Universidad Nacional de Cajamarca
"Norte de la Universidad Peruana"

UNIDAD II:
AUTOMATAS FINITOS (AF)

ING. SANDRA RODRIGUEZ AVILA

Resultado de aprendizaje 3 (RA-3): Desarrolla problemas relacionados con expresiones regulares y de construcción de autómatas para reconocer lenguajes definidos por una Gramática del Tipo 2 y 3 y traductores para traducir lenguajes definidos por una Gramática del Tipo 2 y 3, considerando propiedades del algebra de Boole, definiciones formales de autómatas y traductores y algoritmos de transformación.

CONTENIDO

- REPRESENTACION DE UN AF - DIAGRAMA DE ESTADOS - MATRIZ DE TRANSICION
- EJEMPLOS DE AF
- CLASIFICACIÓN DE LOS AF
- TRANSFORMACIÓN DE UN AFND EN AFD
- TRANSFORMACIÓN DE UNA GLD EN UN AF
- TRANSFORMACIÓN DE UN AF EN UNA GLD
- TRANSFORMACIÓN DE UNA GLI EN UN AF
- TRANSFORMACIÓN DE UN AF EN UNA GLI

INTRODUCCION

DEFINICION FORMAL DE UN AF:

$$AF = (Q, \text{Te}, \delta, q1, F)$$

CONFIGURACION DE UN AF:

(q, w)

q es el estado actual y w es la cadena por leer.

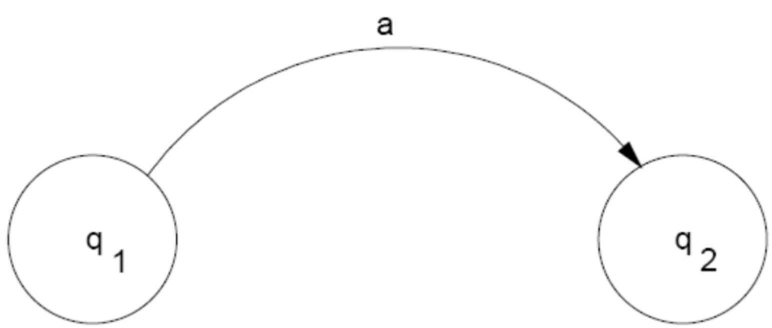
LENGUAJE RECONOCIDO POR UN AF:

$$L(AF) = \{t/t \in Te^*, (q1,t) \rightarrow (qi, \lambda), qi \in F\}$$

REPRESENTACION DE UN AF DIAGRAMA DE ESTADOS

- Los nodos son los estados y las ramas están marcadas con los símbolos de entrada. Las ramas se construyen según la función de transición, así debe de cumplir :

$\delta(q1, a) \rightarrow q2$



- Los nodos que representan los *estados finales*, se marcan con un doble círculo, y el *estado inicial* también se marca con una flecha.

REPRESENTACION DE UN AF
MATRIZ DE TRANSICION

- Los estados se ubican en las filas y los símbolos de entrada en las columnas. En las entradas de la matriz se coloca el estado hacia el cual se transita, según la función δ:

$\delta(q1, a) \to q2$

δ	a	b
q_1	q_2	q_4
q_2	q_2	q_3
q_3	q_4	q_3
q_4	q_4	q_4

EJEMPLOS AF

Ejemplo 1

Sea el autómata finito *A*1 = (*Q*, Te, *δ*, *q*1,*F)* donde *Te* = {*a*,*b*} U {λ}; *Q* = {*q*1, *q*2, *q*3, *q*4};y la función *δ* viene dada por la tabla siguiente y el conjunto de estados finales es F={q3}

δ	*a*	*b*
q_1	q_2	q_4
q_2	q_2	q_3
q_3	q_4	q_3
q_4	q_4	q_4

EJEMPLOS AF

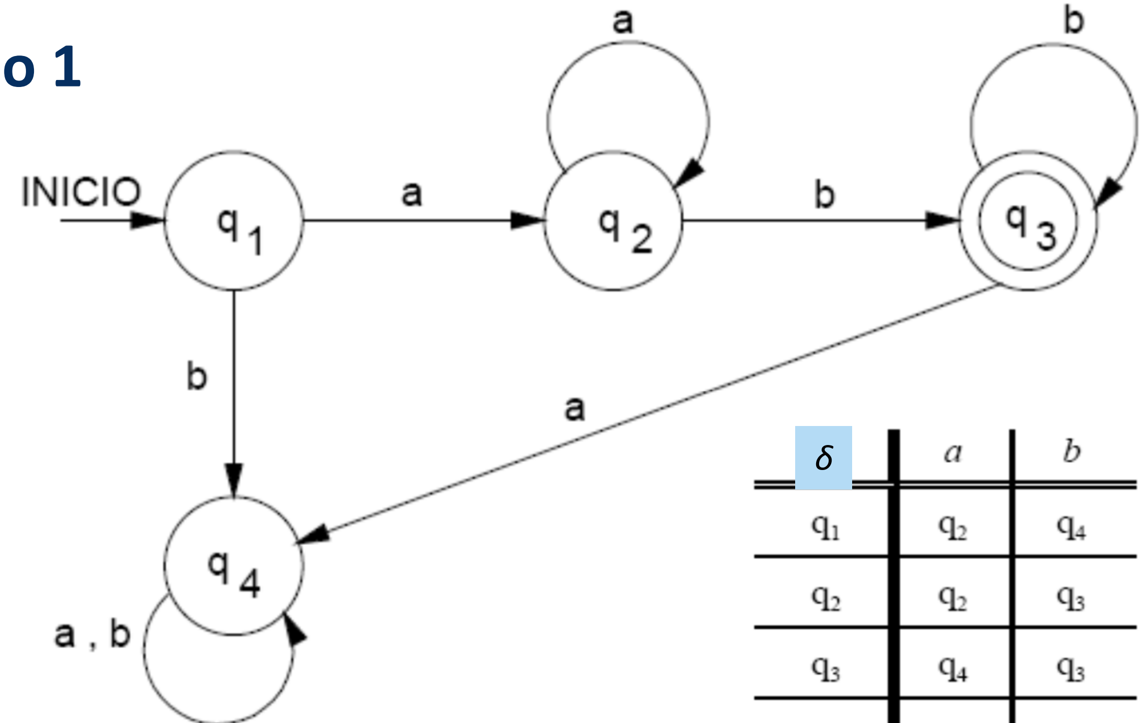
Ejemplo 1

Determinar el lenguaje que reconoce, representar el diagrama de Moore, e indicar la expresión regular que representa al lenguaje.

Solución : Para construir las ramas, nos situamos en el primer estado de la tabla de transiciones y se observa que δ(q1,a)=q2, entonces se traza una flecha entre q1 y q2, apuntando a q2, y se coloca encima de la flecha el símbolo del vocabulario de entrada a. De igual forma se recorre la tabla de transiciones para cada estado y entrada completándose el diagrama de Moore.

EJEMPLOS AF

Ejemplo 1



δ	a	b
q_1	q_2	q_4
q_2	q_2	q_3
q_3	q_4	q_3
q_4	q_4	q_4

EJEMPLOS AF

Ejemplo 1

- El lenguaje generado se obtiene partiendo del estado inicial y recorriendo todos los caminos posibles para alcanzar el estado final. Así se obtiene que este autómata reconoce el lenguaje :

  $L(A1) = \{ab, aab, \ldots, abbb, \ldots, aabb, \ldots\}$

  $L(A1) = \{a^n b^m / n \geq 1, m \geq 1\}$

- La expresión regular que denota el lenguaje es a^+b^+ o también aa^*bb^*.

EJEMPLOS AF

Ejemplo 2

Construir un autómata finito que reconozca un identificador de un lenguaje de programación, definido en EBNF de la forma :

<identificador> ::= <letra> {<letra>|<dígito>}

<letra> ::= a|b|...|z

<dígito> ::= 0|1|2|...|9

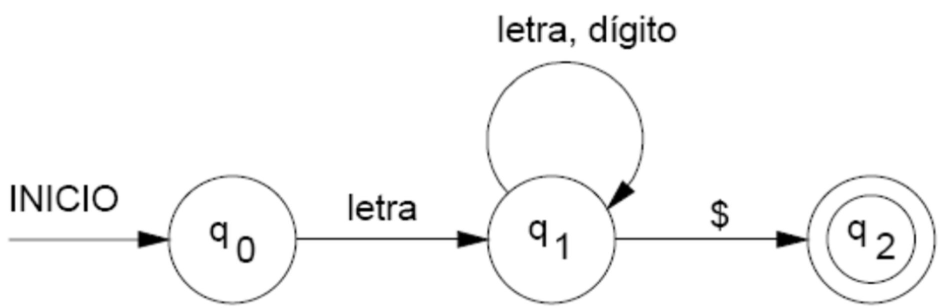
EJEMPLOS AF

Ejemplo 2

Solución : Este ejemplo es inverso al anterior, pues se da un lenguaje y se pide el autómata que lo reconoce. En primer lugar se construye un diagrama de estados, de tal forma que a partir del estado inicial, después de leer una letra, acepte letras o dígitos de forma variable, y cuando encuentre un carácter diferente de letra o dígito alcance el estado final. El diagrama de estados es el que se muestra en la figura:

EJEMPLOS AF

Ejemplo 2



- $ representa a todos los caracteres diferentes de letra o dígito.
- El AF se deduce del diagrama de estados y es:

$$AF = (Q = \{q0, q1, q2\}, Te = \{a, b, \ldots, z, 0, 1, \ldots 9, \$\}, \delta, q0, F = \{q2\})$$

EJEMPLOS AF

Ejemplo 2

δ	<letra>	<dígito>	$
q_0	q_1	-	-
q_1	q_1	q_1	q_2
q_2	-	-	-

CLASIFICACIÓN DE LOS AF: AF NO DETERMINISTA o AFND

- Se caracteriza por la posibilidad de que dada una entrada *e* en un estado q_i, se pueda pasar a un estado q_j, q_k,...,q_n sin saber a ciencia cierta, a cual de esos estados pasará.
- La definición formal de un AFND es:

$$AFND = (Q, Te, \delta, q1, F)$$

con la salvedad de que es no determinista.

$$\delta : Q \times Te^* \rightarrow Q$$

CLASIFICACIÓN DE LOS AF: AF NO DETERMINISTA o AFND

Ejemplo 3

- Sea el AFND:

$$AFND = (Q, Te, \delta, q1, F)$$

donde Te={a,b}, Q = {$q1$, $q2$, $q3$, $q4$} F = {$q4$}, y la función δ viene dada por la siguiente tabla :

δ	a	b
q_1	$\{q_2,q_3\}$	λ
q_2	λ	$\{q_2,q_4\}$
q_3	q_3	q_4
q_4	q_4	λ

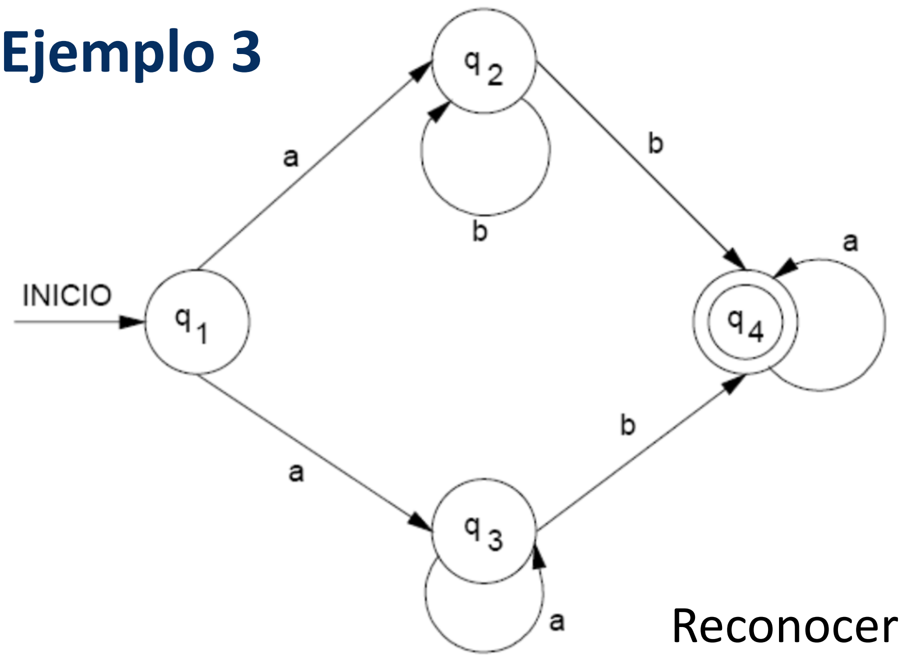
CLASIFICACIÓN DE LOS AF: AF NO DETERMINISTA o AFND

Ejemplo 3

- Determinar el lenguaje que reconoce, y dar su expresión regular.
  Solución : El diagrama de estados se construye al igual que en los ejemplos anteriores de autómatas finitos, con la salvedad de que para una entrada a un estado puede salir más de una flecha de un determinado estado.
  El lenguaje reconocido es el siguiente : a(b*b | a*b)a*o también a(b* | a*)ba*

CLASIFICACIÓN DE LOS AF: AF NO DETERMINISTA o AFND

Ejemplo 3



Algunas tiras reconocidas por el AFND:

ab, abb, abaa,aaba

abba, aaab, aab

E.R. (q1 →q2 →q4): ab*ba*

E.R. (q1 →q3 →q4): aa*ba*

E.R.: ab*ba* + aa*ba*=

a(b*+a*)ba*

Reconocer formalmente la tira aab:

(q1,aab)→(q3, ab) →(q3,b) →(q4, λ)

CLASIFICACIÓN DE LOS AF: AF DETERMINISTA o AFD

- Un AFD es un caso particular de los AF, en el que la función de transición δ no presenta ambigüedad en las transiciones de estados para una símbolo de entrada.
- Un AFD es una quíntupla de elementos:

$$AFD=(Q, Te, \delta, q1, F)$$

donde la función $\delta : Q \times Te^* \rightarrow Q$ es determinista.

TRANSFORMACIÓN DE UN AFND EN AFD

Teorema: "Para todo AFND=(Q, Te, δ, q1,F) se puede construir un AFD=(Q', Te, δ', q'1, F') tal que el lenguaje reconocido por el AFD coincida con el lenguaje reconocido por el AFND, es decir L(AFD) = L(AFND)".

Demostración :

- Se determina en primer lugar Q' que es el conjunto de las partes del conjunto de estados Q.

Q' = P(Q) = { conjunto de las partes de Q }

TRANSFORMACIÓN DE UN AFND EN AFD

- El cardinal de Q’ o número de estados del conjunto Q’ es :

$$\text{cardinal (Q')} = 2^{\text{cardinal(Q)}}$$

- Al estado de Q’ que corresponde a {qa,qb,...ql} se denotará por [qa,qb,...ql] es decir que se define δ’ de la forma :

$$\delta'([qa,qb,...ql],e)=[qm,qn,...qk] \text{ si y sólo si}$$
$$\delta(\{qa,qb,...ql\},e)=\{qm,qn,...qk\}$$

TRANSFORMACIÓN DE UN AFND EN AFD

- Es decir se calcula δ'(q',e) aplicando δ a cada estado q de los que figuran en q' y haciendo la unión de todos los resultantes.

q'1=[q1]

F'={ }

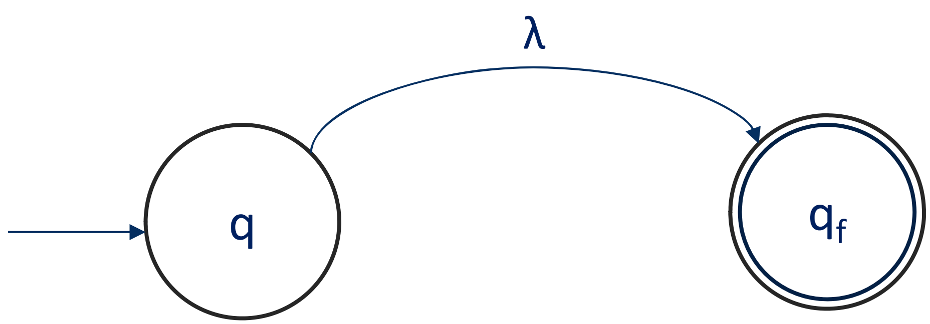
- Es decir, para que q' sea estado final basta que uno o más de los estados de Q que lo componen sea final.
- Ver ejemplo en:

TRANSFORMACIÓN DE UNA EXPRESIÓN REGULAR EN UN AF

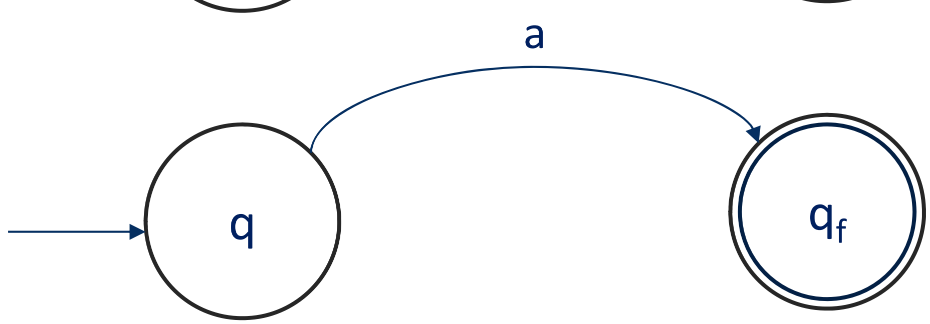
- Dada una expresión regular existe un AF capaz de reconocer el lenguaje que ésta define y viceversa.
- Para la transformación de una expresión regular en un AF, se definirán en un principio las equivalencias entre las expresiones regulares básicas y sus autómatas finitos.

TRANSFORMACIÓN DE UNA EXPRESIÓN REGULAR EN UN AF

1. Expresión regular λ

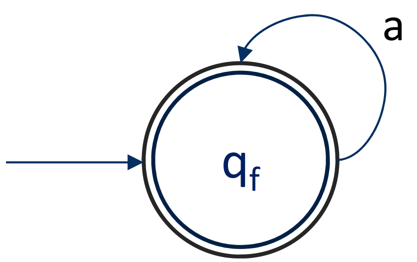


2. Expresión regular a

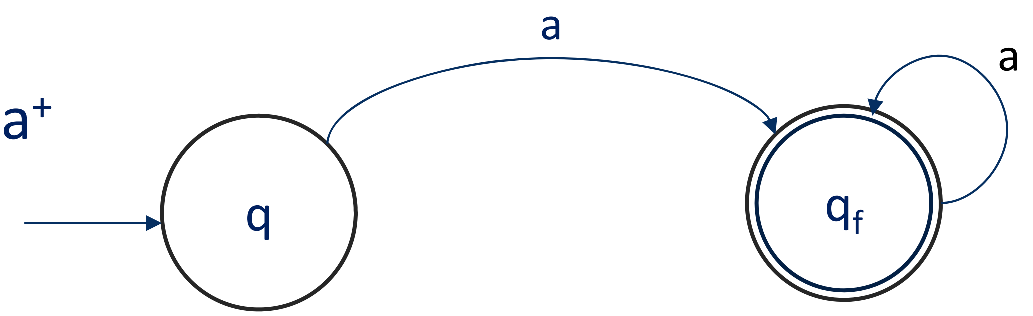


TRANSFORMACIÓN DE UNA EXPRESIÓN REGULAR EN UN AF

3. Expresión regular a*
L(ER)={λ,a, aa, aaa, …}



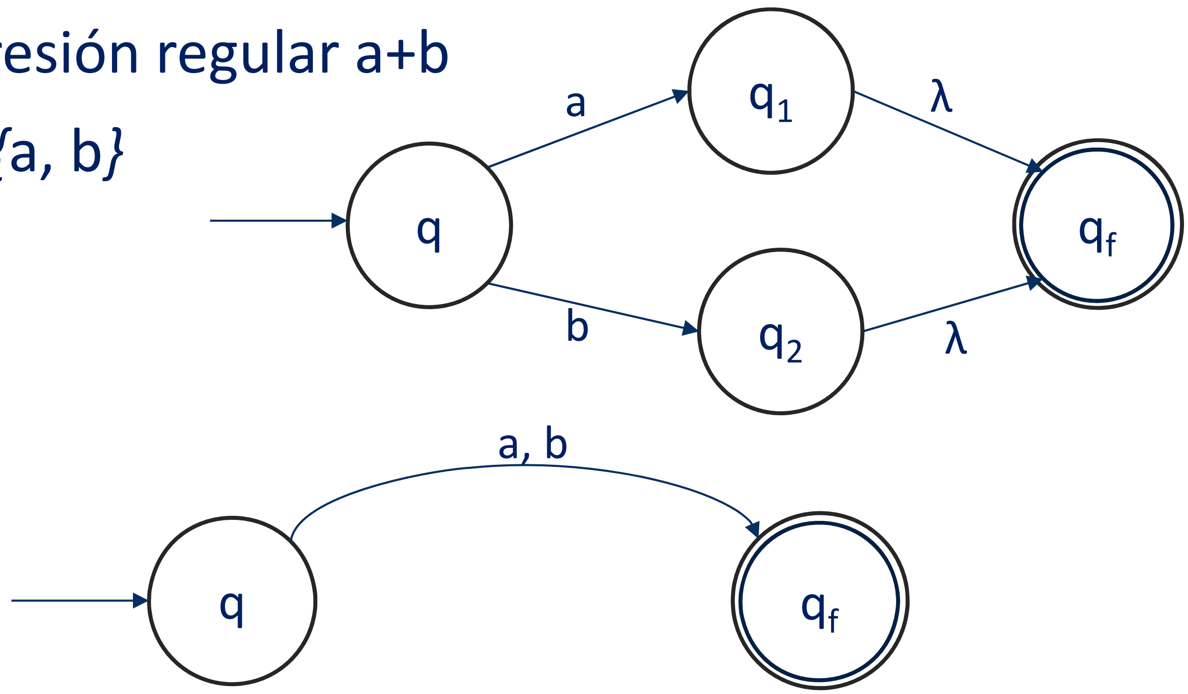
4. Expresión regular a^+
L(ER)={a,aa,aaa, …}



TRANSFORMACIÓN DE UNA EXPRESIÓN REGULAR EN UN AF

5. Expresión regular a+b

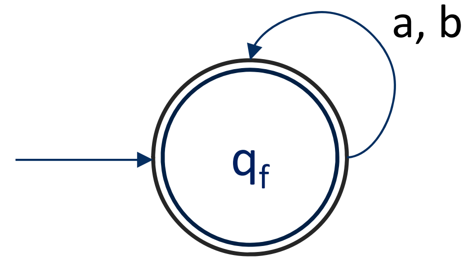
L(ER)={a, b}



TRANSFORMACIÓN DE UNA EXPRESIÓN REGULAR EN UN AF

6. Expresión regular (a + b)*

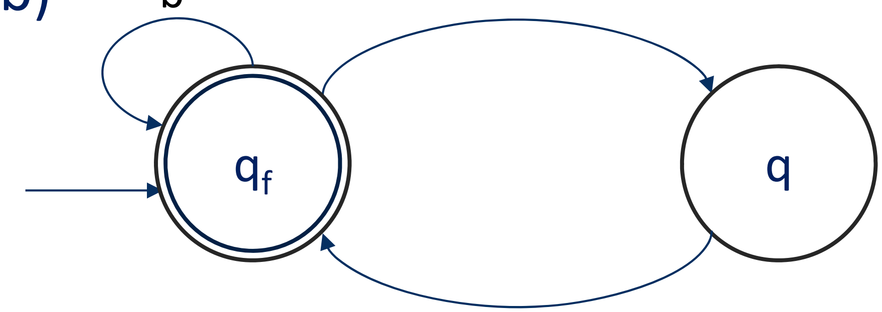
L(ER)={λ,a,b,ab,bba,aabb ...}



7. Expresión regular (ac + b)*

L(ER)={λ, b, ac, bac, bb, bacb...}

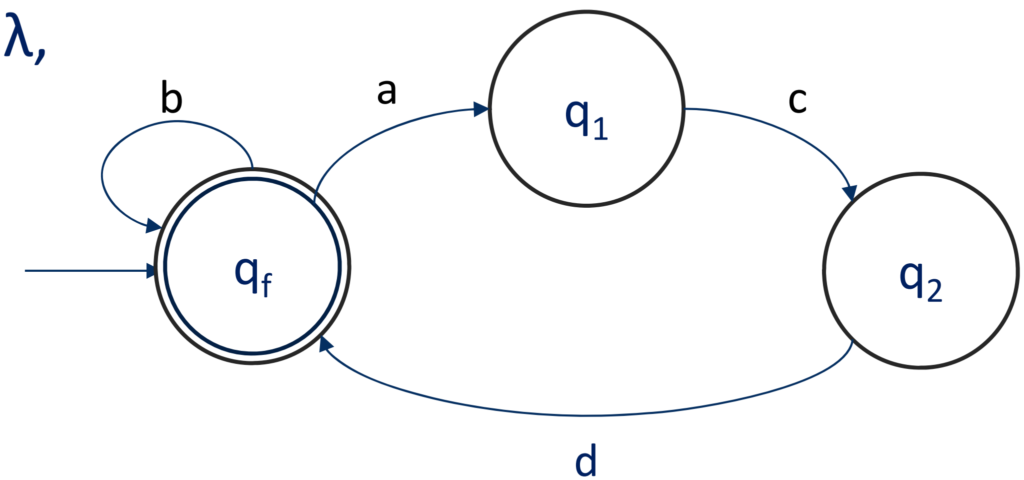
Te= {a,b,c}



TRANSFORMACIÓN DE UNA EXPRESIÓN REGULAR EN UN AF

8. Expresión regular (acd + b)*

L(ER)= {bbacd, acdb, λ,



TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA DERECHA EN UN AF

Sea una GLD=(N,T,P,S) y se desea obtener un AF=(Q, Te, δ, q1, F).

Solución: Se determinan los distintos elementos del AF.

Te=T

Q=N U {*qf*}

q1=S

F={qf}

TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA DERECHA EN UN AF

1. A cada símbolo N de la gramática se le asocia un estado del AF. Se introduce un nuevo estado, denominado qf, que será el único estado final del autómata.

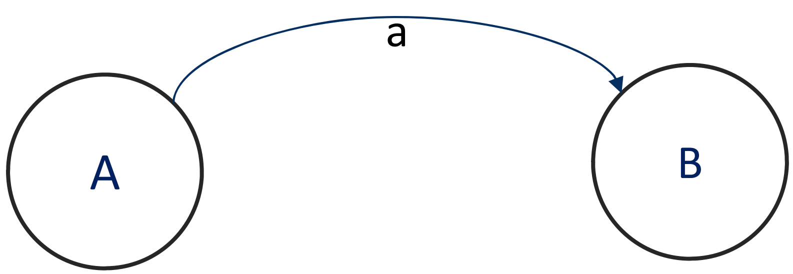
$$q1=S$$

$$F=\{qf\}$$

TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA DERECHA EN UN AF

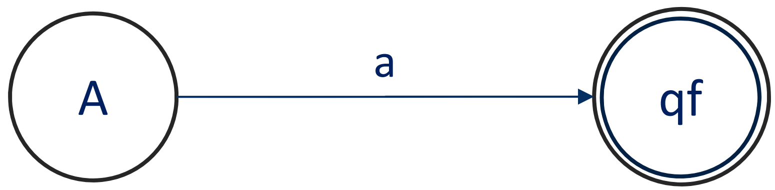
2. La función de transición δ se determina a partir de la forma de las reglas de producción P, de la manera siguiente:

a) Para reglas de la forma $A \rightarrow aB$ se obtiene $\delta(A,a)=B$.

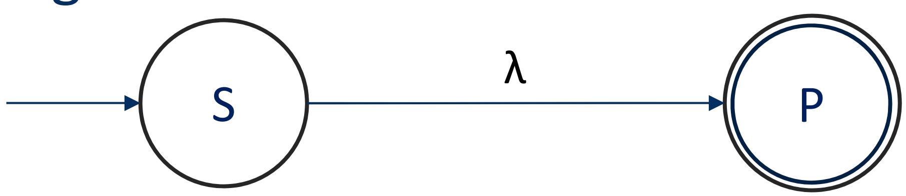


TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA DERECHA EN UN AF

b) Para reglas de la forma A → a se obtiene δ(A,a)=qf



c) Para reglas de la forma S → λ se obtiene δ(S, λ)=qf



TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA DERECHA EN UN AF

Ejemplo: Sea GLD = (N={A,S}, T={*a*,*b*}, P, S)

P: S →aS S →aA A →bA A →b

Obtener un AFND y otro AFD.

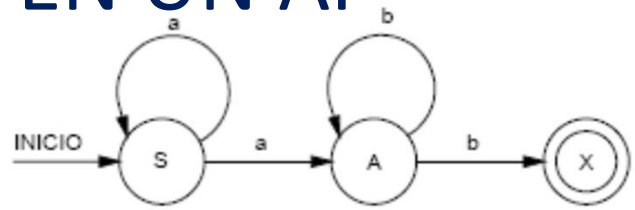
Solución : Se define el AFND, donde Te={*a*,*b*}, Q={A,S,X}, q1=S, F={X}, y *δ* viene dada por la tabla siguiente :

1. δ(S,a) = S
2. δ(S,a) = A
3. δ(A,b) =A
4. δ(A,b) =X

f	*a*	*b*
S	{S,A}	-
A	-	{A,X}
X	-	-

TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA DERECHA EN UN AF

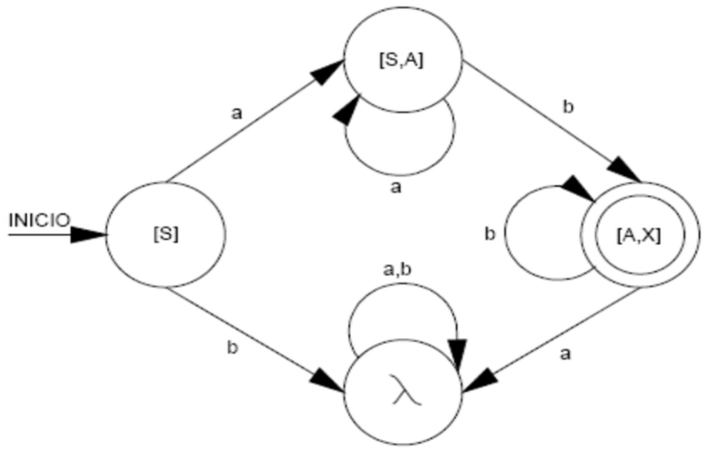
- El diagrama de estados es:

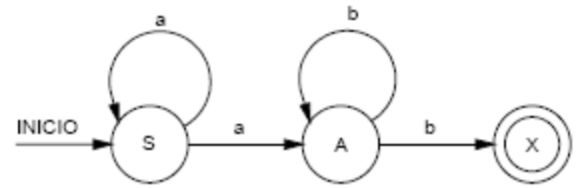


- El lenguaje que reconoce : a*ab*b
- El AFD=(Q’, Te , δ’, q1, F’) donde δ’:

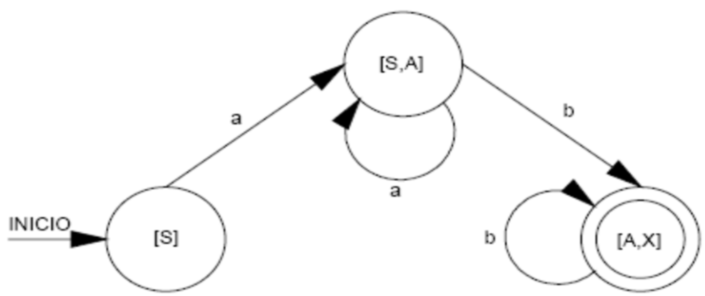
	f'	a	b
	λ	λ	λ
	[S]	[S,A]	λ
→	[A]	λ	[A,X]
→	[X]	λ	λ
	[S,A]	[S,A]	[A,X]
→	[S,X]	[S,A]	λ
	[A,X]	λ	[A,X]
→	[S,A,X]	[A,X]	[A,X]

TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA DERECHA EN UN AF





ER (AFND) = a*ab*b



ER (AFD) = aa*bb*

TRANSFORMACIÓN DE UN AF EN UNA GRAMATICA LINEAL POR LA DERECHA

Las reglas de producción P se determina a partir de la forma de función de transición δ, de la manera siguiente:

a) Para $\delta(A, a)=B$ se obtiene la regla de la forma $A \rightarrow aB$ se obtiene.

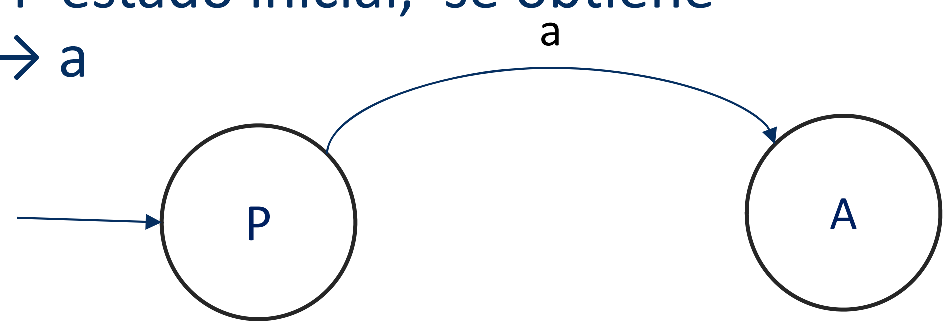
TRANSFORMACIÓN DE UN AF EN UNA GRAMATICA LINEAL POR LA DERECHA

b) Para δ(A, a)=qf siendo qf estado final se obtiene la regla de la forma A → a

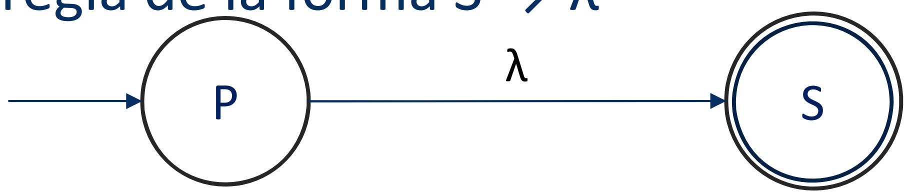
c) Para δ(S, λ)=qf siendo S estado inicial se obtiene la regla de la forma S → λ

TRANSFORMACIÓN DE UN AF EN UNA GRAMATICA LINEAL POR LA IZQUIERDA

b) Para $\delta(P, a)=A$, siendo P estado inicial, se obtiene las reglas de la forma $A \rightarrow a$



c) Para $\delta(P, \lambda)=S$ siendo P estado inicial y S estado final, se obtiene la regla de la forma $S \rightarrow \lambda$



TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA IZQUIERDA EN UN AF

1. El estado final del AF se corresponderá con el símbolo inicial de la gramática.
2. Agregar un estado inicial P que no se corresponda con ningún símbolo No terminal de la GLI

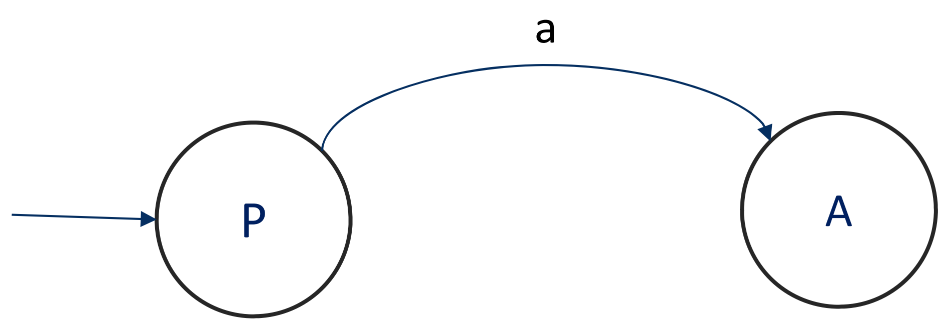
$$q1=P$$

$$F=\{S\}$$

TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA IZQUIERDA EN UN AF

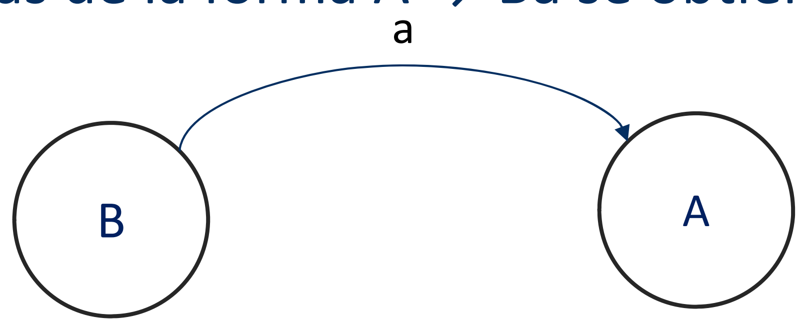
3. La función de transición δ se determina a partir de la forma de las reglas de producción P, de la manera siguiente:

a) Para reglas de la forma $A \rightarrow a$ se obtiene $\delta(P, a)=A$.

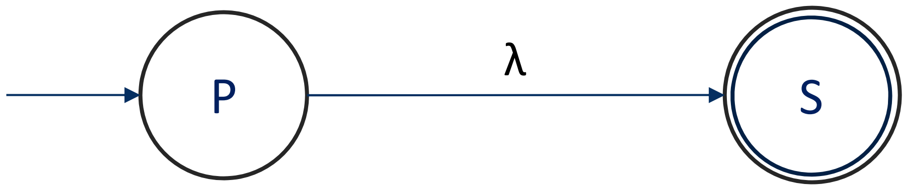


TRANSFORMACIÓN DE UNA GRAMÁTICA LINEAL POR LA IZQUIERDA EN UN AF

b) Para reglas de la forma A → Ba se obtiene δ(B, a)=A



c) Para reglas de la forma S → λ se obtiene δ(P, λ)=S



TRANSFORMACIÓN DE UN AF EN UNA GRAMATICA LINEAL POR LA IZQUIERDA

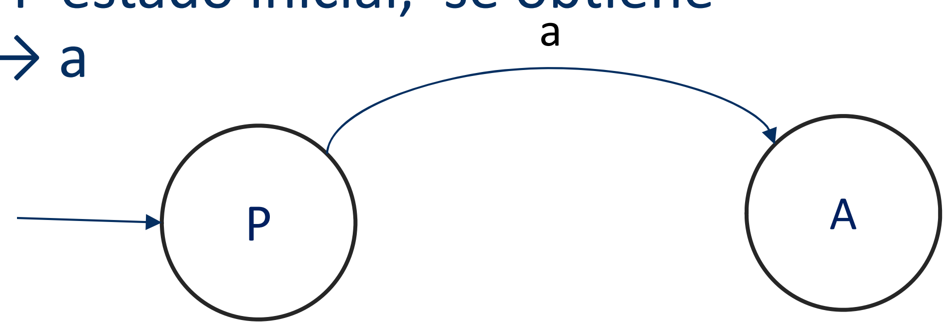
Las reglas de producción P se determina a partir de la forma de función de transición δ, de la manera siguiente:

a) Para $\delta(B, a)=A$ se obtiene la regla de la forma

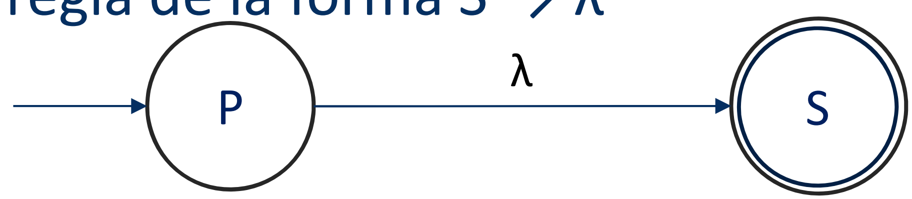
$$A \rightarrow Ba \text{ se obtiene.}$$

TRANSFORMACIÓN DE UN AF EN UNA GRAMATICA LINEAL POR LA IZQUIERDA

b) Para $\delta(P, a)=A$, siendo P estado inicial, se obtiene las reglas de la forma $A \rightarrow a$

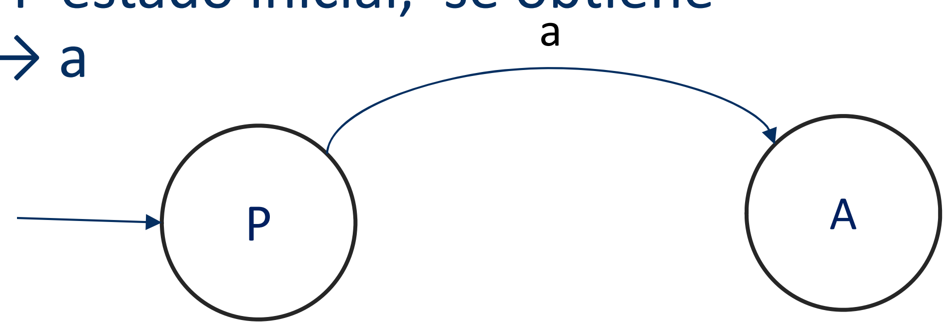


c) Para $\delta(P, \lambda)=S$ siendo P estado inicial y S estado final, se obtiene la regla de la forma $S \rightarrow \lambda$

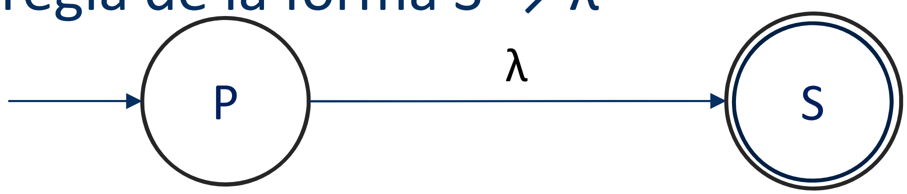


TRANSFORMACIÓN DE UN AF EN UNA GRAMATICA LINEAL POR LA IZQUIERDA

b) Para δ(P, a)=A, siendo P estado inicial, se obtiene las reglas de la forma A → a



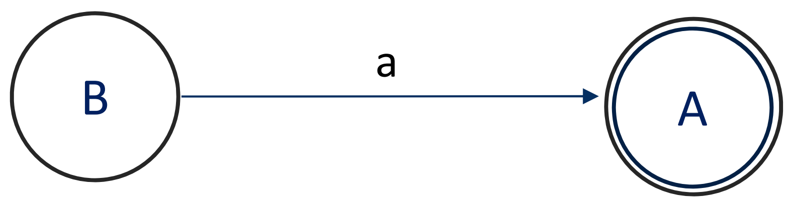
c) Para δ(P, λ)=S siendo P estado inicial y S estado final, se obtiene la regla de la forma S → λ



TRANSFORMACIÓN DE UN AF EN UNA GRAMATICA LINEAL POR LA IZQUIERDA

d) Si hay un estado final, se obtiene la regla de la forma $S \rightarrow Ba$

Si hay varios estados finales, se obtienen las reglas de la forma $A \rightarrow Ba$ y $S \rightarrow Ba$



CONCLUSIONES

- Los AF se pueden representar a través de diagramas de estados y/o matrices que representan a la función de transición δ.
- Los AF se clasifican en AFND y AFD. Es incierto hacer reconocimientos con AFND, por lo que es necesario transformarlo a AFD equivalente
- Se puede realizar la transformación entre GL y AF que reconocen lenguajes definidos por dichas gramáticas.

BIBLIOGRAFIA

- ALFONSECA Enrique, ALFONSECA Manuel y MORIYON Roberto**.** ***Teoría de Autómatas y Lenguajes Formales***. 2007. Madrid. Editorial Mc Graw Hill.
- http://dehesa.unex.es/bitstream/10662/2367/1/978-84-691-6345-0.pdf
- http://di002.edv.uniovi.es/~cueva/publicaciones/libros/36_LGA.pdf

RECURSOS GRAFICOS

- Pixabay
- Pexels
- Icon-Icons