Alfabetos

Un alfabeto es un conjunto finito no vacío, sin elementos que se obtengan por yuxtaposición, es decir que ningún simbolo se pueda formar colocando uno a continuación de otro. Se los denota con : Σ o V

Palabra: secuencia finita de letras formada por caracteres de un alfabeto.

Palabra Nula: está formada por 0 letras, se denota λ o ε

Longitud: cantidad de letras que forman una palabra.

Clausura de Kleene de un alfabeto

Se define:

$$V^* = V^0 \cup V^0 \cup V^1 \cup V^2 \cdots V^n \cup \cdots$$

V^i: conjunto de palabras de longitud i formadas con las letras del alfabeto V

V^* es el conjunto de todas las palabra de cualquier longitud, que se pueden escribir con letras del alfabeto V

Concatenación de palabras

Sea $w_1 \in V^*$, $w_2 \in V^*$ y $w_3 = w_1 \cdot w_2$ entonces w_3 está formada por las letras de w_1 a continuación por las letras de w_2.

$$long(w_1 \cdot w_2) = long(w_1) + long(w_2)$$

Además:

$$(w_1 \cdot w_2) \cdot w_3 = w_1 \cdot (w_2 \cdot w_3)$$

$(V^*; \cdot)$ es un semigrupo con neutro λ y se lo llama **Semigrupo libre** generado por V

Inversión, reflexión o transposición

Dada $w \in V^*$, si $w = x_1 x_2 x_3 \cdots x_{n-1} x_n$ se define:

$$w^R = x_n x_{n-1} \cdots x_3 x_2 x_1$$

Propiedades de la reflexión

1. $(w^R)^R = w$
2. $\lambda^R = \lambda$
3. $(w \cdot y)^R = y^R \cdot w^R$
4. $long(w^R) = long(w)$

Palíndrome

Sea $w \in V^*$, w es palindrome $\Leftrightarrow$ $w = w^R$

Potencia de una palabra

Sea $w \in V^*$, con $n \in \mathbb{N}_0$ se define:

$$\begin{cases} w^0 = \lambda \\ w^1 = w \\ w^n = w \cdot w^{n-1} \end{cases}$$

Propiedad: $long(w^n) = n \cdot long(w)$

Lenguaje

Sea L un conjunto y V un alfabeto.

Diremos que L es un lenguaje $\Leftrightarrow$ $L \subseteq V^*$

O sea un lenguajes es todo subconjunto de V^*

Observaciones

1. Los elementos de L son palabras.
2. Los lenguajes pueden ser finitos o infinitos.
3. Como $L \subseteq V^*$ entocnes $L \in P(V^*)$
4. Se pueden aplicar todas las propiedades y operaciones de conjuntos

Algunos lenguajes especiales son $L = \{\lambda\} = \Lambda$ se llama **Lenguaje nulo** y $L = \emptyset$ se llama **Lenguaje vacío**

Operaciones con lenguajes

Concatenacion de lenguajes

$L = L_1 \cdot L_2 = \{x \cdot y \ \ / \ \ x \in L_1 \quad \wedge \quad y \in L_2\}$

Son todas las palabras que se pueden formar concatenando cualquier palabra de L_1 con cualquier palabras de L_2

Propiedades

1. $|L_1 \cdot L_2| \leq |L_1| \cdot |L_2|$
2. $(P(V^*); \cdot)$ semigrupo con neutro Λ
3. $L_1 \cdot L_2 \neq L_2 \cdot L_1$
4. $L = \emptyset$ es elemento **absorbente** de la concatenación
5. Si $L_1 \subset L_2$ y $L_3 \subset L_4$ entonces $L_1 \cdot L_3 \subset L_2 \cdot L_4$

Lenguaje inverso, reflejo o traspuesto

$$L^R = \{w^R \ \ / \ \ w \in L\}$$

Es decir L^R tiene todas las palabras de L pero reflejadas.

Potencia de un lenguaje

Sea L un lenguaje, con $n \in \mathbb{N}_0$ se define:

$$\begin{cases} L^0 = \Lambda \\ L^1 = L \\ L^n = L \cdot L^{n-1} \end{cases}$$

Clausura de Kleene de un lenguaje

$$L^* = \bigcup_{n=0}^{\infty} L^n = L^0 \cup L^1 \cup L^2 \cdots \cup L^n \cup \cdots$$

O sea, en la clausura de un lenguaje están todas las palabras que se obtienen concatenando las de L cualquier cantidad de veces.

Observaciones:

1. $\Lambda^* = \Lambda$
2. $\varnothing^* = \Lambda$
3. $\forall L : \lambda \in L^*$

Clausura positiva de un lenguaje

$$L^+ = \bigcup_{n=1}^{\infty} L^n = L^1 \cup L^2 \cdots \cup L^n \cup \cdots$$

Observaciones:

1. $\Lambda^+ = \Lambda$
2. $\varnothing^+ = \varnothing$

Complemento de un lenguaje

$$\bar{L} = V^* - L$$

Ej:

Sea $V = \{a, b\}$

$L_1 = \{w \in V^* \ \ / \ \ w \ \ \text{comienza con} \ \ a\}$

Entonces el complemento de L_1 es:

$\bar{L}_1 = \{w \in V^* \ \ / \ \ w \ \ \ \text{no comienza con} \ \ a\}$

Gramaticas

Una gramatica es una cuaterna $G = (V_n; V_t; P; S)$ siendo:

V_n: Vocabulario o alfabeto de **no terminales**

V_t: Vocabulario o alfabeto de **terminales**

P: Producciones

S: simbolo variable incial

Requisitos

1. V_n y V_t son finitos.
2. $V_n \cap V_t = \emptyset$
3. P es finito y $P \subset (V^+ - V_t^*)$ siendo $V = V_n \cup V_t$
4. S pertenece a V_n

Observaciones

1) La gramática va a generar palabras formadas por las letras del alfabeto de **terminales** V_t. El otro alfabeto, V_n, contiene las variables o **no terminales** que se usan para ir formando las palabras.

2) Las producciones son reglas gramaticales. En vez de escribirlas en forma de par ordenado $(a; b)$, se escriben como $a \rightarrow b$ y se lee "a produce b". Ello significa que la parte “a” puede reemplazarse por “b”. Por ello, en la primera parte no puede haber terminales solas ni λ. Siempre al menos debe haber una variable para hacer el reemplazo.

3) El lenguaje generado por la gramática G se llama $L(G)$.

4) Puede haber varias gramáticas que generen un mismo lenguaje, pero el lenguaje que genera una gramática es único.

Ej:

Sea la gramatica $G = \{(\{S, X, Y\}, \{a, b, c\}, P, S)\}$ con el conjunto P :

$$S \rightarrow aSb + aX$$

$$X \rightarrow cX + bY$$

$$Y \rightarrow a$$

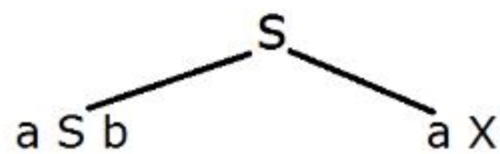
Para hallar palabras de $L(G)$ debemos ir aplicando sucesivas veces las producciones hasta lograr palabras, comenzando por alguna de las producciones que comienzan con el símbolo inicial S.

Árbol de derivación

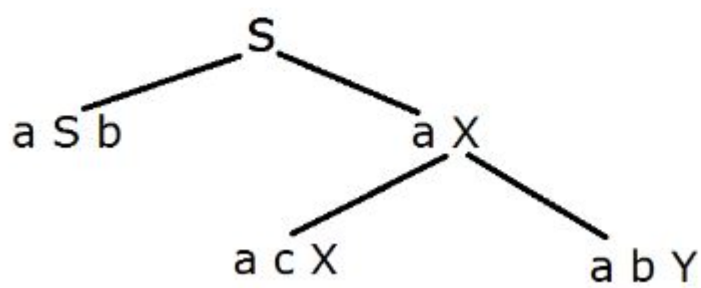
Es un arbol cuya raíz es el simbolo inicial, y cada vértice tiene tantos hijos como producciones diferentes existan que parten de dicho vértice. Es decir: Si $A \to B$, entonces A es padre de B. Si $A \to B + C$, entonces A tiene dos hijos: B y C. En este tipo de arboles, las hojas son las palabras del lenguaje, y cada rama nos da la derivación de dicha palabra.

Construiremos el arbol de derivación de la gramatica de arriba.

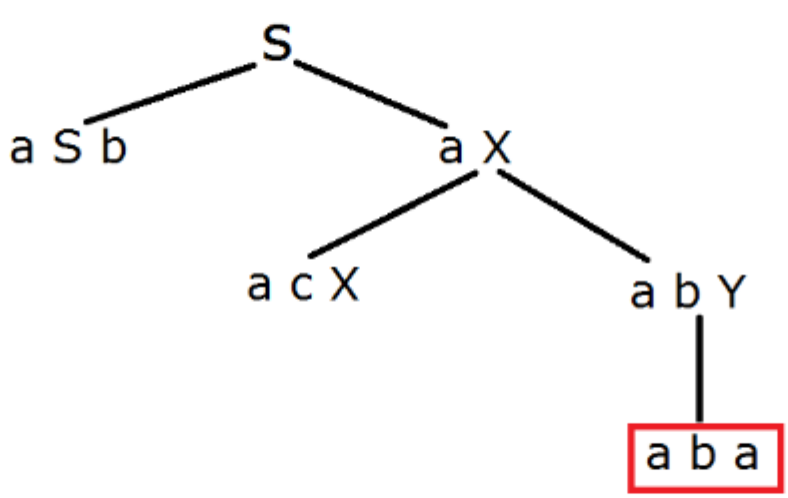
Comenzamos con las producciones de S



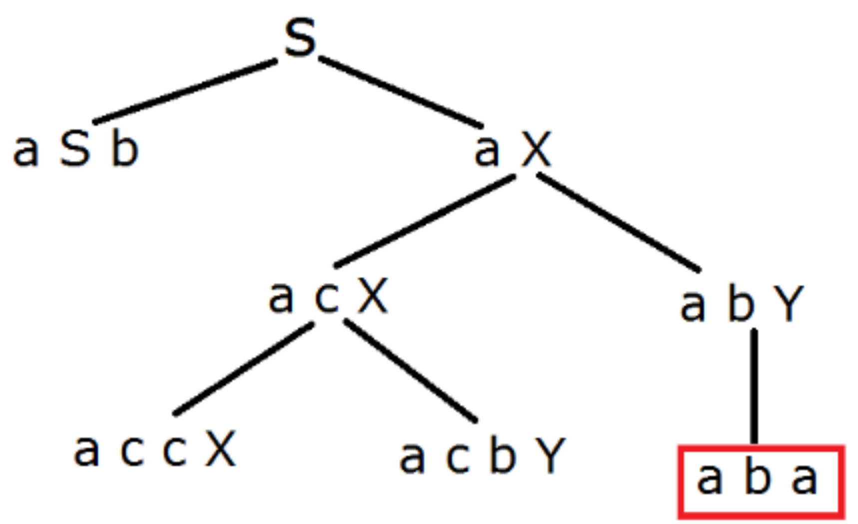
Ahora consideremos las producciones de X, con lo que podemos agregar:



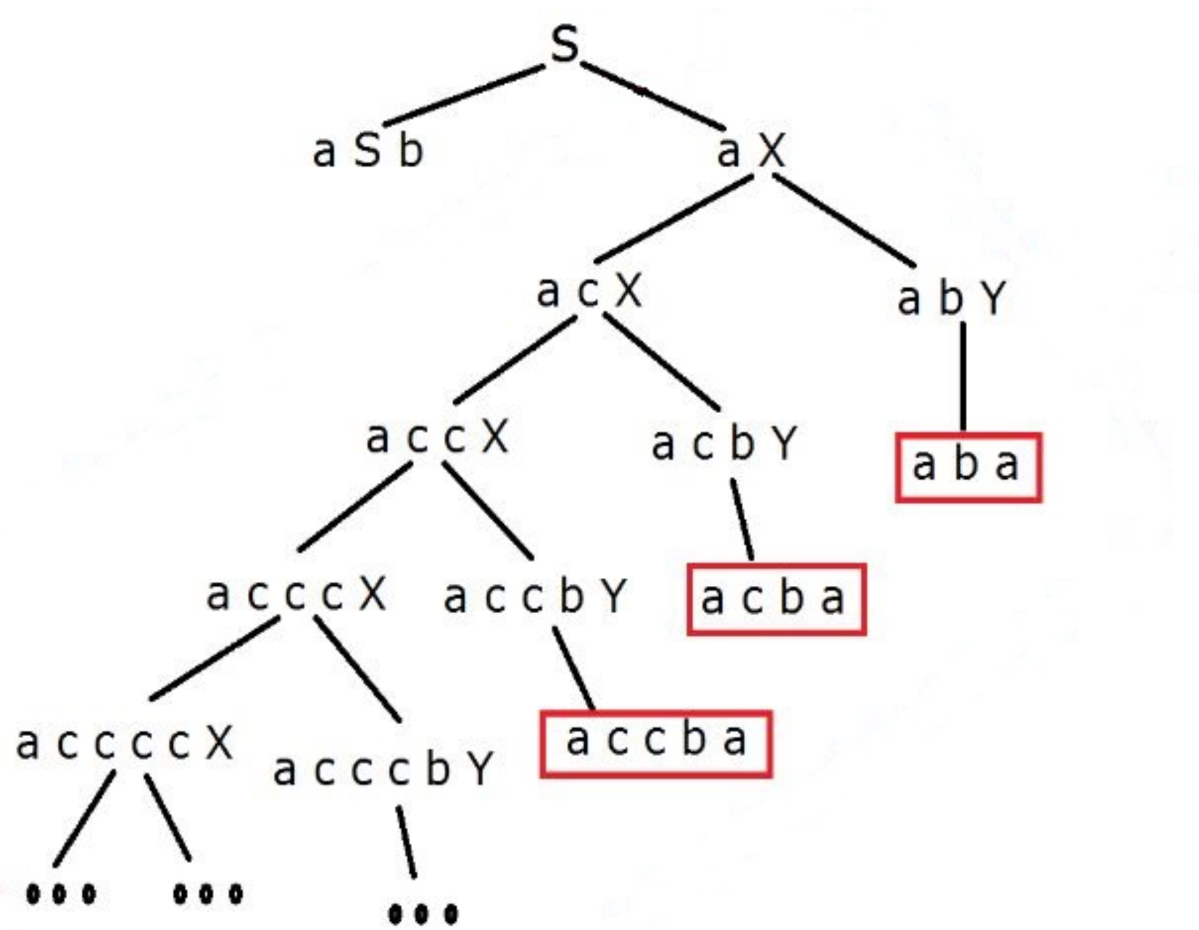
Si consideramos ahora la producción de Y, ya obtenemos una expresión formada solamente por terminales, es decir ua palabra del lenguaje, la recuadramos:



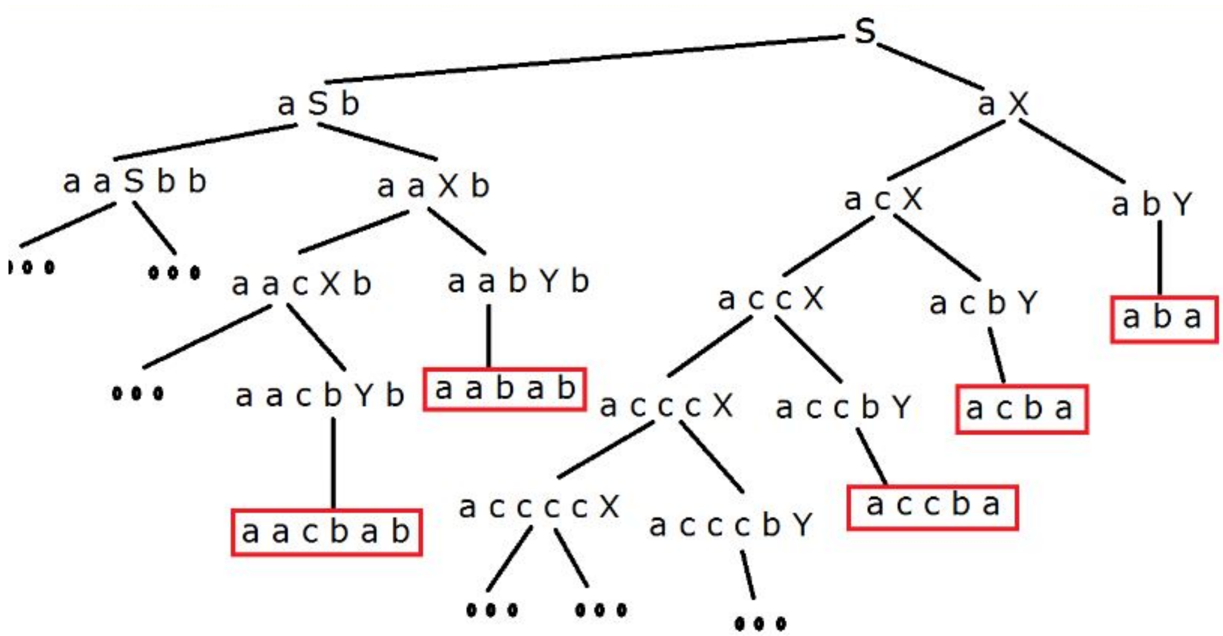
Pero debemos continuar con la otra rama, otra vez la variable X



Y seguimos de esta forma ahora con X y con Y:



Podemos ver que todo el subárbol derecho del árbol principal genera las palabras de la forma $ac^n ba$, con $n \geq 0$ pero aún falta el subárbol izquierdo, es decir el que tiene por raíz. aSb



Luego, por cada sub-árbol de mas a la izquierda, lo que agregan es una "a" mas a izquierda y una "b" a derecha simultáneamente. Por lo tanto el lenguaje que genera esta Gramática es:

$$L(G) = \{a^m ac^n bab^m \ \ / \ \ n \geq 0, m \geq 0\}$$

Dos gramaticas son **equivalentes** si generan el mismo Lenguaje.

Tipos de Gramáticas

Tipo	Nombre	Producciones
0	Irrestricta	Cualquier forma
1	Sensible al contexto	$aXb \to aYb$ donde $a, b, Y \in V^*, X \in V_n$
2	Independiente del contexto	$X \to Y$ donde $X \in V_n$
3	Regular	$X \to Y$ donde $X \in V_n, Y$ puede ser Vt, tV, t, λ

Se dice que un Lenguaje L es regular si existe una gramatica regular que lo genera.

Ej:

Sea $G = \{(A, B, C), \{0, 1, 2\}, P, A\}$

Tipo 0:

$$P = \begin{cases} A \to A2 + 11B \\ 1B \to 01 + 0C21 \\ C22 \to 2 + 2C \end{cases}$$

Tipo 1:

$$P = \begin{cases} A \rightarrow 0BC \\ 0B \rightarrow 011 + 0B1 \\ 1C \rightarrow 12 + 12C \end{cases}$$

Tipo 2:

$$P = \begin{cases} A \rightarrow BC \\ B \rightarrow 01 + 0B1 \\ C \rightarrow 2 + 2C \end{cases}$$

Tipo 3:

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

Expresiones regulares

Una $E.\ R.$ es una secuencia de elementos que verfica:

$$\lambda \text{ es } ER$$
$$a \in V \Rightarrow a \text{ es } ER$$
$$\text{Si } X, Y \text{ son } ER \Rightarrow X \cdot Y \text{ es } ER$$
$$\text{Si } X, Y \text{ son } ER \Rightarrow X + Y \text{ es } ER$$
$$\text{Si } X \text{ es } ER \Rightarrow X^* \text{ es } ER$$

O sea, las expresiones regulares sólo pueden contener letras del alfabeto, la palabra nula λ, concatenaciones $(\cdot)$, disyunciones $(+)$ y clausuras de Kleene $(*)$

Propiedad: Para cada Lenguaje regular, existe una expresión regular que lo define.

Ej:

El lenguaje regular: $L = \{1^n 0^m 2^{2p+1}(0\ \vee\ 1)\ /\ n \geq 0, m \geq 1, p \geq 0\}$ Se puede indicar con la ER:

$$1^* 0 0^* (22)^* 2(0\ +\ 1)$$

Automatas

Cada lenguaje tiene su propia maquina reconocedora del mismo.

Lenguaje Tipo	Maquina que lo reconoce
0	Maquina de Turing
1	Automata linealmente acotado
2	Automata de pila (Push Down)
3	Automata Finito

En esta asignatura solo estudiaremos los **Automatas Finitos**

Automatas Finitos

Un automata Finito es una 5-upla: (Q, V, δ, q_0, F) donde:

Q: Conjunto finito de estados.

V: Vocabulario o alfabeto de entrada.

δ: $Q \times V \to Q$ Función de transición.

q_0: Estado inicial.

F: Conjunto de estados finales $F \neq \emptyset$ y $F \subset Q$

Los Automatas Finitos se pueden representar con **Tablas de transición** o con **Diagramas de transición de estados**.

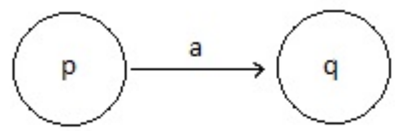
Un Automata acepta una palabra si y solo si al ir ingresando letra por letra desde el estado inicial llega a un estado final cuando termina la palabra.

Clasificación de los A.F.

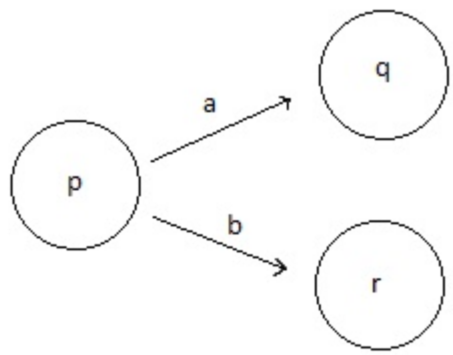
Los Automatas finitos pueden ser **Deterministicos** (A.F.D) o **No Deterministicos** (A.F.N)

Un Automata finito es **Deterministico** si no tiene transiciones por λ y δ cumple unicidad.

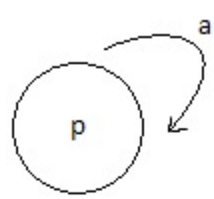
Metodo para obtener la E.R. a partir del A.F.



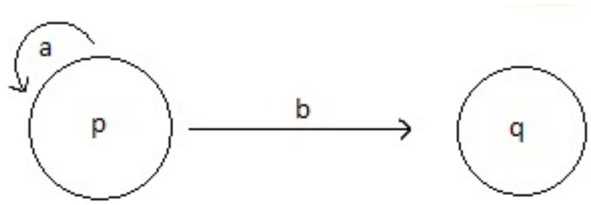
Se escribe $p = aq$



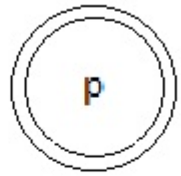
Se escribe: $p = aq + br$



Se escribe $p = a^*$



Se escribe: $p = a^*(bq)$



Se escribe $p = \lambda$

Obtencion de la G.R. a partir del A.F.

Sea $A = (Q, V, \delta, q_0, F)$ queremos hallar una Gramatica $G = (V_n, V_t, P, S)$ que genere el mismo lenguaje que es reconocido por el automata.

Los elementos de la gramatica se obtienen de la siguiente forma:

$V_n = Q$ (Los estados pasan a ser las variables)

$V_t = V$ (El alfabeto de terminales es el alfabeto de entrada del A.F)

$S = q_0$ (El símbolo inicial es el que era estado inicial)

Y las producciones P son tales que: $q \rightarrow a\delta(q, a)$ y $q \rightarrow \lambda$ si q es estado final.

Obtencion del A.F. a partir de la G.R.

Dada $G = (V_n, V_t, P, S)$ queremos hallar un automata $A = (Q, V, \delta, q_0, F)$ que reconozca el lenguaje generado por esta gramatica.

Los elementos del automata se obtienen de la siguiente forma:

$Q = V_n \cup \{f\}$ (Los estados son las variables más un nuevo estado que se agrega)

$V = V_t$ (El alfabeto de entrada es el alfabeto de terminales)

$q_0 = S$ (El estado inicial es el que era el símbolo inicial)

$F = \{q \in V_n \;\; / \;\; q \rightarrow \lambda\} \cup \{f\}$ (Los estados finales son todos los que producían la palabra unla y además el estado que se agrega)

Y la función de transición δ es tal que:

$$\begin{gathered}\delta(q, a) = p \;\; \text{si} \;\; q \rightarrow ap \\ \delta(q, a) = f \;\; \text{si} \;\; q \rightarrow a \\ \delta(t, a) = \varnothing\end{gathered}$$