

Teoría de la Computación y Lenguajes Formales

Máquinas de Turing

Prof. Hilda Y. Contreras

Departamento de Computación

hyelitza@ula.ve

Máquinas de Turing

Contenido

- Introducción
- Historia e importancia
- Modelo básico de la MT
- Ejemplo
- Lenguajes tipo 0
- Variaciones de la MT
- Conclusiones

Máquinas de Turing

Introducción

Jerarquía de Chomsky

- Gramáticas
- “Poder generativo débil”
- Jerarquía implicativa (tipo i incluye tipo $i+1$)



Máquinas de Turing

Introducción

Tip o	Gramática	Lenguaje	Máquina
0	Irrestricta $\lambda_1, \lambda_2 \rightarrow \beta_1, \beta_2$	Recursivamente e enumerables	Máquina de Turing
1	Dependiente del Contexto $x\lambda Z \rightarrow x\beta Z$	Dependiente del contexto	Autómatas linealmente acotados
2	Independiente del Contexto $A \rightarrow \beta$	Independiente del Contexto	Autómatas de Pila
3	Regulares $A \rightarrow aB; A \rightarrow a$	Regulares	Autómatas finitos

Máquinas de Turing

Historia

De la matemática a la computación:

- **Siglo XIX, D. Hilbert**
Validación de fórmulas matemáticas
- **1931, K. Gödel**
Teorema de la incompletitud
- **1936, A. Turing**
Modelo de cualquier proposición posible

Máquinas de Turing

Historia

- Máquinas de Turing → Teoría Matemática de la Computación
- Tesis o hipótesis de Church-Turing

“Cualquier forma general de computación nos permitirá calcular únicamente las *funciones recursivas parciales*”

Máquinas de Turing

Importancia

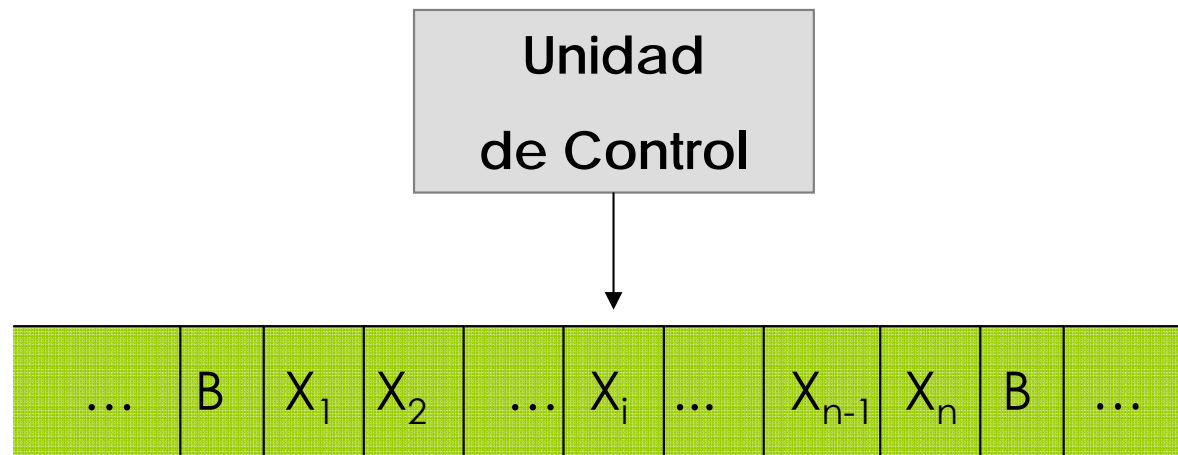
- Modelo teórico
- Máquina simple, sencilla y precisa
- Generalidad
- Determinar características de los problemas
 - Decidable (existe un algoritmo)
 - Tratable (existe un algoritmo rápido)

Máquinas de Turing

Modelo básico

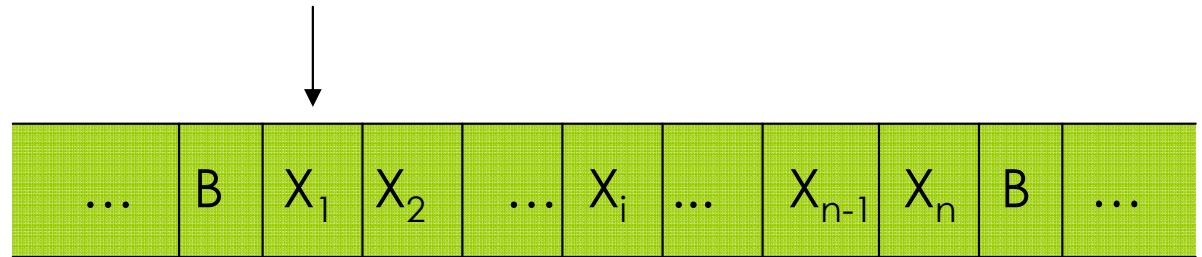
Componentes:

- Unidad de Control (autómata finito)
- Cinta infinita



Máquinas de Turing

Modelo básico



- Cinta infinita
 - Infinita a ambos extremos
 - Dividida en casillas
 - Un símbolo por casilla, inicializada en B
 - Entrada w ubicada sobre la cinta infinita
 - Cabezal que indica posición actual. Inicia en el primer símbolo de la entrada.
 - Cabezal con movimiento secuencial infinitad de veces de izquierda a derecha
 - Leer/escribir en cada transición

Máquinas de Turing

Modelo básico

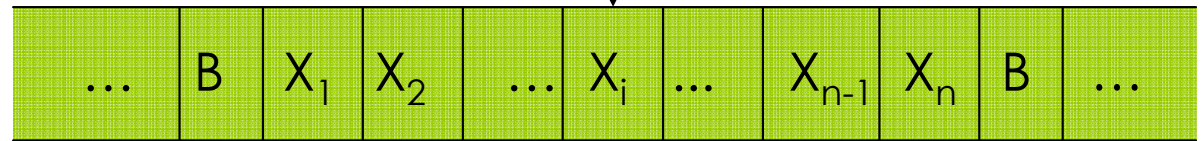
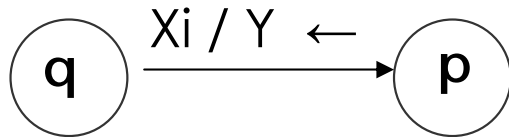
$$\mathbf{M} = (\mathbf{Q}, \Sigma, \Gamma, \delta, \mathbf{q}_0, \mathbf{B}, \mathbf{F})$$

- **Q**: conjunto de estados
- Σ : símbolos de entrada
- Γ : símbolos de la cinta. Γ contiene a $\Sigma \cup \mathbf{B}$
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{D, I\}$
- \mathbf{q}_0 : estado inicial
- **B**: símbolo espacio en blanco
- **F**: conjunto de estados de aceptación

Máquinas de Turing

Modelo básico

- **Transición** $\delta(q, X_i) = (p, Y, l)$



- **Entrada** (Σ^*)

$X_1 X_2 \dots X_{i-1} X_i X_{i+1} \dots X_n$

Descripción instantánea: $X_1 X_2 \dots X_{i-1} \mathbf{q} X_i X_{i+1} \dots X_n$

- **Movimiento** \vdash

$X_1 \dots X_{i-1} \mathbf{q} X_i X_{i+1} \dots X_n \vdash X_1 \dots X_{i-2} \mathbf{p} X_{i-1} Y X_{i+1} \dots X_n$

- **Múltiples movimientos** \vdash^*

$qw \vdash^* \beta_1 \mathbf{p} \beta_2$

Máquinas de Turing

Ejemplo

- Diseñar una máquina de Turing que acepte el lenguaje:

$$L = \{ 0^n 1^n \mid n \geq 1 \}$$

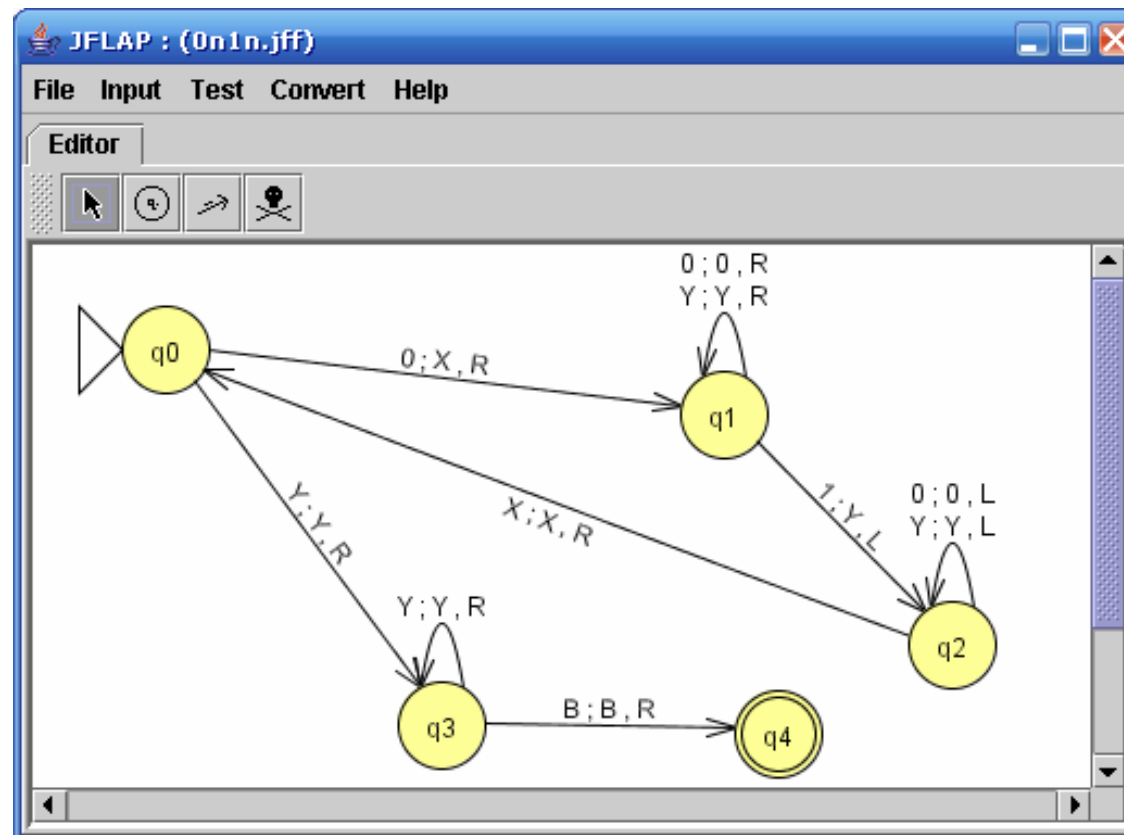
Idea:

- Cambiar **0** por **X** y **1** por **Y** en una pasada
- Repetir hasta el final de la cadena de entrada
- Si hay **XⁿYⁿ** la máquina acepta
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, X, Y, B\}$

Máquinas de Turing

Ejemplo

- La máquina de Turing que acepta el lenguaje: $\{ 0^n 1^n \mid n \geq 1 \}$ (archivo 0n1n.jff)



Máquinas de Turing

Ejemplo

	0	1	X	Y	B
$\rightarrow q_0$	(q_1, X, D)			(q_3, Y, D)	
q_1	$(q_1, 0, D)$	(q_2, Y, I)		(q_1, Y, D)	
q_2	$(q_2, 0, I)$		(q_0, X, D)	(q_2, Y, I)	
q_3				(q_3, Y, D)	(q_4, B, D)
$*q_4$					

para $w = 01$ se tiene $q_0 01 \vdash Xq_1 1 \vdash q_2 XY \vdash Xq_0 Y \vdash XYq_3 B \vdash XYBq_4 B$
 ¿Cuál es la secuencia de \vdash para $w = 0011$ y $w = 0010$?

Máquinas de Turing

Lenguaje tipo 0

Sea $\mathbf{M} = (\mathbf{Q}, \Sigma, \Gamma, \delta, \mathbf{q}_0, \mathbf{B}, \mathbf{F})$ una Máquina de Turing

Lenguaje aceptado se define:

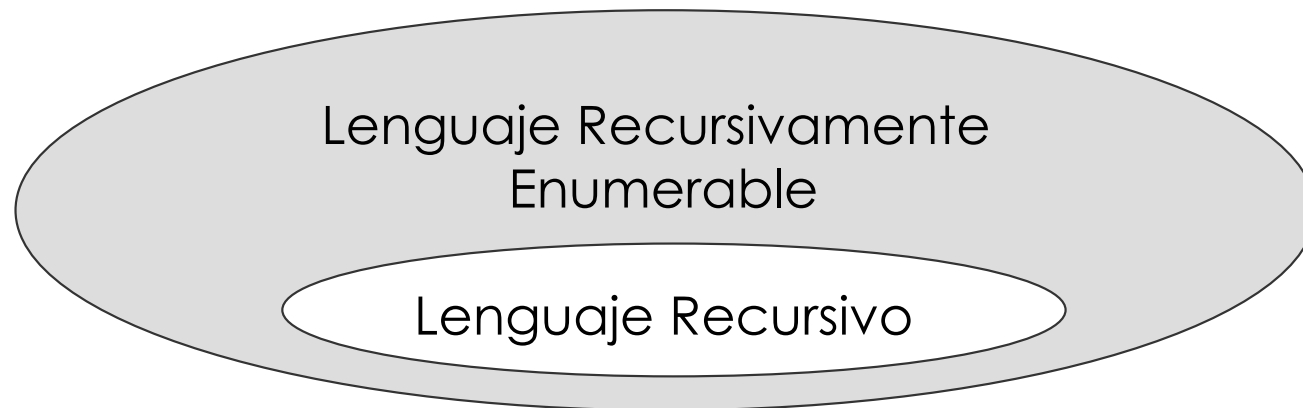
$$L(\mathbf{M}) = \{ w \mid w \text{ en } \Sigma^*, q_0 w \vdash^* \alpha_1 q \alpha_2, \alpha_1 \text{ y } \alpha_2 \text{ en } \Gamma^* \text{ y } q \text{ en } \mathbf{F} \}$$

$L(\mathbf{M})$ es un lenguaje recursivamente enumerable

Máquinas de Turing

Lenguaje tipo 0

- Lenguaje **Recursivamente Enumerable**
Existe una MT
- Lenguaje **Recursivo** (decidable = algoritmo)
Existe una MT que siempre para



Máquinas de Turing

Utilidad

- **Aceptador** Lenguajes

¿w en L?

- **Generador** de Lenguajes

01B0011B000111B...

- **Computador** de funciones matemáticas sobre enteros

$0^m 1 0^n$ generar $0^{m \times n}$

Máquinas de Turing

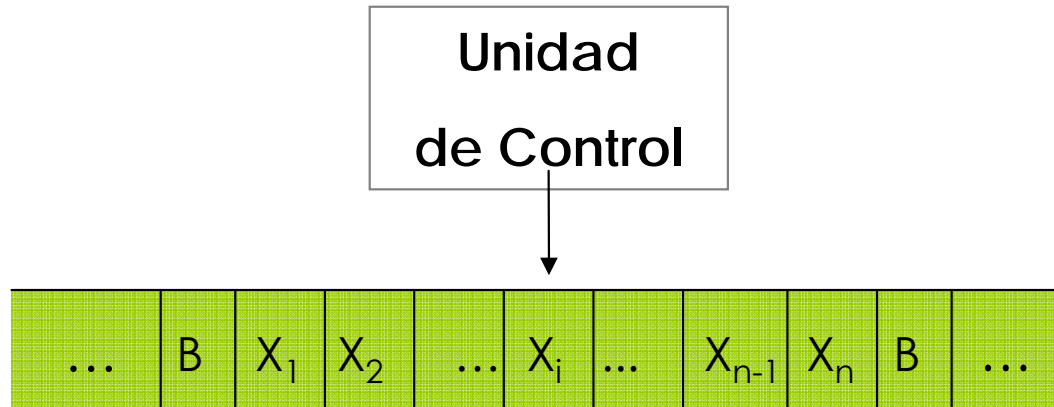
Conclusiones

- MT es la máquina abstracta más **poderosa**
- MT consta de un **autómata finito** y una **cinta infinita**
- MT reconoce los lenguajes tipo 0 **recursivamente enumerables**
- Si MT se detiene para toda entrada, el **lenguaje es recursivo**

Contenido

- Técnicas de construcción de la MT:
 - Memoria en la unidad de control
 - Multipistas
 - Multicintas
- MT generadora de lenguajes y MT calculadora de funciones
- Lenguajes recursivamente enumerables
- Lenguajes tipo 1
- Gramáticas sin restricciones

Modelo básico de MT



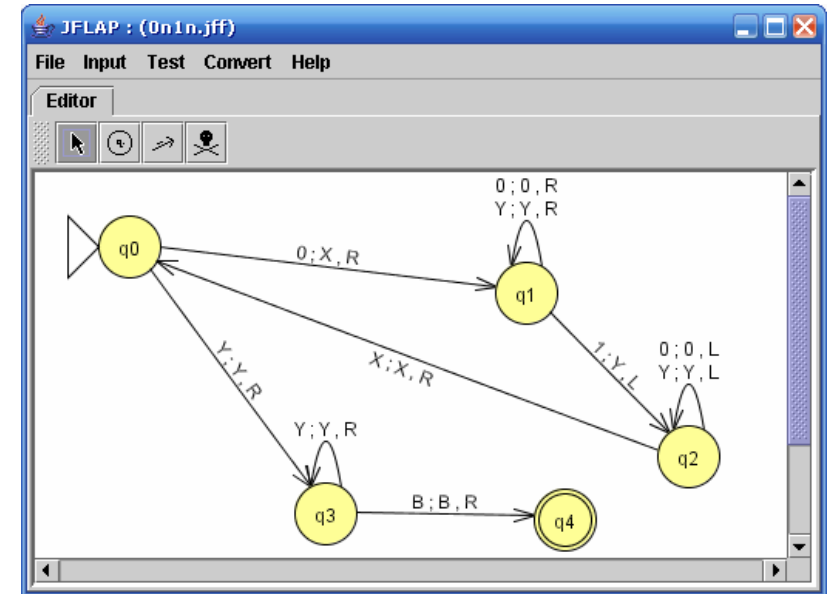
$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

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

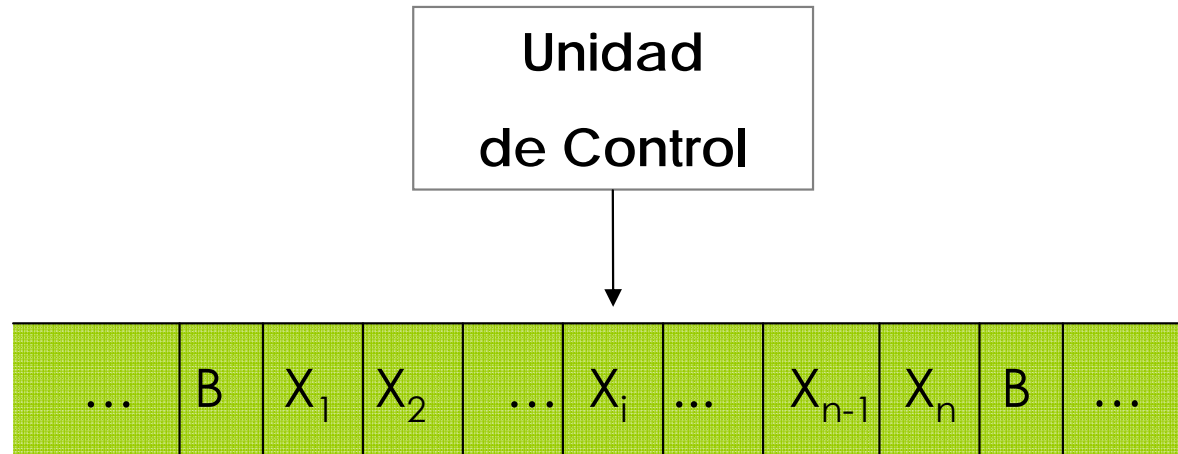
$$\delta(q, X_i) = (p, Y_i, D)$$

$$L = \{ 0^n 1^n \mid n \geq 1 \}$$

JFLAP: [0n1n.jff](#)



Técnicas de construcción de MT



- Extensiones del modelo básico (una cinta, una pista, sin memoria en el control)
- Ninguna técnica aumenta el poder de la MT
- Técnicas facilitan la programación, disminuye el número de transiciones, mejora la comprensión del algoritmo, etc.

Almacenamiento en la UC

$$M = (Q', \Sigma, \Gamma, \delta, q_0, B, F)$$

Unidad de Control
0

- Utiliza el estado de control para almacenar una cantidad **finita** de información.
- Conjunto Q de estados cambia
 $Q = \{q_0, q_1\}$ y $\Delta = \{0, 1\}$ (Δ información finita)
 $Q' = Q \times \Delta = \{[q_0, 0], [q_0, 1], [q_1, 0], [q_1, 1]\}$

Almacenamiento en la UC

Por ejemplo: $L = \{ 01^* + 10^* \}$

- Almacenar el primer símbolo leído
- $\Delta = \{0,1,B\}$
- JFLAP: [01a10a.jff](#)

$$\delta([q_0, B], 0) = ([q_1, 0], 0, D)$$

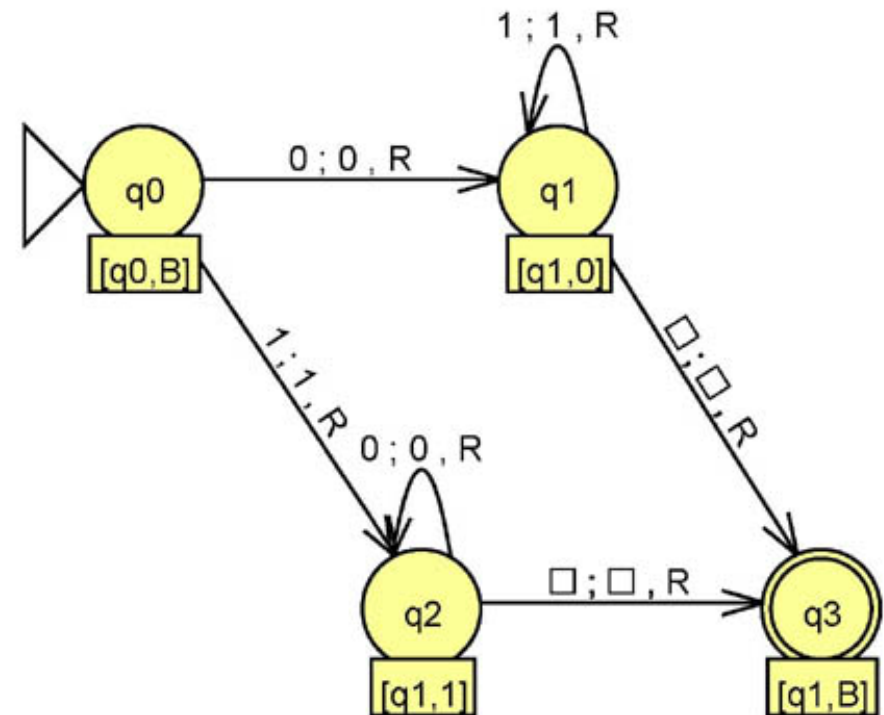
$$\delta([q_1, 0], 1) = ([q_1, 0], 1, D)$$

$$\delta([q_1, 0], B) = ([q_1, B], B, D)$$

$$\delta([q_0, B], 1) = ([q_1, 1], 1, D)$$

$$\delta([q_1, 1], 0) = ([q_1, 1], 0, D)$$

$$\delta([q_1, 1], B) = ([q_1, B], B, D)$$



MT con multiples pistas

- Un solo cabezal. La cinta está dividida en un número finito de k pistas, la función de transición tiene la siguiente forma:

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

Unidad
de Control

$$W_1 = [X_1, B, \dots, B], [X_2, B, \dots, B], \dots [X_n, B, \dots, B]$$

$$\delta(q, [X_i, B, \dots, B]) = (p, [Y_i, B, \dots, B], D)$$



...	B	X_1	X_2	...	X_i	...	X_{n-1}	X_n	B	...
...			
...	B	B	B	...	B	...	B	B	B	...

Pista 1

...

Pista k

MT con multiples pistas

- Pistas múltiples (*multipistas*)
- La cinta almacena en cada celda un vector ***k*** dimensional de símbolos a los que se accede simultáneamente.

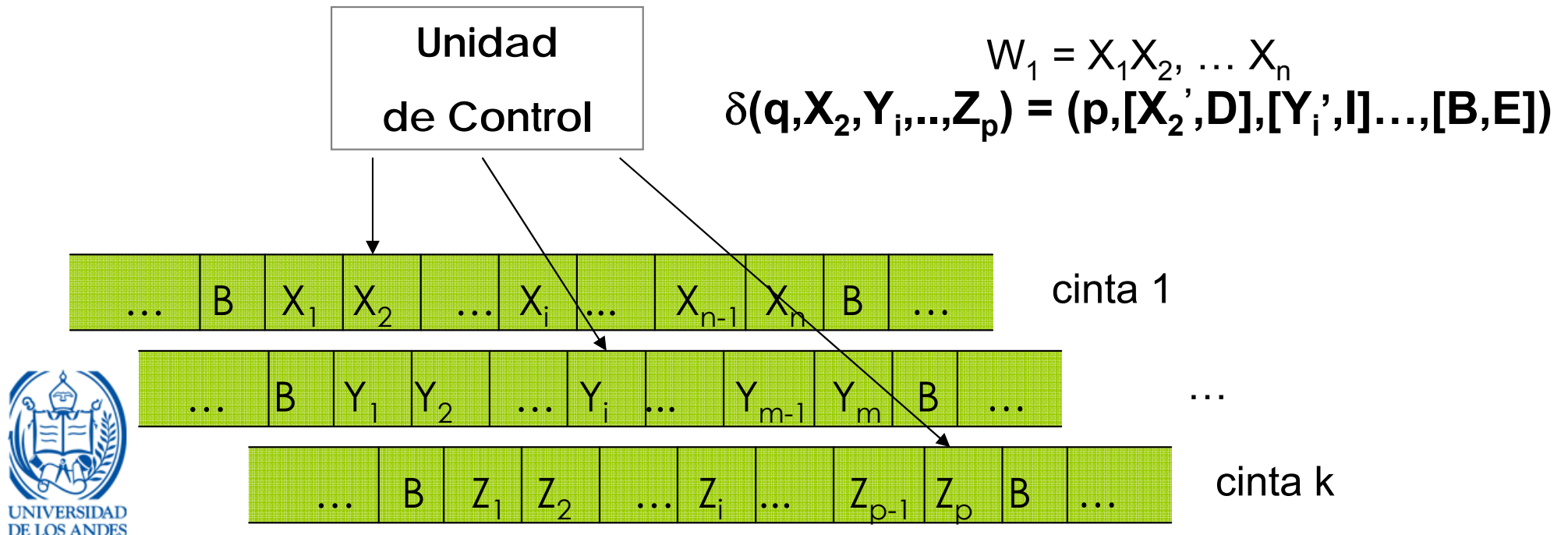
Un *movimiento* de la máquina implica:

- (a) Cambiar el estado del control finito
- (b) Escribir ***k*** símbolos en la celda analizada
- (c) Mover el cabezal de la cinta a la izquierda o a la derecha.

MT con múltiples cintas

- Hay k cintas diferentes y k cabezales. La función de transición para máquinas de Turing con n cintas:

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



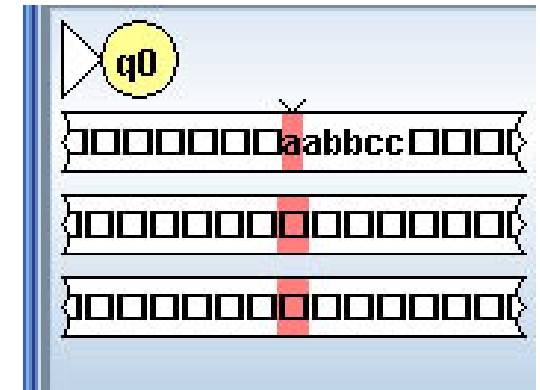
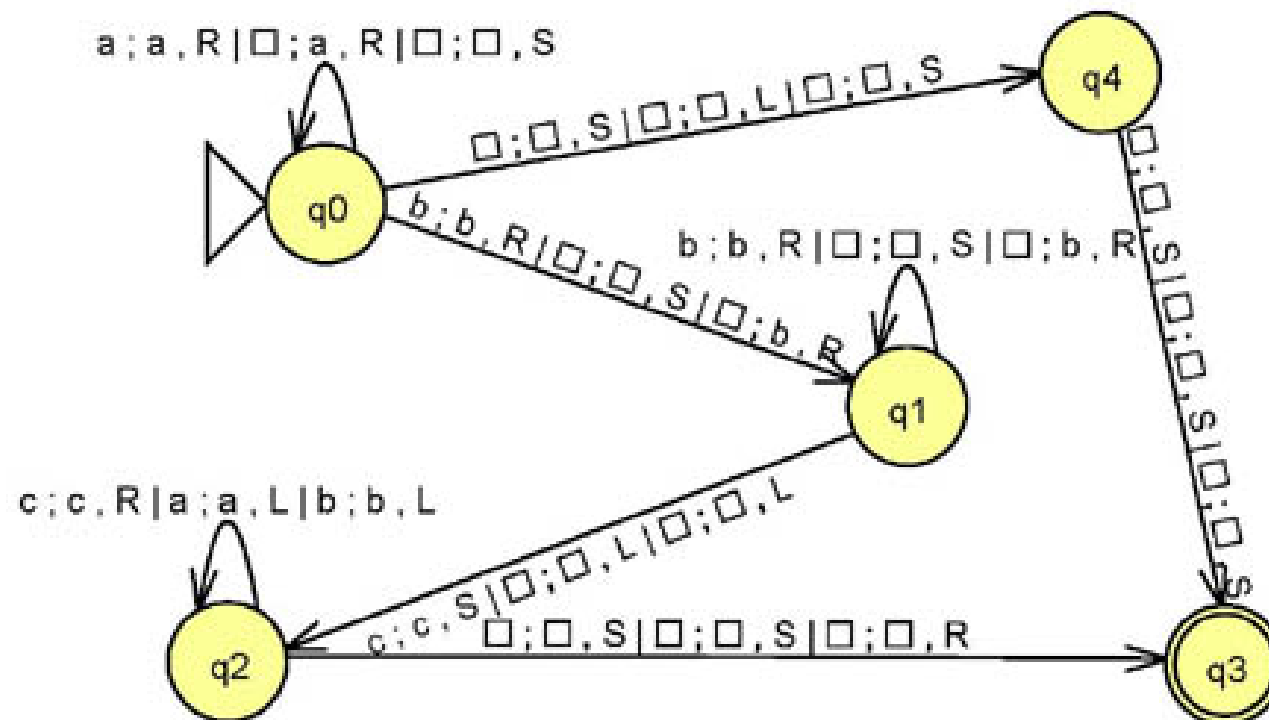
MT con múltiples cintas

- Un movimiento de la máquina *multicinta* depende del estado del control finito y de los símbolos analizados por cada cabezal de cada cinta
- Un movimiento de la máquina *multicinta* implica:
 - (a) Cambiar el estado del control finito
 - (b) Escribir un símbolo en cada una de las celdas analizadas
 - (c) Mover cada cabezal de cinta a la izquierda o a la derecha independientemente

MT con múltiples cintas

- Por ejemplo: $L = \{ a^n b^n c^n \mid n \geq 0 \}$

JFLAP: [turingAnBnCnMulti.jff](#)



MT calculadora

- Calcula funciones $f: \Sigma^* \rightarrow \Gamma^*$, es decir $f(w) = v$, donde w en Σ^* y v en Γ^*
- $q_0 w \vdash^* q_f v$, donde $v = f(w)$
- Uso de notación “unaria” para codificar los valores del dominio y el rango.

$X = 0^x$, p.e. $3 = 0^3 = 000$

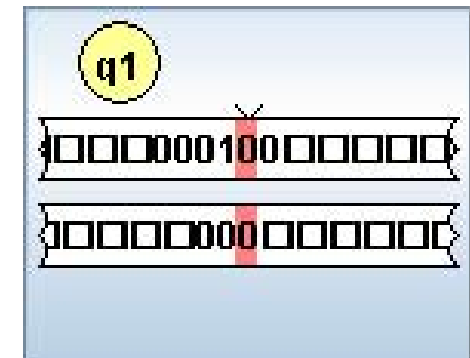
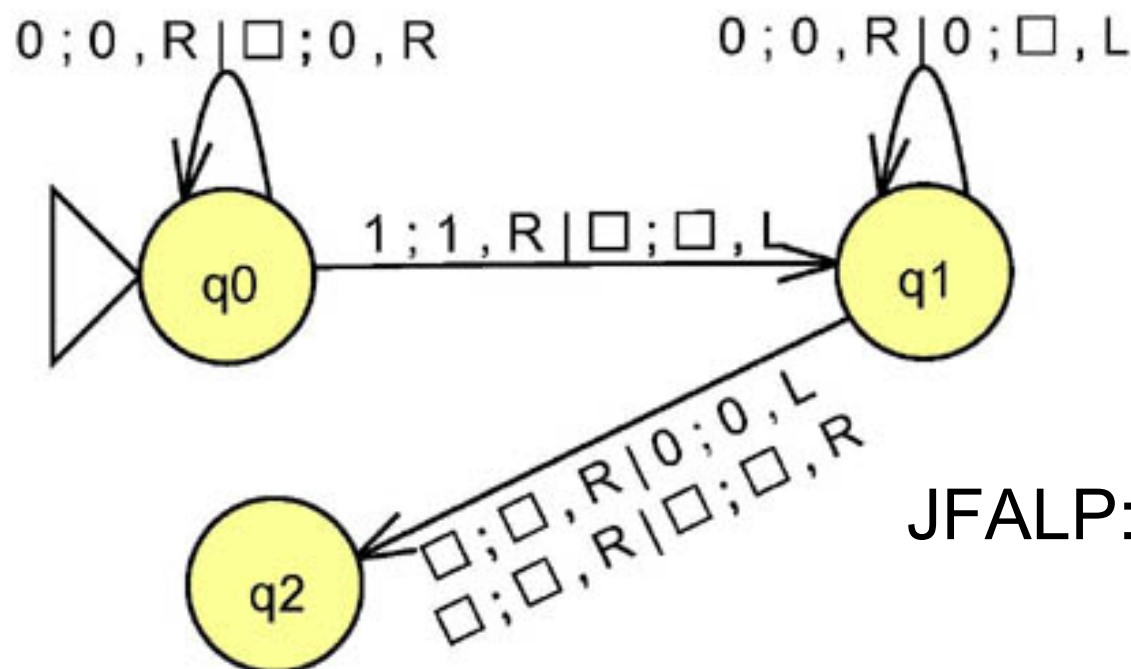
- No tiene estados de aceptación

MT calculadora

- Por ejemplo: sustracción propia

$$m - n \begin{cases} m-n & \text{si } m \geq n \\ 0 & \text{si } m < n \end{cases}$$

p.e: operandos 3 y 2 para la operación 3 -2
 Entrada = 000100
 Salida = 0



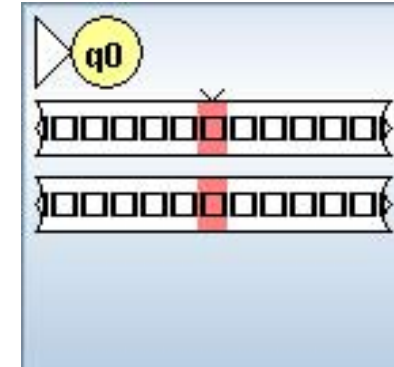
JFALP: calculam-nMulti.jff

MT generadora de lenguajes

- MT no necesita cadena de entrada
- MT comienza a operar con la cinta en blanco en el estado inicial q_0 .
- Cada vez que MT retorna al estado inicial q_0 , hay una cadena w en L escrita sobre la cinta (salida de solo escritura).
- Todas las cadenas de L son, eventualmente, generadas por M .

MT generadora de lenguajes

Por ejemplo: $L = \{ 0^{2n} \mid n > 0 \}$



MT de 2 cintas:

$\delta(q_0, B, B) = (q_1, (0, I), (\#, D))$ // inicio de la cadena salida

$\delta(q_1, 0, B) = (q_1, (0, I), (B, E))$ // busca inicio del contador n

$\delta(q_1, B, B) = (q_2, (B, D), (B, E))$ // busca inicio del contador n

$\delta(q_2, 0, B) = (q_3, (0, E), (0, D))$ // coloca cinda 2 el 0 impar

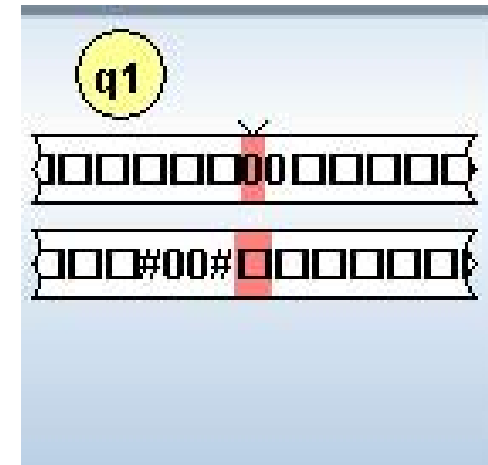
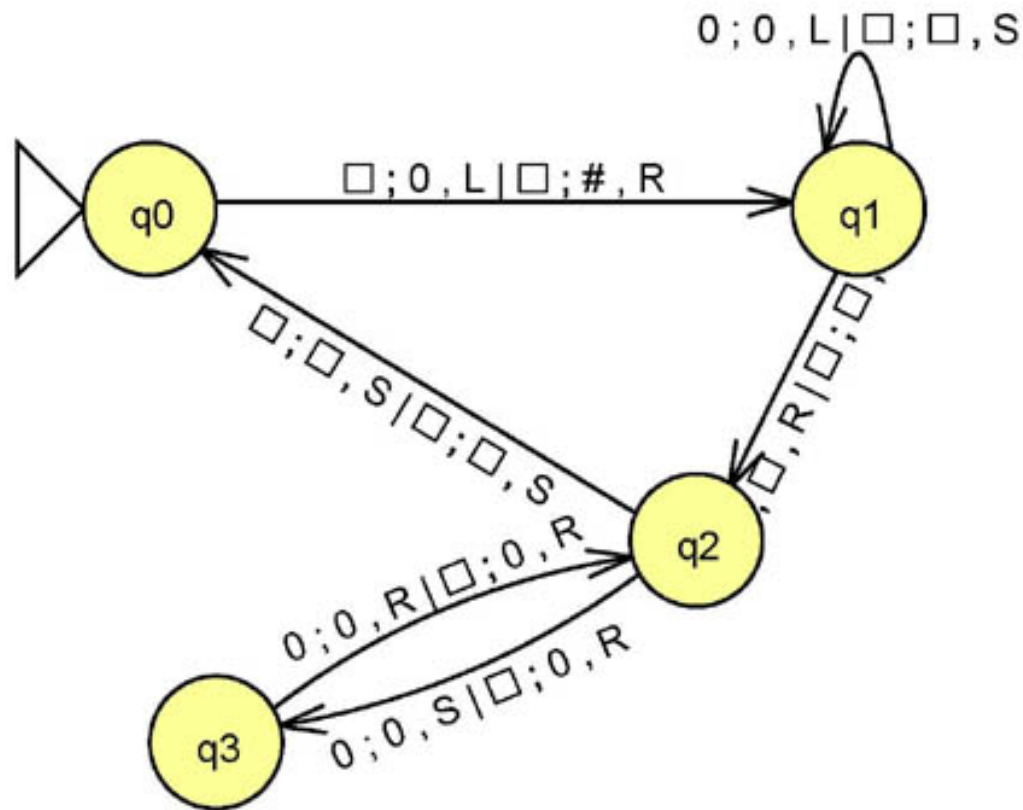
$\delta(q_3, 0, B) = (q_2, (0, D), (0, D))$ // coloca cinda 2 el 0 par

$\delta(q_2, B, B) = (q_0, (B, E), (B, E))$ // fin del contador n, próxima cadena

MT generadora de lenguajes

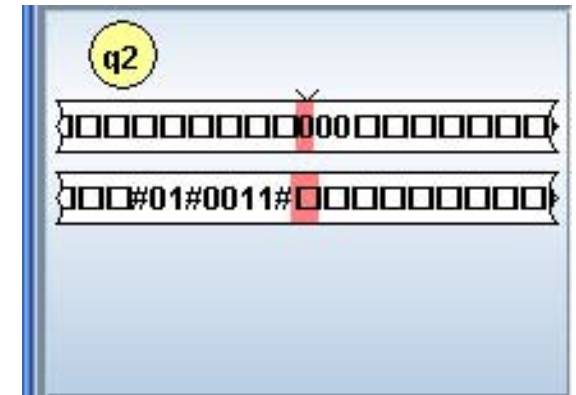
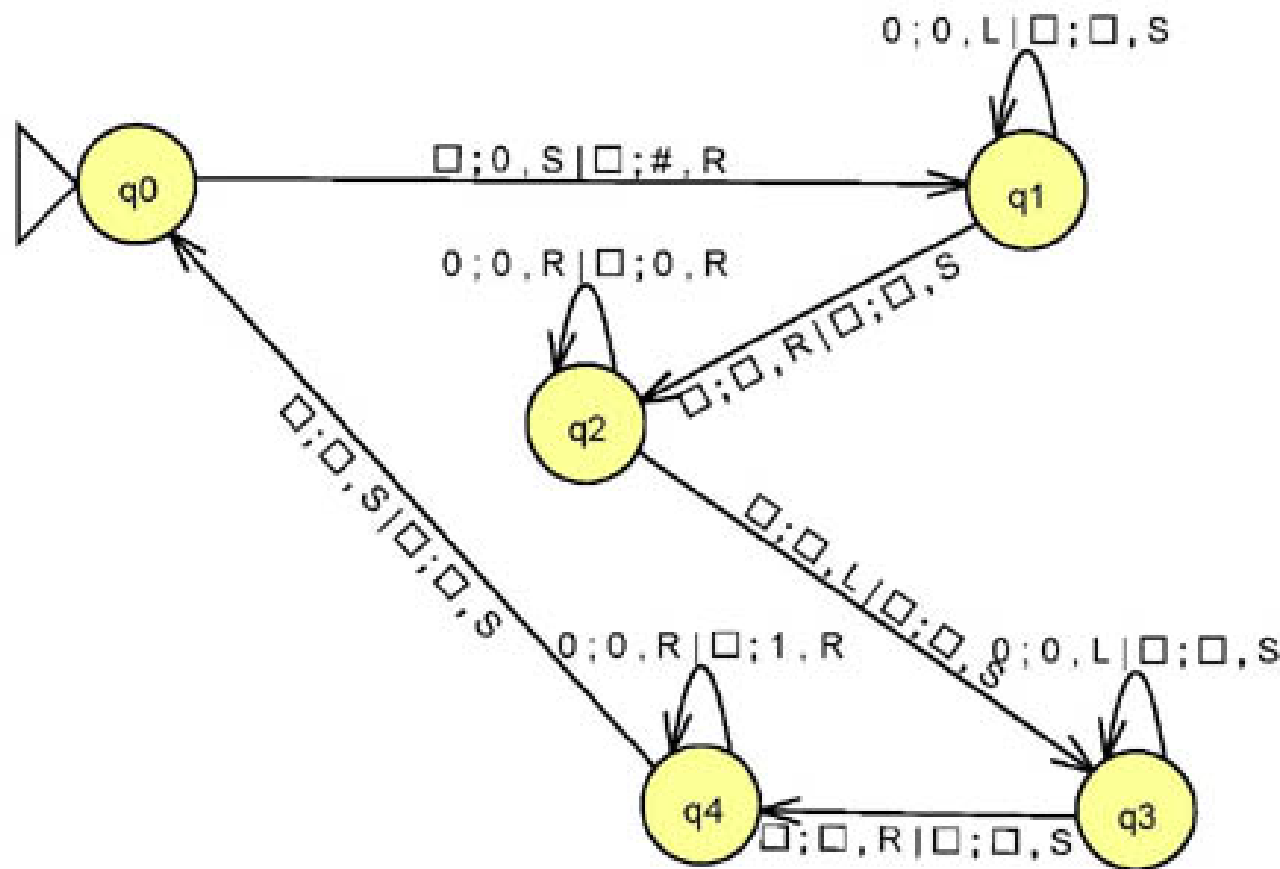
Por ejemplo: $L = \{ 0^{2n} \mid n > 0 \}$

JFLAP: [genera02nMulti.jff](#)



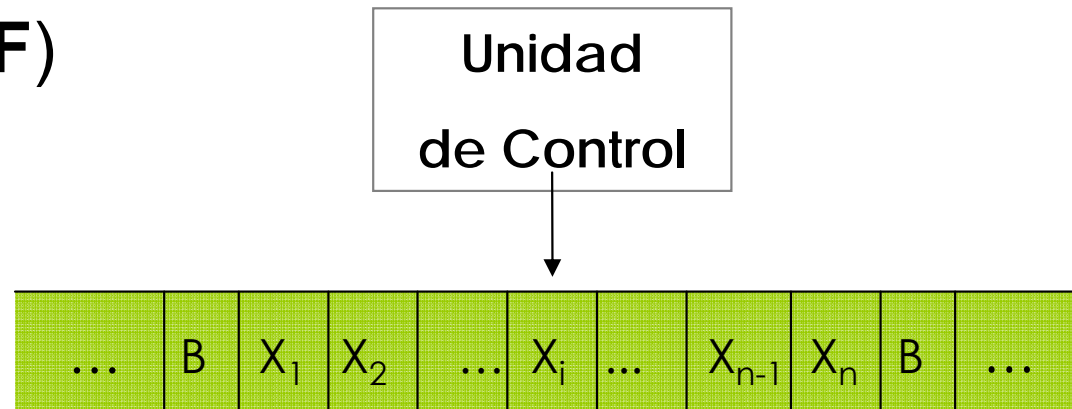
MT generadora de lenguajes

Por ejemplo: $L = \{ 0^n 1^n \mid n > 0 \}$



MT No determinista

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$



$$\delta: Q \times \Gamma \rightarrow (Q \times \Gamma \times \{D, I\})^*$$

$$\delta(q, X_i) = \{ (p, Y_i, D), (r, Z_i, D), \dots, (q, X_i, I) \}$$

- *El indeterminismo* de MT se refiere a las opciones que la máquina pueda tener en cualquier configuración por la función δ
- M acepta w si existe un camino desde q_0 hasta un estado final

Equivalencias de MT

Teorema: para cualquier máquina de Turing M_1 con k cintas existe otra equivalente M_2 con una sola cinta

Teorema: para cualquier máquina de Turing M_1 no determinista existe otra equivalente M_2 determinista

Teorema: para cualquier máquina de Turing M_1 con cinta infinita en ambos sentidos existe otra equivalente M_2 con cinta limitada por la izquierda

Lenguajes recursivamente enumerables

Lenguaje recursivamente enumerable: si existe una máquina de Turing M tal que $L(M) = L$. Define la clase $Lr.e$.

Lenguaje recursivo: si existe una máquina de Turing M tal que $L(M)=L$ y M se detiene ante cualquier entrada. Define la clase $Lrec$

$Lrec$ es subconjunto de $Lr.e$

Una función es *computable* si puede ser calculada por una máquina de Turing.

Propiedades de clausura

Propiedad	LR	LLC	LRE
U (unión)	S	S	S
\cap (intersección)	S	N	S
complemento	S	N	N
concatenación	S	S	S
\cap LR	S	S	S

Propiedades de clausura

Propiedad	LR	LLC	LRE
Kleene-clausura	S	S	S
Reflejo	S	S	S
Morfismo	S	S	S
Morfismo ⁻¹	S	S	S
Diferencia	S	N	S

Algoritmos de decisión

Problemas	LR	LLC	LRE
Equidad	D	N	N
Inclusión	D	N	N
Membresía	D	D	N
Vacuidad	D	D	N
Finitud	D	D	N

Jerarquía de Chomsky

Tipo	Lenguaje	Máquina	Gramática $G = (V, T, P, S)$
0	Recursivamente enumerable	Máquina de Turing	Gramática sin restricciones $\alpha \rightarrow \beta$ $(\alpha, \beta \text{ en } (V \cup T)^*, \alpha \text{ contiene una variable})$
1	Dependiente del Contexto	Autómata linealmente acotado	<i>Gramática sensible al contexto</i> $\alpha \rightarrow \beta$ $(\alpha, \beta \text{ en } (V \cup T)^*, \alpha \text{ contiene una variable, } \beta \geq \alpha)$
2	Independiente del Contexto	Autómata de Pila	<i>Gramática libre de contexto</i> $A \rightarrow \alpha$ $(A \text{ en } V \text{ y } \alpha \text{ en } (V \cup T)^*)$
3	Lenguaje Regular	Autómata finito	<i>Gramática Regular</i> $A \rightarrow aB$ $A \rightarrow a$ $(A, B \text{ en } V \text{ y } a \text{ en } T)$



Lenguajes sensibles al contexto

Gramáticas sensibles al contexto

- Una gramática $G = (V, T, P, S)$ es una gramática sensible de contexto si todas las producciones son de la forma

$$\alpha \rightarrow \beta$$

- donde α, β en $(V+T)^*$ y $|\alpha| \leq |\beta|$, α contiene al menos una variable V

Lenguajes sensibles al contexto

Gramáticas sensibles al contexto

Por ejemplo: $L = \{ a^n b^n c^n \mid n > 0 \}$

$S \rightarrow A_0 B C S_1 \mid A_0 B C$

$S_1 \rightarrow A B C S_1 \mid A B C$

$B A \rightarrow A B$

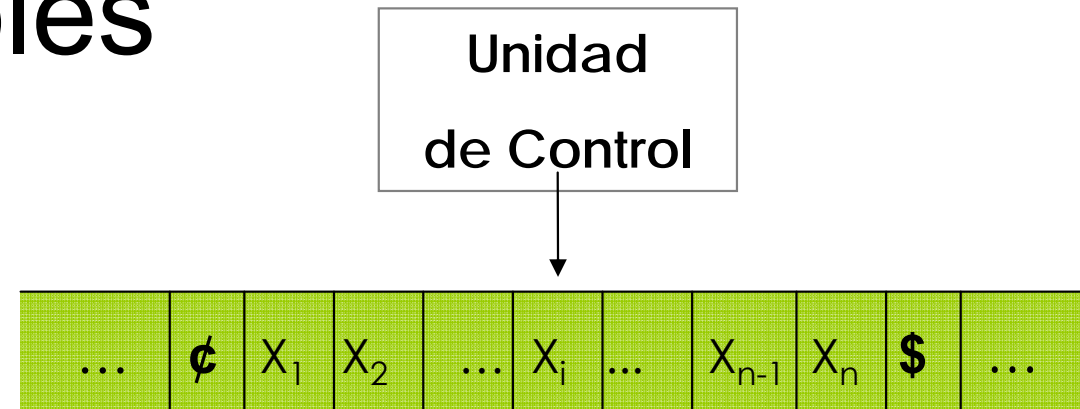
$C A \rightarrow A C$

$C B \rightarrow B C$

$A_0 \rightarrow a, \quad a A \rightarrow a a, \quad a B \rightarrow a b$

$b B \rightarrow b b, \quad b C \rightarrow b c, \quad c C \rightarrow c c$

Lenguajes sensibles al contexto



Autómatas linealmente acotadas

Un *autómata de memoria limitada linealmente* (ALL) es una máquina de Turing no determinista monocinta que satisface las siguientes condiciones:

1. El alfabeto de entrada incluye los limitadores ¢ (izquierda) y \$ (derecha)
2. El ALL no mueve el cabezal de la cinta fuera de los limitadores ni escribe sobre ellos

Teorema: L es un lenguaje sensible al contexto (de tipo 1) si L es aceptado por un ALL

Gramáticas sin restricciones

- Una gramática $G = (V, T, P, S)$ es una gramática de tipo 0 (irrestringida) si todas las producciones son de la forma

$$\alpha \rightarrow \beta$$

- donde α en $(V+T)^+$, β en $(V+T)^*$

Teorema: L es un lenguaje recursivamente enumerable sii $L=L(G)$ donde G es de tipo 0.

Gramática irrestricta

Gramáticas irrestricta

Por ejemplo: $L = \{ ww \mid w \text{ en } (a+b)^* \}$

$S \rightarrow FM$

$F \rightarrow FaA \mid FbB$

$Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB$

$AM \rightarrow Ma, BM \rightarrow Mb$

$F \rightarrow \lambda, M \rightarrow \lambda$