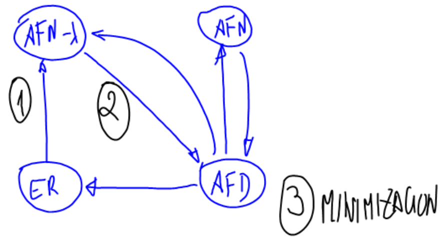
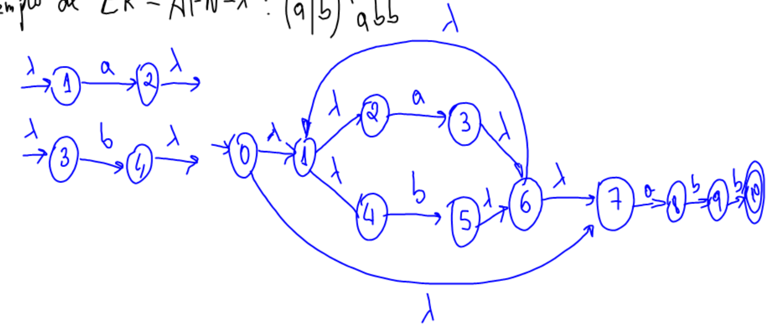
AUTOMATAS FINITOS Y EXPRESIONES REGULARES (ER)



Ejemplo de ER - AFN-λ : $(a|b)^*abb$

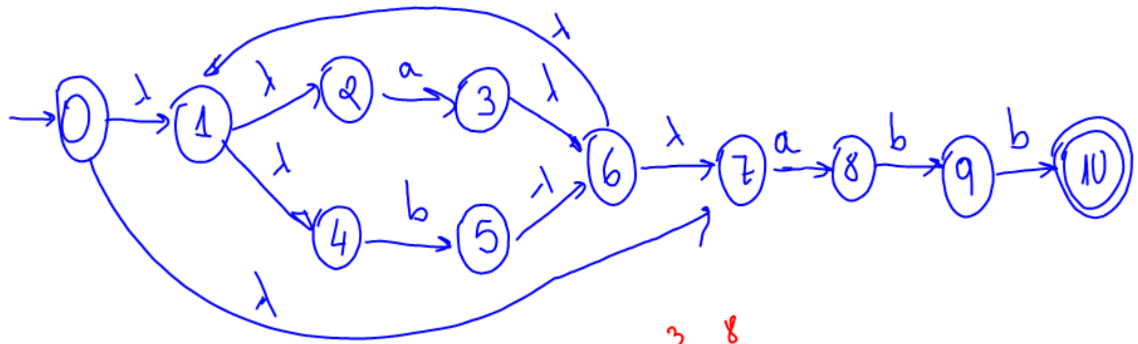


- Algoritmo para convertir un AFN en AFD.
  - Entrada: AFN N.
  - Salida: AFD D.
  - Método: Se construye una tabla de transiciones tranD para D, de manera que tranD simula "en paralelo" todos los posibles movimientos que se pueden dar en N ante una determinada cadena de entrada.
  - Además del cierre-λ, se utiliza la operación mueve(T, a), que dará como resultado un conjunto de estados del AFN hacia los cuales existe una transición con el símbolo a desde algún estado s ∈ T.
  - Tomamos 0 como el estado inicial del AFN.

  - Pseudocódigo:

```
construir estado U := cierre-λ(0)
añadir U a estadosD como estado no marcado
mientras haya un estado no marcado T en estadosD hacer
    marcar T
    para cada símbolo de entrada a hacer
        U := cierre-λ(mueve(T, a))
        si U no está en estadosD entonces
            añadir U a estadosD como estado no marcado
        tranD[T, a] := U
    fin
fin
```

abb



ESTADO DE INICIO: cierre-λ(0) = {0, 1, 2, 4, 7} = A (estado no marcado en estadosD)

a → 3, a → 8, b → 5

estadosD = {A*} marcamos T

Para cada símbolo de la entrada (a, b) hacer

U:: cierre-λ(mueve(T, a)) A*

mueve(A, a) = {3, 8}

cierre-λ(mueve A, a) = cierre-λ({3, 8}) = {1, 2, 3, 4, 6, 7, 8} = B

a → 3, b → 5, a → 8, b → 9

estadosD = {A*, B}

tranD[A, a] = B

U := cierre-λ(mueve A*, b) = cierre-λ({5}) = {1, 2, 4, 5, 6, 7} = C

a → 3, a → 8

estadosD = {A*, B, C}

tranD[A, b] = C

marcamos B $\Rightarrow$ estadosD $= \{A^*, B^*, C\}$

$U := \text{cierre-}\lambda(\text{mueve}(B,a)) = \text{cierre-}\lambda(\{3,8\}) = \{1,2,3,4,6,7,8\} = B$

$\text{tranD}[B,a] = B$

$U := \text{cierre-}\lambda(\text{mueve}(B,b)) = \text{cierre-}\lambda(\{5,9\}) = \{1,2,4,5,6,7,9\} = D$

estadosD $= \{A^*, B^*, C, D\}$

$\text{tranD}[B,b] = D$

marcamos C $\Rightarrow$ estadosD $= \{A^*, B^*, C^*, D\}$

$U := \text{cierre-}\lambda(\text{mueve}(C,a)) = \text{cierre-}\lambda\{3,8\} = B$

$\text{tranD}[C,a] = B$

$U := \text{cierre-}\lambda(\text{mueve}(C,b)) = \text{cierre-}\lambda\{5\} = C$

$\text{tranD}[C,b] = C$

marcamos D $\Rightarrow$ estadosD $= \{A^*, B^*, C^*, D^*\}$

$U := \text{cierre-}\lambda(\text{mueve}(D,a) = \{3,8\} = B$

$\text{tranD}[D,a] = B$

$U := \text{cierre-}\lambda(\text{mueve}(D,b) = \{5,10\} = \{1,2,4,5,6,7,10\} = E$

estadosD $= \{A^*, B^*, C^*, D^*, E\}$

$\text{tranD}[D,b] = E$

marcamos E $\Rightarrow$ estadosD $= \{A^*, B^*, C^*, D^*, E^*\}$

$U := \text{cierre-}\lambda(\text{mueve}\{E,a\}) = \text{cierre-}\lambda\{3,8\} = B$

$\text{tranD}[E,a] = B$

$U := \text{cierre-}\lambda(\text{mueve}\{E,b\}) = \text{cierre-}\lambda\{5\} = C$

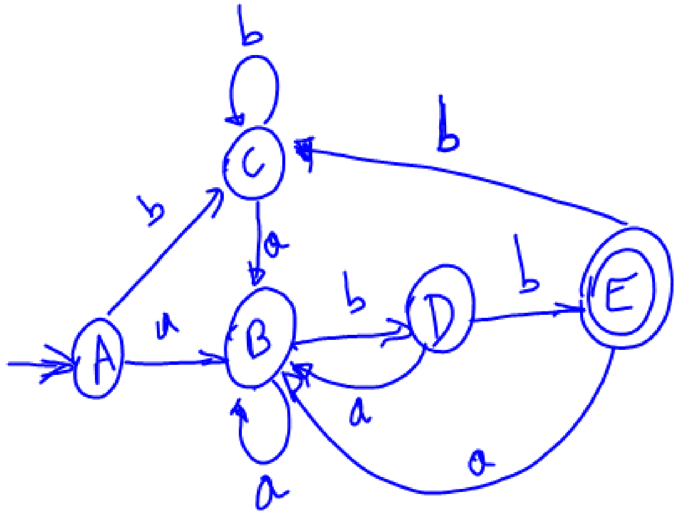
$\text{tranD}[E,b] = C$

Como todos los estados están marcados (estadosD = {A*, B*, C*, D*, E*}) termina el método

→ A será el estado de inicio

→ E será el estado final

	SIMBOLO ENTRADA	
ESTADOS	a	b
→ A	B	C
B	B	D
C	B	C
D	B	E*
E*	B	C



$(a|b)^*abb$ ⇒ lexemas válidos: abb ✓

aa ✗

babb ✓

bbabb ✓

- Minimización de estados de un AFD.
  - Todo conjunto regular es reconocido por un AFD con el mínimo número de estados que es único.
  - Algoritmo.
    1. Se construye una partición inicial P del conjunto de estados con dos grupos: Estados de aceptación F y estados de no-aceptación S-F.

1.- Tenemos dos grupos: F={E}   b E*

S-F={A,B,C,D}

2: Con la entrada a: Todos saltan a miembros del grupo S-F

Con la entrada b: Tenemos al estado D, que nos salta al grupo de estados finales

{E}

{A,B,C}

{A,B,C}   {D}

3:- Con la entrada a: Se queda todo igual

" " " b: Vemos que B salta a un miembro de otro grupo {D}

{E}

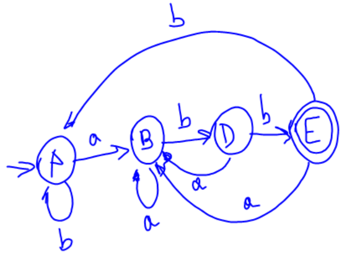
{A,C} ← Mirando la tabla A y C son idénticos

{B}

{D}

Quedaría:

ESTADOS	SIMBOLO ENTRADA a	b
A	B	A
B	B	D
D	B	E
E	B	A



Comprobamos los lexemas: abb ✓

aa ✗

ababb ✓

bbabb ✓