Ciencias de la Computación I

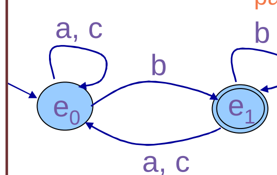
Autómatas Finitos No Determinísticos

Minimización de Autómatas Finitos Determinísticos

Autómatas Finitos: Determinismo – No determinismo

Determinísticos: Para cada estado y cada símbolo se puede pasar a un único estado:

Un único camino



Ejemplo: aab

$e_0 \xrightarrow{a} e_0 \xrightarrow{a} e_0 \xrightarrow{b} e_1$ Llega a e_f => aab $\in$ L

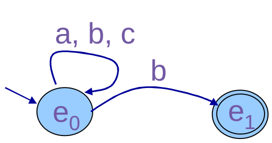
Ejemplo: aba

$e_0 \xrightarrow{a} e_0 \xrightarrow{b} e_1 \xrightarrow{a} e_0$ No llega a e_f => aba $\notin$ L

No Determinísticos: Para algunos estados, dado un símbolo, se puede pasar a más de un estado:

Varios caminos

Ejemplo: aab



$e_0 \xrightarrow{a} e_0 \xrightarrow{a} e_0 \xrightarrow{b} e_0$

$\phantom{e_0 \xrightarrow{a} e_0 \xrightarrow{a}} e_0 \xrightarrow{b} e_1$

Al menos un camino llega a e_f => aab $\in$ L

Ejemplo: aba

$e_0 \xrightarrow{a} e_0 \xrightarrow{b} e_0 \xrightarrow{a} e_0$

$\phantom{e_0 \xrightarrow{a}} e_0 \xrightarrow{b} e_1$

Ningún camino llega a e_f => aba $\notin$ L

Autómatas Finitos No Determinísticos

Formalmente, un AF reconocedor no determinístico (AFND) se define como una quintupla

$$M = <E, A, \delta, e_i, F>$$

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

$\delta: E \times A \rightarrow P(E)$ P(E) conjunto potencia de E

$\delta(e_j, a) = \{e_k, e_s, e_t, \dots\}$ $e_j, e_k, e_s, e_t, \in E;\ a \in A$

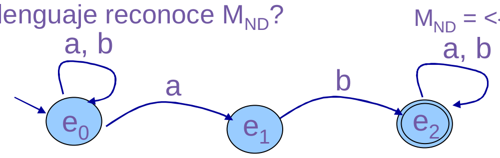
- ✓ e_i es el estado inicial; $e_i \in E$
- ✓ F es el conjunto de estados finales; $F \subseteq E$

Autómatas Finitos No Determinísticos

Ejemplo:

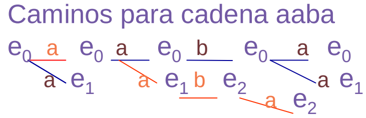
¿Qué lenguaje reconoce M_{ND}?

$M_{ND} = <\{e_0, e_1, e_2\}, \{a, b\}, \delta, e_0, \{e_2\}>$

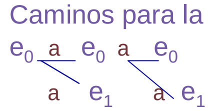


Las cadenas aaba y aa, ¿pertenecen o no a $L(M_{ND})$?

Caminos para cadena aaba



Caminos para la cadena aa



$L = \{ x / x \in \{a, b\}^* \text{ y } x \text{ contiene la subcadena ab} \}$

Aceptación de cadena por AFND

Un AFND acepta una cadena si existe alguna secuencia de transiciones que a partir del primer símbolo de la cadena y empezando en el estado inicial, permite alcanzar un estado final luego de leer todos los símbolos de la cadena.

Determinismo y No determinismo

Determinismo $\rightarrow$ existe una alternativa válida, o no hay alternativa.

No Determinismo $\rightarrow$ puede haber varias alternativas válidas.

Importante distinguir si el no determinismo agrega o no poder computacional

En los Autómatas Finitos, todo se puede resolver con un Autómata Finito Determinístico

EL NO DETERMINISMO NO AGREGA PODER COMPUTACIONAL

Equivalencia entre AFND y AFD

Teorema: Sea L un lenguaje aceptado por un AFND. Entonces existe un AFD que acepta el mismo lenguaje L.

Algoritmo para obtener AFD a partir de AFND:

Dado $M_{ND} = \langle E_{ND}, A_{ND}, \delta_{ND}, e_{0ND}, F_{ND} \rangle$ AFND se define
$M_D = \langle E_D, A_D, \delta_D, e_{0D}, F_D \rangle$ AFD tal que $L(M_{ND})=L(M_D)$

-$E_D = P(E_{ND})$ (conjunto potencia de E_{ND}).

Cada elemento de E_D se representa como
$[e_1, e_2, \ldots, e_i]$ donde $e_1, e_2, \ldots, e_i \in E_{ND}$
$[e_1, e_2, \ldots, e_i]$ es un único estado de M_D

$P(E_{ND})$		E_D
$\varnothing$	$\longrightarrow$	$[\varnothing]$
$\{e_1\}$	$\longrightarrow$	$[e_1]$
$\{e_2\}$	$\longrightarrow$	$[e_2]$
$\{e_1, e_2\}$	$\longrightarrow$	$[e_1,e_2]$
….		…

- $A_D = A_{ND}$

- $e_{0D} = [e_{0ND}]$

-F_D: subconjuntos de $P(E_{ND})$ que contienen al menos un estado $e_i \in F_{ND}$.

Equivalencia entre AFND y AFD

<u>Algoritmo para obtener AFD a partir de AFND</u>:

-δ_D: E_D x A $\rightarrow$ E_D, se define como

$\delta_D([e_1,..., e_i], a) = [e_1, ..., e_k]$ sii

$\delta_{ND}(\{e_1,..., e_i\}, a) = \delta^G (\{e_1,..., e_i\}, a) = \{e_l,..., e_k\}$,

donde

$\delta^G (C, a) = \bigcup_{e \in C} \delta (e, a)$ (C: conj. de estados)

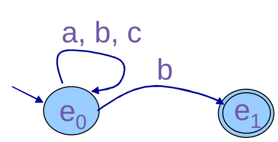
$\delta^G (\emptyset, a) = \emptyset$

δ_D aplicada a un elemento $[e_1, e_2, ..., e_i]$ de E_D se obtiene calculando δ_{ND} para cada estado de E_{ND} que está en $[e_1, e_2, ..., e_i]$.

Equivalencia entre AFND y AFD

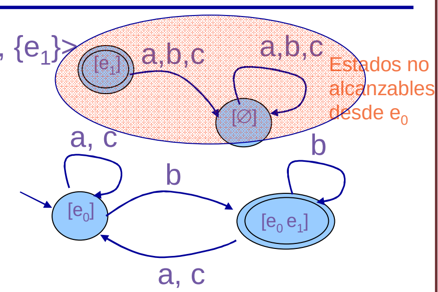
Sea $M_{ND} = \langle\{e_0, e_1\}, \{a, b, c\}, \delta_{ND}, e_0, \{e_1\}\rangle$

$L(M_{ND}) = \{ x / x \in \{a, b, c\}^*$ y x termina en b$\}$



δ_{ND}	a	b	c
e_0	$\{e_0\}$	$\{e_0, e_1\}$	$\{e_0\}$
e_1	$\emptyset$	$\emptyset$	$\emptyset$

Función de transición no determinística



δ_D	a	b	c
$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
$[e_0]$	$[e_0]$	$[e_0, e_1]$	$[e_0]$
$[e_1]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
$[e_0, e_1]$	$[e_0]$	$[e_0, e_1]$	$[e_0]$

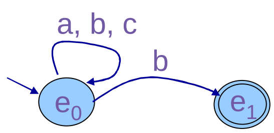
$\delta_{ND}(e_0, a) \cup \delta_{ND}(e_1, a)$ $\delta_{ND}(e_0, b) \cup \delta_{ND}(e_1, b)$ $\delta_{ND}(e_0, c) \cup \delta_{ND}(e_1, c)$

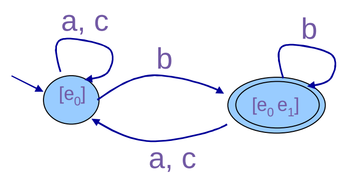
Función de transición determinística

Equivalencia entre AFND y AFD

Sea $M_{ND} = <\{e_0, e_1\}, \{a, b, c\}, \delta, e_0, \{e_1\}>$

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





En la práctica no se trabaja con todo el conjunto potencia $P(E_{ND})$

δ_{ND}	a	b	c
e_0	$\{e_0\}$	$\{e_0, e_1\}$	$\{e_0\}$
e_1	$\varnothing$	$\varnothing$	$\varnothing$

Función de transición no determinística

δ_D	a	b	c
$[e_0]$	$[e_0]$	$[e_0, e_1]$	$[e_0]$
$[e_0, e_1]$	$[e_0]$	$[e_0, e_1]$	$[e_0]$

Función de transición determinística

Minimización de AFD

Teorema

Para cada AFD existe un AFD_{min} con cantidad mínima de estados que acepta el mismo lenguaje, es decir $L(AFD)=L(AFD_{min})$

Algoritmo para minimizar un AFD

(divide al conjunto de estados del AFD en clases de estados equivalentes)

Dado un $AFD = <E, A, \delta, e_0, F>$,

dos **estados** $p, q \in E$ son **equivalentes** $\Leftrightarrow$ para toda cadena $x \in A^*$,

$$\delta^* (p,x) \in F \Leftrightarrow \delta^* (q,x) \in F$$

ó

$$\delta^* (p,x) \notin F \Leftrightarrow \delta^* (q,x) \notin F$$

Minimización de AFD

Algoritmo para minimizar un AFD

1) Eliminar estados no alcanzables desde el estado inicial (estados inalcanzables)

2) Eliminar estados desde los que no se alcanza un estado final (estados muertos)

3) Construir una partición Π_0 del conjunto de estados, que consiste en dos grupos: estados finales y estados no finales.

4) Sea K = 0.

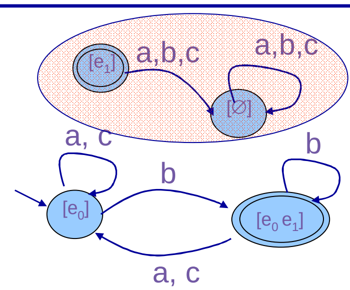
5) Definir Π_{K+1} de la siguiente manera:
para cada grupo G de una partición Π_K, dividir a G en subgrupos tales que dos estados s y t están en el mismo grupo sí y sólo sí para todo símbolo a del alfabeto de entrada, los estados s y t van al mismo grupo de Π_K.

6) K = K + 1.

7) Si $\Pi_K \neq \Pi_{K-1}$ volver al paso 5. En caso contrario, terminar.

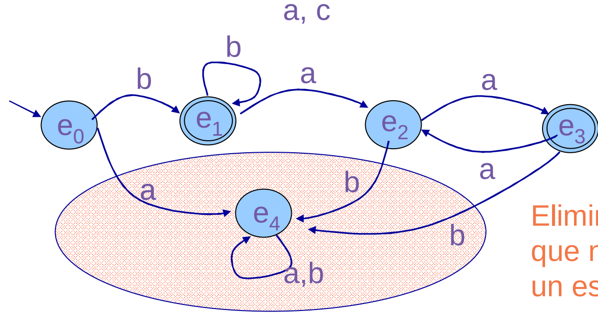
Minimización

Ejemplo 1



Eliminar estados no alcanzables desde el estado inicial

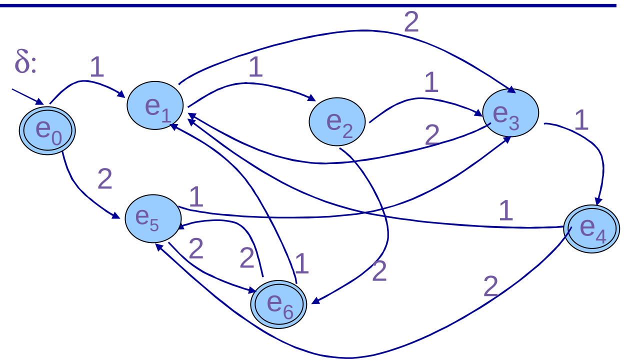
Ejemplo 2



Eliminar estados que no alcanzan un estado final

Minimización

Ejemplo 3



δ	1	2
e_0	e_1	e_5
e_1	e_2	e_3
e_2	e_3	e_6
e_3	e_4	e_1
e_4	e_1	e_5
e_5	e_3	e_6
e_6	e_1	e_5

$M= <\{e_0,e_1,e_2,e_3,e_4,e_5,e_6\},\{1, 2\},\delta, e_0, \{e_0,e_4,e_6\}>$

L={ x / x $\in$ {1, 2}* y la suma de los símbolos en x es múltiplo de 4}

Minimización

$M= <\{e_0,e_1,e_2,e_3,e_4,e_5,e_6\},\{1, 2\},\delta, e_0, \{e_0,e_4,e_6\}>$

Ejemplo 3 (continuación)

δ	1		2	
e_0	e_1	G1	e_5	G1
e_1	e_2	G1	e_3	G1
e_2	e_3	G1	e_6	G2
e_3	e_4	G2	e_1	G1
e_4	e_1	G1	e_5	G1
e_5	e_3	G1	e_6	G2
e_6	e_1	G1	e_5	G1

- No hay estados inalcanzables
- No hay estados muertos

Π_0 $\underbrace{e_1e_2e_3e_5}_{G1}$ $\underbrace{e_0e_4e_6}_{G2}$

Π_1 $\underbrace{e_1}_{G11}$ $\underbrace{e_2e_5}_{G12}$ $\underbrace{e_3}_{G13}$ $\underbrace{e_0e_4e_6}_{G2}$

Π_0 distinto de Π_1

$\Downarrow$

volver al paso 5

Minimización

Ejemplo 3 (continuación)

δ	1		2	
e_0	e_1	G11	e_5	G12
e_1	e_2	G12	e_3	G13
e_2	e_3	G13	e_6	G2
e_3	e_4	G2	e_1	G11
e_4	e_1	G11	e_5	G12
e_5	e_3	G13	e_6	G2
e_6	e_1	G11	e_5	G12

Π_1

G11	G12	G13	G2
e_1	e_2e_5	e_3	$e_0e_4e_6$

Π_2

G11	G12	G13	G2
e_1	e_2e_5	e_3	$e_0e_4e_6$

$\Pi_1 = \Pi_2$ terminar

Minimización

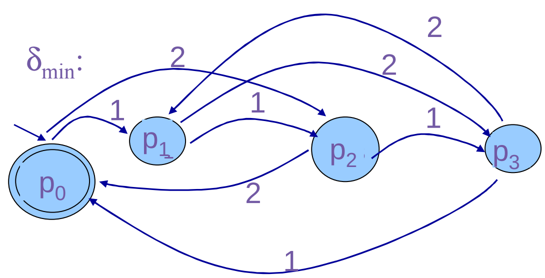
Ejemplo 3 (continuación)

Π_2

G11	G12	G13	G2
e_1	e_2e_5	e_3	$e_0e_4e_6$
p_1	p_2	p_3	p_0

δ	1	2
e_0	e_1	e_5
e_1	e_2	e_3
e_2	e_3	e_6
e_3	e_4	e_1
e_4	e_1	e_5
e_5	e_3	e_6
e_6	e_1	e_5

δ_{min}:



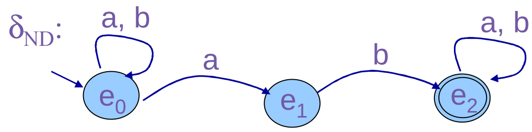
M_{min}= <{p_0, p_1, p_2, p_3}, {1, 2}, δ_{min}, p_0, {p_0}>

L={ x / x $\in$ {1, 2}* y la suma de símbolos en x es múltiplo de 4}

Ejemplo: Pasaje AFND-AFD y Minimización

Para el siguiente AFND:
- Hacer pasaje AFND-AFD
- Minimizar AFD

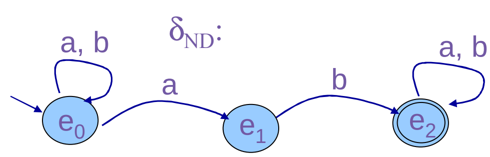
$L = \{ x \,/\, x \in \{a, b\}^* \text{ y } x \text{ contiene la subcadena ab} \}$



AFND= $<\{e_0, e_1, e_2\}, \{a, b\}, \delta_{ND}, e_0, \{e_2\}>$

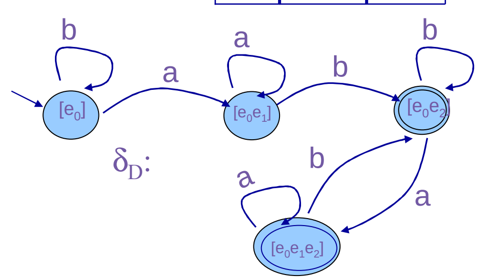
δ_{ND}	a	b
e_0	$\{e_0, e_1\}$	$\{e_0\}$
e_1	$\varnothing$	$\{e_2\}$
e_2	$\{e_2\}$	$\{e_2\}$

Ejemplo: Pasaje AFND-AFD



δ_{ND}	a	b
e_0	$\{e_0, e_1\}$	$\{e_0\}$
e_1	$\varnothing$	$\{e_2\}$
e_2	$\{e_2\}$	$\{e_2\}$

δ_D	a	b
$[e_0]$	$[e_0, e_1]$	$[e_0]$
$[e_0, e_1]$	$[e_0, e_1]$	$[e_0, e_2]$
$[e_0, e_2]$	$[e_0, e_1, e_2]$	$[e_0, e_2]$
$[e_0, e_1, e_2]$	$[e_0, e_1, e_2]$	$[e_0, e_2]$



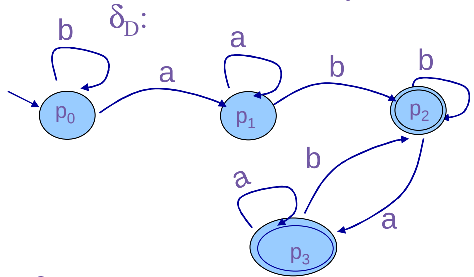
AFD= $<\{[e_0], [e_0e_1], [e_0e_2], [e_0e_1e_2]\}, \{a, b\}, \delta_D, [e_0], \{[e_0e_2], [e_0e_1e_2]\}>$

Ejemplo: Minimización

$L = \{ x \,/\, x \in \{a, b\}^* \text{ y } x \text{ contiene la subcadena ab} \}$

δ_D	a	b
p_0	p_1	p_0
p_1	p_1	p_2
p_2	p_3	p_2
p_3	p_3	p_2

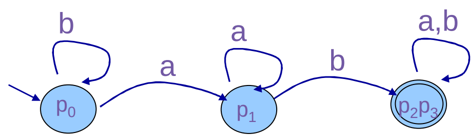
δ_D:



- No hay estados inalcanzables
- No hay estados muertos

Π_0	G1: p_0p_1		G2: p_2p_3
Π_1	G11: p_0	G12: p_1	G2: p_2p_3
Π_2	G11: p_0	G12: p_1	G2: p_2p_3

δ_{min}:



$AFD_{Min} = <\{[p_0], [p_1], [p_2p_3]\}, \{a, b\}, \delta_{Min}, [p_0], \{[p_2p_3]\}>$