# Sintaxis y Semántica del Lenguaje

UTN – FRLP

# Clase 7 - Gramáticas

Gramáticas regulares a partir de un lenguaje

Convertir AF a gramática regular

Resolución de ejercicios TP 5

# Gramáticas Regulares

# Gramáticas Regulares

Son las más restringidas. Generan los lenguajes regulares. Reciben este nombre porque presentan ‘regularidades’ o repeticiones en la conformación de sus palabras.

**Características de una gramática regular:**

El lado derecho de la RP debe contener un símbolo terminal, y a lo sumo, un símbolo no terminal.

Los símbolos **terminales** se escriben en minúscula; los **no terminales** en mayúscula

# Gramáticas Regulares

A → a <A>::= a en BNF
A → aA <A>::= a <A> en BNF

Se puede incluir la producción S → ε si el lenguaje debe incluir la cadena vacía.

Lineal a derecha: A → aB o A → a ‘B aparece a derecha de a’

Lineal a izquierda: A → Ba o A → a

# Gramáticas Regulares - Ejemplos

Ejemplo 1: Sea L={w/ w= $ab^n$ , n >=1}   ∑={a,b}

*Cuando n=1*   $w_1=ab$

*Para n=2* $w_2=abb$

*Para n=3* $w_3=abbb$

# Gramáticas Regulares - Ejemplos

**Solución**

Es decir que las palabras empiezan con una sola ‘a’ seguida de una o más letras ‘b’.

$S \rightarrow aB$ la regla S debe decir que la palabra empieza con ‘a’, seguidas de letras ‘b’

$B \rightarrow b$ ahora la regla para B debe permitirme tener 1 sola ‘b’ o varias

$B \rightarrow bB$ entonces uso reglas recursivas

# Gramáticas Regulares - Ejemplos

Ejemplo 2:

Sea L={ w/ w termina en 1} sobre ∑={0,1}. La palabra más corta es $w_1$=1; $w_2$=01; $w_3$=11; $w_4$=001; $w_5$=101; …..

# Gramáticas Regulares - Ejemplos

**Solución**

$S \rightarrow 1$ la primera regla representa la palabra más corta
$S \rightarrow A1$ luego la regla inicial debe decir que las palabras terminan en '1'

Ahora hay que pensar que puede venir antes del '1' final: un solo '0', varios '0', un solo '1' o varios '1', o varios '0' y '1' mezclados.

$A \rightarrow 0$ permite que venga un solo '0' antes del '1' final.
$A \rightarrow A0$ permite que vengan varios '0' seguidos antes del '1' final.
$A \rightarrow 1$ permite que venga un solo 1 antes del '1' final.
$A \rightarrow A1$ permite que vengan varios 1 seguidos antes del '1' final.

# Gramáticas Regulares - Ejemplos

Ejemplo 3:

Sea L={ w/ w=$0^{2m}$, m>=1}  sobre ∑={0}
Para m=1 $w_1$=00; m=2 $w_2$=0000; m=3 $w_3$=000000, ……

# Gramáticas Regulares - Ejemplos

**Solución**

a) $S \rightarrow 0A$ la regla inicial S debe permitirme escribir palabras con varios 0 seguidos

b) $A \rightarrow 0$ esta regla me asegura obtener w1, la palabra más corta

Ahora debo pensar cómo escribir la regla siguiente para asegurarme que las palabras tengan un número par de letras ‘0’.

La regla recursiva $A \rightarrow 0A$ no me sirve porque no controla que sea cantidad par.

c) $A \rightarrow 0S$ hago una regla con recursión indirecta.

Partiendo de la regla a), desde S ya tengo un 0 seguido de A; si aplico la regla c),y reemplazo la A, tengo 00 seguido de S, y usando la regla a) de nuevo para reemplazar S tengo 000A. Si finalmente para reemplazar la A aplico la regla b), obtengo 0000 que es la palabra w2

# Gramáticas Regulares - Ejemplos

Ejemplo 4:

Sea L={ w/ w=$bc^n$, n>=0} sobre ∑={b,c}
Para n=0 $w_1$=b; n=1 $w_2$=bc; n=2 $w_3$=bcc; …….
Las palabras comienzan con una sola ‘b’, seguidas de cero o más letras ‘c’.

# Gramáticas Regulares - Ejemplos

**Solución**

La regla inicial S debe reflejar la palabra más corta w1=b

$S \rightarrow b$

$S \rightarrow bC$ la regla inicial debe permitirme generar el resto de las palabras que tienen una o varias letras ‘c’, después de la primer ‘b’

$C \rightarrow cC$ esta regla debe permitirme escribir varias ‘c’ seguidas

$C \rightarrow c$ caso base de la regla recursiva

# Ejercicios para resolver

Resolver los ejercicios del TP 5

Propuesto en clase:

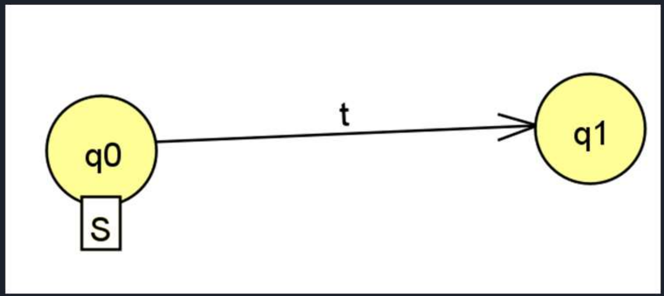
1.4) Escribir la gramática regular para el siguiente lenguaje
L={w/w empieza con 'aa'} sobre $\sum$={a,b}

# Conversión AF a Gramática regular

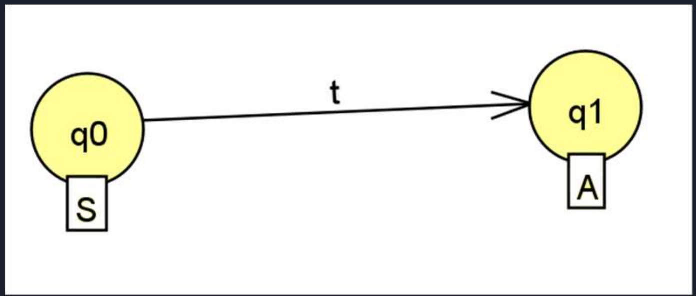
# Conversión AF - Gramática Regular

**Pasos**:

1- Asociar al estado inicial $q_0$ el símbolo S

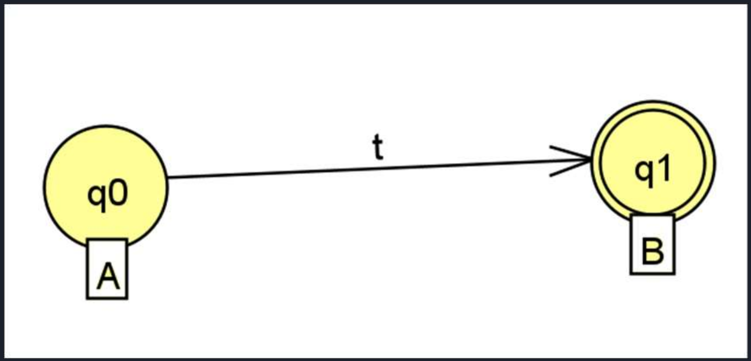



2- Asociar a cada estado $q_i$ restante del autómata un símbolo no terminal



# Conversión AF - Gramática Regular

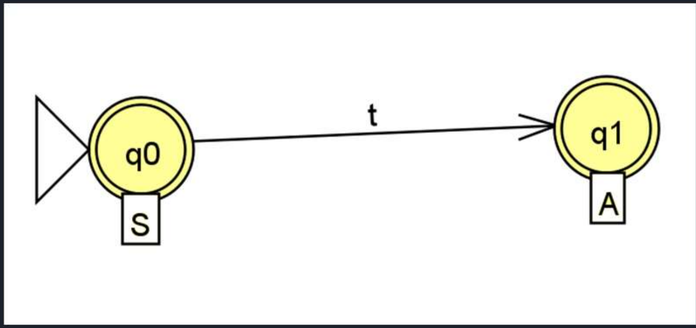
3- Para cada transición 't' que lleva del estado A al estado B, agregar a las reglas de producción, la regla $A \rightarrow tB$ siendo A y B los símbolos no terminales.



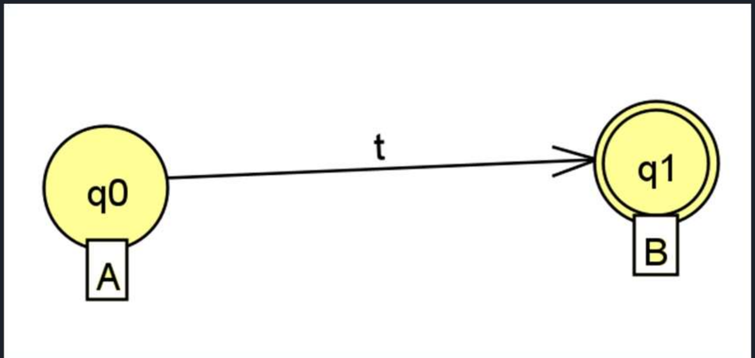

4- Si B es un estado final, agregar la regla de producción: $B \rightarrow \varepsilon$

# Conversión AF - Gramática Regular

5- Si el estado inicial también fuese final, agregar la regla : S → ε

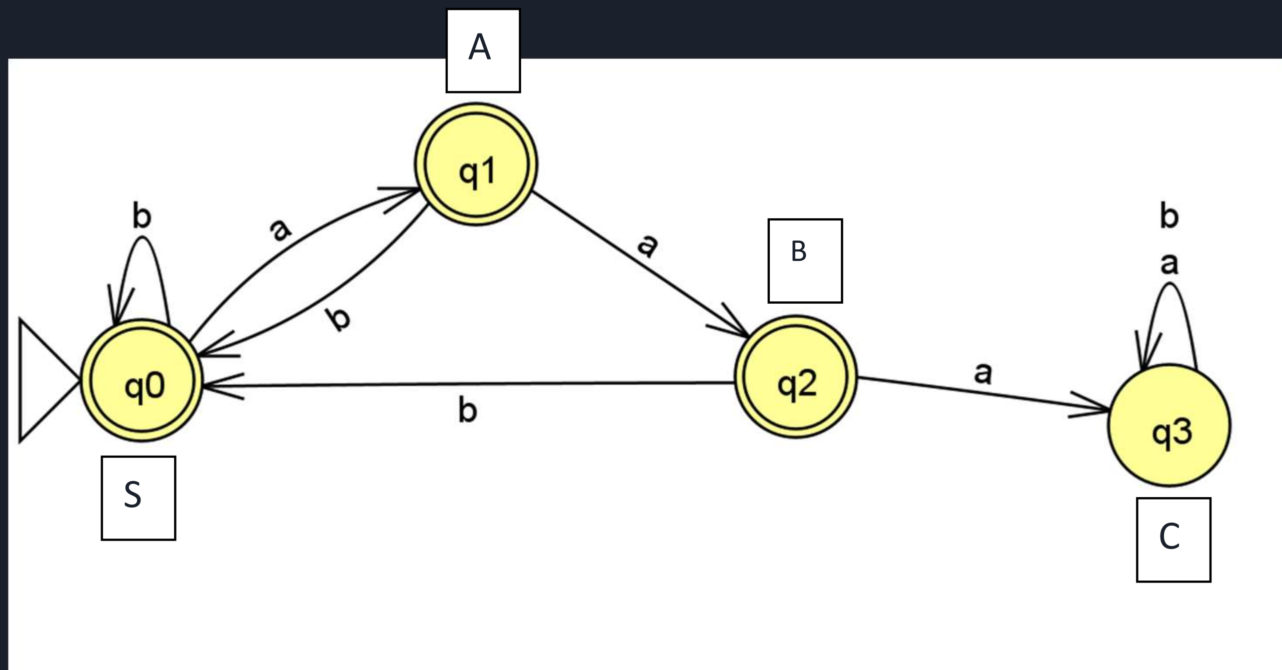



6- Si el estado B es final y hay una transición ‘t’ que lleva del estado A al B, se agrega la regla A → t



# AF - Gramática Regular. Ejemplos

Ejemplo 1



# AF - Gramática Regular. Ejemplos

Ejemplo 1:

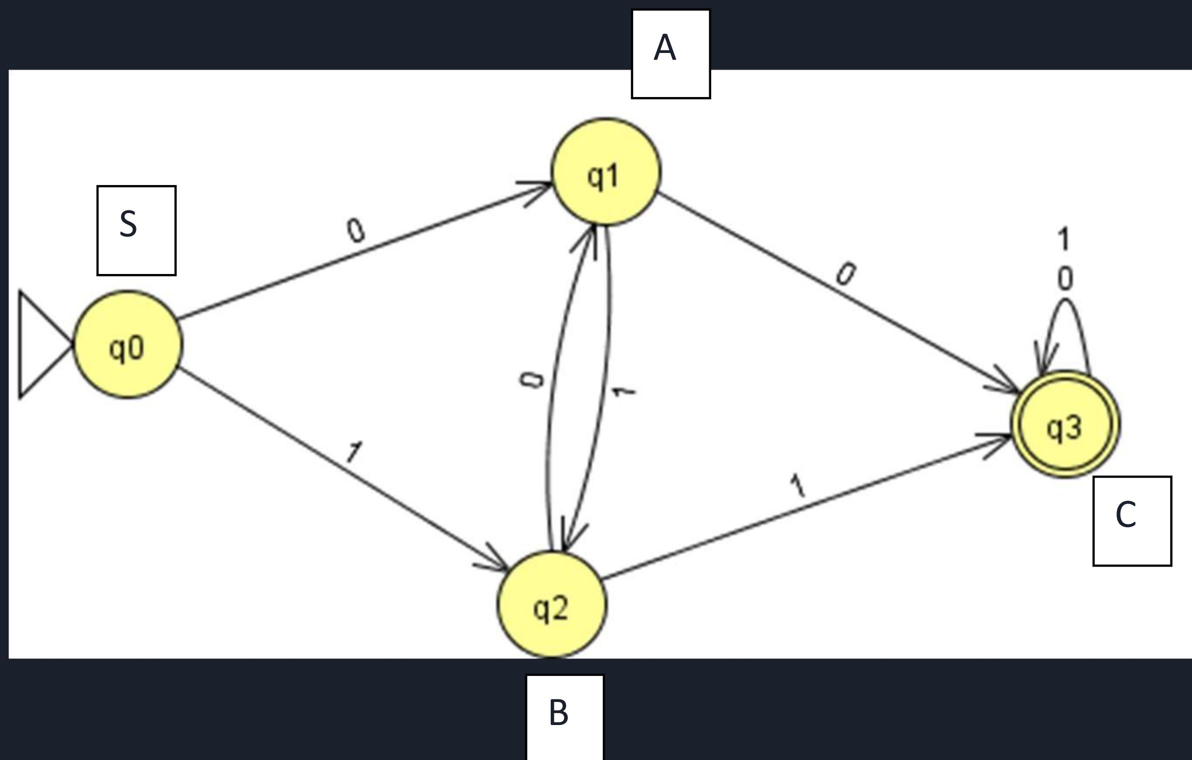
RP={

$S \rightarrow aA$ $S \rightarrow \varepsilon$ /* por ser S, A y B finales*/
$S \rightarrow bS$ $A \rightarrow \varepsilon$
$A \rightarrow aB$ $B \rightarrow \varepsilon$
$A \rightarrow bS$ $A \rightarrow a$ /*por llegar con 'a' al estado final B */
$B \rightarrow aC$ $A \rightarrow b$ /* por llegar con 'b' al estado final S*/
$B \rightarrow bS$ $S \rightarrow b$ /* por llegar con 'b' al estado final S (rulo)*/
$C \rightarrow aC$ $S \rightarrow a$ /* por llegar con 'a' al estado final A */
$C \rightarrow bC$ $B \rightarrow b$ } /* por llegar con 'b' al estado final S*/

Definición formal: ∑={a,b} G={ NT={A,B,C}, T={a,b}, RP, S}

# AF - Gramática Regular. Ejemplos

Ejemplo 2

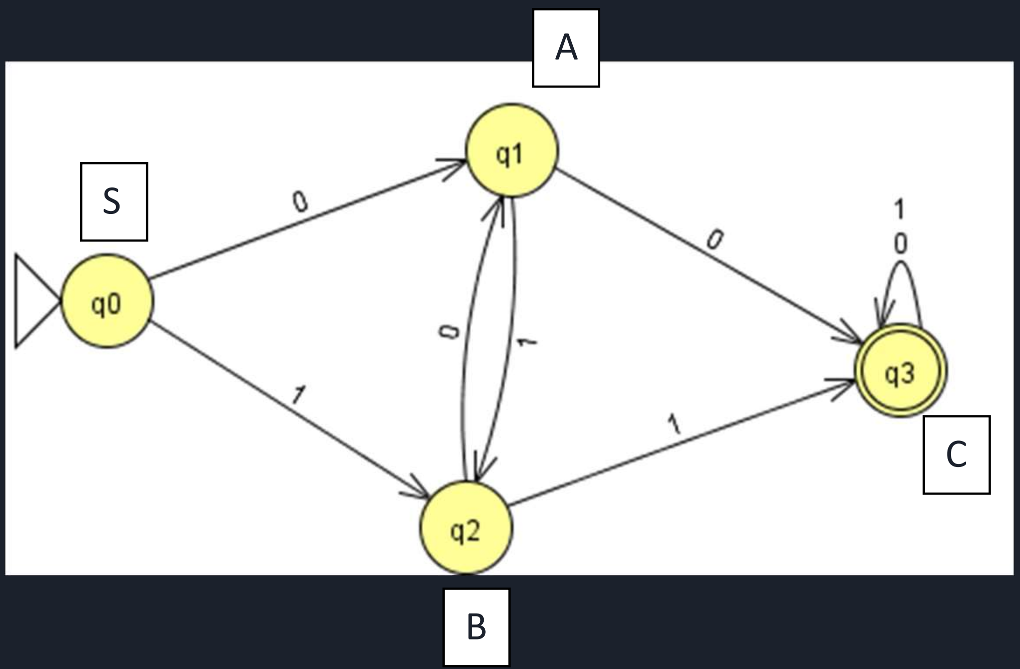


# AF - Gramática Regular. Ejemplos

Ejemplo 2:

RP={

| | |
|---|---|
| S → 0A | C → 0C |
| S → 1B | C → 1C |
| A → 0C | C → 0 |
| A → 1B | C → 1 |
| B → 1C | A → 0 |
| B → 0A | B → 1 |
| | C → ε } |



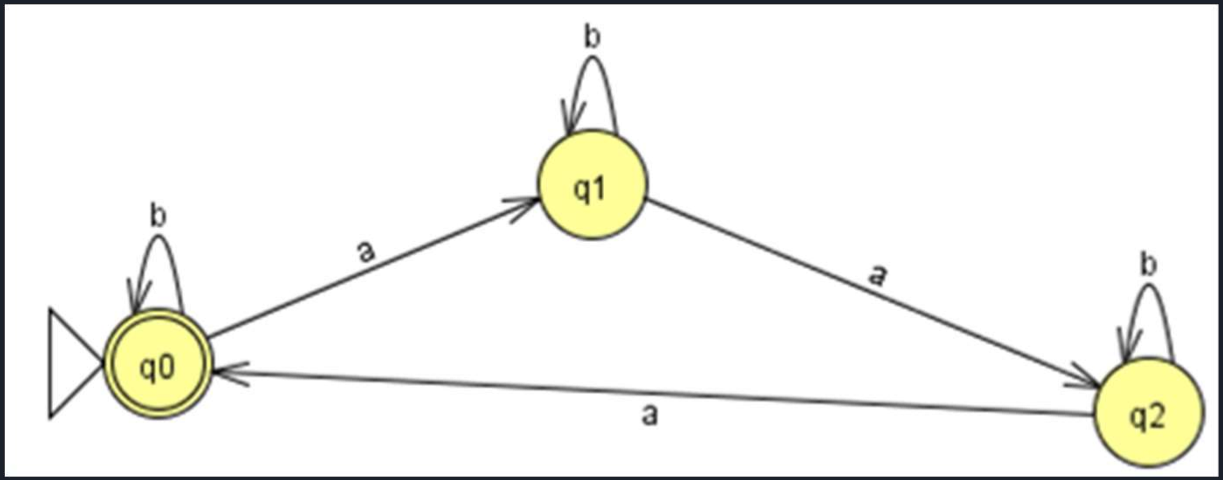

Definición formal: ∑={0,1} G={ NT={A,B,C}, T={0,1}, RP, S}

# Ejercicios para resolver

Resolver los ejercicios del TP 5

Propuesto en clase:

2.a) Escribir la gramatica del siguiente AF

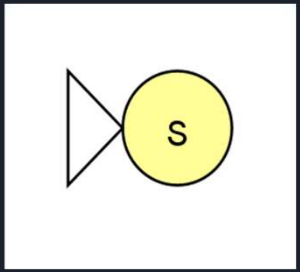


# Conversión Gramática regular a AF

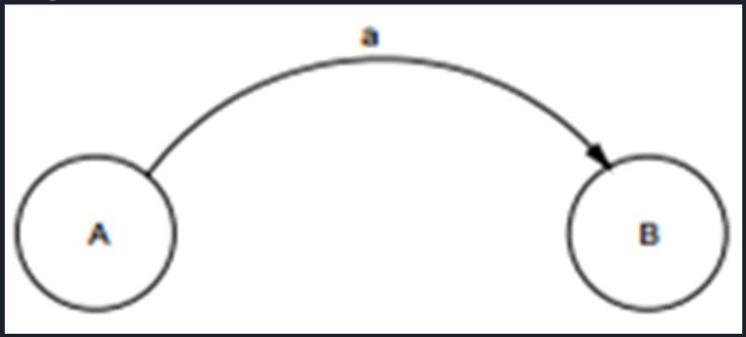
# Conversión Gramática Regular - AF

1- A cada símbolo **no terminal** se le asocia 1 estado (nodo) del autómata.

S es el símbolo inicial que será el nodo inicial del AF

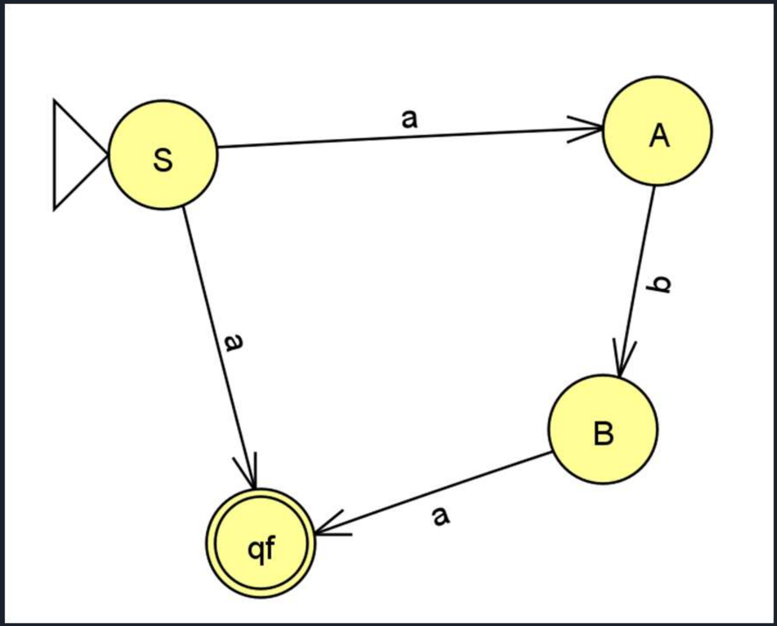



2- Si hay una regla de la forma: $A \rightarrow aB$ se la grafica:



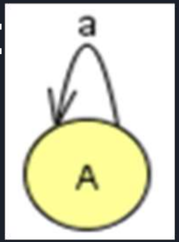
# Conversión Gramática Regular - AF

3- Se introduce el estado qf como **único** estado final

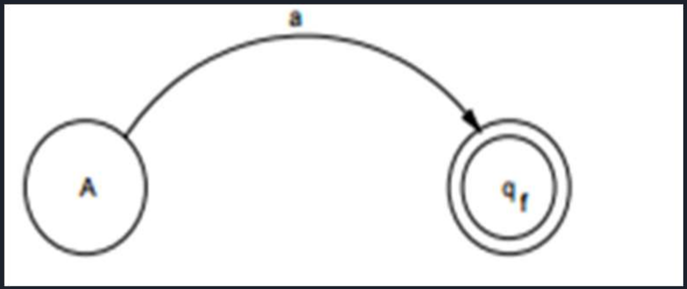


# Conversión Gramática Regular - AF

4- Si hay una regla de la forma:
$A \to aA$ se la grafica:




Si hay una regla de la forma:
$A \to a$ se la grafica




Donde qf es estado final y **no corresponde a ningún no terminal de la gramática.**

**El autómata que se obtiene puede ser un AFD o un AFN (en general son AFN)**

# Gramática Regular - AF. Ejemplos

Ejemplo 1

RP = {

$S \rightarrow aS$

$S \rightarrow aA$

$A \rightarrow bA$

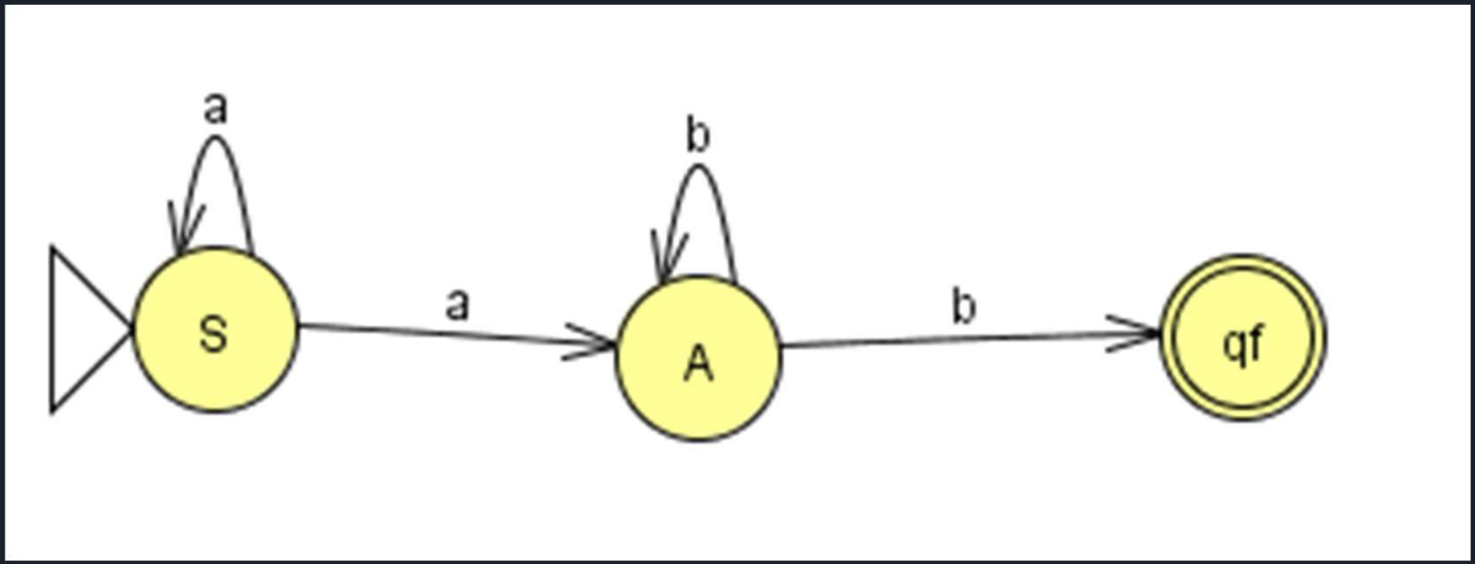
$A \rightarrow b$

}

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

# Gramática Regular - AF. Ejemplos

Solución: observar que el AF obtenido no es determinístico



# Gramática Regular - AF. Ejemplos

Ejemplo 2

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

RP = {

$A \rightarrow aA$

$A \rightarrow bB$

$B \rightarrow aA$

$B \rightarrow bC$

$C \rightarrow aC$

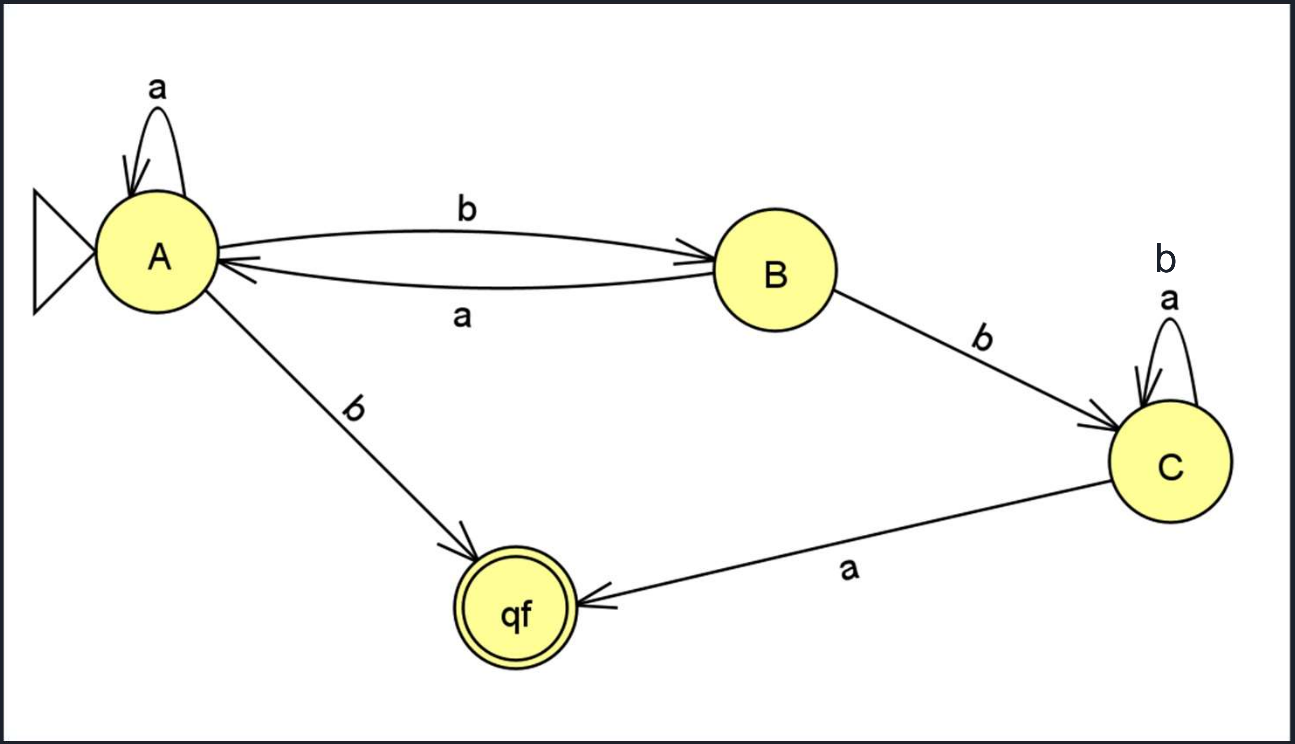
$C \rightarrow bC$

$C \rightarrow a$

$A \rightarrow b$ }

# Gramática Regular - AF. Ejemplos

Solución:  Observar que el AF obtenido es no determinístico



# Ejercicios para resolver

Resolver los ejercicios del TP 5

Propuesto en clase:
3.b) Diseñar el AF correspondiente a la siguiente gramática:

$S \rightarrow xX$

$S \rightarrow yY$

$X \rightarrow xY$

$X \rightarrow yZ$

$Y \rightarrow xZ$

$Y \rightarrow y$

$Z \rightarrow x$