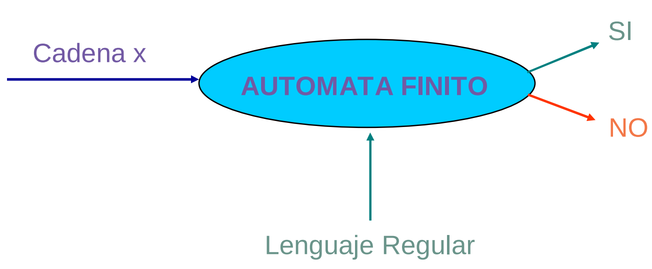
Ciencias de la Computación I

Autómatas Finitos y Lenguajes Regulares

Autómatas Finitos y Lenguajes Regulares

Problema:

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



Autómatas Finitos

• Un **Autómata Finito** es un modelo matemático de una máquina abstracta con entradas y salidas discretas.

• Dos puntos de vista:

➢ Como dispositivo **reconocedor** de la pertenencia de una cadena a un lenguaje regular.

➢ Como **traductor** de una cadena en otra.

• Un AF puede leer símbolos de una cinta, y puede estar en un número finito de estados.

Autómatas Finitos

Aplicaciones:

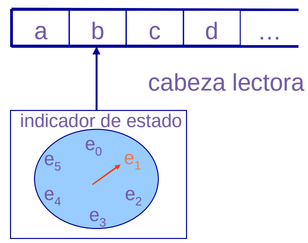
✓Análisis de cadenas de caracteres (búsqueda de una cadena en un archivo de texto, reconocimiento de cadenas que satisfacen ciertos criterios, etc.)

✓ Reproductor de video, máquina expendedora de boletos, etc.

Ciencias de la Computación I - Filminas de Clase – Facultad Cs. Exactas – UNCPBA - 2012

Autómatas Finitos Reconocedores

cinta de entrada (contiene cadena a ser leída)



cabeza lectora (se mueve a derecha)

mecanismo de control

Estados del AF:

- ✓ Cantidad finita.
- ✓ Representan la “memoria” del autómata.
- ✓ Un estado inicial.
- ✓ Al menos un estado final o de aceptación.

Dada una cadena x en la cinta de entrada, si el AF:

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

Autómatas Finitos Reconocedores

L = { x / x $\in$ {a, b, c}* y x termina en b }

Cadenas que pertenecen a L	Cadenas que no pertenecen a L
b	ε
ab	a
bb	c
cb	aa
aab	ba
bab	…
…	

Dos situaciones para distinguir:

- el último símbolo leído es distinto de b
- el último símbolo leído es b

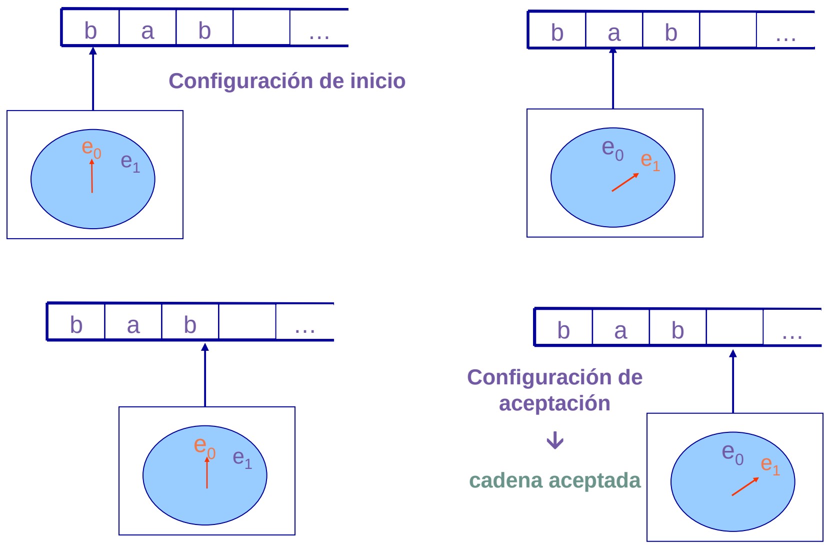
Dos estados:

e_0

e_1

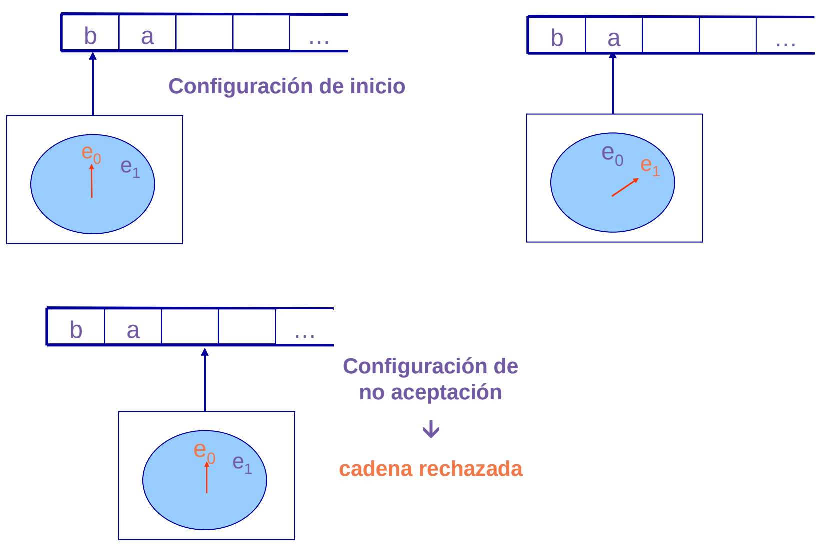
Autómatas Finitos Reconocedores

e_0: estado inicial (último símbolo leído ≠ b) **e_1**: estado final (último símbolo leído es b)



Autómatas Finitos Reconocedores

e_0: estado inicial (último símbolo leído ≠ b) **e_1**: estado final (último símbolo leído es b)



Autómatas Finitos Reconocedores

Para definir un AF reconocedor es necesario indicar:

- ✓ el alfabeto de entrada: A
- ✓el conjunto finito de estados: $E=\{e_0, e_1, \ldots, e_n\}$
- ✓ de estos estados, un único estado inicial: e_0
- ✓ de estos estados, uno o varios estados finales: F
- ✓ una función de transición de estados: δ (indica a qué estado pasar luego de leer un símbolo en la cinta de entrada)

Autómatas Finitos Reconocedores

Formalmente, un AF reconocedor determinístico (AFD) se define como una quintupla

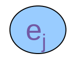
$$M = \langle E, A, \delta, e_i, F \rangle$$

- ✓ E es un conjunto finito de estados; $E \neq \emptyset$
- ✓ A es el alfabeto de entrada
- ✓ δ es la función de transición de estados; $\delta: E \times A \rightarrow E$
  $\delta(e_j, a) = e_k$ la máquina puede pasar del estado e_j al e_k después de leer el símbolo a en la cinta ($e_j, e_k \in E$; $a \in A$)
- ✓ e_i es el estado inicial; $e_i \in E$
- ✓ F es el conjunto de estados finales o de aceptación; $F \subseteq E$

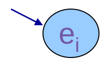
Autómatas Finitos Reconocedores

Un AF reconocedor determinístico se puede representar gráficamente usando un diagrama de transición de estados.

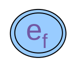
- cada estado $e_j \in E$



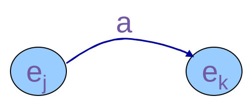
- estado inicial e_i



- cada estado final $e_f \in F$



-cada transición entre estados
$\delta(e_j, a) = e_k$ para $e_j, e_k \in E, a \in A$



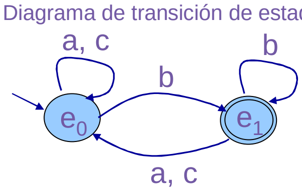
Autómatas Finitos Reconocedores

$L = \{ x \,/\, x \in \{a, b, c\}^* \text{ y } x \text{ termina en b} \}$

e_0: estado inicial (último símbolo leído $\neq$ b)

e_1: estado final (último símbolo leído es b)

Diagrama de transición de estados



Descripción instantánea

$\alpha\ e_i\ \beta$ donde e_i estado actual, α cadena ya leída, β cadena que falta leer ($\alpha, \beta \in A^*$)

Ejemplos

$e_0abcb \vdash ae_0bcb \vdash abe_1cb \vdash abce_0b \vdash abcbe_1$ → lee abcb y termina en estado final. Luego, abcb $\in$ L

$e_0ba \vdash be_1a \vdash bae_0$ → lee ba y termina en estado no final. Luego, ba $\notin$ L

Autómatas Finitos Reconocedores

L = { x / x $\in$ {a, b, c}* y x termina en b }

Diagrama de transición de estados

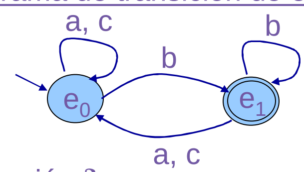


Tabla de transición de estados:

δ	a	b	c
e_0	e_0	e_1	e_0
e_1	e_0	e_1	e_0

Función δ

$\delta(e_0, a)=e_0$ $\delta(e_0, b)=e_1$ $\delta(e_0, c)=e_0$

$\delta(e_1, a)=e_0$ $\delta(e_1, b)=e_1$ $\delta(e_1, c)=e_0$

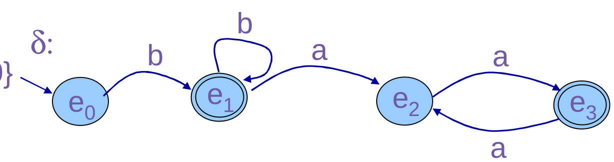
AFD = <{e_0, e_1}, {a, b, c}, δ, e_0, {e_1}>

El AFD = <{e_0, e_1}, {a, b, c}, δ, e_0, {e_1}> acepta una cadena x si la secuencia de transiciones correspondientes a los símbolos de x conduce desde e_0 (el estado inicial) a e_1 (el único estado final).

Definición parcial de la función δ

Ejemplo

L ={$b^n a^{2m}$/ n>0 y m≥0}



$\delta(e_0$, a) no está definida
$\delta(e_2$, b) no está definida
$\delta(e_3$, b) no está definida

Dada una cadena x, si el autómata no puede leer un símbolo de x entonces la cadena x $\notin$ L

$e_0$aa ├ indefinido no termina de leer toda la cadena entonces aa $\notin$ L

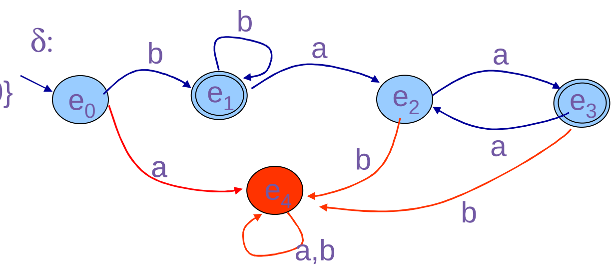
$e_0$baab ├ b$e_1$aab ├ ba$e_2$ab ├ baae_3b ├ indefinido no termina de leer la cadena entonces baab $\notin$ L

Definición total de la función δ

Ejemplo

L ={$b^n a^{2m}$/ n>0 y m≥0}

δ:



• Todas las transiciones indefinidas se pueden hacer llegar a un estado absorbente (e_4) que nunca alcanza un estado final

• Los estados como e_4 no se usan en la práctica (se define δ parcial)

Lenguaje aceptado por un AFD

Sea M = <E, A, δ, e_0, F> un AF donde

δ: E x A → E

Se define la función δ^*: E x A* → E (función de transición para cadenas)

- $\delta^*(e_j, \varepsilon) = e_j$
- $\delta^*(e_j, a) = \delta(e_j, a)$ $e_i \in E, a \in A$
- $\delta^*(e_j, ax) = \delta^*(\delta(e_j, a), x)$ $e_j \in E, x \in A^*, a \in A$

Una cadena x es aceptada por un AFD M = <E, A, δ, e_i, F> si:

$\delta^*(e_i, x) = e_f$ para algún $e_f \in F$

Luego, el lenguaje aceptado por un AFD M = <E, A, δ, e_i, F> es:

L(M) = { x / $x \in A^*$ y $\delta^*(e_i, x) = e_f$ y $e_f \in F$}

Los lenguajes aceptados por los **Autómatas Finitos** se denominan **Lenguajes Regulares** o de **Tipo 3**.

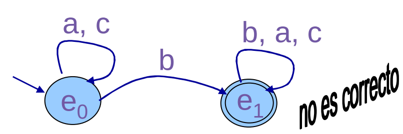
Diseño de Autómatas Finitos

↓ No es conveniente proceder por "prueba y error", pueden cometerse dos tipos de errores:

1) "sobren" cadenas (el AF acepta cadenas que no debería aceptar)

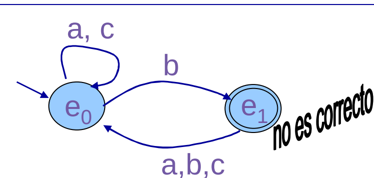
2) "falten" cadenas" (el AF no acepta todas las cadenas del lenguaje)

Ejemplo $L = \{ x / x \in \{a, b, c\}^* \text{ y } x \text{ termina en b} \}$



"sobran" reconoce cadenas que $\notin$ L

ab	$\in$ L	la acepta	correcto
bc	$\notin$ L	la acepta	incorrecto
baa	$\notin$ L	la acepta	incorrecto



"faltan" no reconoce cadenas que $\in$ L

ab $\in$ L	la acepta	correcto
bb $\in$ L	no la acepta	incorrecto
cbb $\in$ L	no la acepta	incorrecto

Diseño de Autómatas Finitos

↑ Importante para un diseño sistemático:

1) Proponer un conjunto de estados que "recuerdan" condiciones importantes en el problema considerado

2) De estos estados, determinar cuál representa la condición inicial y cuál/cuáles la condición de aceptación

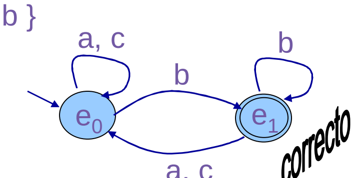
3) Proponer las transiciones que permiten pasar de un estado a otro

Ejemplo

$L = \{ x / x \in \{a, b, c\}^* \text{ y } x \text{ termina en b} \}$

$\mathbf{e_0}$: "recuerda" que último símbolo leído es $\neq$ b

$\mathbf{e_1}$: "recuerda" que último símbolo leído es b

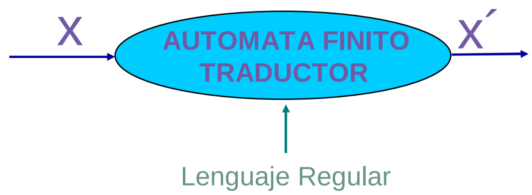


"correcto" no sobran ni faltan cadenas
Reconoce todas y sólo las que pertenecen a L

Autómatas Finitos Traductores

- Producen una salida diferente de SI o NO

- Permiten realizar “cálculos” a partir de una cadena de entrada → “traducen” una cadena de entrada en una cadena de salida



Ejemplos:

- AF que calcule la función f(x) = 2x +3
- Analizador léxico de un compilador

Autómatas Finitos Traductores

Formalmente, un AF traductor determinístico (AFT) se define como una 7-tupla

$$M_T = <E, A, \delta, e_i, F, S, \gamma>$$

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

✓ A es el alfabeto de entrada

✓ δ es la función de transición de estados; $\delta: E \times A \rightarrow E$

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

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

✓ S es el alfabeto de salida

✓ γ es la función de traducción; $\gamma : E \times A \rightarrow S^*$

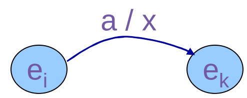
Ciencias de la Computación I - Filminas de Clase – Facultad Cs. Exactas – UNCPBA - 2012

Autómatas Finitos Traductores

Si existen $\delta(e_i, a) = e_k$ y $\gamma(e_i, a) = x$

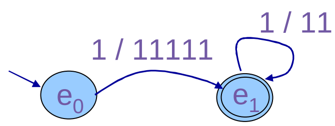
donde $e_i, e_k \in E$; $a \in A$; $x \in S^*$

se representa en el diagrama de transición de estados



Ejemplo:

Autómata finito traductor que calcula f(x) = 2x + 3 para $x \in N$, $x > 0$, x representado en unario



Ejemplos
si x =1 traduce 11111
si x =11 traduce 1111111
si x =111 salida 1^9
si x =11111 salida 1^{13}

AFT = $<\{e_0, e_1\}, \{1\}, \delta, e_0, \{e_1\}, \{1\}, \gamma>$

Autómatas Finitos Traductores

Definición de γ^* (función de traducción para cadenas)

Sea $M_T = <E, A, \delta, e_0, F, S, \gamma>$ un AFT. Se define la función

$\gamma^*: E \times A^* \rightarrow S^*$ tal que $\gamma^*(e_i, w)$ es la cadena que traducirá el autómata luego de leer w comenzando en e_i ($w \in A^*$)

- $\gamma^*(e_i, \varepsilon) = \varepsilon$
- $\gamma^*(e_i, ax) = \gamma(e_i, a) \,.\, \gamma^*(\delta(e_i, a), x)$ $e_i \in E$, $x \in A^*$, $a \in A$

Nota:
El autómata solo define la traducción, si el autómata finito reconocedor subyacente “acepta” la cadena.

Es decir, la traducción $T(w): A^* \rightarrow S^*$ asociada a M_T está definida como:
$T(w) = \gamma^*(e_0, w) \Leftrightarrow \delta^*(e_0, w) \in F$ donde $w \in A^*$

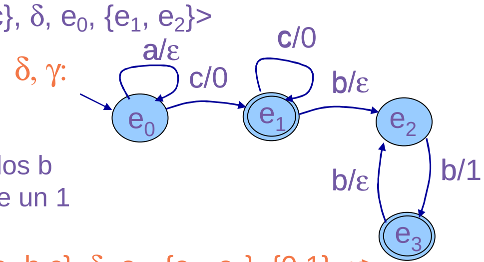
Ejemplo Autómata Finito Traductor

$L = \{a^n c^k b^{2m} / \ n, m \geq 0 \ \ y \ \ k > 0 \}$

$M = <\{e_0, e_1, e_2, e_3\}, \{a, b, c\}, \delta, e_0, \{e_1, e_2\}>$

Traducir las cadenas $a^n c^k b^{2m}$ como $0^k 1^m$

cada a no traduce — cada c traduce 0 — cada dos b traduce un 1



AFDT = $<\{e_0, e_1, e_2, e_3\}, \{a, b, c\}, \delta, e_0, \{e_1, e_2\}, \{0,1\}, \gamma>$

$c \in L$ traduce 0

$\varepsilon \notin L$ no traduce

$b \notin L$ no traduce

$ca \notin L$ no traduce

$cbb \in L$ traduce 01

$ac \in L$ traduce 0

$aaaccccbb \in L$ traduce 00001

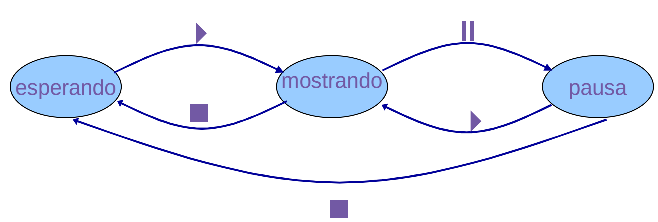
Autómatas Finitos Modelos

Formalmente, un AF modelo se define como una 3-upla

$M_M = < E, A, \delta >$

- ✓ E es un conjunto finito de estados; $E \neq \emptyset$
- ✓ A es el alfabeto de entrada
- ✓ δ es la función de transición de estados; $\delta: E \times A \rightarrow E$

• Ejemplo modelo de Videograbadora



M_M = <{esperando, mostrando, pausa},{ ▶ , II , ■ }, δ>