Autómatas y Lenguajes Formales

Tema 13: Máquinas de Turing y computabilidad

Dr. Favio Ezequiel Miranda Perea
favio@ciencias.unam.mx

Facultad de Ciencias UNAM[1]

27 de abril de 2019

[1] Con el apoyo del proyecto PAPIME PE102117

Máquinas de Turing

Introducción

- Las máquinas de Turing (MT)son máquinas idealizadas capaces de realizar cómputos.

Máquinas de Turing

Introducción

- Las máquinas de Turing (MT)son máquinas idealizadas capaces de realizar cómputos.

- Una MT consiste de una cinta infinita dividida en sectores (cuadros) y una cabeza de lectoescritura.

Máquinas de Turing

Introducción

- Las máquinas de Turing (MT)son máquinas idealizadas capaces de realizar cómputos.
- Una MT consiste de una cinta infinita dividida en sectores (cuadros) y una cabeza de lectoescritura.
- Cada sector de la cinta contiene un símbolo de cierto alfabeto de entrada o bien el símbolo blanco.

Máquinas de Turing

Introducción

- Las máquinas de Turing (MT)son máquinas idealizadas capaces de realizar cómputos.
- Una MT consiste de una cinta infinita dividida en sectores (cuadros) y una cabeza de lectoescritura.
- Cada sector de la cinta contiene un símbolo de cierto alfabeto de entrada o bien el símbolo blanco.
- La cabeza lee el sector y puede escribir sobre él.

Máquinas de Turing

Introducción

- Las máquinas de Turing (MT)son máquinas idealizadas capaces de realizar cómputos.
- Una MT consiste de una cinta infinita dividida en sectores (cuadros) y una cabeza de lectoescritura.
- Cada sector de la cinta contiene un símbolo de cierto alfabeto de entrada o bien el símbolo blanco.
- La cabeza lee el sector y puede escribir sobre él.
- La cabeza puede moverse a izquierda o derecha.

Máquinas de Turing
Definición

Una máquina de Turing es una septupla:

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

Máquinas de Turing
Definición

Una máquina de Turing es una septupla:

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

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

Máquinas de Turing
Definición

Una máquina de Turing es una septupla:

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

- $Q \neq \varnothing$ es un conjunto finito de estados.
- Σ es el alfabeto de entrada.

Máquinas de Turing
Definición

Una máquina de Turing es una septupla:

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

- $Q \neq \varnothing$ es un conjunto finito de estados.
- Σ es el alfabeto de entrada.
- Γ es el alfabeto de la cinta, el cual incluye a Σ, es decir, $\Sigma \subseteq \Gamma$.

Máquinas de Turing

Definición

Una máquina de Turing es una septupla:

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

- $Q \neq \varnothing$ es un conjunto finito de estados.
- Σ es el alfabeto de entrada.
- Γ es el alfabeto de la cinta, el cual incluye a Σ, es decir, $\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \to\}$ es la función de transición

Máquinas de Turing

Definición

Una máquina de Turing es una septupla:

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

- $Q \neq \varnothing$ es un conjunto finito de estados.
- Σ es el alfabeto de entrada.
- Γ es el alfabeto de la cinta, el cual incluye a Σ, es decir, $\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \to\}$ es la función de transición
- $q_0 \in Q$ es el estado inicial.

Máquinas de Turing
Definición

Una máquina de Turing es una septupla:

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

- $Q \neq \varnothing$ es un conjunto finito de estados.
- Σ es el alfabeto de entrada.
- Γ es el alfabeto de la cinta, el cual incluye a Σ, es decir, $\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \to\}$ es la función de transición
- $q_0 \in Q$ es el estado inicial.
- $\sqcup \in \Gamma$ es el símbolo blanco tal que $\sqcup \notin \Sigma$.

Máquinas de Turing

Definición

Una máquina de Turing es una septupla:

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

- $Q \neq \varnothing$ es un conjunto finito de estados.
- Σ es el alfabeto de entrada.
- Γ es el alfabeto de la cinta, el cual incluye a Σ, es decir, $\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ es la función de transición
- $q_0 \in Q$ es el estado inicial.
- $\sqcup \in \Gamma$ es el símbolo blanco tal que $\sqcup \notin \Sigma$.
- $F \subseteq Q$ es el conjunto de estados finales. F podría ser vacío.

Función de transición

Máquinas de Turing

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

$$\delta(q, a) = (p, b, D)$$

Función de transición

Máquinas de Turing

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

$$\delta(q, a) = (p, b, D)$$

- El estado actual es q y el símbolo a leer es a.

Función de transición

Máquinas de Turing

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

$$\delta(q, a) = (p, b, D)$$

- El estado actual es q y el símbolo a leer es a.
- La transición es hacia el estado p

Función de transición

Máquinas de Turing

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \to\}$$

$$\delta(q, a) = (p, b, D)$$

- El estado actual es q y el símbolo a leer es a.
- La transición es hacia el estado p
- b es el símbolo escrito en lugar de a.

Función de transición

Máquinas de Turing

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

$$\delta(q, a) = (p, b, D)$$

- El estado actual es q y el símbolo a leer es a.
- La transición es hacia el estado p
- b es el símbolo escrito en lugar de a.
- La cabeza se mueve una celda según la dirección dada por $D \in \{\leftarrow, \rightarrow\}$. Dichos movimientos se realizan después de leer a y escribir b.

Ejemplos
Máquinas de Turing

$$\delta(q, a) = (p, b, \rightarrow)$$

- Estado actual: q

Ejemplos

Máquinas de Turing

$$\delta(q, a) = (p, b, \rightarrow)$$

- Estado actual: q
- Símbolo a leer: a.

Ejemplos
Máquinas de Turing

$$\delta(q, a) = (p, b, \rightarrow)$$

- Estado actual: q
- Símbolo a leer: a.
- La cabeza borra a y escribe b.

Ejemplos

Máquinas de Turing

$$\delta(q, a) = (p, b, \rightarrow)$$

- Estado actual: q
- Símbolo a leer: a.
- La cabeza borra a y escribe b.
- El nuevo estado es p.

Ejemplos

Máquinas de Turing

$$\delta(q, a) = (p, b, \rightarrow)$$

- Estado actual: q
- Símbolo a leer: a.
- La cabeza borra a y escribe b.
- El nuevo estado es p.
- La cabeza se mueve una celda a la derecha.

Ejemplos
Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \{q_1\} \rangle$$

$$\delta(q_0, a) = (q_0, b, \rightarrow)$$

Ejemplos

Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \{q_1\} \rangle$$

$$\delta(q_0, a) = (q_0, b, \rightarrow)$$

$$\delta(q_0, b) = (q_0, b, \rightarrow)$$

Ejemplos

Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \{q_1\} \rangle$$

$$\delta(q_0, a) = (q_0, b, \rightarrow)$$

$$\delta(q_0, b) = (q_0, b, \rightarrow)$$

$$\delta(q_0, \sqcup) = (q_1, \sqcup, \leftarrow)$$

$L = \{w \in \{0,1\}^\star \mid w$ tiene un número par de ceros $\}$

Ejemplos

δ	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			

Ejemplos

Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \varnothing \rangle$$

$$\delta(q_0, a) = (q_1, a, \rightarrow)$$

Ejemplos

Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \varnothing \rangle$$

$$\delta(q_0, a) = (q_1, a, \rightarrow)$$

$$\delta(q_0, b) = (q_1, b, \rightarrow)$$

Ejemplos
Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \varnothing \rangle$$

$$\delta(q_0, a) = (q_1, a, \rightarrow)$$

$$\delta(q_0, b) = (q_1, b, \rightarrow)$$

$$\delta(q_0, \sqcup) = (q_1, \sqcup, \rightarrow)$$

Ejemplos

Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \varnothing \rangle$$

$$\delta(q_0, a) = (q_1, a, \rightarrow)$$

$$\delta(q_0, b) = (q_1, b, \rightarrow)$$

$$\delta(q_0, \sqcup) = (q_1, \sqcup, \rightarrow)$$

$$\delta(q_1, a) = (q_0, a, \leftarrow)$$

Ejemplos

Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \textvisiblespace\}, \delta, q_0, \textvisiblespace, \varnothing \rangle$$

$$\delta(q_0, a) = (q_1, a, \rightarrow)$$

$$\delta(q_0, b) = (q_1, b, \rightarrow)$$

$$\delta(q_0, \textvisiblespace) = (q_1, \textvisiblespace, \rightarrow)$$

$$\delta(q_1, a) = (q_0, a, \leftarrow)$$

$$\delta(q_1, b) = (q_0, b, \leftarrow)$$

Ejemplos
Máquinas de Turing

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \varnothing \rangle$$

$$\delta(q_0, a) = (q_1, a, \rightarrow)$$

$$\delta(q_0, b) = (q_1, b, \rightarrow)$$

$$\delta(q_0, \sqcup) = (q_1, \sqcup, \rightarrow)$$

$$\delta(q_1, a) = (q_0, a, \leftarrow)$$

$$\delta(q_1, b) = (q_0, b, \leftarrow)$$

$$\delta(q_1, \sqcup) = (q_0, \sqcup, \leftarrow)$$

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

Ejemplos

δ	a	b	X	Y	$\sqcup$
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				$(q_3, Y, \rightarrow)$	$(q_4, \sqcup, \rightarrow)$
q_4					

Máquina Estandar de Turing

Máquinas de Turing

- La cinta es infinita en ambas direcciones.

Máquina Estandar de Turing

Máquinas de Turing

- La cinta es infinita en ambas direcciones.
- Se permite un número arbitrario de movimientos en cualquier dirección.

Máquina Estandar de Turing

Máquinas de Turing

- La cinta es infinita en ambas direcciones.
- Se permite un número arbitrario de movimientos en cualquier dirección.
- La máquina es determinista, δ define a lo más un movimiento para cada configuración posible.

Máquina Estandar de Turing

Máquinas de Turing

- La cinta es infinita en ambas direcciones.
- Se permite un número arbitrario de movimientos en cualquier dirección.
- La máquina es determinista, δ define a lo más un movimiento para cada configuración posible.
- No hay transiciones desde estados finales, es decir, $\delta(q, a)$ no está definida si $q \in F$.

Máquina Estandar de Turing

Máquinas de Turing

- La cinta es infinita en ambas direcciones.
- Se permite un número arbitrario de movimientos en cualquier dirección.
- La máquina es determinista, δ define a lo más un movimiento para cada configuración posible.
- No hay transiciones desde estados finales, es decir, $\delta(q, a)$ no está definida si $q \in F$.
- No hay un archivo especial de entrada o salida, se asume que la máquina contiene algo al final y al principio del proceso.

Configuraciones

Máquinas de Turing

Una configuración o descripción instantanea

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

Configuraciones

Máquinas de Turing

Una configuración o descripción instantanea

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

está determinada por:

- El estado actual de la unidad de control (cabeza).

Configuraciones

Máquinas de Turing

Una configuración o descripción instantanea

$$a_1 a_2 \ldots a_{k-1} \color{red}{q} \color{black} a_k a_{k+1} \ldots a_n$$

está determinada por:

- El estado actual de la unidad de control (cabeza).
- El contenido de la cinta.

Configuraciones
Máquinas de Turing

Una configuración o descripción instantanea

$$a_1 a_2 \dots a_{k-1} \textcolor{red}{q} a_k a_{k+1} \dots a_n$$

está determinada por:

- El estado actual de la unidad de control (cabeza).
- El contenido de la cinta.
- La posición de la unidad de control.

Configuraciones

Máquinas de Turing

Una configuración o descripción instantanea

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

está determinada por:

- El estado actual de la unidad de control (cabeza).
- El contenido de la cinta.
- La posición de la unidad de control.
- Configuración inicial: $\color{red}{q_0} w$

Cómputos

Máquinas de Turing

Un cómputo o paso de computación es el cambio de una descripción instantanea a otra mediante una transición dada por δ.

$$uqav \vdash ubpv$$

si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

Cómputos

Máquinas de Turing

Un cómputo o paso de computación es el cambio de una descripción instantanea a otra mediante una transición dada por δ.

$$uqav \vdash ubpv$$

si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

$$ucqav \vdash upcbv$$

si y sólo si $\delta(q, a) = (p, b, \leftarrow)$

Cómputos

Máquinas de Turing

Un cómputo o paso de computación es el cambio de una descripción instantanea a otra mediante una transición dada por δ.

$$uqav \vdash ubpv$$

si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

$$ucqav \vdash upcbv$$

si y sólo si $\delta(q, a) = (p, b, \leftarrow)$

$\vdash^\star$ se define de la manera usual.

Cómputos

Casos especiales con ε

$$u{\color{red}q}a \vdash ub{\color{red}p}\sqcup$$

Cómputos

Casos especiales con ε

$$u q a \vdash u b p \sqcup$$

si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

Cómputos

Casos especiales con ε

$$u\color{red}{q}a \vdash ub\color{red}{p}\textvisiblespace$$

si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

$$\color{red}{q}av \vdash \color{red}{p}\textvisiblespace bv$$

Cómputos

Casos especiales con ε

$$uqa \vdash ubp\sqcup$$

si y sólo si $\delta(q, a) = (p, b, \rightarrow)$

$$qav \vdash p\sqcup bv$$

si y sólo si $\delta(q, a) = (p, b, \leftarrow)$

Situaciones especiales

Cómputos

- Cómputos bloqueados: el cómputo se bloquea porque la siguiente transición no está definida.

$$u{\color{red}q}v \not\vdash^{\star}$$

Situaciones especiales

Cómputos

- Cómputos bloqueados: el cómputo se bloquea porque la siguiente transición no está definida.

$$u\textcolor{red}{q}v \not\vdash^{\star}$$

- Cómputos infinitos: el cómputo entra en un ciclo infinito.

$$u\textcolor{red}{q}v \vdash^{\star} \infty$$

Lenguaje de aceptación

Máquinas de Turing

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

$$L(M) = \{w \in \Sigma^\star \mid q_0 w \vdash^\star w_1 q_f w_2 \ \ q_f \in F\}$$

Lenguaje de aceptación

Máquinas de Turing

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

$$L(M) = \{w \in \Sigma^\star \mid q_0 w \vdash^\star w_1 q_f w_2 \;\; q_f \in F\}$$

Observaciones:

- A diferencia con los autómatas se acepta una cadena en el momento en que el proceso llega a un estado final.

Lenguaje de aceptación

Máquinas de Turing

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

$$L(M) = \{w \in \Sigma^\star \mid q_0 w \vdash^\star w_1 q_f w_2 \;\; q_f \in F\}$$

Observaciones:

- A diferencia con los autómatas se acepta una cadena en el momento en que el proceso llega a un estado final.

- No es necesario consumir toda la cadena.

Lenguaje de aceptación

Máquinas de Turing

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

$$L(M) = \{w \in \Sigma^{\star} \mid q_0 w \vdash^{\star} w_1 q_f w_2 \;\; q_f \in F\}$$

Observaciones:

- A diferencia con los autómatas se acepta una cadena en el momento en que el proceso llega a un estado final.
- No es necesario consumir toda la cadena.
- Si no hay estados finales se acepta una cadena en el momento en que la máquina se detiene.

Variaciones

MT

- Existen diversas variaciones en la definición de MT.

Variaciones
MT

- Existen diversas variaciones en la definición de MT.
- Todas ellas resultan equivalentes, es decir, el poder de computación de cualquier modelo resulta equivalente al de la máquina estandar.

Variaciones

MT

- Existen diversas variaciones en la definición de MT.
- Todas ellas resultan equivalentes, es decir, el poder de computación de cualquier modelo resulta equivalente al de la máquina estandar.
- Las variaciones son útiles para simplificar la presentación o programación de diversos problemas.

MT con cabeza lectora estacionaria

Variaciones

- Se permite que al leer y escribir un símbolo la cabeza no realice movimiento alguno.

MT con cabeza lectora estacionaria

Variaciones

- Se permite que al leer y escribir un símbolo la cabeza no realice movimiento alguno.
- El conjunto de direcciones se amplia a $\{\leftarrow, \rightarrow, -\}$.

MT con cabeza lectora estacionaria

Variaciones

- Se permite que al leer y escribir un símbolo la cabeza no realice movimiento alguno.

- El conjunto de direcciones se amplia a $\{\leftarrow, \rightarrow, -\}$.
- La transición
$$\delta(q, a) = (p, b, -)$$
significa que la cabeza lee a, escribe b y no se mueve

MT con cabeza lectora estacionaria

Variaciones

- Se permite que al leer y escribir un símbolo la cabeza no realice movimiento alguno.
- El conjunto de direcciones se amplia a $\{\leftarrow, \rightarrow, -\}$.
- La transición

$$\delta(q, a) = (p, b, -)$$

  significa que la cabeza lee *a*, escribe *b* y no se mueve
- Tales transiciones pueden simularse mediante un nuevo estado y movimientos consecutivos a la izquierda y a la derecha.

MT con múltiples pistas

Variaciones

- Idea: la cinta se divide en multiples pistas.

MT con múltiples pistas
Variaciones

- Idea: la cinta se divide en multiples pistas.
- La función de transición es:

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

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

MT con múltiples cintas

Variaciones

- Idea: se agregan más cintas a la máquina.

MT con múltiples cintas

Variaciones

- Idea: se agregan más cintas a la máquina.
- La función de transición es:

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

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

MT No-determinista

Variaciones

- La función de transición es:

$$\delta : Q \times \Gamma^n \to \mathcal{P}(Q \times \Gamma \times \{\leftarrow, \rightarrow\})$$

$$\delta(q, \langle a_1, \ldots, a_n \rangle) = \{\langle b_1, D_1 \rangle, \ldots, \langle b_n, D_n \rangle\}$$

MT No-determinista

Variaciones

- La función de transición es:

$$\delta : Q \times \Gamma^n \to \mathcal{P}(Q \times \Gamma \times \{\leftarrow, \rightarrow\})$$

$$\delta(q, \langle a_1, \ldots, a_n \rangle) = \{\langle b_1, D_1 \rangle, \ldots, \langle b_n, D_n \rangle\}$$

- Las máquinas no-deterministas juegan un papel central en la teoría de la complejidad.

Generación de lenguajes

Máquinas de Turing

- Las MT tienen suficiente poder como para generar lenguajes y calcular funciones.

Generación de lenguajes

Máquinas de Turing

- Las MT tienen suficiente poder como para generar lenguajes y calcular funciones.
- Una MT M genera al lenguaje $L \subseteq \Sigma^{\star}$ si

Generación de lenguajes

Máquinas de Turing

- Las MT tienen suficiente poder como para generar lenguajes y calcular funciones.
- Una MT M genera al lenguaje $L \subseteq \Sigma^\star$ si
  - M comienza a operar con la cinta en blanco en q_0.

Generación de lenguajes

Máquinas de Turing

- Las MT tienen suficiente poder como para generar lenguajes y calcular funciones.
- Una MT M genera al lenguaje $L \subseteq \Sigma^\star$ si
  - M comienza a operar con la cinta en blanco en q_0.
  - Cada vez que M regresa a q_0 hay una cadena de L escrita sobre la cinta.

Generación de lenguajes

Máquinas de Turing

- Las MT tienen suficiente poder como para generar lenguajes y calcular funciones.
- Una MT M genera al lenguaje $L \subseteq \Sigma^{\star}$ si
  - M comienza a operar con la cinta en blanco en q_0.
  - Cada vez que M regresa a q_0 hay una cadena de L escrita sobre la cinta.
  - Eventualmente se generan todas las cadenas de L.

Cálculo de funciones
Máquinas de Turing

La máquina de Turing $M = \langle\{q_0, q_f\}, \Sigma, \Gamma, \delta, q_0, \sqcup, \varnothing\rangle$ calcula una función $f : \Sigma^\star \to \Gamma^\star$ si

Cálculo de funciones

Máquinas de Turing

La máquina de Turing $M = \langle \{q_0, q_f\}, \Sigma, \Gamma, \delta, q_0, \sqcup, \varnothing \rangle$ calcula una función $f : \Sigma^\star \to \Gamma^\star$ si

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

Cálculo de funciones

Máquinas de Turing

La máquina de Turing $M = \langle \{q_0, q_f\}, \Sigma, \Gamma, \delta, q_0, \sqcup, \varnothing \rangle$ calcula una función $f : \Sigma^\star \to \Gamma^\star$ si

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

Observaciones:

- No hay estados finales, el estado q_f se usa para detener la máquina, es decir, no hay transiciones desde q_f.

Cálculo de funciones

Máquinas de Turing

La máquina de Turing $M = \langle \{q_0, q_f\}, \Sigma, \Gamma, \delta, q_0, \sqcup, \varnothing \rangle$ calcula una función $f : \Sigma^\star \to \Gamma^\star$ si

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

Observaciones:

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

Lenguajes recursivos y recursivamente enumerables

Aceptación en MT

- Un lenguaje L es recursivamente enumerable si es reconocido por una máquina de Turing, es decir, si existe una máquina de Turing M tal que $L = L(M)$.

Lenguajes recursivos y recursivamente enumerables

Aceptación en MT

- Un lenguaje L es recursivamente enumerable si es reconocido por una máquina de Turing, es decir, si existe una máquina de Turing M tal que $L = L(M)$.
- un lenguaje L es recursivo si es reconocido por una máquina de Turing que siempre se detiene, es decir, si existe una máquina de Turing M que se detiene con todas las cadenas de entrada y $L = L(M)$.

Propiedades de Cerradura

Lenguajes recursivos y R.E.

- Si L es recursivo entonces $\overline{L}$ es recursivo.

Propiedades de Cerradura

Lenguajes recursivos y R.E.

- Si L es recursivo entonces $\overline{L}$ es recursivo.
- Si L, M son recursivos entonces $L \cup M$ es recursivo.

Propiedades de Cerradura

Lenguajes recursivos y R.E.

- Si L es recursivo entonces $\overline{L}$ es recursivo.
- Si L, M son recursivos entonces $L \cup M$ es recursivo.
- Si L, M son rec. enumerables entonces $L \cup M$ es rec. enumerable.

Propiedades de Cerradura

Lenguajes recursivos y R.E.

- Si L es recursivo entonces $\overline{L}$ es recursivo.
- Si L, M son recursivos entonces $L \cup M$ es recursivo.
- Si L, M son rec. enumerables entonces $L \cup M$ es rec. enumerable.
- L es recursivo si y sólo si L y $\overline{L}$ son rec. enumerables.

Jerarquía de Chomsky

su relación con autómatas

- Las gramáticas tipo 3 (regulares) son equivalentes a los autómatas finitos.

Jerarquía de Chomsky

su relación con autómatas

- Las gramáticas tipo 3 (regulares) son equivalentes a los autómatas finitos.
- Las gramáticas tipo 2 (libres de contexto) son equivalentes a los autómatas de pila (no-determinísticos).

Jerarquía de Chomsky

su relación con autómatas

- Las gramáticas tipo 3 (regulares) son equivalentes a los autómatas finitos.
- Las gramáticas tipo 2 (libres de contexto) son equivalentes a los autómatas de pila (no-determinísticos).
- ¿Qué pasa con las gramáticas tipo 1 (sensibles al contexto)?

Jerarquía de Chomsky

su relación con autómatas

- Las gramáticas tipo 3 (regulares) son equivalentes a los autómatas finitos.
- Las gramáticas tipo 2 (libres de contexto) son equivalentes a los autómatas de pila (no-determinísticos).
- ¿Qué pasa con las gramáticas tipo 1 (sensibles al contexto)?
- ¿Qué pasa con las gramáticas tipo 0 (irrestrictas)?

$L = \{a^i b^i c^i \mid i \geq 0\}$

Gramática sensible al contexto

- $S \to A$

$L = \{a^i b^i c^i \mid i \geq 0\}$

Gramática sensible al contexto

- $S \to A$
- $A \to aABC \mid abC$

$L = \{a^i b^i c^i \mid i \geq 0\}$

Gramática sensible al contexto

- $S \to A$
- $A \to aABC \mid abC$
- $CB \to BC$

$L = \{a^i b^i c^i \mid i \geq 0\}$

Gramática sensible al contexto

- $S \to A$
- $A \to aABC \mid abC$
- $CB \to BC$
- $bB \to bb$

$L = \{a^i b^i c^i \mid i \geq 0\}$

Gramática sensible al contexto

- $S \to A$
- $A \to aABC \mid abC$
- $CB \to BC$
- $bB \to bb$
- $bC \to bc$

$L = \{a^i b^i c^i \mid i \geq 0\}$

Gramática sensible al contexto

- $S \to A$
- $A \to aABC \mid abC$
- $CB \to BC$
- $bB \to bb$
- $bC \to bc$
- $cC \to cc$

Autómatas Linealmente Acotados

Definición

- Un autómata linealmente acotado (ALA) es una máquina de Turing que satisface las siguientes condiciones:

Autómatas Linealmente Acotados

Definición

- Un autómata linealmente acotado (ALA) es una máquina de Turing que satisface las siguientes condiciones:
- El alfabeto de entrada Σ incluye dos símbolos especiales $[,]$ que sirven como marcas de fin de cinta izquierda y derecha respectivamente.

Autómatas Linealmente Acotados

Definición

- Un autómata linealmente acotado (ALA) es una máquina de Turing que satisface las siguientes condiciones:
- El alfabeto de entrada Σ incluye dos símbolos especiales $[,]$ que sirven como marcas de fin de cinta izquierda y derecha respectivamente.
- La cabeza lectora no puede desplazarse más allá de dichos límites y no puede sobreescribir tales sectores.

Autómatas Linealmente Acotados

Definición

- Formalmente tenemos

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

con $[,] \in \Sigma$.

Autómatas Linealmente Acotados

Definición

- Formalmente tenemos

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

con $[,] \in \Sigma$.

- El lenguaje de aceptación es

$$L(M) = \{w \in \Sigma^\star - \{[,]\} \mid q_0[w] \vdash^\star w_1 q_f w_2 \;\; q_f \in F\}$$

Las marcas $[,]$ no son consideradas como parte de la cadena a procesar.

Autómatas Linealmente Acotados

Definición

- Formalmente tenemos

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

  con $[,] \in \Sigma$.

- El lenguaje de aceptación es

$$L(M) = \{w \in \Sigma^\star - \{[,]\} \mid q_0[w] \vdash^\star w_1 q_f w_2 \;\; q_f \in F\}$$

  Las marcas $[,]$ no son consideradas como parte de la cadena a procesar.

- Un ALA no puede moverse fuera de la cadena de entrada.

ALA y Gramáticas

Equivalencia

- Dada una gramática sensible al contexto G, existe un autómata linealmente acotado M tal que $L(G) = L(M)$. Es decir, los lenguajes sensibles al contexto son reconocidos por autómatas linealmente acotados.

ALA y Gramáticas

Equivalencia

- Dada una gramática sensible al contexto G, existe un autómata linealmente acotado M tal que $L(G) = L(M)$. Es decir, los lenguajes sensibles al contexto son reconocidos por autómatas linealmente acotados.
- Si $L = L(M)$ es un lenguaje reconocido por un autómata linealmente acotado M entonces existe una gramática sensible al contexto G tal que $L(M) = L(G)$. Es decir, los lenguajes reconocidos por ALA son sensibles al contexto.

Máquinas de Turing y gramáticas irrestrictas

Equivalencia

- Para toda gramática G de tipo 0 existe una máquina de Turing M tal que $L(M) = L(G)$. Es decir, los lenguajes tipo 0 son recursivamente enumerables.

Máquinas de Turing y gramáticas irrestrictas

Equivalencia

- Para toda gramática G de tipo 0 existe una máquina de Turing M tal que $L(M) = L(G)$. Es decir, los lenguajes tipo 0 son recursivamente enumerables.
- Para toda máquina de Turing M existe una gramática G de tipo 0 tal que $L(G) = L(M)$. Es decir, los lenguajes recursivamente enumerables son lenguajes tipo 0.

Jerarquía de Chomsky

su relación con autómatas

- Las gramáticas tipo 3 (regulares) son equivalentes a los autómatas finitos.

Jerarquía de Chomsky

su relación con autómatas

- Las gramáticas tipo 3 (regulares) son equivalentes a los autómatas finitos.

- Las gramáticas tipo 2 (libres de contexto) son equivalentes a los autómatas de pila (no-determinísticos).

Jerarquía de Chomsky
su relación con autómatas

- Las gramáticas tipo 3 (regulares) son equivalentes a los autómatas finitos.
- Las gramáticas tipo 2 (libres de contexto) son equivalentes a los autómatas de pila (no-determinísticos).
- Las gramáticas tipo 1 (sensibles al contexto) son equivalentes a los autómatas linealmente acotados.

Jerarquía de Chomsky

su relación con autómatas

- Las gramáticas tipo 3 (regulares) son equivalentes a los autómatas finitos.
- Las gramáticas tipo 2 (libres de contexto) son equivalentes a los autómatas de pila (no-determinísticos).
- Las gramáticas tipo 1 (sensibles al contexto) son equivalentes a los autómatas linealmente acotados.
- Las gramáticas tipo 0 (irrestrictas) son equivalentes a las máquinas de Turing.