Autómata finito determinista (AFD)

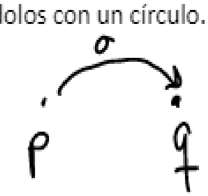
viernes, 13 de noviembre de 2020   03:01 p. m.

Formalmente, un autómata finito determinista M es una colección de cinco elementos:

- Un alfabeto de entrada Σ.
- Una colección finita de estados Q.
- Un estado inicial s.
- Una colección de estados finales o de aceptación F.
- Una función de transición δ: QxΣ → Q que determina el único estado siguiente para el par (qi, σ) correspondiente al estado actual y la entrada.

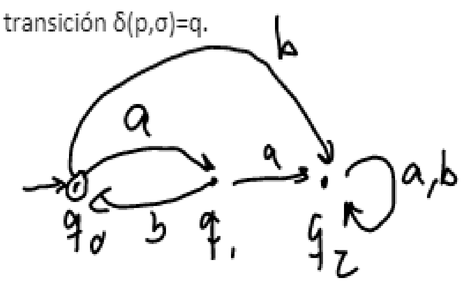
M = (Q, Σ, s, F, δ)

Un AFD suele representarse gráficamente con un diagrama de transiciones. Usamos puntos para representar los estados
de Q, el estado inicial s lo señalamos con una flecha. Los estados finales (elementos de F)
los señalamos rodeándolos con un círculo. Las transiciones se señalan con flechas de
la siguiente manera:



Esta flecha representa la transición δ(p,σ)=q.

Ejemplo:



Q={ q0, q1, q2 }
Σ={ a, b}
s= q0
F={q0}

δ(q0,a)=q1
δ(q0,b)=q2
δ(q1,a)=q2
δ(q1,b)=q0
δ(q2,a)=q2
δ(q2,b)=q2

Σ*={ ε, a, b, aa,bb, ab,ba , aba, abb, bbb, abab , ... }
L(M)=Aceptadas={ ε, ab, abab, ababab, ...}
Σ*-L(M)=No aceptadas={ a, b, ba, aa, bb, ba, aba, abb, bbb, ...}
L(M)=L((ab)*)

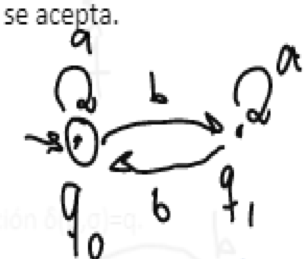
δ	a	b
q0	q1	q2
q1	q2	q0
q2	q2	q2

w= ba

δ(q0,b)=q2
δ(q2,a)=q2
En este caso se llegó a q2, que no es un elemento de F, entonces la cadena no se acepta.

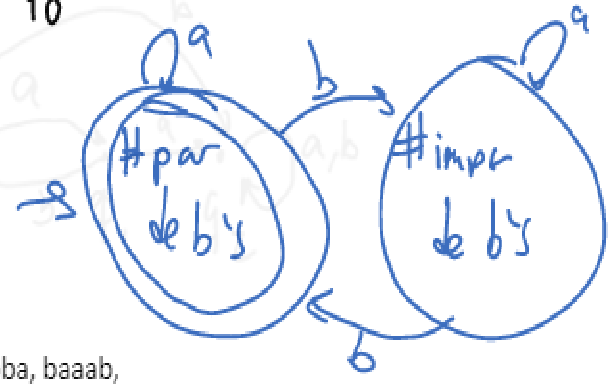
w= ab

Iniciamos en s (en este caso es q0). El primer símbolo de la cadena es a; usando la transición δ(q0,a)=q1, transitamos a q1, consumiendo este símbolo. El siguiente símbolo es b. Usando la transición δ(q1,b)=q0, transitamos a q0 consumiendo este símbolo.
Ya no hay más símbolos. Revisamos si el último estado al que se llegó es un elemento de F.
En caso de que así sea, la cadena se acepta.



Otro ejemplo:

Q={ q0, q1 }
Σ={ a,b }
F={q0}
s= q0

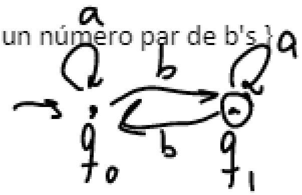


δ	a	b
q0	q0	q1
q1	q1	q0

L(M)={ ε, a, aa, bb, aaaa, aaaaaaa, abba, baaab,
Σ*-L(M)={ b, ab, ba, bbb,
babab

L(M)={ cadenas que tengan un número par de b's }

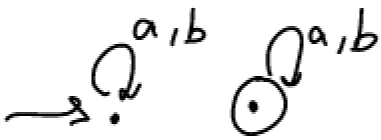
Otro ejemplo:



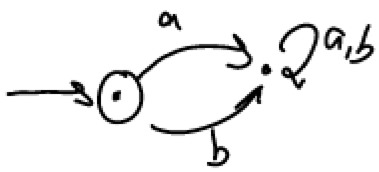
L(M)={ b, ba, ab, … }

Σ^*-L(M)={ ε, aaaaaa, bb, aaabaaaaab, … }

L(M) = Σ^*

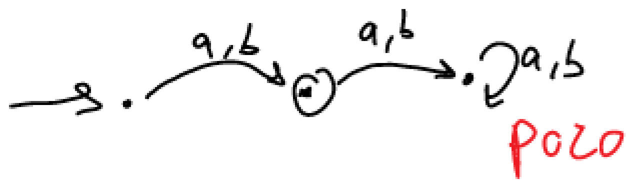


L(M)= {} = L(Ø)

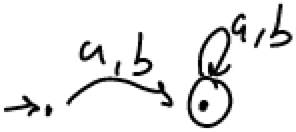


L(M)= { ε }

=L (ε)



L(M)=Σ



L(M)=Σ^+

$L(M)=\Sigma^+$

Q={q0,q1,q2}
Σ={a,b}
F={q0}
s=q0

δ	a	b
q0	q0	q1
q1	q2	q0
q2	q2	q2

Determinar la secuencia de estados recorridos al procesar la cadena bbabb.

Respuesta: q0, q1, q0, q0, q1, q0

L(M)={ ε, a, bba, bbabb
Σ*-L(M)={ bab,

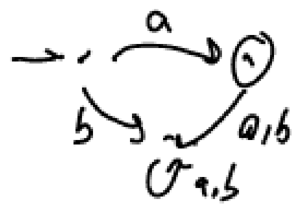
L(M)=L((a*(bb)*)*)

Construir un AFD que acepte el lenguaje L.

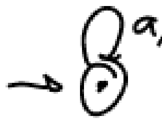
L={ cadenas de {a,b}* que empiecen con a }

L={ a, aa, ab, abababa, ...}
Σ*-L={ ε, b, ba, babab ab, ...}

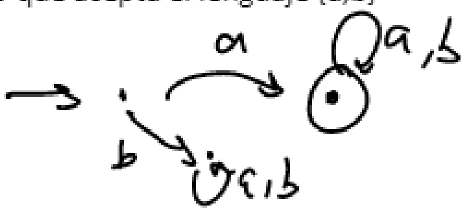
a(aUb)*

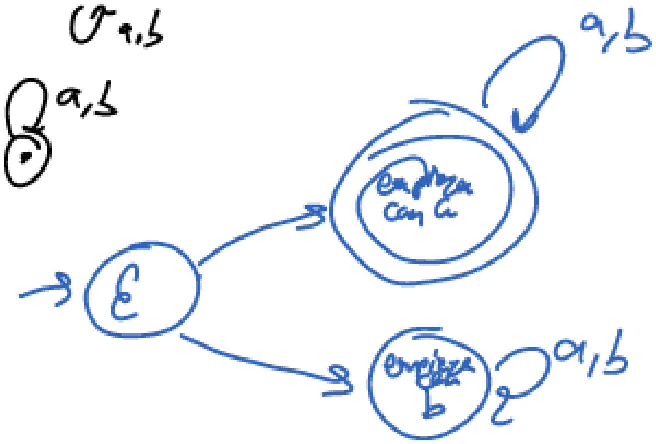


AFD que acepta el lenguaje {a}



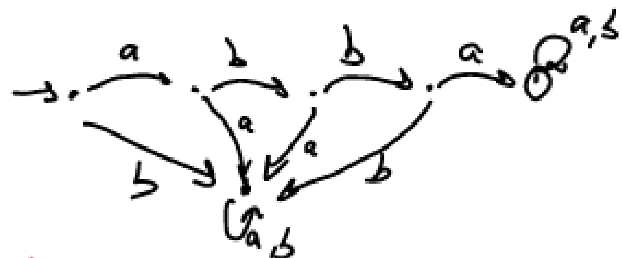
AFD que acepta el lenguaje {a,b}*





L={ cadenas que empiecen con abba }

L={ abba, abbaa, abbab, abbabababbaaab, … }
Σ*-L={ ε, a, aaa, bbb, baab, …}



L={ cadenas que terminen con b }

L={ b, ab, bb, babababab, … }
Σ*-L={ ε, a, ba, aba, abababaa, …}

(aUb)*b

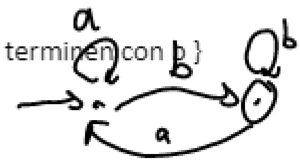
δ	a	b
q0	q0	q0, q1
q1	q0	

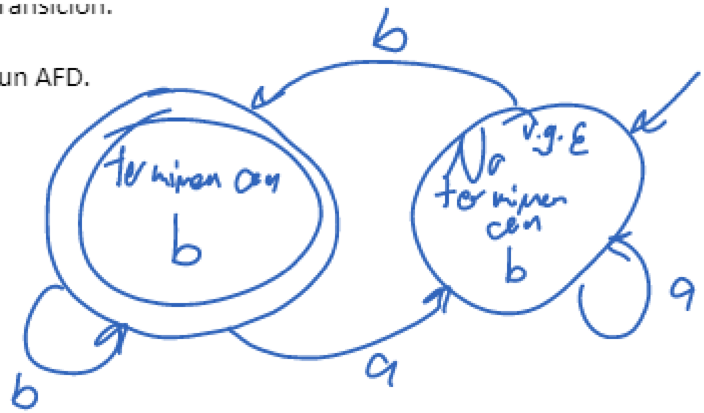
δ(q0,b)={q0,q1} Más de un estado destino posible para esta transición.
δ(q1,b)={}   Cero estados posibles para esta transición.

δ(q1,b)={} Cero estados posibles para esta transición.

Ninguno de estos casos debe presentarse en un AFD.

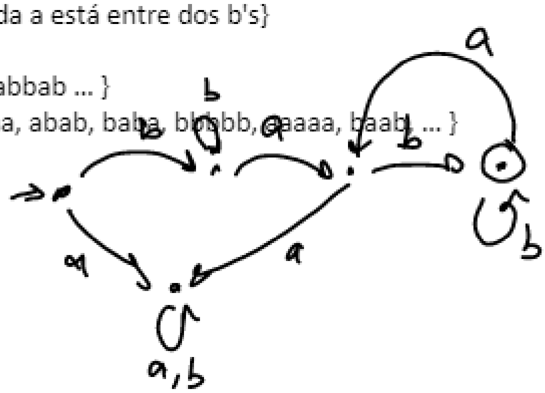
L={ cadenas que terminen con b }



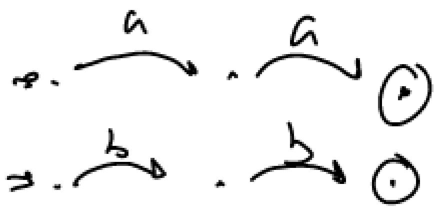


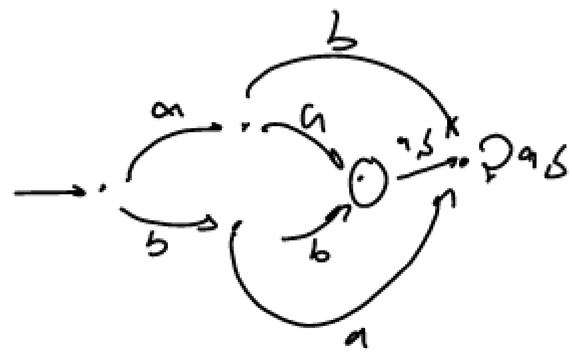
L={ Cadenas en las que toda a está entre dos b's}

L={ bab, bbab, babab, bbabbab ... }
Σ*-L={ ε, a, b, ab, ba, bb, aa, abab, baba, bbbbb, aaaaa, baab, ... }

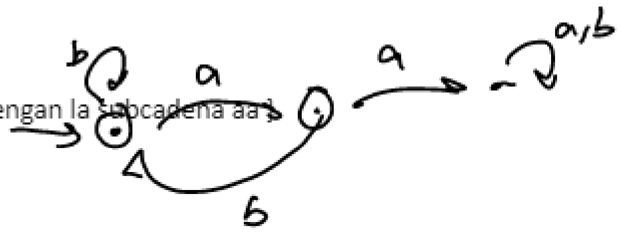


L={ aa, bb}





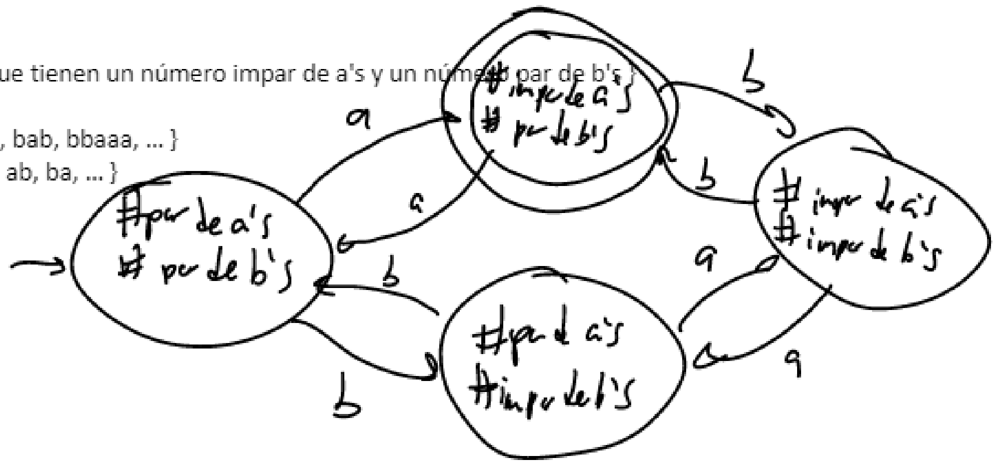
L={ cadenas que no contengan la subcadena aa }



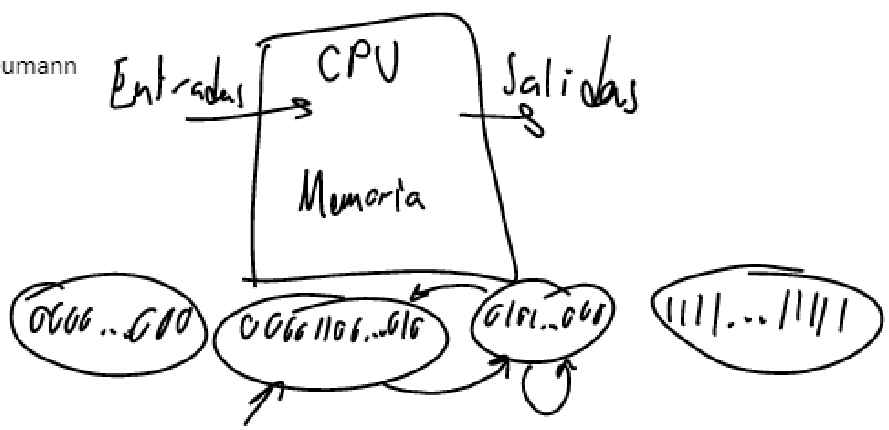
L={ cadenas que no contengan ninguna de las subcadenas aa o bb }

L={ cadenas que tienen un número impar de a's y un número par de b's }

L={ a, aaa, abb, bab, bbaaa, … }
Σ*-L={ ε, b, aa, ab, ba, … }



Modelo de Von Neumann



Autómata Finito No determinista (AFN)

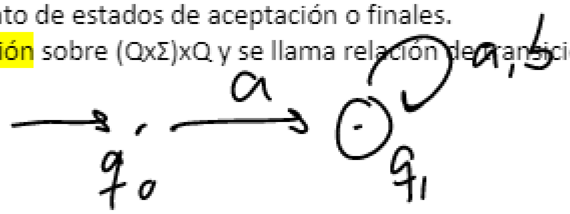
Un AFN es una colección de cinco objetos

$M=(Q,\Sigma,s,F,\Delta)$

Donde:

Q es un conjunto finito de estados.
Σ es un alfabeto de entrada.
s es uno de los estados de Q designado como estado inicial.
F es un conjunto de estados de aceptación o finales.
Δ es una relación sobre $(Q\times\Sigma)\times Q$ y se llama relación de transición.

Ejemplo:

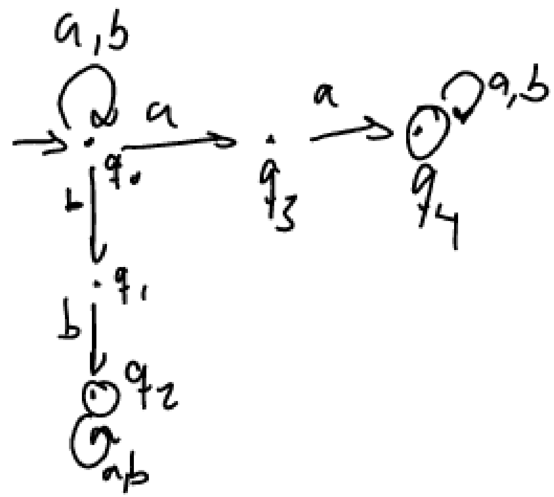


L(M)={ cadenas que empiecen con a } = L(a(a*b*)*)

Q={ q0,q1 }
Σ={ a, b}
s=q0
F={q1}

Δ(q0,a)=q1
Δ(q1,a)=q1
Δ(q1,b)=q1

Δ	a	b
q0	Q1	
q1	q1	q1



aabb

Δ(q0,aabb)=Δ(Δ(q0,a), abb)
=Δ(Δ(Δ(q0,a), a), bb)
=Δ(Δ(Δ(Δ(q0,a), a), b), b)
=Δ(Δ(Δ({q0,q3}, a), b), b)
=Δ(Δ({q0,q3,q4},b), b)
=Δ({q0,q1,q4} , b)
= {q0,q1,q2,q4}

q2ϵF (al igual que q4), entonces aabb se acepta.

En general, decimos que si tenemos un AFN M=(Q,Σ,s,F,Δ) y wϵΣ*, w se acepta sí y solo sí

∃ x ϵ Δ(s,w), xϵF.

abab

Δ(q0,abab)={q0,q3}
Δ({q0,q3},bab)={q0,q1}
Δ({q0,q1},ab)={q0,q3}
Δ({q0,q3},b)={q0,q1}

Ningún elemento del conjunto de estados finales
es miembro de F. Entonces abab se rechaza.

Procesar la cadena abba

$\Delta(q0, abba) = \{q0,q3\}$
$\Delta(\{q0,q3\},bba) = \{q0,q1\}U\{\}=\{q0,q1\}$
$\Delta(\{q0,q1\},ba) = \{q0,q1\}U\{q2\}=\{q0,q1,q2\}$
$\Delta(\{q0,q1,q2\},a) = \{q0,q3\}U\{\}U\{q2\} = \{q0,q2,q3\}$

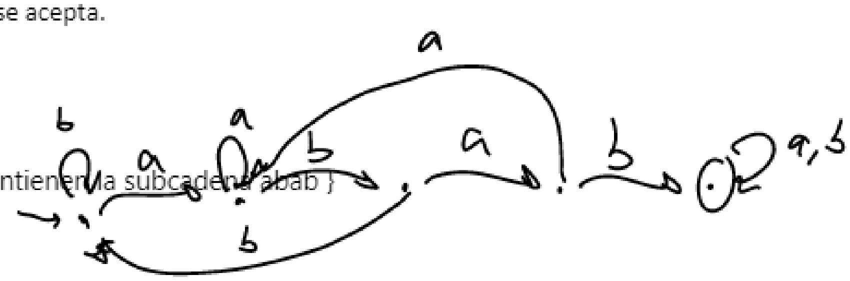
$q2 \epsilon F$, entonces abba se acepta.

Procesar la cadena abaab.

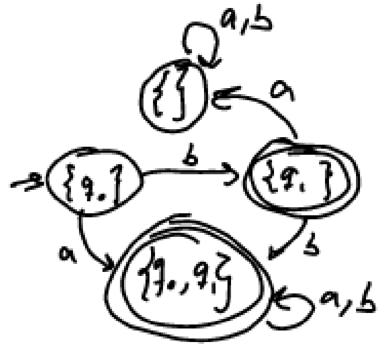
$\Delta(q0,a) = \{ q0, q3 \}$
$\Delta(\{q0, q3\}, b) = \Delta(q0,b) \ U \ \Delta(q3,b) = \{q0,q1\}U\{\}=\{q0,q1\}$
$\Delta(\{q0,q1\},a) = \Delta(q0,a) \ U \ \Delta(q1,a) = \{q0,q3\}U\{\}=\{q0,q3\}$
$\Delta(\{q0,q3\}, a) = \Delta(q0,a)U\Delta(q3,a)=\{q0,q3\}U\{q4\}=\{q0,q3,q4\}$
$\Delta(\{q0,q3,q4\}, b) =\Delta(q0,b)U\Delta(q3,b)U\Delta(q4,b) = \{q0,q1\}U\{\}U\{q4\}$
$=\{q0,q1,q4\}$

$q4 \ \epsilon \ F$, entonces abaab se acepta.

L1 = { cadenas que contienen la subcadena abab }



AFN A AFD



M=(Q,Σ,s,F,Δ)

Q={q0,q1}
Σ={a,b}
s=q0
F={q1}

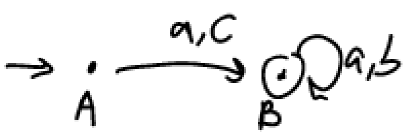
Δ	a	b
q0	{q0,q1}	{q1}
q1	{}	{q0,q1}

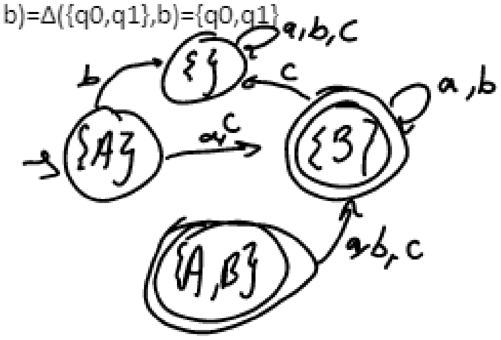
Construiremos el autómata finito determinista M'
M'=(Q',Σ',s',F',Δ') equivalente a M (L(M)=L(M')).

Q'=2^Q={ {}, {q0}, {q1}, {q0,q1} }
Σ' = Σ = {a,b}
s'={q0}
F'={ q ϵQ' | ∃ xϵq, xϵF } ={ {q1},{q0,q1} }

Δ'(q',σ)= Δ(q',σ)

Δ'({}, a)=Δ({},a)={}
Δ'({}, b)=Δ({},b)={},
Δ'({q0}, a)=Δ({q0},a)={q0,q1}
Δ'({q0}, b)=Δ({q0},b)={q1}
Δ'({q1}, a)=Δ({q1},a)={}
Δ'({q1}, b)=Δ({q1},b)={q0,q1}
Δ'({q0,q1}, a)=Δ({q0,q1},a)={q0,q1}
Δ'({q0,q1}, b)=Δ({q0,q1},b)={q0,q1}





F={ B }

Q'={ {}, {A}, {B}, {A,B} }
Σ'={a,b,c}
s'={A}
F'={ {B}, {A,B} }

LEMA DEL BOMBEO

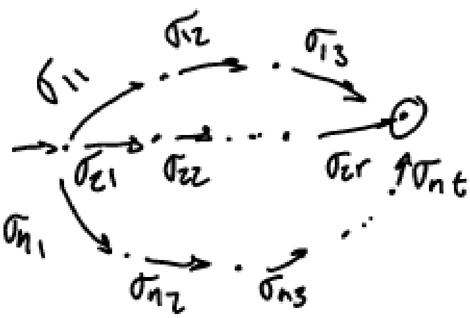
Todo lenguaje finito es regular.

L={ w1, w2, ..., wn }

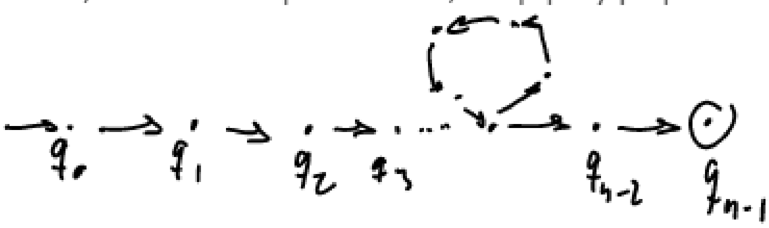
L= {w1} U {w2} U ... U {wn}

Para cualquier w_i

w_i = ({σ_i1})({σ_i2})...({σ_im})



Sea L un lenguaje regular infinito. Entonces hay una constante n de forma que, si w es una cadena de L cuya longitud es mayor o igual que n, se tiene que w=uvx, siendo uv^ix ϵ L para todo i≥0, con |v|≥1 y |uv|≤n.



Pensemos que el autómata tiene n estados. Pensemos que queremos procesar una cadena de longitud 2n.

Ejemplo:
L1={ a^n }
L1={ ε, a, aa, aaa, ... }
Σ*-L1={ b, ab, ba, bbb, ...}

w= aaa   n≤|w|=3

u=a
v=a
x=a
w = u v x = (a)(a)(a)

{uv^ix }= { aa, aaa, aaaa, ... }

L1 sí cumple con el lema del bombeo.

L2={ a^n b^m }
L2={ ε, a, b, ab, aaaab, abbb, ...}
Σ*={ ba, baaaaa, abaaaaa, abababab, ... }

w=abbb

u=a
v=b
x=bb
{ uv^ix } = { abb, abbb, abbbb, abbbbb, ...}

u=ε
v=ab
x=bb
{ uv^ix } = { bb, abbb, ababbb, ... }

u=a
v=bb
x=b
{ uv^ix } = { ab, abbb, abbbbb, abbbbbbb, ... }

L2 sí cumple con el lema del bombeo.

L3={ a^n b^n }
L3={ ε, ab, aabb, aaabbb, ...}
Σ*-L3={ a, b, ba, aba, abbba, ... }

w=aaabbb

u=aa
v=a
x=bbb

{ uv^ix } = { aabbb, aaabbb, aaaabbb, aaaaabbb, ... }

u=aa
v=ab
x=bb
{ uv^ix } = { aabb, aaabbb, aaababbb, aaabababbb , ... }

L3 no cumple con el lema del bombeo. Por lo tanto, L3 no es un lenguaje regular.

L4={ (ab)^n | n>0 }
L4={ ab, abab, ababab, abababab,... }
Σ*-L4={ ε, a, b, ba, aba, aab, abb, ... }

w= abababab

u=aba
v=b
x=abab
{ uv^ix } = { abaabab, abababab, ababbabab, ...}

u= ab
v= aba
x=bab
{ uv^ix } = { abbab, ... }

u= ab
v= ababab
x=ε
{ uv^ix } = { ab, abababab, ababababababab, ...}

u= ε
v= ab
x= abab
{ uv^ix } = {abab, ababab, abababab, ... }

L4 sí cumple con el lema del bombeo.

L5={ a^n | n=i^2 }
L5={ ε, a, aaaa, aaaaaaaaa, aaaaaaaaaaaaaaaa, ... }
Σ*-L5={ b, aa, bb, ba, ab, aaa, ...}

Supongamos que L5 sí cumple con el lema del bombeo.
Entonces debe existir una cadena de longitud mínima (digamos m)
de manera que cadenas de longitud mayor o igual a la mínima se puedan
dividir en tres partes de manera que una sea "bombeable".

Proponemos w=a^(m^2) = aaaaaaaa...aaaa (m^2 veces)
w=uvx
|w|=|uvx|= m^2

Recordemos que |uv|≤ m
1≤|v|≤m

|uvx|= m^2
$|uv^2x|=|uvx|+|v|$

$|uv^2x| \leq$ m^2 +m

Pero, a continuación de w=a^(m^2), debería seguir la cadena a^((m+1)^2).
| a^((m+1)^2) | = (m+1)^2 = m^2 + 2m +1

$|uv^2x| \leq$ m^2 +m < m^2 + 2m +1

L5 no cumple con el lema del bombeo. Por lo tanto, L5 no es un lenguaje regular.

TEOREMA DE MYHILL NERODE

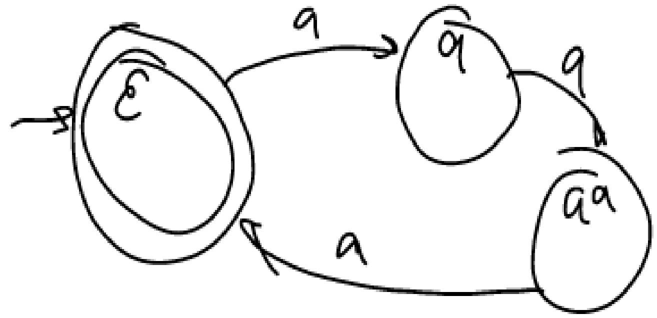
Teorema de Myhill-Nerode (1958)

martes, 1 de diciembre de 2020 03:03 p. m.

Dado un lenguaje L y un par de cadenas x,y ∈Σ*, se define una **extensión discriminante** como una cadena z∈Σ* tal que exactamente una de las dos cadenas xz, yz, pertenecen a L.

Definimos la relación R sobre cadenas mediante la regla de que xRy si no hay una extensión discriminante para x,y. R es una relación de equivalencia. Por lo tanto, divide Σ* en clases de equivalencia.

El teorema de Myhill-Nerode afirma que L es regular si, y solamente si, R tiene un número finito de clases de equivalencia. Además, el número de estado del menor AFD que reconoce a L es igual al número de clases de equivalencia de R.



Σ={a}

L={ a^{3n}, n≥0}

L={ ε, aaa, aaaaaa, aaaaaaaaa, ... }

Σ*-L={ a, aa, aaaa, ... }

Σ*={ ε, a, aa, aaa, aaaa, aaaaa, aaaaaa, ... }

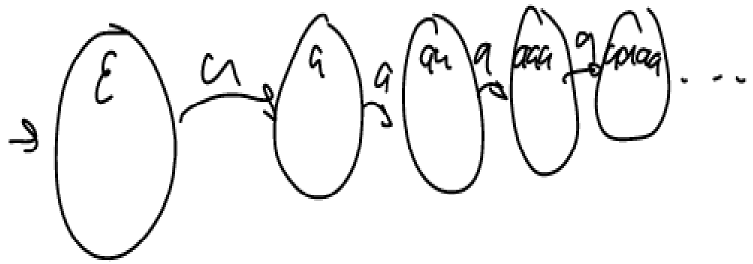
x= aaa	x= a	x= ε	x= a	x= a	x= aa
y= aa	y= ε	y= aaa	y= aa	y= aaaa	y= aaaaa
z= a	z= ε	z= ε	z= ε	z= aaaaa	z= a
xz= aaaa ∉L	xz= a ∉ L		xz= a ∉L	xz= aaaaaa ∈L	xz= aaa ∈L
yz= aaa ∈L	yz= ε ∈ L	xRy= εRaaa	yz= aa ∉L	yz= aaaaaaaaa ∈L	yz= aaaaaa ∈L
				aRaaaa	aaRaaaaa
z= aaa	z= aa		z= a		
xz= aaaaaa ∈L	xz= aaa ∈L		xz= aa ∉L		
yz= aaaaa ∉L	yz= aa ∉L		yz= aaa ∈L		

R define 3 clases de equivalencia. Por lo tanto, L es un lenguaje regular y puede ser reconocido con un AFD con 3 estados.

$L=\{ a^n b^n \}$

$L=\{ \varepsilon$, ab, aabb, aaabbb, aaaabbbb, ….}

Σ^*-L={ a, b, ba, abb, aab, … }



x= a	x= aa	x=a
y= aa	y= aaa	y=aaa
z= b	z= bb	z=b
xz= ab $\in$L	xz=aabb $\in$L	xz= ab $\in$L
Yz= aab $\notin$L	yz=aaabb $\notin$L	yz= aaab $\notin$L

Sean r,s$\in$N, r≠s

$x = a^r$

$y= a^s$

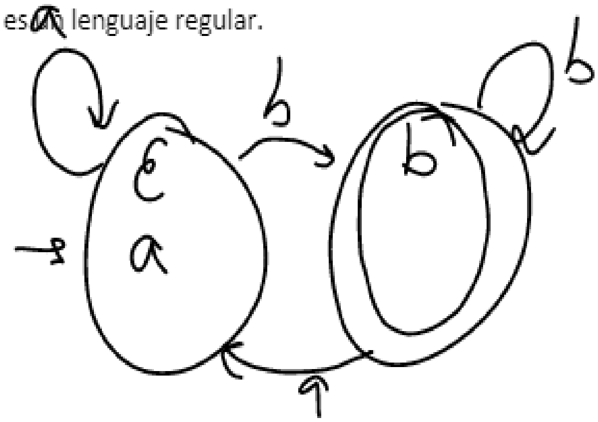
$z= b^s$

$xz= a^r b^s \notin L$

$yz= a^s b^s \in L$

R tiene infinitas clases de equivalencia. Por lo tanto L no es un lenguaje regular.

R tiene infinitas clases de equivalencia. Por lo tanto L no es un lenguaje regular.



L={ cadenas que terminen con b }

L={ b, ab, bb, aab, abb, bab, bbbb, … }
Σ*-L={ ε, a, aa, ba, baa, bba,bbbbbbbbba, …}

x= ε	x= a	x=ε	x=ab
y=b	y=b	y=a	y= bbb
z= ε	z=ε	z=ε	z= ba
xz= ε ∉L	xz= a ∉L	xz= ε ∉L	xz=abba ∉L
yz= b ϵL	yz=b ϵL	yz= a ∉L	yz=bbbba ∉L
		εRa	abRbbb

x=ba	x=bb
y= a	y=b
z=	z=
baRa	bbRb

R tiene 2 clases de equivalencia. Por lo tanto, L es regular.

L={ cadenas con el mismo número de a's que de b's }

L={ ε, ab, ba, aabb, abab, baba, abba, bbaa,… }
Σ*-L={ a, b, aab, aba, bba, …}

w = uvx = (ε)(ab)(ε)

Sí cumple con el lema del bombeo.

x= b
y= bb
z= a

Para cualquier r,sϵN r≠s,

$x = b^r$

$y = b^s$

$z = a^r$

Entonces tenemos un número infinito de clases de equivalencia y L no es regular.

L={ a^p, p es un número primo }

Supongamos que sí cumple con el lema del bombeo.

W = uvx

$|u|=l_u$
$|v|=l_v$
$|x|=l_x$

$|uv^ix| = l_u+(i)l_v+l_x$ supongamos que $i= l_u+l_x$

$|uv^ix| = l_u+(l_u+l_x)l_v+l_x$
$= (l_u+l_x)+(l_u+l_x)l_v$
$=(l_u+l_x)(1 + l_v)$ no es un número primo.

Por lo tanto, L no cumple con el lema del bombeo.

AUTOMATA DE PILA NO DETERMINISTA (ADPND)

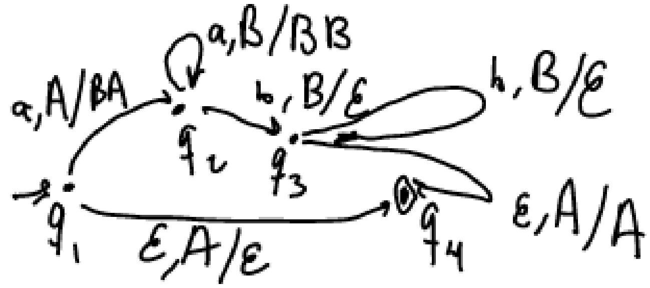
Un autómata de pila no determinista (ADPND) es una 7-tupla
$M=(Q,\Sigma,\Gamma,\Delta,s,F,z)$

Q es un conjunto finito de estados
Σ es un alfabeto de entrada
Γ es un alfabeto llamado alfabeto de la pila
Δ es una regla de transición $\Delta \subseteq Qx(\Sigma U\{\varepsilon\})x\Gamma xQx\Gamma^*$
$s\epsilon Q$ es el estado inicial
$z\epsilon\Gamma$ es el símbolo inicial de la pila
$F\subseteq Q$ es el conjunto de estados finales o de aceptación

En la representación gráfica de un ADPND, los elementos de Q, s, F se representan de la misma manera que en los AFD. En las transiciones, ilustraremos con una flecha los estados inicial y final, y la etiqueta tendrá la forma $\sigma,X/\gamma$, donde $\sigma\epsilon(\Sigma U\{\varepsilon\})$ es el símbolo consumido de la cadena de entrada, $X\epsilon\Gamma$ es el símbolo que se encuentra visible en la pila (este símbolo es retirado de la pila) y $\gamma\epsilon\Gamma^*$ es lo que se introduce a la pila.

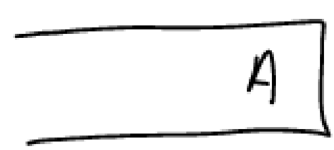
$\Delta(q,\sigma,X)= (p,\gamma)$

Ejemplo:

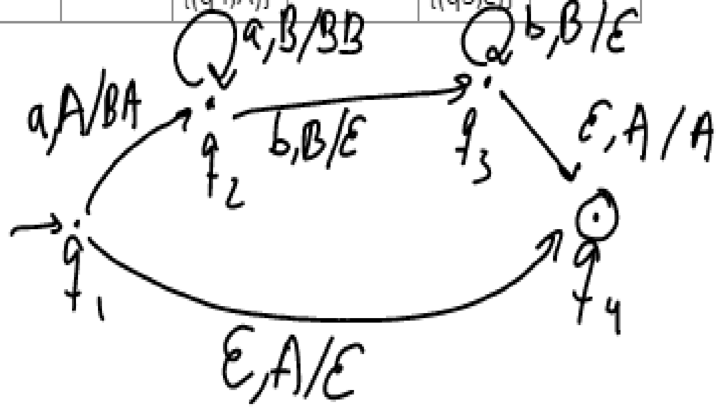


Q={ q1,q2,q3,q4 }
Σ={a,b}
Γ={A,B}
z=A
F={ q4 }
s=q1

Estado	Entrada	Pila
q1	aaabbb	A
q2	aabbb	BA
q2	abbb	BBA
q2	bbb	BBBA
q3	bb	BBA
q3	b	BA
q3	ε	A
q4	ε	A

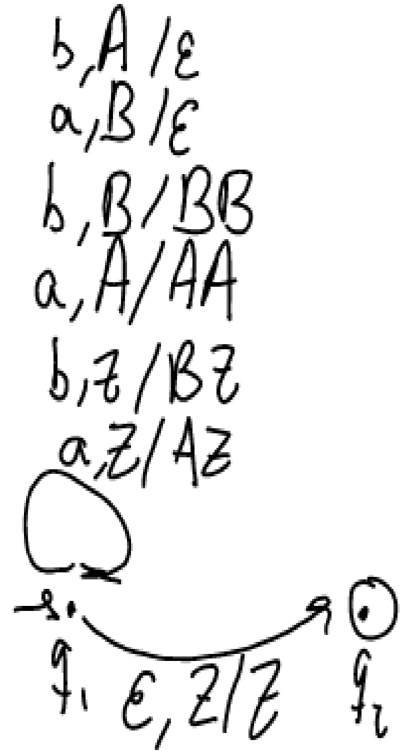


Δ	(a,A)	(b,A)	(ε,A)	(a,B)	(b,B)	(ε,B)
q1	{(q2,BA)}		{(q4,ε)}			
q2				{(q2,BB)}	{(q3,ε)}	
q3			{(q4,A)}		{(q3,ε)}	



Δ(q1,a,A)={ (q2,BA) }
Δ(q2,a,B)={ (q2,BB) }
Δ(q2,b,B)={ (q3,ε) }
Δ(q3,b,B)={ (q3,ε) }
Δ(q3,ε,A)={ (q4,A) }
Δ(q1,ε,A)={ (q4,ε) }

L={ cadenas con el mismo número de a's que de b's }
L={ ε, ab, ba, abba, bbaa, baba, abab, ... }
Σ*-L={ a, b, aaa, bbb, aaabb, ... }



Q={q1,q2}
Σ={a,b}
Γ={A,B,Z}
s=q1
z=Z
F={q2}

(q1,ababa,Z)⊢(q1,baba,AZ)⊢(q1,aba,Z)
⊢(q1,ba,AZ)⊢(q1,a,Z)⊢(q1,ε,AZ)

(q1,bbaa,Z)⊢(q1,baa,BZ)⊢(q1,aa,BBZ)⊢(q1,a,BZ)⊢(q1,ε,Z)⊢(q2,ε,Z)

abba	Z
bba	AZ
ba	Z
a	BZ
ε	Z

La terna (q,w,u) donde q es el estado actual, w es la cadena de entrada restante y u el contenido de la pila
(con el símbolo de la cima en el extremo de la izquierda) se llama descripción instantánea del autómata.

(p,aw,bx) ⊢ (q, w, yx)

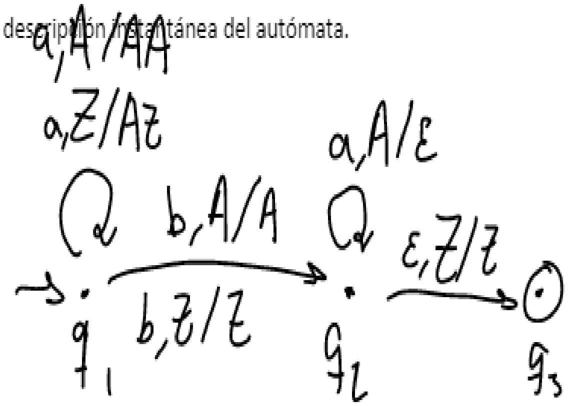
Representa δ(p,a,b)=(q,y)

L1={ (a^n)b(a^n) } = {b, aba, aabaa, aaabaaa, ...}

Σ*-L={ ε, a, aa, bb, abaa, ... }

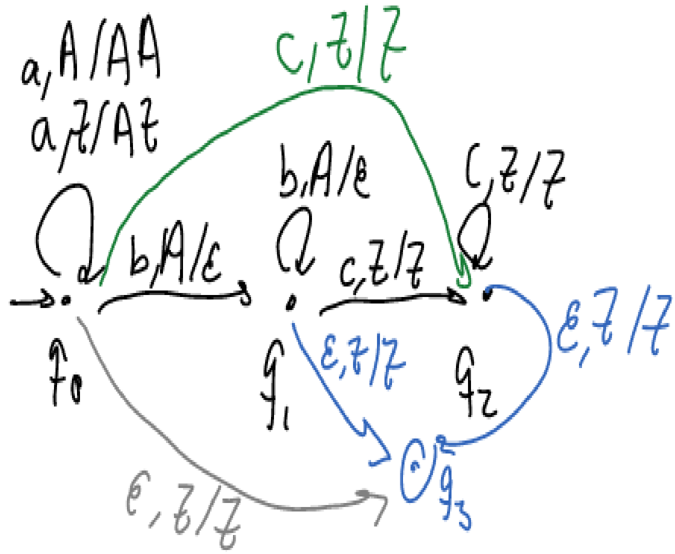
(q1, aabaa, Z) ⊢ (q1, abaa, AZ) ⊢(q1, baa, AAZ)
⊢(q2, aa, AAZ)⊢(q2, a, AZ)⊢ (q2, ε, Z)⊢(q3,ε,Z)

(q1, b, Z)⊢(q2,ε,Z)⊢(q3,ε,Z)



Q={q1,q2,q3}
Σ={a,b}
Γ={A,Z}
s=q1
F={q3}
z=Z

Δ={ (q1,a,Z,q1,AZ), (q1,a,A,q1,AA), (q1,b,A,q2,A),
(q1,b,Z,q2,Z), (q2,a,A,q2,ε), (q2,ε,Z,q3,Z) }



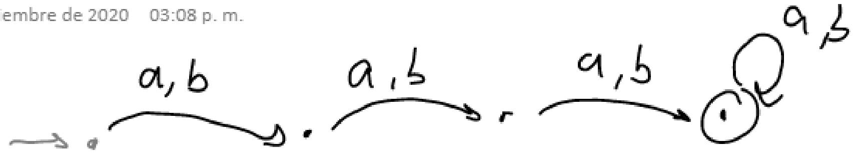
L2={ (a^n)(b^n)(c^m) } = { ε, c, ab, abc, aabbccccc, ... }

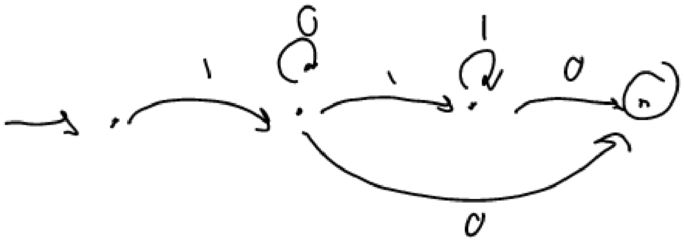
(q0,abc,Z)⊢(q0,bc,AZ)⊢(q1,c,Z)⊢(q2,ε,Z)

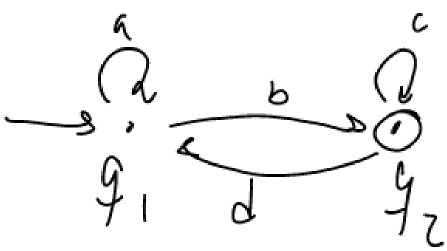
(q0,ab,Z)⊢(q0,b,AZ)⊢(q1,ε,Z)⊢(q3,ε,Z)

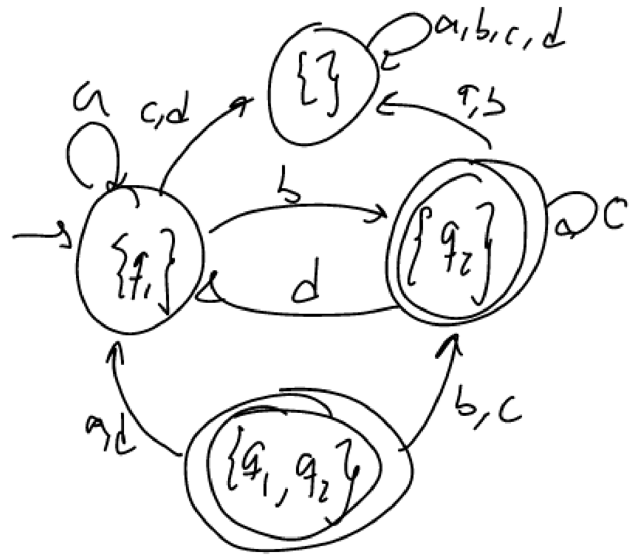
GUIA DE EXAMEN

miércoles, 9 de diciembre de 2020 03:08 p. m.









Q'={ {}, {q1}, {q2}, {q1,q2} }
Σ'={a,b,c,d}
s'={q1}
F'={ {q2}, {q1,q2} }

Δ'={ ({},a,{}), ({},b,{}), ({},c,{}), ({},d,{}),
({q1},a,{q1}), ({q1},b,{q2}), ({q1},c,{}), ({q1},d,{}),
({q2},a,{}), ({q2},b,{}), ({q2},c,{q2}), ({q2},d,{q1}),
({q1,q2},a,{q1}), ({q1,q2},b,{q2}), ({q1,q2},c,{q2}), ({q1,q2},d,{q1}) }

w=0000
u=00
v=00
x=ε

$uv^i x= 00(00)^i \in L1 \ \forall n \in \mathbf{N}$

L1 sí cumple con el lema del bombeo.

$a^n b^{2n} c^{3n}$

$x= a^n$

$y= a^{n+1}$

$z= b^{2n} c^{3n}$

$xz \in L2$

$yz \notin L2$ Entonces $z= b^{2n} c^{3n}$ es una extensión discriminante.

Dados $n,m \in N$, $n \neq m$

$x= a^n$

$y= a^m$

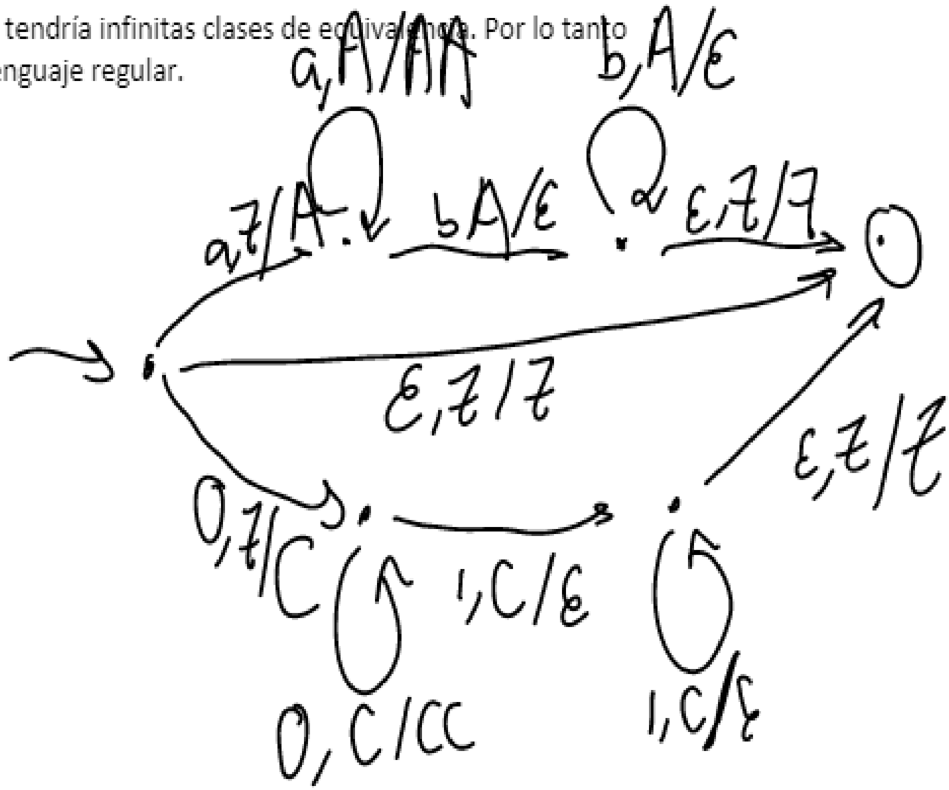
$z= b^{2n} c^{3n}$

$xz \in L2$

$yz \notin L2$ Entonces $z= b^{2n} c^{3n}$ es una extensión discriminante.
Este lenguaje tendría infinitas clases de equivalencia. Por lo tanto
L2 no es un lenguaje regular.

a, A/AA   b, A/ε

$yz \notin L2$ Entonces $z = b^{2n}c^{3n}$ es una extensión discriminante.
Este lenguaje tendría infinitas clases de equivalencia. Por lo tanto L2 no es un lenguaje regular.



$\Sigma=\{a,b,0,1\}$

$\Gamma=\{A,C,Z\}$

z=Z

$L=\{ a^n b^n \cup 0^m 1^m \}$

$a^n b^n =\{ \varepsilon, ab, aabb, aaabbb, aaaabbbb, \dots \}$
$0^m 1^m =\{ \varepsilon, 01, 0011, 000111, 00001111, \dots\}$

$\{ a^n b^n \cup 0^m 1^m \}= \{ \varepsilon, ab, 01, aabb, 0011, aaabbb, 000111, \dots \}$