PROPIEDADES DE LOS LENGUAJES ACEPTADOS POR

AUTÓMATAS FINITOS

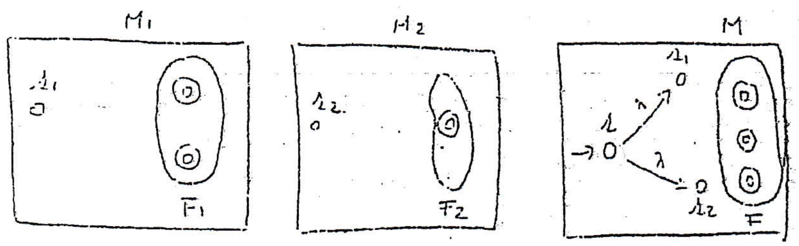
La clase de lenguajes aceptados por autómatas finitos es cerrado bajo s

No deter. {
- a → La Unión
- b → Concatenación
- c → Estrella de Kleene
- d → Complementación
- e → Intersección

Prueba: Sean $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$ y $M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$ AFN que aceptan $L(M_1)$ y $L(M_2)$ respectivamente

a) UNIÓN

Construimos el autómata $M = (K, \Sigma, \Delta, s, F)$ tal que $L(M) = L(M_1) \cup L(M_2)$



donde: $K = K_1 \cup K_2 \cup \{s\}$

s es un nuevo estado inicial

$F = F_1 \cup F_2$

$\Delta = \Delta_1 \cup \Delta_2 \cup \{(s, \lambda, s_1), (s, \lambda, s_2)\}$

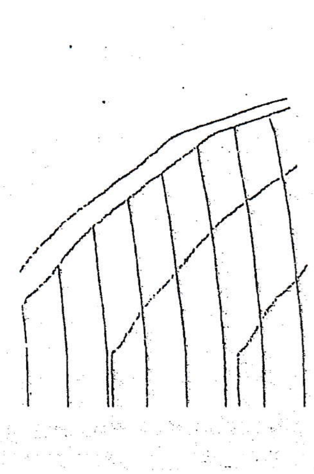
La Unión de lenguajes aceptados por autómatas finitos es también un lenguaje aceptado por autómatas finitos

Para construir autómata unión -

→ Se añade un nuevo estado inicial

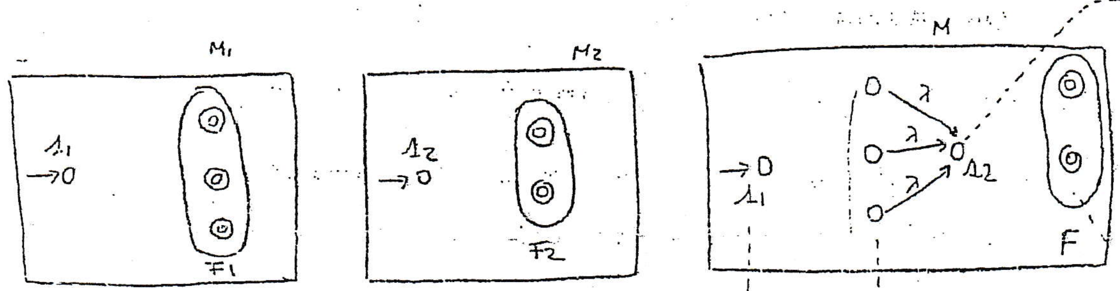
→ de ese estado inicial se mandan aristas λ a los viejos estados iniciales

→ El conjunto de estados finales es unión de los 2 conjuntos de estados finales viejos



(b) CONCATENACION

Construimos $M = (K, \Sigma, \Delta, s, F)$ tal que $L(M) = L(M_1)L(M_2)$



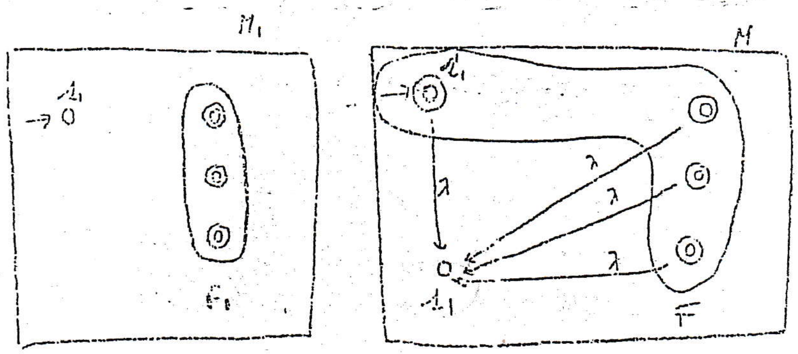
$K = K_1 \cup K_2$
$s = s_1$
$F = F_2$
$\Delta = \Delta_1 \cup \Delta_2 \cup (F_1 \times \{\lambda\} \times \{s_2\})$

"la concatenación de lenguajes aceptados por autómatas finitos es nuevamente un lenguaje aceptado por autómata finito"

1 → El estado inicial es el estado inicial del primero (M_1)
2 → El conjunto de estados finales es el conjunto de estados finales de 2do autómata (M_2)
3 → Del conjunto des estados finales del 1ro mandamos aristas λ al estado inicial del segundo (M_2) y dicho conjunto de estados finales deja de ser estados finales ahora son no terminales

(c) ESTRELLA DE KLEENE

Construimos un autómata $M = (K, \Sigma, \Delta, s, F)$ tal que $L(M) = L(M_1)^*$



① → Se agrega un nuevo estado que es inicial y final a la vez

② → El conjunto de estados finales es el antiguo F_1 mas el estado agregado (s_1')

③ → De F (conjunto de estados finales) se mandan aristas λ al antiguo estado inicial (s_1)

$K = K_1 \cup \{s_1'\}$
$s = s_1'$
$F = F_1 \cup \{s_1'\}$
$\Delta = \Delta_1 \cup \{F \times \{\lambda\} \times \{s_1\}\}$

(d) COMPLEMENTO

Sea $M = (K, \Sigma, \delta, s, F)$ un AFD que acepta $L(M)$

El autómata que acepta $\Sigma^* - L(M)$ es $\overline{M} = (K, \Sigma, \delta, s, K - F)$.
En los estados finales F los que eran ya no son estados finales y los estados que no eran estados finales se convierten en estados finales

(e) INTERSECCION

$$A \cap B = \overline{\overline{A \cap B}} = \overline{\overline{A} \cup \overline{B}}$$

$$L_1 \cap L_2 = \Sigma^* - ((\Sigma^* - L_1) \cup (\Sigma^* - L_2))$$

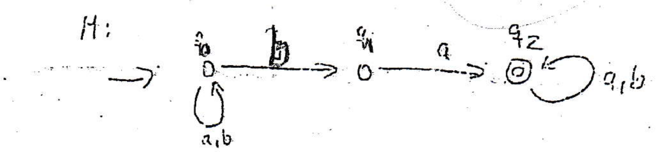
La intersección de lenguajes por autómatas finitos es nuevamente un lenguaje aceptado por autómata finito.

Ejemplo (1) $L(M) = \Sigma^*$?

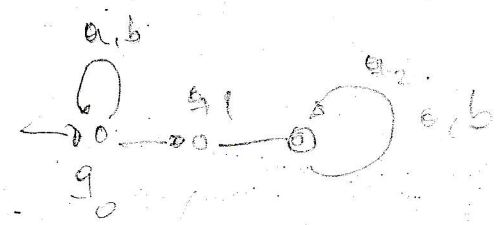
$\Sigma^* - L(M) = \emptyset$

$L(M) = \Sigma^*$ sí $\Sigma^* - L(M) = \emptyset$

M:



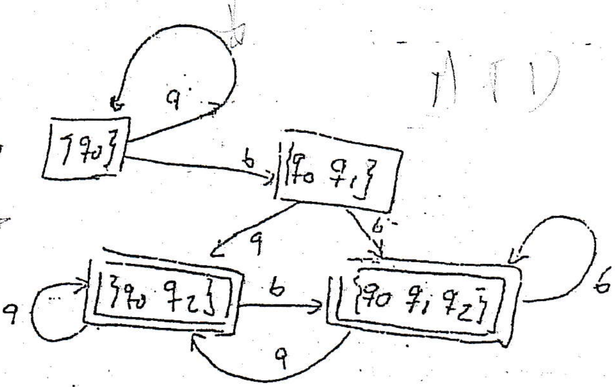
AFN



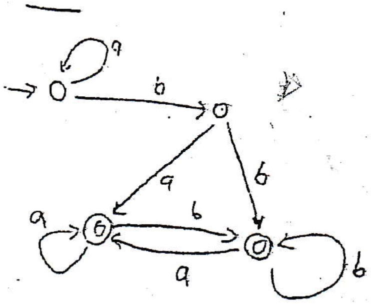
Haciendolo Deterministico

Estado	Entrada λ	Entrada a	Entrada b
$\rightarrow q_0$	q_0	q_0	$q_0\ q_1$
q_1	q_1	q_2	q_2
q_2	q_2	q_2	q_2

$\Rightarrow$

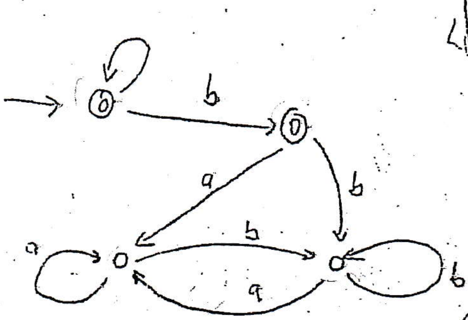


M



$\Rightarrow$

$\overline{M}$ (complemento de M)



- Como existen palabras que pertenecen tal lenguaje $L(\overline{M})$ ($\Sigma^* - L(M)$) entonces $\Sigma^* - L(M) \neq \emptyset$

por lo tanto

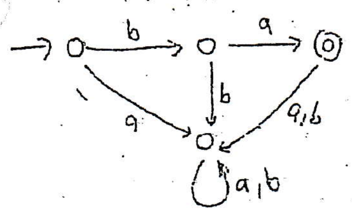
$\therefore L(M) \neq \Sigma^*$

$\Sigma^* \ldots \neq \Sigma^*$

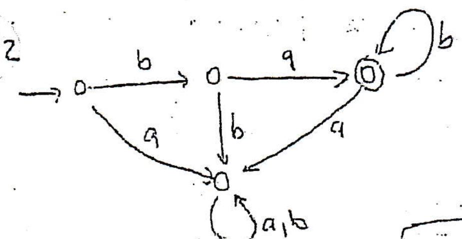
Σ M

② ¿Es $L(M_1) \subseteq L(M_2)$? Justificar la respuesta

M1.



M2



$L_1 \subseteq L_2$ tal que
$\overline{L_2} \cap L_1 = \phi$
equivale a
$\overline{\overline{L_2 \cup \overline{L_1}}} = \phi$

$L_1 = L(M_1)$
$L_2 = L(M_2)$
$L_1 \subseteq L_2$
$\overline{L_2} \cap L_1 = \phi$

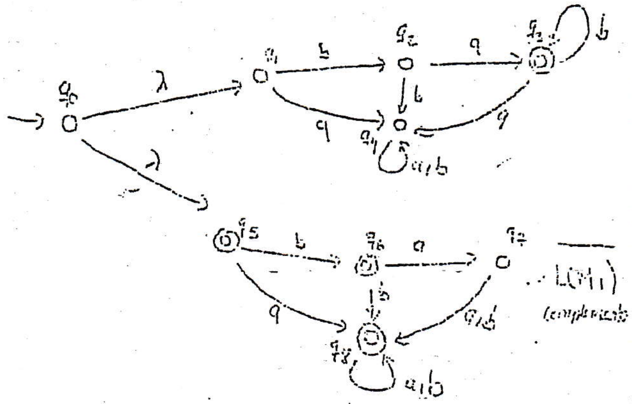
Sí $L_1 \subseteq L_2 \Rightarrow \overline{L_2} \cap L_1 = \phi$
y apl. propiedad de intersección queda
$\overline{L_2} \cap L_1 \Rightarrow \overline{\overline{\overline{L_2} \cap L_1}} \Rightarrow \overline{L_2 \cup \overline{L_1}} = \phi$ //

$\therefore \Sigma^* - (L(M_2) \cup (\Sigma^* - L(M_1))) = \phi$

$L(M_3) = \Sigma^* - (L(M_2) \cup (\Sigma^* - L(M_1)))$ → Aplicamos

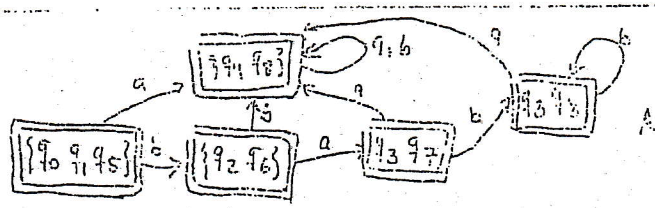
1 → $L(M_1)$ tiene que ser AFD y luego sacarle complemento

M3



Estado	Entrada λ	Entrada a	Entrada b
→ q_0	$q_0\ q_1\ q_5$	$q_4\ q_8$	$q_2\ q_6$
q_1	q_1	q_4	q_2
q_2	q_2	q_3	q_4
q_3	q_3	q_4	q_3
q_4	q_4	q_4	q_4
q_5	q_5	q_8	q_6
q_6	q_6	q_7	q_8
q_7	q_7	q_8	q_8
q_8	q_8	q_8	q_8

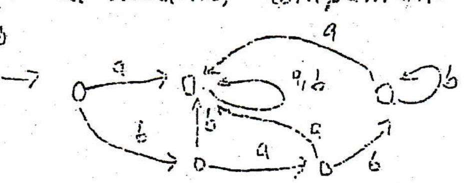
M3



AFD

Ahora sacamos complemento de M3

$\overline{M_3}$



//

$\overline{M_3} = \Sigma^* - (L(M_2) \cup (\Sigma^* - L(M_1)))$

$\overline{M_3}$ No tiene estados finales

Como no hay estados finales, no existen palabras que pertenezcan al lenguaje

$\Sigma^* - (L(M_2) \cup (\Sigma^* - L(M_1)))$

Entonces

$\Sigma^* - (L(M_2) \cup (\Sigma^* - L(M_1))) = \phi$

$\therefore L(M_1) \subseteq L(M_2)$

///

$\boxed{\text{Cuando } L(M_1) = L(M_2) \text{ ?}}$

$\Rightarrow \underbrace{L(M_1) \subseteq L(M_2)}_{\text{I}} \wedge \underbrace{L(M_2) \subseteq L(M_1)}_{\text{I}}$

y el ($\subseteq$) ya se lo explico en la anterior hoja

$\boxed{\text{* Cuando } L(M_1) = \phi \text{ ?}}$

Res

- Cuando el lenguaje $L(M_1)$ No le pertenecen palabras
- Cuando No existen Estados Finales

$\boxed{\text{☆ Cuando } L(M_1) = \Sigma^* \text{ ?}}$

Sol

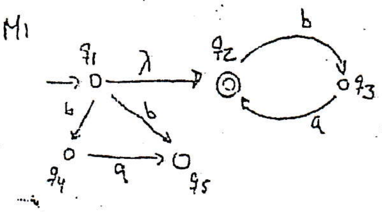
Cuando El complemento del lenguaje sea $= \phi$

$\Sigma^* - L(M_1) = \phi$ y para hacer complemento $L(M_1)$ tiene que ser un AFD

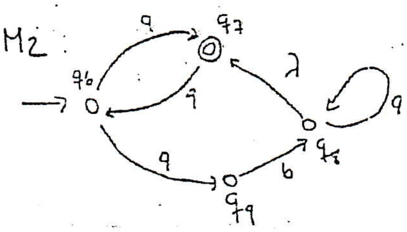
PROPIEDADES DE LOS LENGUAJES

ⓐ $\boxed{L(M) = L(M_1) \cup L(M_2)}$

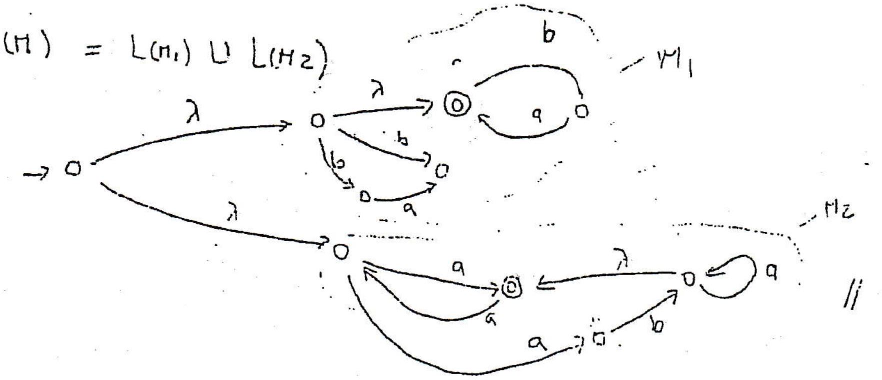
Sí M_1



Sí M_2



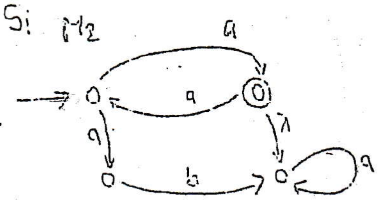
$L(M) = L(M_1) \cup L(M_2)$



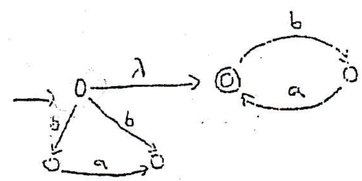
//

ⓑ $\boxed{L(M) = L(M_2)\,L(M_1)}$

Si M_2

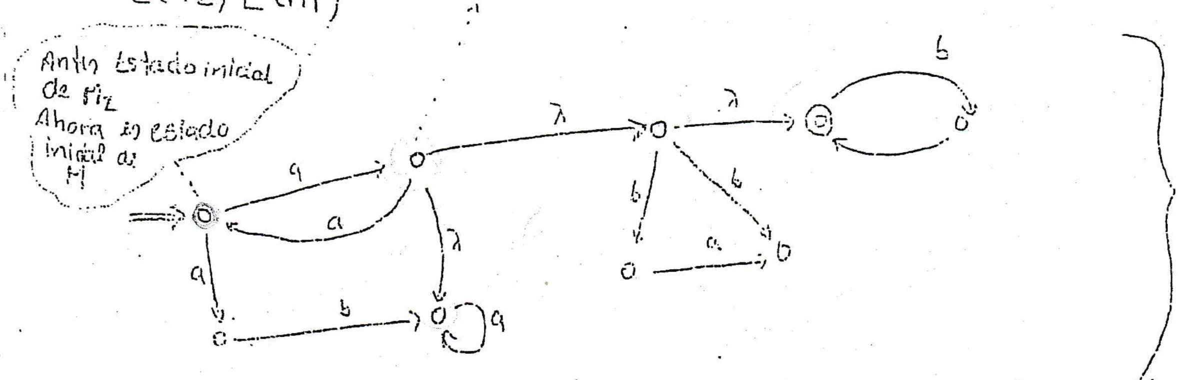


Si M_1



Antes era estado final

$L(M_2)\,L(M_1)$



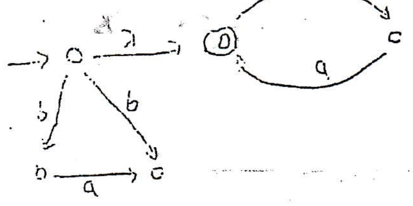
El nuevo $L(M)$

$L(M_2)\,L(M_1)$ ó $L(M_1)\,L(M_2)$

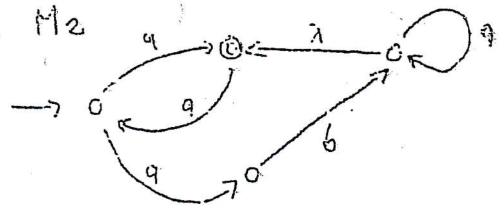
-- Se mantiene como estado inicial el de M_2 ó M_1

ⓒ $L(M) = \left(L(M_1) \cup L(M_2) \right)^*$

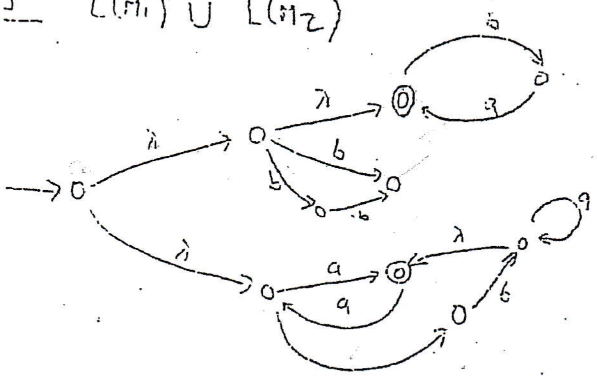
Sea M_1



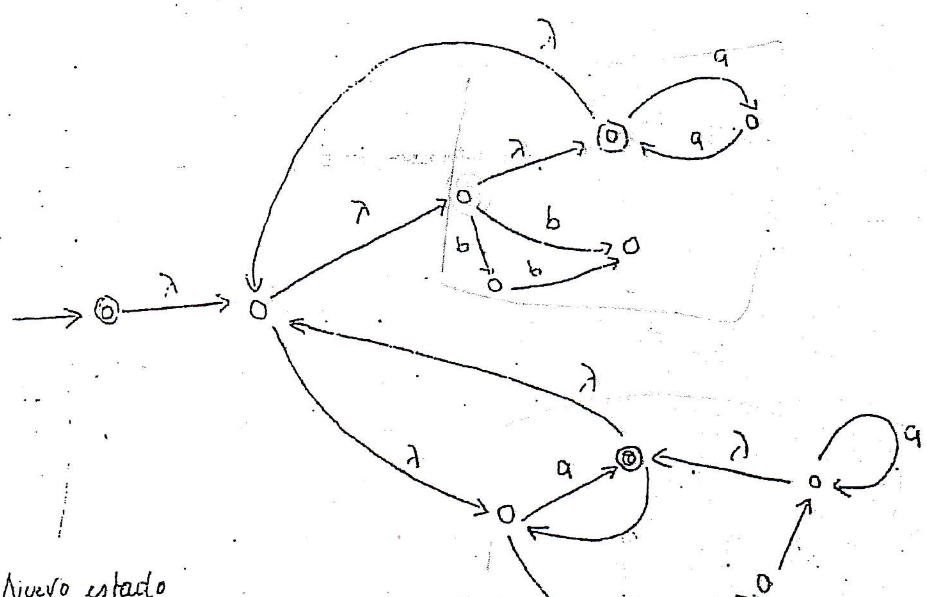
Sea M_2



1^{ro} $L(M_1) \cup L(M_2)$



2^{do} $\left(L(M_1) \cup L(M_2) \right)^*$



Nuevo estado
Inicial y final a la vez

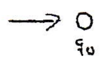
AUTÓMATAS FINITOS Y EXPRESIONES REGULARES

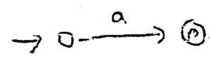
Teorema Un lenguaje es Regular sí y solo si es aceptado por un autómata finito

$\Longrightarrow$ Sí un lenguaje es regular entonces es aceptado por un A.F.

contiene al ϕ y a los conjuntos unitarios $\{a\}$ $\forall a \in \Sigma$

$L(\phi) = \phi$ $\qquad\qquad\qquad$ $L(a) = \{a\}$





α y β son ER
$(\alpha \cup \beta)$
$(\alpha\beta)$
α^*

Por Inducción sobre ER γ

H.I $\longrightarrow$ la propiedad se cumple para γ tal que $|\gamma| < n$

P.d para γ tal que $|\gamma| = n$

- Si $\gamma = \alpha \cup \beta$

$$\begin{aligned} \underline{L(\gamma)} = L(\alpha \cup \beta) &= L(\alpha) \cup L(\beta) \\ &= \underbrace{L(M_1) \cup L(M_2)} \\ &= \underline{L(M)} \end{aligned}$$

- Si $\gamma = \alpha\beta$

$$\begin{aligned} \underline{L(\gamma)} = L(\alpha\beta) &= L(\alpha)\, L(\beta) \\ &= \underbrace{L(M_1)\, L(M_2)} \\ &= \underline{L(M)} \end{aligned}$$

- Si $\gamma = \alpha^*$

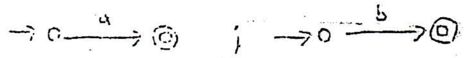
$$\begin{aligned} \underline{L(\gamma)} = L(\alpha^*) &= L(\alpha)^* \\ &= L(M_1)^* \\ &= \underline{L(M)} \end{aligned}$$

Ejemplo Sea $\alpha = (ab \cup aab)^*$

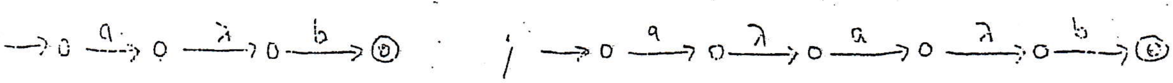
Construir un AF M tal que $L(\alpha) = L(M)$.

Solución

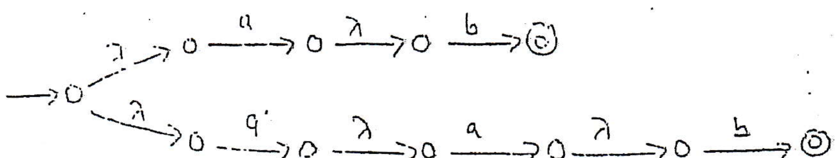
Paso 1 → a ; b



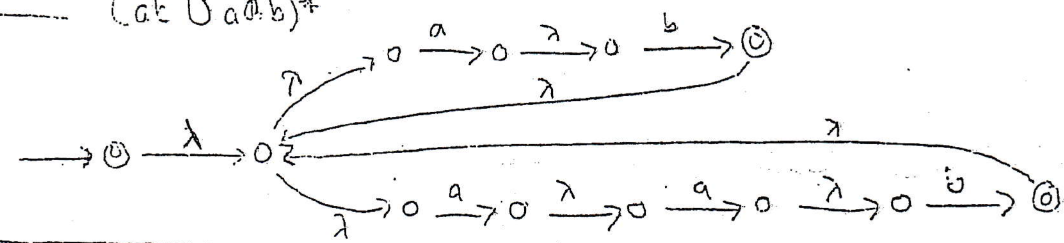
Paso 2 ab ; aab



Paso 3 $ab \cup aab$

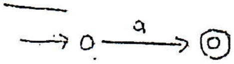


Paso 4 $(ab \cup aab)^*$

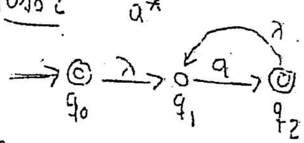


Sea $\alpha = a^*$ ER sobre $\Sigma = \{a, b\}$ Construir un AFD, M tal que $L(\alpha) = L(M)$

Paso 1 a



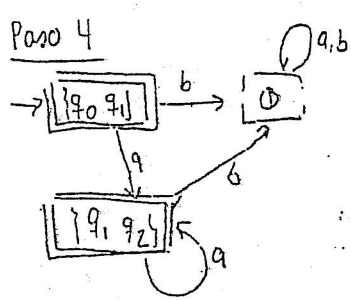
Paso 2 a^*

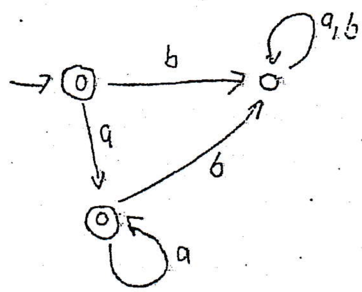


Paso 3 AFD

Estado	λ	Entrada a	Entrada b
$\to q_0$	$q_0 q_1$	q_1, q_2	—
q_1	q_1	q_1, q_2	—
q_2	$q_2 q_1$	q_1, q_2	—

Paso 4

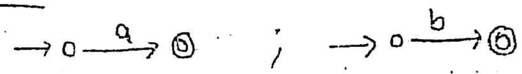




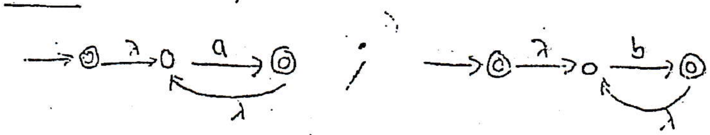
AFD

Sea $\alpha = a^*b^*$ Construir un AFD M tal que $L(\alpha) = L(M)$

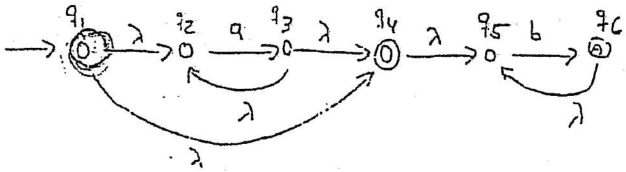
Paso 1 a ; b



Paso 2 a^* ; b^*



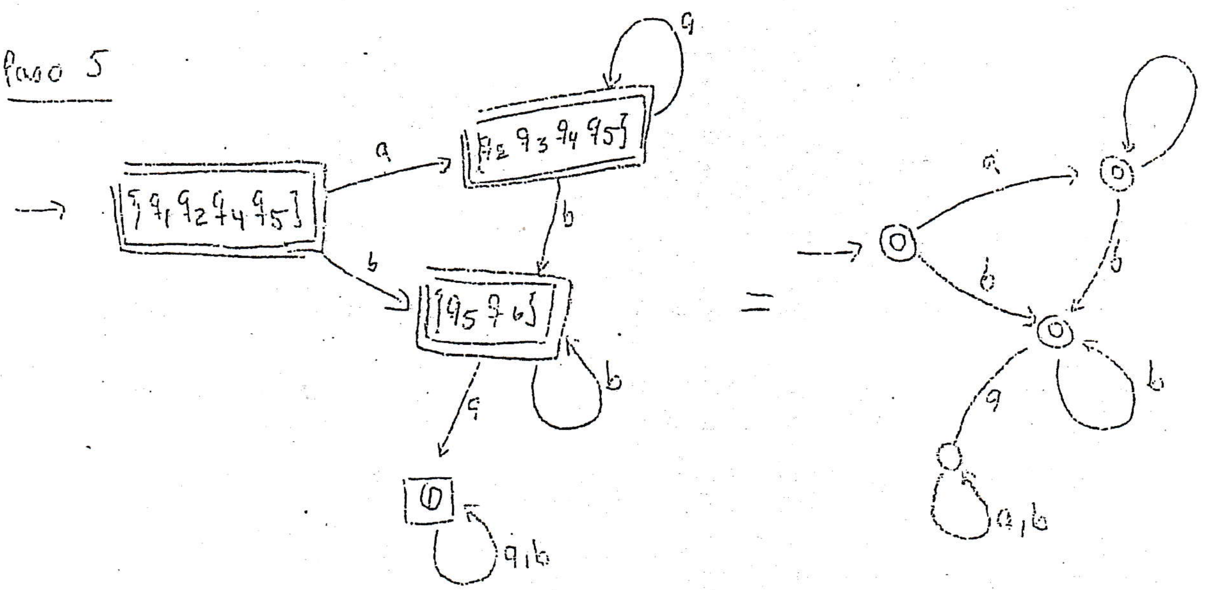
Paso 3 a^*b^*



Paso 4. AFD convertir

	Entrada λ	Entrada a	Entrada b
q_1	$q_1 q_2 q_4 q_5$	$q_3 q_2 q_4 q_5$	$q_6 q_5$
q_2	q_2	$q_3 q_2 q_4 q_5$	—
q_3	$q_3 q_2 q_4$	$q_3 q_2 q_4 q_5$	$q_6 q_5$
q_4	$q_4 q_5$	—	$q_6 q_5$
q_5	q_5	—	$q_6 q_5$
q_6	$q_6 q_5$	—	$q_6 q_5$

Paso 5



AFD

$\Longrightarrow$) Sí un lenguaje es aceptado por AF entonces es un lenguaje regular.

Prueba: cualquier lenguaje finito es un lenguaje regular

Sea $M = (K, \Sigma, \delta, s, F)$ un AFD que acepta $L(M)$.

p.d. existe un lenguaje Regular R tal que $R = L(M)$

La idea es expresar $L(M)$ como la unión de muchos (pero el número finito) lenguajes simples

Sea $K = \{q_1, q_2, \ldots, q_n\}$ donde $s = q_i$

para $i, j = 1, 2, \ldots, n$; $k = 1, 2, \ldots, n+1$

Definimos los conjuntos $R(i,j,k)$ de todas las palabras que llevan de q_i a q_j sin pasar por estados con subíndice k o mayores

Formalmente:

$$R(i,j,k) = \{x \in \Sigma^* : (q_i, x) \vdash_M^* (q_j, \lambda) \text{ y si } (q_j, x) \vdash_M^* (q_l, y)$$
$$\text{para algún } y \in \Sigma^* \; l < k \text{ ó } (l = j \wedge y = \lambda) \text{ ó } (l = i \wedge y = x)\}$$

$$R(i, j, n+1) = \{x \in \Sigma^* : (q_i, x) \vdash_M^* (q_j, \lambda)\}$$

$$L(M) = \bigcup \{R(1, j, n+1) ; q_j \in F\}$$

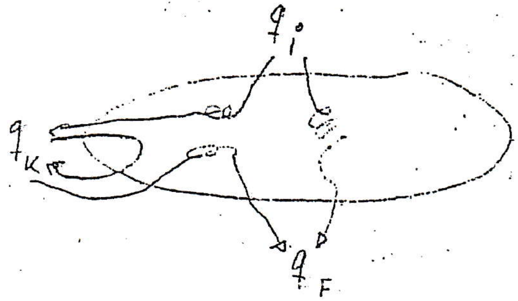
p.d. $R(i,j,k)$ son regulares

por inducción sobre k

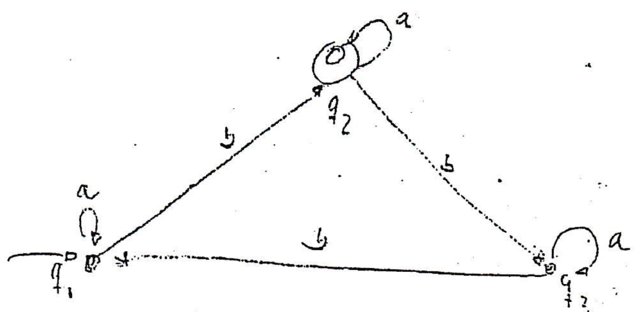
para $k = 1$

$$R(i,j,1) = \begin{cases} \{\sigma \in \Sigma : \delta(q_i, \sigma) = q_j\} & \text{si } i \neq j \\ \{\lambda\} \cup \{\sigma \in \Sigma : \delta(q_i, \sigma) = q_j\} & \text{si } i = j \end{cases}$$

Son regulares por ser FINITOS



$R(i,j,k+1) = (i,j,k) \cup (i,k,k)(k,k,k)$
(k,j,k)



$a^* b \ldots a^* b$.

$$L(M) = R(1,2,4)$$

$i \longrightarrow K \longrightarrow$

Por la ecuación.

$$R(1,2,4) = R(1,2,3) \cup R(1,3,3)(3,3,3)^*(3,2,3)$$

$$R(1,2,3) = R(1,2,2) \cup R(1,2,2)\, R(2,2,2)^*\, R(2,2,2)$$
$$R(1,3,3) = R(1,3,2) \cup R(1,2,2) R(2,2,2)^* R(2,3,2)$$
$$R(3,3,3) = R(1,3,2) \cup R(3,2,2)(2,2,2)^* R(2,3,2)$$
$$R(3,2,3) = R(3,2,2) \cup R(3,2,2)\, R(2,2,2)^*\, R(2,2,2)$$

$$R(1,2,2) = R(1,2,1) \cup R(1,1,1) R(1,1,1)^* R(1,2,1)$$
$$R(2,2,2) = R(2,2,1) \cup R(2,1,1) R(1,1,1)^* R(1,2,1)$$
$$R(1,3,2) = R(1,3,1) \cup R(1,1,1) R(1,1,1)^* R(1,3,1)$$
$$R(2,3,2) = R(2,3,1) \cup R(2,1,1) R(1,1,1)^* R(1,3,1)$$
$$R(3,2,2) = R(3,2,1) \cup R(3,1,1) R(1,1,1)^* R(1,2,1)$$

$$R(1,2,1) = R(1,2,0) \cup R(1,1,1)\, R(1,1,1)^*\, R(1,2,1)$$
$$R(1,1,1) = R(1,1,0) \cup R(1,0,0)\, R(0,0,0)^*\, R$$

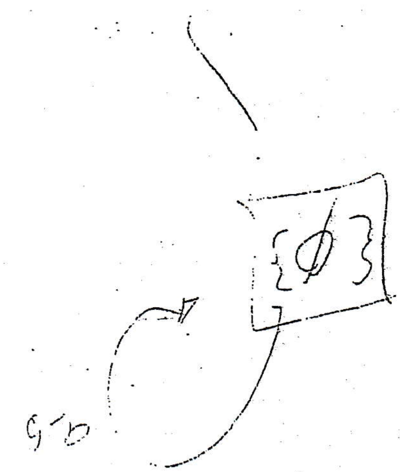
Lenguajes No Regulares

$(i,j,k) \cup (i,k,k)($

$= \{aab, aba,$

$(a \cup b \cup c)$

$a \cup b \cup a^*$



GRAMATICAS

Ejemplo $\rightarrow$ $a(a^* \cup b^*)b$

$S \rightarrow aMb$
$M \rightarrow A$
$M \rightarrow B$
$A \rightarrow \lambda$
$A \rightarrow aA$
$B \rightarrow \lambda$
$B \rightarrow bB$

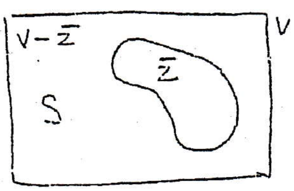
Definición

Una gramática libre de contexto (GLC) es una cuádrupla $G = (V, \Sigma, R, S)$

$V \rightarrow$ Es un alfabeto

$\Sigma \rightarrow$ Es un subconjunto de V (Simbolos terminales) (minusculas)

$R \subseteq (V - \Sigma) * V^*$ (conjunto de reglas)
(R es un subconjunto finito) $(A, v) \in R$
$A \in V - \Sigma$ $v \in V^*$



$S \rightarrow V - \Sigma$ Simbolo inicial

- A los elementos de $V - \Sigma$ se les llama simbolos No terminales
- Un elemento de R tiene la forma:

$(A, v) \in R$ $A \in V - \Sigma$, $v \in V^*$

Ejemplo

$V = \{S, M, A, B, a, b\}$
$\Sigma = \{a, b\}$
$S \in V - \Sigma$

$R = \{(S, aMb), (M, A), (M, B), (A, \lambda), (A, aA), (B, \lambda), (B, bB)\}$

Notación si $(A, v) \in R$; $A \in V - \Sigma$, $v \in V^*$
$(A, v) \in R \iff A \underset{G}{\rightarrow} v$

Derivaciones para todo par de palabras $u, v \in V^*$ escribimos
$u \underset{G}{\Rightarrow} v$ (se lee v deriva directamente de u)

Si y solo si existen palabras $x, y, v' \in V^*$ y $A \in V - \Sigma$ tales que:
$u = xAy$ $v = xv'y$ y $A \underset{G}{\rightarrow} v'$ o $(A, v') \in R$

Denotamos por $\overset{*}{\underset{G}{\Rightarrow}}$ a la clausura reflexiva transitiva de $\underset{G}{\Rightarrow}$

Sea $G = (V, \Sigma, R, S)$ un GLC y sea $w \in \Sigma^*$ se dice que G genera w si y solo si
$S \overset{*}{\underset{G}{\Rightarrow}} w$

Lenguaje Generado por G

$L(G) = \{w \in \Sigma^* / S \overset{*}{\underset{G}{\Rightarrow}} w\}$

L es un lenguaje libre de contexto si existe una GLC G tal que $L(G) = L$

- $S \Rightarrow aMb \Rightarrow aAb \Rightarrow a\lambda b = ab$

  $\therefore$ $S \overset{*}{\Rightarrow} ab$ luego $ab \in L(G)$

- $S \Rightarrow aMb \Rightarrow aAb \Rightarrow aaAb \Rightarrow aaaAb \Rightarrow aaa\lambda b = aaab$

  $\therefore$ $S \overset{*}{\Rightarrow} aaab$ luego $aaab \in L(G)$

$L(G) = \{ w \in \Sigma^* / w = aa^m b \vee w = ab^n b \; ; \; m, n \geq 0 \}$

Ejemplo (1) Sea $G = (V, \Sigma, R, S)$

$V = \{S, a, b\}$, $\Sigma = \{a, b\}$ $R = \{S \rightarrow aSb \; , \; S \rightarrow \lambda\}$

Escribir $L(G)$ por comprensión

Sol

- $S \rightarrow \lambda$ $\therefore$ $\lambda \in L(G)$
- $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aa\lambda bb$ $\therefore$ $S \overset{*}{\underset{G}{\Rightarrow}} aabb$ $aabb \in L(G)$

$L(G) = \{ w \in \Sigma^* / w = a^n b^n \; n \geq 0 \}$

$L(G) = \{ a^n b^n \; ; \; n \geq 0 \}$ luego libre de contexto

Ejemplo 2 Sea $G = (W, \Sigma, R, S)$

$W = \{S, A, N, V, F\} \cup \Sigma$, $\Sigma = \{$ jaime, verde, grande, queso, comió

$R = \{ F \rightarrow N \; , \; F \rightarrow FA \; , \; S \rightarrow FVF \; , \; A \rightarrow$ grande , $A \rightarrow$ verde , $N \rightarrow$ queso ,

$N \rightarrow$ jaime , $V \rightarrow$ comió $\}$

Generar una palabra "sin sentido y otra con sentido"

- $S \Rightarrow FVF \Rightarrow FVFA \Rightarrow NVFA \Rightarrow$ queso $VFA \Rightarrow$ queso comió FA

  $\Rightarrow$ queso comió $NA \Rightarrow$ queso comió queso $A \Rightarrow$ queso comió queso verde

GRAMATICAS REGULARES

Def. Una gramática libre de contexto es una Gramatica Regular sí:

terminales

$$R \subseteq (V-\Sigma) \times \Sigma^* ((V-\Sigma) \cup \{\lambda\})$$

// (May , minuscula) ~ Regular.
// (May , minuscula M) regular

$A = \{a, ab, ba\}$ $B = \{M, N\}$

$AB = \{aM, abM, baM, aN, abN, ba$

Ejemplo $G = (V, \Sigma, R, S)$

$V = \{S, A, B, a, b\}$

$\Sigma = \{a, b\}$

$R = \{S \to bA,\ S \to aB,\ A \to abaS,\ B \to babS,\ S \to \lambda\}$

$L(G) = ?$

$S \Rightarrow bA \Rightarrow babaS \Rightarrow baba\lambda = baba$

$S \Rightarrow aB \Rightarrow ababS \Rightarrow abab\lambda = abab$

$S \Rightarrow \lambda = \lambda$

$S \Rightarrow aB \Rightarrow ababS \Rightarrow ababaB \Rightarrow abababab S \Rightarrow abababab\lambda = abababab$

$L(G) = (bab \cup abab)^*$

Teorema Un lenguaje es regular sí y solo sí es generado por una gramática Regular

($\Longrightarrow$) si un lenguaje es regular entonces es generado por una G.R.

Sea L un lenguaje regular entonces existe un AFD $M = (K, \Sigma, \delta, s, F)$

tal que $L(M) = L$

estados

Construimos $G = (V, \Sigma, R, S)$ tal que $L(G) = L(M)$ donde

$V = \Sigma \cup K$

$S = s$

$R = \{q \to ap : \delta(q, a) = p\} \cup \{q \to \lambda : q \in F\}$

($\Longleftarrow$) Si un lenguaje es generado por una G.R. entonces es lenguaje Regular

Sea $G = (V, \Sigma, R, S)$ la gramática que genera $L(G)$

Construimos $M = (K, \Sigma, \Delta, s, F)$ tal que $L(M) = L(G)$

$K = (V - \Sigma) \cup \{f\}$ donde f es un nuevo elemento

$s = S$

$F = \{f\}$

$\Delta = \{(A, w, B) : A \to wB,\ A, B \in (V-\Sigma),\ w \in \Sigma^*\} \cup$

$\{(A, w, f) : A \to w\ \ A \in (V-\Sigma)\ w \in \Sigma^*\}$

Ejercicios de Gramaticas Regulares

(1) Ejemplo Dado un Automata, hallar la Gramática

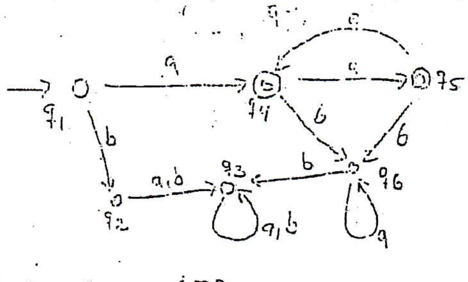
- El automata tiene que ser AFD

Sea $M = (K, \Sigma, S, s, F)$ construimos $G = (V, \Sigma, R, S)$ tal que

$V = \Sigma \cup K$

$S = s$

$R = \{ q \rightarrow ap : \delta(q, a) = p \} \cup \{ q \rightarrow \lambda : q \in F \}$



AFD

$G = (V, \Sigma, R, S)$

$V = \{q_1, q_2, q_3, q_4, q_5, q_6, a, b\}$

$\Sigma = \{a, b\}$

$S = q_1$

$R = \{ q_1 \rightarrow a q_4, q_1 \rightarrow b q_2, q_2 \rightarrow a q_3, q_2 \rightarrow b q_3,$
$q_3 \rightarrow a q_3, q_3 \rightarrow b q_3, q_4 \rightarrow a q_5, q_4 \rightarrow b q_6,$
$q_5 \rightarrow a q_4, q_5 \rightarrow b q_6, q_6 \rightarrow a q_6, q_6 \rightarrow b q_3,$
$q_4 \rightarrow \lambda, q_5 \rightarrow \lambda \}$

Ejemplo → Dado un Gramatica, encontrar un Automata

Sea $G = (V, \Sigma, R, S)$ construimos $M = (K, \Sigma, \Delta, s, F)$ tal que

$K = \underbrace{(V - \Sigma)}_{} \cup \{f\}$ → Solo estados (Mayusculos)

AFN

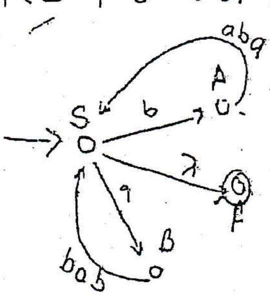
$s = S$

$F = \{f\}$

$\Delta = \{ (A, w, B) : A \rightarrow wB : A, B \in (V - \Sigma), w \in \Sigma^* \} \cup \{ (A, w, f) : A \rightarrow w, f \in (V - \Sigma), w \in \Sigma^* \}$

$R = \{ S \rightarrow bA, S \rightarrow aB, A \rightarrow abaS, B \rightarrow babS, S \rightarrow \lambda \}$

$B \rightarrow \lambda \}$



$K = \{S, A, B\} \cup \{f\}$

$s = S$

$F = \{f\}$

$\Delta = \{ (S, b, A), (S, a, B), (A, aba, S), (B, bab, S),$
$(S, \lambda, f) \}$

//

Ejemplo   Considere una Gramática Regular $G = (V, \Sigma, R, S)$

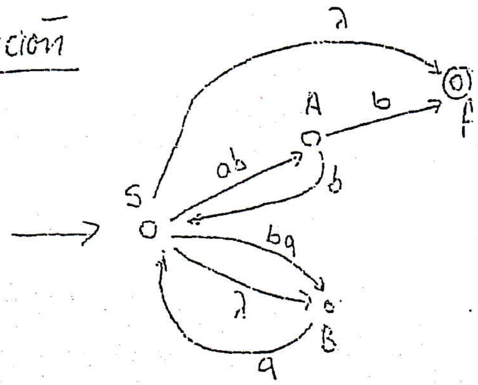
donde $V = \{a, b, A, B, S\}$

$\Sigma = \{a, b\}$

$R = \{ S \rightarrow abA, S \rightarrow B, S \rightarrow baB, S \rightarrow \lambda, A \rightarrow bS,$
$B \rightarrow aS, A \rightarrow b \}$

Construir un Autómata NO determinístico tal que $L(G) = L(M)$.

Solución



$K = \{S, A, B\} \cup \{f\}$

$\Sigma = \{a, b\}$

$s = S$

$F = \{f\}$

$\Delta = \{ (S, ab, A), (S, ba, B), (S, \lambda, B),$
$(A, b, S), (A, b, f), (B, a, S),$
$(S, \lambda, f) \}$

Ejemplo Considere una Gramática Regular $G = (V, \Sigma, R, S)$

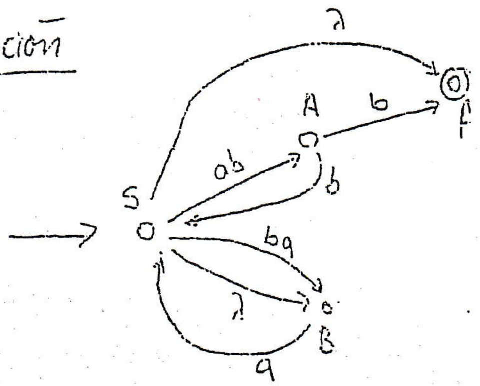
donde $V = \{a, b, A, B, S\}$

$\Sigma = \{a, b\}$

$R = \{ S \to abA,\ S \to B,\ S \to baB,\ S \to \lambda,\ A \to bS,$
$B \to aS,\ A \to b \}$

Construir un Autómata NO determinístico tal que $L(G) = L(M)$.

Solución



$K = \{S, A, B\} \cup \{f\}$

$\Sigma = \{a, b\}$

$s = S$

$F = \{f\}$

$\Delta = \{ (S, ab, A), (S, ba, B), (S, \lambda, B),$
$(A, b, S), (A, b, f), (B, a, S),$
$(S, \lambda, f) \}$

AUTÓMATAS CON PILA

Con los lenguajes libres de contexto

Definición

Un autómata con Pila es una sextupla
$M = (K, \Sigma, \Gamma, \Delta, s, F)$ donde:

K = Conjunto de Estados
Σ = Conjunto No vacío llamado alfabeto
Γ = Conjunto finito No vacío llamado alfabeto de la Pila
$s \in K$ Estado inicial
$F \subseteq K$
Δ = es un subconjunto finito de
$(K \times \Sigma^* \times \Gamma^*) \times (K \times \Gamma^*)$

$((p, u, \beta), (q, \gamma)) \in \Delta$

Estando en "p" con u en la cinta y β en el tope de la pila, pasa a q y reemplaza γ con β

Empujar → Un elemento a la pila significa añadirlo a la cabeza de la pila

$$((p, u, \lambda), (q, a)) \in \Delta$$

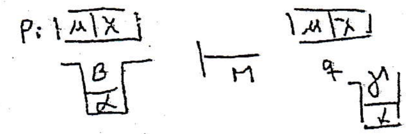
Sacar → Un elemento de la pila significa remover de la cabeza dicho elemento

$$((p, u, a), (q, \lambda)) \in \Delta$$

Configuración

Una configuración es un elemento de $K \times \Sigma^* \times \Gamma^*$
La relación $\vdash_M$ para cada transición $((p, u, \beta), (q, \gamma)) \in \Delta$ y para cada $x \in \Sigma^*$ y $\alpha \in \Gamma^*$ definimos

$$(p, ux, \beta\alpha) \vdash_M (q, x, \gamma\alpha)$$



Denotamos por $\vdash^{*}_{M}$ a la clausura reflexiva transitiva de $\vdash_M$

Si $w \in \Sigma^*$ decimos que M acepta a w si y solo si

$$(s, w, \lambda) \vdash^{*}_{M} (q, \lambda, \lambda) ; \; q \in F$$

LENGUAJE ACEPTADO POR M

$$L(M) = \{ w \in \Sigma^* / (s, w, \lambda) \vdash^{*}_{M} (q, \lambda, \lambda) : q \in F \}$$

(Eje) Un AF es un autómata con pila?

Sea $M = (K, \Sigma, \Delta, s, F)$ un AFN

el autómata con pila $M' = (K, \Sigma, \Gamma, \Delta', s, F)$ donde $\Gamma = \phi$

$$\Delta = \{ ((p, u, \lambda), (q, \lambda)) / (p, u, q) \in \Delta \}$$

(Eje) $M = (K, \Sigma, \Gamma, \Delta, s, F)$ donde $K = \{s, f\}$; $\Sigma = \{a, b, c\}$

$\Gamma = \{a, b\}$ $\quad F = \{f\}$

$$\Delta = \{ ((s, a, \lambda), (s, a)) \quad 1$$
$$((s, b, \lambda), (s, b)) \quad 2$$
$$((s, c, \lambda), (f, \lambda)) \quad 3$$
$$((f, a, a), (f, \lambda)) \quad 4$$
$$((f, b, b), (f, \lambda)) \} \quad 5$$

Procesar abbcbba

Estado	Entrada No leida	Pila	transicion
s	abbcbba	λ	—
s	bbcbba	a	1
s	bcbba	ba	2
s	cbba	bba	2
s	bba	bba	3
f	ba	ba	5
f	a	a	5
f	λ	λ	4

AUTÓMATAS CON PILA Y GRAMÁTICAS LIBRE DE CONTEXTO

La clase de lenguajes aceptados por autómatas con pila es exactamente la clase de lenguajes libres de contexto.

Teorema ($\Longleftarrow$) Sea $G = (V, \Sigma, R, S)$ una GLC construimos un autómata con pila M tal que $L(M) = L(G)$

$M = (\{p, q\}, \Sigma, V, \Delta, p, \{q\})$

Δ contiene

1. $((p, \lambda, \lambda), (q, S))$
2. $((q, \lambda, A), (q, x))$ $\forall A \to x \in R$

   $((q, a, a), (q, \lambda))$ $\forall a \in \Sigma$

Ejemplo $G = (V, \Sigma, R, S)$

$V = \{S, a, b, c\}$; $\Sigma = \{a, b, c\}$

$R = \{S \to aSa, S \to bSb, S \to c\}$ $\Rightarrow$ 3 Reglas + Alfabeto + 1 obligatoria

$=$ 7 transiciones

$M = (\{p, q\}, \{a, b, c\}, \{S, a, b, c\}, \Delta, p, \{q\})$

transición $((K \times \Sigma^* \times \Gamma^*) \times (K \times \Gamma^*))$ $*$ construir M y procesar $abbcbba$

$\Delta = \{ ((p, \lambda, \lambda), (q, S))$ 1 ....> obligatorio

$((q, \lambda, S), (q, aSa))$ 2
$((q, \lambda, S), (q, bSb))$ 3
$((q, \lambda, S), (q, c))$ 4
} los 3 reglas

$((q, a, a), (q, \lambda))$
$((q, b, b), (q, \lambda))$
$((q, c, c), (q, \lambda))$
} alfabeto Σ

Estando en el estado p

Sin importar lo que hay en la pila ②

$(p, \lambda, \lambda), (q, S)$

Pila

Cambia al estado q ③

cambio λ por S ④

con el ejemplo anterior procesar $|abbc\,bba|$

Estado	Entrada No leida	Pila	transición Usada
p	abbcbba	λ	—
q	abbcbba	S	1
q	abbcbba	aSa	2
q	bbcbba	Sa	5
q	bbcbba	bSba	3
q	bcbba	Sba	6
q	bcbba	bSbba	3
q	cbba	Sbba	6
q	cbba	cbba	4
q	bba	bba	7
q	ba	ba	6
q	a	a	6
q	λ	λ	5

//

estado inicial

quita de la entrada

Esta al que pasara

mete a la pila

~~e~~ saca

(, ,), (,)

lo que hay en la cabeza de la pila

PROPIEDADES

Los lenguajes L.C son cerrados bajo:

- UNION
- Concatenacion
- Estrella de Kleene

Prueba Sean $G_1 = (V_1, \Sigma_1, R_1, S_1)$ y $G_2 = (V_2, \Sigma_2, R_2, S_2)$
GLC tales que generan $L(G_1)$ y $L(G_2)$ respectivamente
Sin perdida de generalidad supongamos que $\underbrace{(V_1 - \Sigma_1)}_{\text{Mayúsculas Simbolos No terminales}} \cap (V_2 - \Sigma_2) = \emptyset$

Si tengo 2 lenguajes con A, B, C en común rebautizo las de G_2

Mayúsculas
Simbolos No terminales

1 → $\boxed{\text{Construimos } L(G) = L(G_1) \cup L(G_2)}$

$$G = (\underbrace{V_1 \cup V_2 \cup \{S\}}_{V}, \underbrace{\Sigma_1 \cup \Sigma_2}_{\Sigma}, \underbrace{R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}}_{R}, \underbrace{S}_{S})$$

Donde S es un nuevo símbolo

2 → $\boxed{\text{Construimos } G \text{ tal que } L(G) = L(G_1)L(G_2)}$

$$G = (\underbrace{V_1 \cup V_2 \cup \{S\}}_{V}, \underbrace{\Sigma_1 \cup \Sigma_2}_{\Sigma}, \underbrace{R_1 \cup R_2 \cup \{S \to S_1 S_2\}}_{R}, \underbrace{S}_{S})$$

donde S es un nuevo simbolo

3 → Construimos G tal que $L(G) = L(G_1)^*$

$$G = (V_1, \Sigma_1, R_1 \cup \{S_1 \to \lambda, S_1 \to S_1 S_1\}, S_1)$$

Teorema La intersección de un LLC con un lenguaje regular es un lenguaje libre de contexto.

Arnold Rosales - Toledo.
1er PARCIAL..

① Demostrar: $(xy)' = x'y'$ ⟶

② Hallar la ER de $L = \{ b^n a b^m a b^p \;\; n, m, p \geq 0 \}$

$b^* a b^* a b^*$.

⑤ Construir un modelo

④ Demostrar: $f^7(s, uv) = f^7[f^7(s,u), v]$ = ?

5t4bh8bt

$(P \cup Q)' = P' \cup Q'$

divisores(N, S) :- divisores(N, S, 1).

divisores(N, S, 1) :- S is 1.
divisores(N, S, T) :- divisor(I, N), ~~[illegible]~~ Diego
divisores(N, S1, T+1) 77082613
S is S1.

3541079

② dado $M = (K, \Sigma, \Delta, s,$

① $L_1 = \{ w \in \Sigma^* ; |w|_a = 2 \}$ y

$L_2 = \{ w \in \Sigma^* : |w|_a \equiv 0 \; (\text{mod } 2) \}$

Demostrar: $L_1^* \cup \{b\}^* = L_2$

③ Construir los lenguajes $|L_1| = 1$ y $|L_2| = 2$
Demostrar $L_1 \subseteq L_2$

0,0,102
3