros}\}$$

está determinada por :

| $\delta$ | $0$ | $1$ | $\sqcup$ |
|---|---|---|---|
| $q_0$ | $(q_1, 0, \rightarrow)$ | $(q_0, 1, \rightarrow)$ | $(q_f, \sqcup, -)$ |
| $q_1$ | $(q_0, 0, \rightarrow)$ | $(q_1, 1, \rightarrow)$ | |
| $q_f$ | | | |

# Máquinas de Turing

Ejemplo: Una máquina de Turing que acepta al lenguaje

$$L = \{a^n b^n | n \geq 1\}$$

está determinada por :

| $\delta$ | a | b | X | Y | ␣ |
|---|---|---|---|---|---|
| $q_0$ | $(q_1, X, \rightarrow)$ | | | $(q_3, Y, \rightarrow)$ | |
| $q_1$ | $(q_1, a, \rightarrow)$ | $(q_2, Y, \leftarrow)$ | | $(q_1, Y, \rightarrow)$ | |
| $q_2$ | $(q_2, a, \leftarrow)$ | | $(q_0, X, \rightarrow)$ | $(q_2, Y, \leftarrow)$ | |
| $q_3$ | | | | $(q_3, Y, \rightarrow)$ | $(q_4, ␣, \rightarrow)$ |
| $q_4$ | | | | | |

# Máquinas de Turing

Ejemplo: La siguiente máquina recibe como entrada $w \in \{1\}^*$, ¿Qué genera como salida?

| $\delta$ | $1$ | $0$ | $\sqcup$ |
|---|---|---|---|
| $q_0$ | $(q_1, 0, \rightarrow)$ | $(q_0, 0, \leftarrow)$ | $(q_2, \sqcup, \rightarrow)$ |
| $q_1$ | $(q_1, 1, \rightarrow)$ | $(q_1, 0, \rightarrow)$ | $(q_0, 0, \leftarrow)$ |
| $q_2$ | | $(q_2, 1, \rightarrow)$ | $(q_f, \sqcup, -)$ |
| $q_f$ | | | |

## Variantes de Máquinas de Turing

Existen diversas variantes en la definición de máquinas de Turing. Todas ellas resultan equivalentes, es decir, no añaden poder computacional; sin embargo, sí representan ventajas prácticas en la presentación o programación de diversos problemas.

**MT con cabeza lectora estacionaria.**

- Es permitido que al leer y escribir un símbolo, la cabeza no realice desplazamiento alguno.

# Variantes de Máquinas de Turing

Existen diversas variantes en la definición de máquinas de Turing. Todas ellas resultan equivalentes, es decir, no añaden poder computacional; sin embargo, sí representan ventajas prácticas en la presentación o programación de diversos problemas.

**MT con cabeza lectora estacionaria.**

- Es permitido que al leer y escribir un símbolo, la cabeza no realice desplazamiento alguno.
- El conjunto de direcciones se amplía a $\{\leftarrow, \rightarrow, -\}$

# Variantes de Máquinas de Turing

Existen diversas variantes en la definición de máquinas de Turing. Todas ellas resultan equivalentes, es decir, no añaden poder computacional; sin embargo, sí representan ventajas prácticas en la presentación o programación de diversos problemas.

**MT con cabeza lectora estacionaria.**

- Es permitido que al leer y escribir un símbolo, la cabeza no realice desplazamiento alguno.
- El conjunto de direcciones se amplía a $\{\leftarrow, \rightarrow, -\}$
- La transición $\delta(q, a) = (p, b, -)$ , significa que se pasa del estado $q$ a $p$, se lee $a$, se escribe $b$, y no se mueve la cabeza.

# Variantes de Máquinas de Turing

**Máquina de Turing con múltiples pistas.**

- La idea es dividir la cinta en múltiples pistas, por lo que la función de transición es:

$$\delta : Q \, x \, \Gamma^n \to Q \, x \, \Gamma^n \, x \, \{\leftarrow, \rightarrow\}$$

  es decir, $\delta(q, \langle a_1, ..., a_n \rangle) = (p, \langle b_1, ..., b_n \rangle, D)$.

**Máquina de Turing con múltiples cintas.**

# Variantes de Máquinas de Turing

**Máquina de Turing con múltiples pistas.**

- La idea es dividir la cinta en múltiples pistas, por lo que la función de transición es:

$$\delta : Q \, x \, \Gamma^n \rightarrow Q \, x \, \Gamma^n \, x \, \{\leftarrow, \rightarrow\}$$

es decir, $\delta(q, \langle a_1, ..., a_n \rangle) = (p, \langle b_1, ..., b_n \rangle, D)$.

**Máquina de Turing con múltiples cintas.**

- La idea es agregar cintas a la máquina, por lo que la función de transición es:

$$\delta : Q \, x \, \Gamma^n \rightarrow Q \, x \, (\Gamma^n \, x \, \{\leftarrow, \rightarrow\})^n$$

es decir, $\delta(q, \langle a_1, ..., a_n \rangle) = (p, (\langle b_1, D_1 \rangle, ..., \langle b_n, D_n \rangle))$.

# Variantes de Máquinas de Turing

**Máquina de Turing no determinista.**

Se caracteriza por la función de transición:

$$\delta : Q \, x \, \Gamma \rightarrow \mathcal{P}(Q \, x \, \Gamma \, x \, \{\leftarrow, \rightarrow\})$$

es decir, $\delta(q, a) = \{\langle p_1, b_1, D_1\rangle, ..., \langle p_n, b_n, D_n\rangle\})$.

# Máquinas de Turing

**Ejercicio 1:** Diseñar una máquina de Turing $M$ que toma como entrada una cadena del lenguaje $a^i b^j \mid 0 \leq j \leq i\}$ y agrega las b's necesarias para igualar el número de b's con el número de a's.

Por ejemplo:
Entrada: *aaab*
Salida (el contenido de la cinta cuando $M$ se detiene): *aaabbb*.

**Ejercicio 2:** Construir una máquina que recibe una cadena $w \in \{1\}$ y cuya salida es $w^3$.

## Máquinas Generadoras

Podemos utilizar una máquina de Turing principalmente con tres objetivos:

- Generar las cadenas de un lenguaje.

# Máquinas Generadoras

Podemos utilizar una máquina de Turing principalmente con tres objetivos:

- Generar las cadenas de un lenguaje.
- Reconocer un lenguaje.

# Máquinas Generadoras

Podemos utilizar una máquina de Turing principalmente con tres objetivos:

- Generar las cadenas de un lenguaje.
- Reconocer un lenguaje.
- Calcular una función.

# Máquinas Generadoras

Podemos utilizar una máquina de Turing principalmente con tres objetivos:

- Generar las cadenas de un lenguaje.
- Reconocer un lenguaje.
- Calcular una función.

# Máquinas Generadoras

Podemos utilizar una máquina de Turing principalmente con tres objetivos:

- Generar las cadenas de un lenguaje.
- Reconocer un lenguaje.
- Calcular una función.

**MT Generadoras.**
Una máquina de Turing $M$ genera el lenguaje $L \subseteq \Sigma^*$ si:

- $M$ inicia en el estado $q_0$ con una cinta en blanco.

## Máquinas Generadoras

Podemos utilizar una máquina de Turing principalmente con tres objetivos:

- Generar las cadenas de un lenguaje.
- Reconocer un lenguaje.
- Calcular una función.

**MT Generadoras.**
Una máquina de Turing $M$ genera el lenguaje $L \subseteq \Sigma^*$ si:

- $M$ inicia en el estado $q_0$ con una cinta en blanco.
- Cada vez que $M$ regresa a $q_0$ hay una cadena de $L$ escrita sobre la cinta.

# Máquinas Generadoras

Podemos utilizar una máquina de Turing principalmente con tres objetivos:

- Generar las cadenas de un lenguaje.
- Reconocer un lenguaje.
- Calcular una función.

**MT Generadoras.**
Una máquina de Turing $M$ genera el lenguaje $L \subseteq \Sigma^*$ si:

- $M$ inicia en el estado $q_0$ con una cinta en blanco.
- Cada vez que $M$ regresa a $q_0$ hay una cadena de $L$ escrita sobre la cinta.
- En algún momento se habrán escrito todas las cadenas de $L$ en la cinta.

# Máquinas Reconocedoras

Sean $w \in \Sigma^*$ y $M$ una máquina de Turing cuyo estado inicial es $q_0$ y dos estados de detención $q_a$ y $q_r$. Decimos que:

- **M acepta w** si y sólo si $q_0 w \vdash^* q_a w'$ , para alguna $w'$.

# Máquinas Reconocedoras

Sean $w \in \Sigma^*$ y $M$ una máquina de Turing cuyo estado inicial es $q_0$ y dos estados de detención $q_a$ y $q_r$. Decimos que:

- **M acepta w** si y sólo si $q_0 w \vdash^* q_a w'$ , para alguna $w'$.
- **M rechaza w** si y sólo si $q_0 w \vdash^* q_r w'$ , para alguna $w'$.

# Máquinas Reconocedoras

Sean $w \in \Sigma^*$ y $M$ una máquina de Turing cuyo estado inicial es $q_0$ y dos estados de detención $q_a$ y $q_r$. Decimos que:

- **M acepta w** si y sólo si $q_0 w \vdash^* q_a w'$ , para alguna $w'$.
- **M rechaza w** si y sólo si $q_0 w \vdash^* q_r w'$ , para alguna $w'$.

# Máquinas Reconocedoras

Sean $w \in \Sigma^*$ y $M$ una máquina de Turing cuyo estado inicial es $q_0$ y dos estados de detención $q_a$ y $q_r$. Decimos que:

- **M acepta w** si y sólo si $q_0 w \vdash^* q_a w'$ , para alguna $w'$.
- **M rechaza w** si y sólo si $q_0 w \vdash^* q_r w'$ , para alguna $w'$.

Importante notar que si $M$ no se detiene, la cadena no se acepta ni se rechaza.

Un lenguaje $L$ es **decidible** si y sólo si existe una máquina de Turing $M$ que lo *decide*.
Sea $\Sigma$ el alfabeto de entrada de $M$. Decimos que $M$ **decide** a $L \subseteq \Sigma^*$, si y sólo si para cualquier cadena $w \in \sigma^*$, sucede que:

- Si $w \in L$ entonces $M$ acepta $w$.

# Máquinas Reconocedoras

Sean $w \in \Sigma^*$ y $M$ una máquina de Turing cuyo estado inicial es $q_0$ y dos estados de detención $q_a$ y $q_r$. Decimos que:

- **M acepta w** si y sólo si $q_0 w \vdash^* q_a w'$ , para alguna $w'$.
- **M rechaza w** si y sólo si $q_0 w \vdash^* q_r w'$ , para alguna $w'$.

Importante notar que si $M$ no se detiene, la cadena no se acepta ni se rechaza.

Un lenguaje $L$ es **decidible** si y sólo si existe una máquina de Turing $M$ que lo *decide*.
Sea $\Sigma$ el alfabeto de entrada de $M$. Decimos que $M$ **decide** a $L \subseteq \Sigma^*$, si y sólo si para cualquier cadena $w \in \sigma^*$, sucede que:

- Si $w \in L$ entonces $M$ acepta $w$.
- Si $w \notin L$ entonces $M$ rechaza $w$.

## Máquinas Calculadoras

El objetivo de estas máquinas es calcular una función sin necesidad de aceptar o generar cadenas de un lenguaje. La máquina de Turing

$$M = \langle \{q_0, q_f\}, \Sigma, \Gamma, \delta, q_0, \sqcup, \emptyset \rangle$$

calcula una función $f : \Sigma^* \to \Gamma^*$ si

$$q_0 w \vdash^* q_f v \quad \text{donde } f(w) = v.$$

Notas:

- No hay estados finales. El estado $q_f$ se utiliza para detener la máquina, es decir, no hay transiciones desde $q_f$.

# Máquinas Calculadoras

El objetivo de estas máquinas es calcular una función sin necesidad de aceptar o generar cadenas de un lenguaje. La máquina de Turing

$$M = \langle \{q_0, q_f\}, \Sigma, \Gamma, \delta, q_0, \textvisiblespace, \emptyset \rangle$$

calcula una función $f : \Sigma^* \to \Gamma^*$ si

$$q_0 w \vdash^* q_f v \quad \text{donde } f(w) = v.$$

Notas:

- No hay estados finales. El estado $q_f$ se utiliza para detener la máquina, es decir, no hay transiciones desde $q_f$.
- El proceso termina en la configuración $q_f v$, es decir, la cabeza debe estar leyendo el primer símbolo de la salida $v$.