Ciencias de la Computación I

Autómatas Linealmente Acotados
Máquinas de Turing

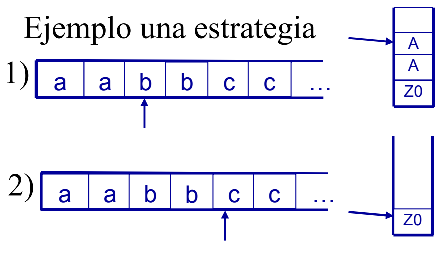
Motivación

-¿Es posible diseñar un AP que reconozca el lenguaje L_1?

$L_1 = \{ a^n b^n c^n / n > 0 \}$

abc
aabbcc
aaabbbccc
…

Ejemplo una estrategia



Apilamos una A con cada a

Desapilamos una A con cada b

No se puede comparar las c´s, ya que la pila quedo vacía

No existe ninguna estrategia para reconocer este tipo de lenguajes con un AP

Máquinas de Turing

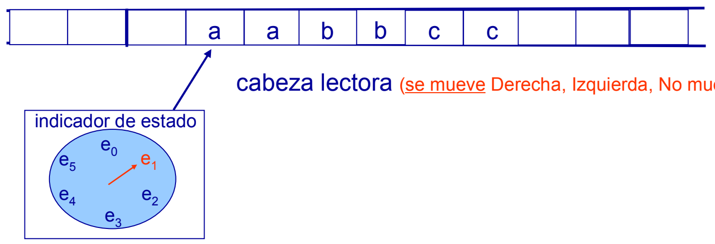
Es necesario agregar algo a los AP para incrementar
su poder computacional

1) Leer y reemplazar símbolos en la cinta de entrada

2) Agregar movimientos a la cabeza lectora (Izq., Der., o no moverse)

Máquinas de Turing

cinta de entrada (espacio infinito a izquierda y derecha) contiene cadena a ser leída



cabeza lectora (se mueve Derecha, Izquierda, No mueve)

mecanismo de control

Estados de MT

- ✓ Cantidad finita.
- ✓ Un estado inicial.
- ✓ Estados finales o de aceptación.

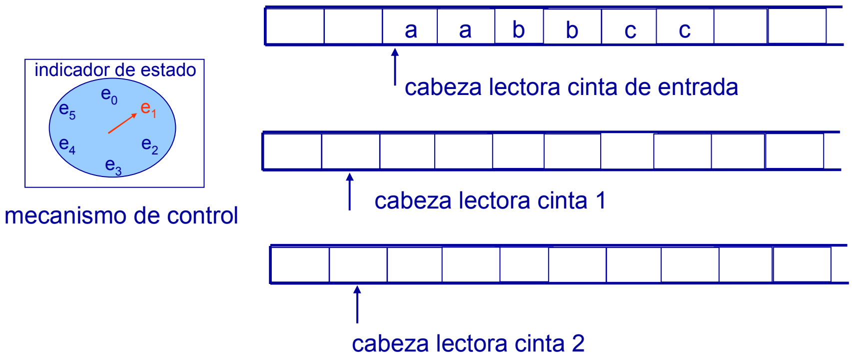
Dada una cadena x en la cinta de entrada, si la MT lee toda la cadena y:

- ➢ **termina en estado final → cadena aceptada**
- ➢ **termina en estado no final → cadena rechazada**

Máquinas de Turing multicinta

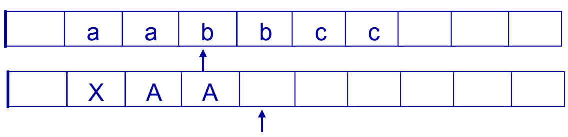
Varias cintas (cada una un espacio ilimitado)

Una cabeza lectora independiente para cada cinta

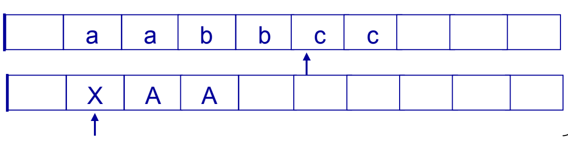


Uso de las cintas

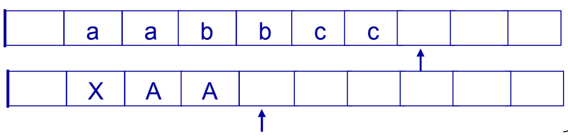
$L_1 = \{ a^n\, b^n\, c^n \mid n > 0 \}$



Marco inicio con X
Copio una A con cada a leída hasta llegar a b y blanco



Retrocedo una A con cada b leída. Si |A| =|b| =>se lee c y X



Avanza una A con cada c leída. Si |A|=|c|=>se lee blanco y blanco

Máquina de Turing

Formalmente, un MT reconocedora determinística se define como una 9-upla

$$MT = \langle E, A, C, \delta, e_0, B, F \rangle$$

- ✓ E es un conjunto finito de estados; $E \neq \emptyset$
- ✓ A es el alfabeto de entrada
- ✓ C es el alfabeto de cinta. $C = A \cup \{B\} \cup \text{Auxiliares}$

$\text{Auxiliares} \cap A = \emptyset$

$A \subseteq C$

- ✓ δ es la función de transición de estados

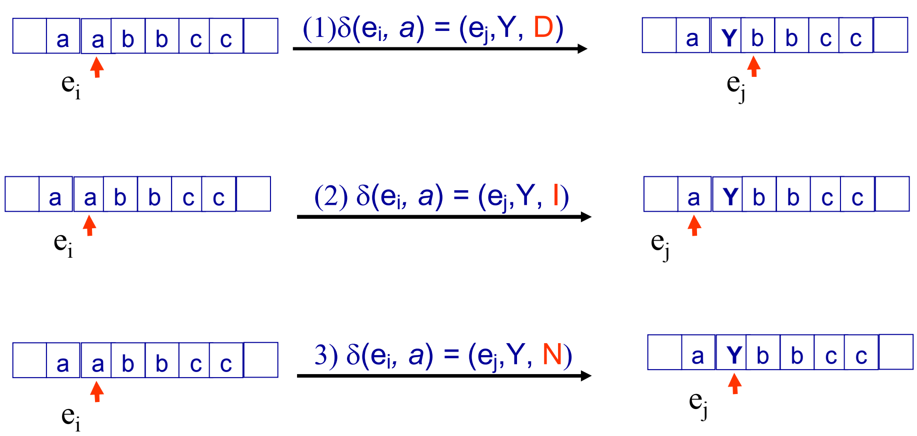
$\delta: E \times C \rightarrow E \times C \times \{I,D,N\}$ 1-cinta

$\delta: E \times C^k \rightarrow E \times (C \times \{I,D,N\})^k$ k-cintas

- ✓ e_0 es el estado inicial; $e_0 \in E$
- ✓ B es el blanco $B \in C$
- ✓ F es el conjunto de estados finales o de aceptación; $F \subseteq E$

Transiciones de MT (1 cinta)

✓ δ es la función de transición de estados $\delta: E \times C \rightarrow E \times C \times \{I,D,N\}$



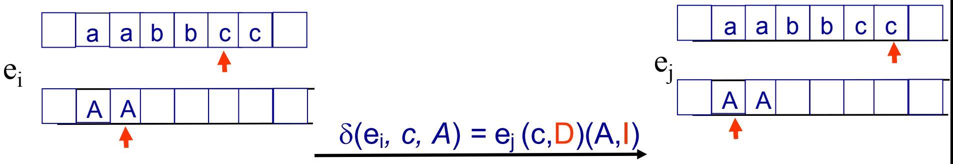
donde $a,b,c,Y \in C$; $e_i, e_j \in E$

Transiciones de MT (k-cintas)

✓ δ es la función de transición de estados $\delta: E \times C^k \rightarrow E \times (C \times \{I,D,N\})^k$

Se lee en cada cinta donde apunta la cabeza lectora y en cada cinta se reemplaza el símbolo leído y se hace un movimiento

Ejemplo 2-cintas



donde $a,b,c,A \in C$; e_i , $e_j \in E$

Relación de transición formal

✓ δ es la función de transición de estados

$\delta: E \times C \rightarrow E \times C \times \{I,D,N\}$ 1-cinta

Formalmente, se define una relación de transición $\vdash$

$x_1x_2\ldots x_{k-1}e_i\, x_k\, x_{k+1}\ldots x_n \vdash x_1x_2\ldots x_{k-1}Y\, e_j\, x_{k+1}\ldots x_n$ si $\delta(e_i, x_k) = (e_j, Y, D)$

$x_1x_2\ldots x_{k-1}e_i\, x_k\, x_{k+1}\ldots x_n \vdash x_1x_2\ldots e_j\, x_{k-1}Y\, x_{k+1}\ldots x_n$ si $\delta(e_i, x_k) = (e_j, Y, I)$

$x_1x_2\ldots x_{k-1}e_i\, x_k\, x_{k+1}\ldots x_n \vdash x_1x_2\ldots x_{k-1}e_j\, Y\, x_{k+1}\ldots x_n$ si $\delta(e_i, x_k) = (e_j, Y, N)$

donde $x_k, Y \in C$; e_i , $e_j \in E$ $x_1,\ldots x_n \in C$

Máquinas de Turing Reconocedoras

Cadena aceptada por MT

Una cadena $\omega \in A^*$ es aceptada por MT = <E, A , C, δ, e_0, B, F> sí y sólo sí

$e_0\, \omega \vdash^* \alpha_1\, e_f\, \alpha_2$

$\vdash^*$ la MT a partir de e_0 con la cabeza lectora apuntando al primer símbolo de w, en varias transiciones usando δ termina de leer toda la cadena w, llega a un estado $e_f \in F$, y en la cinta queda una cadena $\alpha_1\alpha_2 \in C^*$ con la cabeza lectora apuntando al primer símbolo de α_2

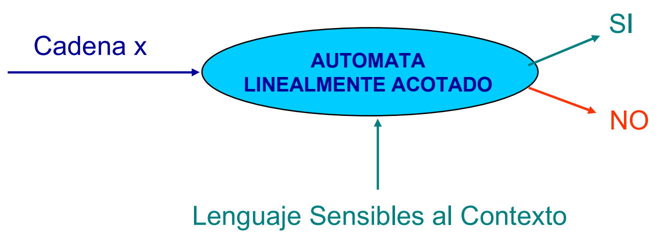
Luego, el lenguaje aceptado por MT es:

$L(MT) = \{ \omega \ / \ e_0\, \omega \vdash^* \alpha_1\, e_f\, \alpha_2 \text{ y } \omega \in A^* \text{ y } e_f \in F \text{ y } \alpha_1, \alpha_2 \in C^* \}$

Los lenguajes aceptados por los **Máquinas de Turing** se denominan **Lenguajes Estructurados por frases o Recursivos enumerables** o de **Tipo 0**.

Autómatas Linealmente Acotados (ALA)

Dado un lenguaje L, sensible al contexto, definido sobre un alfabeto A y una cadena x arbitraria, determinar si $x \in L$ o $x \notin L$.

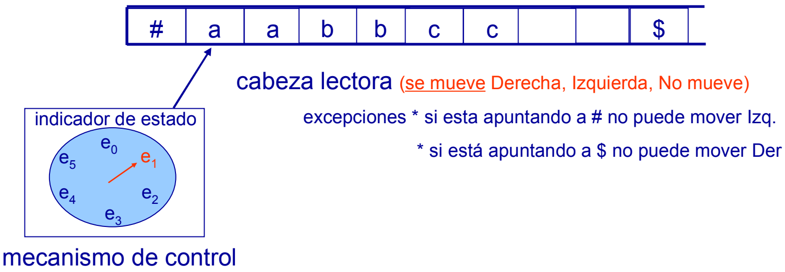


- Dos puntos de vista:
  - ➢ Como dispositivo **reconocedor** de la pertenencia de una cadena a un lenguaje sensible al contexto.
  - ➢ Como **traductor** de una cadena en otra (Ej. Cálculo de funciones)

Autómatas Linealmente Acotados

ALA es una MT especial

cinta de entrada (espacio acotado entre #y $) contiene cadena a ser leída



Estados del ALA

- ✓ Cantidad finita.
- ✓ Un estado inicial.
- ✓ Estados finales o de aceptación.

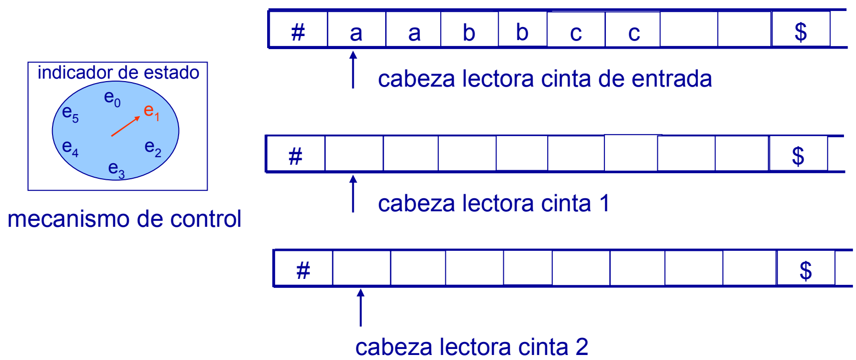
Dada una cadena x en la cinta de entrada, si el ALA lee toda la cadena y:

- ➢ **termina en el estado final → cadena aceptada**
- ➢ **termina en el estado no final → cadena rechazada**

Autómatas Linealmente Acotados multicinta

Varias cintas (cada una en espacio acotado entre #y $)

Una cabeza lectora independiente para cada cinta



Autómata Linealmente Acotado

Formalmente, un ALA reconocedor determinístico se define como una 9-upla

ALA= <E, A, C, δ, e_0, B, F,#, $>

✓ E es un conjunto finito de estados; $E \neq \varnothing$

✓ A es el alfabeto de entrada

Auxiliares $\cap$ A= $\varnothing$

$A \subseteq C$

✓ C es el alfabeto de cinta. C=A $\cup$ {B, $, #} $\cup$ Auxiliares

✓ δ es la función de transición de estados

δ: E x C $\rightarrow$ E x C x {I,D,N} 1-cinta

δ: E x C^k $\rightarrow$ E x (C x {I,D,N}$)^k$ k-cintas

✓ e_0 es el estado inicial; $e_0 \in E$

✓ B es el blanco $B \in C$

✓ F es el conjunto de estados finales o de aceptación; $F \subseteq E$

✓ #, $ inicio y fin espacio δ(e,#)=(e, #,I) y δ(e,$)=(e, $,D) no están definidas

ALA y Máquina de Turing

- En la práctica vamos a usar MT para reconocer lenguajes Sensibles al Contexto (tipo 1). Sin embargo, se debe aclarar que lo correcto es diseñar un ALA para este tipo de lenguajes.

- La razón es que si modelaran con ALA se tendría que calcular el espacio de cinta necesario entre # y $ y no es un tema que se estudia en esta materia.

Máquinas de Turing

- MT multi-cinta y MT 1-cinta

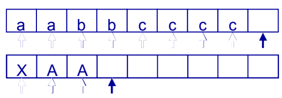
Son modelos equivalentes. Todo lo que se puede hacer con un modelo de k-cintas también se puede diseñar con un modelo de 1-cinta.

- En el modelo multi-cinta resulta más fácil el diseño del autómata

Reconocimiento de lenguajes

$L = \{ a^n b^n c^{2n} / n > 0 \}$

Estado actual	C1	C2	C1		C2		Nuevo estado
e0	a	B	a	N	X	D	e1
e1	a	B	a	D	A	D	e1
	b	B	b	N	B	I	e2
e2	b	A	b	D	A	I	e2
	c	X	c	N	X	D	e3
e3	c	A	c	D	A	N	e4
	B	B	B	N	B	N	e5
e4	c	A	c	D	A	D	e3
e5	-	-	-	-	-	-	-



e_3

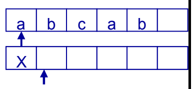
Estado actual

$MT = <\{e_0,e_1,e_2,e_3,e_4,e_5\},\{a,b,c\},\{B,X,A,a,b,c\},\delta, e_0, B, \{e_5\}>$

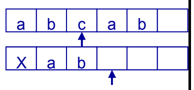
L = { wcw / w ∈ {a,b}* }   {c, abcab,bbcbb…..}

Estado actual	C1	C2	C1		C2		Nuevo estado
e0	c	B	c	D	B	N	e4
	a	B	a	N	X	D	e1
	b	B	b	N	X	D	e1
e1	a	B	a	D	a	D	e1
	b	B	b	D	b	D	e1
	c	B	c	N	B	I	e2
e2	c	a	c	N	a	I	e2
	c	b	c	N	b	I	e2
	c	X	c	D	X	D	e3
e3	a	a	a	D	a	D	e3
	b	b	b	D	b	D	e3
	B	B	B	N	B	N	e5
e4	B	B	B	N	B	N	e5
e5	-	-	-	-	-	-	-

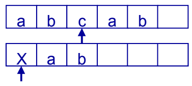
c
marca X
marca X



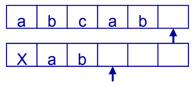
Copia a o b
hasta c



retrocede
a o b
hasta X



compara
igualdad

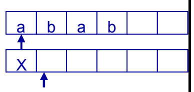


despues de c hay blanco?

L = { ww / w ∈ {a,b}* }   {ε, abab,bbbb, abaaba…..}

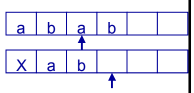
Estado actual	C1	C2	C1		C2		Nuevo estado
e0	B	B	B	N	B	N	e4
	a	B	a	N	X	D	e1
	b	B	b	N	X	D	e1
e1	a	B	a	D	a	D	e1
	b	B	b	D	b	D	e1
	a	B	a	N	B	I	e2
	b	B	b	N	B	I	e2
e2	a	a	a	N	a	I	e2
	a	b	a	N	b	I	e2
	b	b	b	N	b	I	e2
	b	a	b	N	a	I	e2
	a	X	a	N	X	D	e3
	b	X	b	N	X	D	e3
e3	a	a	a	D	a	D	e3
	b	b	b	D	b	D	e3
	B	B	B	N	B	N	e4
e4	-	-	-	-	-	-	-

B
marca X
marca X

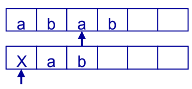


Copia a o b

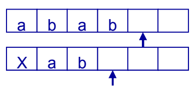
No copia
mas



retrocede
a o b
hasta X



compara
igualdad



Máquina de Turing (determinística y no det.)

Formalmente, un MT se define como antes una 7-upla
MT= $\langle E, A, C, \delta, e_0, B, F \rangle$ sólo cambia la definición de la función δ

✓ δ: E x $C^k \rightarrow$ E x (C x {I,D,N}$)^K$ k-cintas (determinística)

✓ δ: E x $C^k \rightarrow P_f$(E x (C x {I,D,N}$)^k$) k-cintas (no determinística)

P_f: subconjuntos finitos

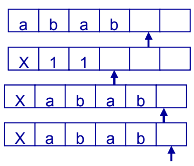
$$\text{Ejemplo no det } \delta(e_1, a, X, X) = \begin{cases} e_1\ (a, D)\ (X, D)\ (X, D) \\ e_2\ (A, N)\ (Y, D)\ (Y, D) \\ \ldots \qquad\qquad a,A,X,Y \in C \quad e_1,e_2 \in E \end{cases}$$

•Existe equivalencia entre: el modelo MT determinístico y MT no determinístico, ya que aceptan los mismos lenguajes.

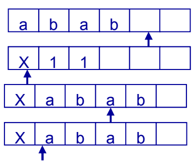
Solución con MT determinística para el ejemplo

L = { ww / w $\in$ {a,b}* } {ε, abab,bbbb, abaaba…..}

Generalmente el enunciado dice que en la cinta de entrada solo se permiten movimientos D



C1: la cadena de entrada
C2: Calcular en unario la longitud de la cadena dividido 2
C3: Copiar la cadena
C4: Copiar la cadena



C1: la cadena de entrada
C2: Usarla para retroceder hasta el principio de la 2da w en C3 y al principio de la 1er w en C4

Comparar la 2da w en C3 y la 1er w en C4

Ciencias de la Computación I - Filminas de Clase – Facultad Cs. Exactas – UNCPBA - 2009