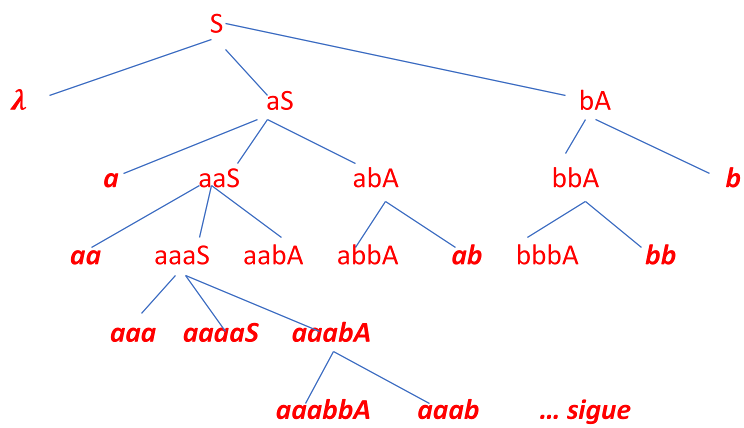
***GRAMATICAS Y AUTÓMATAS FINITOS***

Dada la gramática G = ({S;A}; {a;b}; P;S)

$$P = \begin{cases} S \;\rightarrow \lambda \,/\, aS \,/\, bA \\ \;A \;\; \rightarrow \;\; bA \,/\, \lambda \end{cases}$$

1) Clasificarla
   G es una gramática de ***tipo 3*** o sea ***regular***, ya que en las derivaciones de cada paso de las producciones, está $\lambda$, o un elemento del alfabeto no terminal junto a uno del alfabeto terminal, en este caso a derecha. Por eso se llama ***gramática regular a derecha.***

2) Hallar el lenguaje regular que genera G , L(G)
   Para hallar el lenguaje generado construiremos el árbol de derivación en base a las producciones.




   Puede verse Que el ***árbol de derivación*** nos da las ***hojas*** que son las palabras del lenguaje generado por la gramática G, o sea ***L(G).***
   También podemos ver que ***L(G)*** es ***un lenguaje infinito*** por lo tanto no podremos nombrarlo por extensión, entonces tendremos que dar su ***expresión regular,***o sea dar la forma y la estructura que tiene las palabras de L(G).
   Observando el árbol vemos que las palabras generadas son del tipo $a^n b^m$ con n ≥ 0, m ≥ 0 ya que hay palabras que solo tienen a ***a*** ó a ***b***,
   Recordemos que todo carácter $x^0 = \lambda$, por lo tanto, por ejemplo $aab^0 = aa$
   El lenguaje es entonces

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

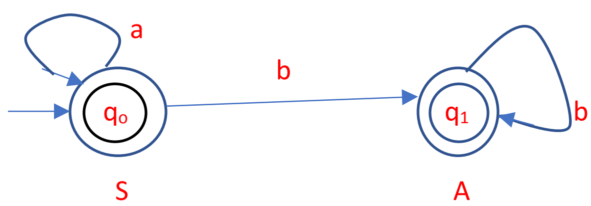
   que también puede expresarse como $L(G) = a^*b^*$

3) Hallar el Autómata Finito ***AF*** que reconoce dicho lenguaje
   Recordemos que un Autómata Finito es una máquina de estado que permite reconocer un lenguaje de tipo 3.
   Su estructura es $AF = (Q \,;\, V \,;\, \delta \,;\, q_0 \,;\, F)$
   Vamos a partir del conjunto de producciones de la gramática G, o sea :

$$P = \begin{cases} S \rightarrow \lambda \;/\; aS \;/\; bA \\ \quad A \rightarrow bA \;/\; \lambda \end{cases}$$

- Cada elemento del alfabeto $\boldsymbol{V_N}$ será un estado del ***AF***
- Las transiciones las haremos de acuerdo a las reglas de derivación en P.
- Si el elemento distinguido deriva en $\lambda$, entonces el estado inicial será también final.
- Siempre apelaremos a la observación, la intuición y la prueba y error.

Quedaría entonces:




***AF = ({$q_0$;$q_1$} ; {a;b} ; $\delta$ ; $q_0$ ; {$q_0$;$q_1$})***

La función de transición $\delta$ queda

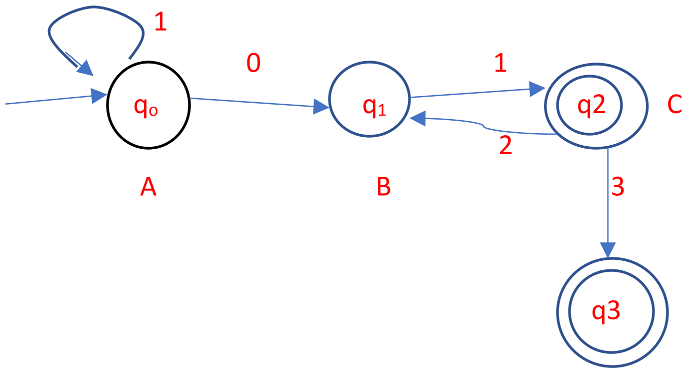
| $\boldsymbol{\delta}$ | ***a*** | ***b*** |
|---|---|---|
| $\boldsymbol{q_0}$ | $\boldsymbol{q_0}$ | $\boldsymbol{q_1}$ |
| $\boldsymbol{q_1}$ | - -- | $\boldsymbol{q_1}$ |

Podemos verificar que el ***AF*** reconoce las palabras del lenguaje ***L(G),*** y además ***AF*** es un autómata finito determinístico.

Ejercicio

Dado el AF = ({$q_0$;$q_1$,$q_2$, $q_3$};{0, 1, 2, 3}; $\delta$ ; $q_0$ ; {$q_2$, $q_3$})

Sea su diagrama de estados.




W = 01 ; w = 111013 ; w = 110121; w = 0121213

a) Definir la función de transición $\delta$

| $\delta$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $q_0$ | $q_1$ | $q_0$ | -- | -- |
| $q_1$ | -- | $q_2$ | -- | -- |
| $q_2$ | -- | -- | $q_1$ | $q_3$ |
| $q_3$ | --- | -- | -- | -- |

b) Indicar si es o no determinístico

El ***AF es determinístico*** ya que de un estado cualquiera a través de un caracter solo va a un estado.

En la función de transición no hay dos estados en ningún casillero, o sea como dice la definición:

$|\delta\ (q;a)| \leq 1\ ;\ \forall\ q \in Q\ ;\ \forall\ a \in V$

c) Indicar la gramática que genera el L(G) que reconoce AF

***G = ( {0, 1, 2, 3} ; {A, B, C} ; P ; A)***

$$P = \begin{cases} A \rightarrow 1A \,/\, 0B \\ B \rightarrow 1C \\ C \rightarrow 2B \,/\, 3 \end{cases}$$

d) Indicar el lenguaje generado por ***G*** dando su expresión regular.

$L(G) = 1^*0(12)^*1 \vee 1^*0(12)^*13$