<u>Alfabeto o Vocabulario</u>

<u>Lenguajes</u>

<u>Gramáticas</u>

<u>Tipos de Gramáticas</u>

<u>Autómatas</u>

¿Cómo nos comunicamos?

A través del Lenguaje

Naturales

y

Formales

El **lenguaje natural** es la lengua o idioma hablado o escrito por humanos para propósitos generales de comunicación.

- Ejemplos: Lenguaje de humanos: chino, español, inglés.

El **lenguaje formal** es un lenguaje cuyos símbolos primitivos y reglas para unir esos símbolos están formalmente especificados

- Ejemplos: Lenguajes de programación.

Lenguajes Formales

Un **alfabeto o vocabulario** es un conjunto no vacío y finito de símbolos.

Los símbolos del alfabeto son las **letras o caracteres.**

Suele indicarse con la letra V **$V = \{ a_1, a_2, a_3, \ldots\ldots., a_n \}$**

Ejemplos: V = { 0, 1 } alfabeto binario

V = { a,b,c,..., x, y, z} alfabeto romano

V = {1, 2, 3.....,9}

V = {a,b}

Palabra o Cadena Toda secuencia finita de letras de un alfabeto se denomina palabra o cadena.

Ejemplo: Sea V = { 0, 1 } un alfabeto

Son palabras w1 = 1001;   w2 = 00110

Palabra nula o vacía: palabra que carece de letras. Se simboliza con λ.

Longitud de una palabra: La longitud de la palabra o cadena es la cantidad de símbolos que la componen.

Ejemplo: Si w = 0011001 entonces long w = 7 se indica $|w| = 7$

Si w = λ entonces long w = 0 se indica $|w| = 0$

Clausura del vocabulario: Al conjunto de todas las palabras que se pueden construir con las letras de un alfabeto se lo indica **V***

- Tengamos en cuenta que si bien el alfabeto es finito, V* no lo es.
- La palabra nula siempre pertenece a V*, sin importar cuál es el alfabeto.

Clausura positiva del Alfabeto: Se representa V^+

$$V^+ = V^* - \{\lambda\}$$

no contiene la palabra vacía

.

Operaciones con Palabras

Concatenación o Producto: Si **x** $\in$ V* e **y** $\in$ V* son palabras, la concatenación **x.y** es una palabra formada por los símbolos de x seguidos por los símbolos de y.

Ejemplo: Sea V = { 0, 1 } el alfabeto y sean x = 0110 y y = 1011 palabras de V*
entonces x.y = 01101011

Observaciones

- La concatenación es una operación cerrada en V*
- La longitud de (x.y) = long v + long y
- La concatenación no es conmutativa, es decir x.y $\neq$ y.x
- La concatenación es asociativa, es decir (x.y).z = x.(y.z)
- Elemento neutro, la hilera nula λ es el neutro de la concatenación x . λ = λ . x= x

Propiedad: Si V es un alfabeto entonces el conjunto V* bajo la concatenación es un semigrupo con neutro.

Potenciación: Si concatenamos n veces una cadena w, es decir w.w.w……w n veces se obtiene w^n, siendo $w^0 = \lambda$

Ejemplo: Sea w = 0011
Si concatenamos 2 veces la palabra w obtenemos $w^2 = 00110011$
Si concatenamos 3 veces la palabra w obtenemos $w^3 = 001100110011$

Observación: long w^n = n long w

Inversión o Trasposición: Sea $w \in V^*$ formada por los símbolos $w = w_1w_2w_3.....w_r$ entonces la palabra inversa de w que se indica con w^R, se forma invirtiendo el orden de los símbolos del a palabra, $w^R = w_r.....w_3w_2w_1$

Ejemplo: Sea w = 10011 entonces $w^R = 11001$

Observaciones: $(w^R)^R = w$ $\lambda^R = \lambda$ $(w.z)^R = z^R.w^R$

Lenguajes

Sea V un alfabeto y V* el conjunto de todas las palabras que se pueden formar con los símbolos de V, llamamos lenguaje L a todo subconjunto de V*, es decir que $L \subseteq V^*$

Observaciones

- $L = \{ \lambda \}$ se llama lenguaje nulo y se indica Λ entonces $|\Lambda| = 1$
- $L = \varnothing$ se llama lenguaje vacío (no tiene palabras), entonces $|L| = 0$

Ejemplos de lenguajes: Sea $V = \{0, 1\}$

$L_1 = \{ 110, 0110, 01001 \}$

$L_2 = \{ \lambda, 0, 1, 00, 11, 01, 10 \}$

Operaciones con Lenguajes

Concatenación o Producto de 2 Lenguajes: Dados dos lenguajes $L_1 \subseteq V^*$ y $L_2 \subseteq V^*$, su concatenación $L_1.L_2$ contendrá todas las palabras que se puedan formar por la concatenación de una palabra de L_1 y otra de L_2.

$L = L_1.L_2 = \{ w_1.w_2 / w_1 \in L_1 \wedge w_2 \in L_2 \}$

Ejemplo: Dados $L_1 = \{ 01, 001 \}$ $L_2 = \{ 1, 01 \}$ y $L_3 = \{ \lambda, 0 \}$

$L = L_1.L_2 = \{ 011, 0101, 0011, 00101 \}$

$L = L_2.L_3 = \{ 1, 10, 01, 010 \}$

Observaciones:

- Si L_1 y L_2 son lenguajes finitos entonces $|L_1.L_2| \leq |L_1||L_2|$
- La concatenación no es conmutativa, es decir $L_1.L_2 \neq L_2.L_1$
- La concatenación es asociativa, es decir $(L_1.L_2).L_3 = L_1.(L_2.L_3)$
- Λ es elemento neutro de la concatenación $\Lambda .L = L. \Lambda = L$

Potencia: La potencia i-ésima de un lenguaje corresponde a la concatenación i veces del lenguaje en él mismo.
$L^i = L.L.L.....L$ i veces

Ejemplo: Si L = {0, 1} entonces L^2 = { 00, 01, 10, 1 }

Observaciones

- Si i = 0 obtenemos el lenguaje nulo $\Lambda = \{ \lambda \}$
- Si i = 1 obtenemos el mismo lenguaje L

Reflexión, Inversión o Trasposición: La reflexión de un lenguaje L está formada por la aplicación de la reflexión a cada una de las palabras del lenguaje
$L^R = \{ x^R / x \in L \}$

Ejemplo: Sea L = { 0, 11, 01 } entonces L^R = { 0, 11, 10 }

Clausura de Kleene

Sea V un alfabeto, n $\in \mathbb{N} \cup \{ 0 \}$ y L un lenguaje de V*, entonces

$$L^* = \bigcup_{n=0}^{\infty} L^n = L^0 \cup L \cup L^2 \cup L^3 \cup \ldots\ldots \cup L^n \cup \ldots\ldots$$

Observaciones

- Si el lenguaje L es el lenguaje nulo $\Lambda = \{ \lambda \}$ entonces $\Lambda^* = \{ \lambda \} = \Lambda$

$$\Lambda^* = \bigcup_{n=0}^{\infty} \Lambda^n = \Lambda^0 \cup \Lambda \cup \ldots\ldots \cup \Lambda^n \cup \ldots\ldots =$$

$$= \{\lambda\}^0 \cup \{\lambda\} \cup \{\lambda\}^2 \cup \ldots\ldots \cup \{\lambda\}^n \cup \ldots\ldots = \{\lambda\}$$

- Si el lenguaje L es el lenguaje vacío entonces $\varnothing^* = \{ \lambda \} = \Lambda$

$$\varnothing^* = \bigcup_{n=0}^{\infty} \varnothing^n = \varnothing^0 \cup \varnothing \cup \varnothing^2 \cup \ldots\ldots \cup \varnothing^n \cup \ldots\ldots =$$

$$= \{ \lambda \} \cup \varnothing \cup \varnothing \cup \ldots\ldots \cup \varnothing \cup \ldots\ldots = \{ \lambda \}$$

Clausura Positiva

$\varnothing^{+} = \varnothing$

Sea V un alfabeto, n $\in \mathbb{N}$ y L un lenguaje de V*, entonces

$$L^{+} = \bigcup_{n=1}^{\infty} L^{n} = L \cup L^{2} \cup L^{3} \cup \ldots\ldots \cup L^{n} \cup \ldots\ldots$$

Observaciones

- Si el lenguaje L es el lenguaje nulo Λ = { λ } entonces $\Lambda^{+} = \{\lambda\} = \Lambda$

$$\Lambda^{+} = \bigcup_{n=1}^{\infty} \Lambda^{n} = \Lambda \cup \ldots\ldots \cup \Lambda^{n} \cup \ldots\ldots =$$

$$= \{\lambda\} \cup \{\lambda\}^{2} \cup \ldots\ldots \cup \{\lambda\}^{n} \cup \ldots\ldots = \{\lambda\}$$

- Si el lenguaje L es el lenguaje vacío entonces

$$\varnothing^{+} = \bigcup_{n=1}^{\infty} \varnothing^{n} = \varnothing \cup \varnothing^{2} \cup \ldots\ldots \cup \varnothing^{n} \cup \ldots\ldots =$$

$$= \varnothing \cup \varnothing \cup \ldots\ldots \cup \varnothing \cup \ldots\ldots = \varnothing$$

Gramáticas

La gramática es una estructura que tiene la pasibilidad de generar palabras que forman un lenguaje o sea es un sistema generador de un lenguaje. Son descripciones de las sentencias de los lenguajes

Definición formal: Se define a la Gramática G como la cuádrupla: G = (Vn; Vt ; P ; s) donde:

Vn es un conjunto de elementos llamados no terminales, suele llamarse vocabulario de elementos no terminales (se usan para describir).
Vt es un conjunto de elementos llamados terminales, suele llamarse vocabulario de elementos terminales (se usan para formar).
P es un conjunto de producciones (reglas de sustitución).
s es un elemento de Vn llamado símbolo inicial

Observaciones

- Los conjuntos Vt y Vn son finitos.
- $Vn \cap Vt = \varnothing$
- El conjunto de producciones es finito.
- Si (x; y) $\in$ P escribimos $x \rightarrow y$, decimos que es una producción de la gramática G
- En la producción $x \rightarrow y$, x no puede ser la palabra nula.
- Una gramática es un conjunto finito de reglas que generan un lenguaje.
- Al lenguaje generado por la gramática G se lo indica L(G).
- . Se llama derivación al proceso de generar palabras usando una gramática
- Gramáticas que generan el mismo lenguaje son equivalentes.

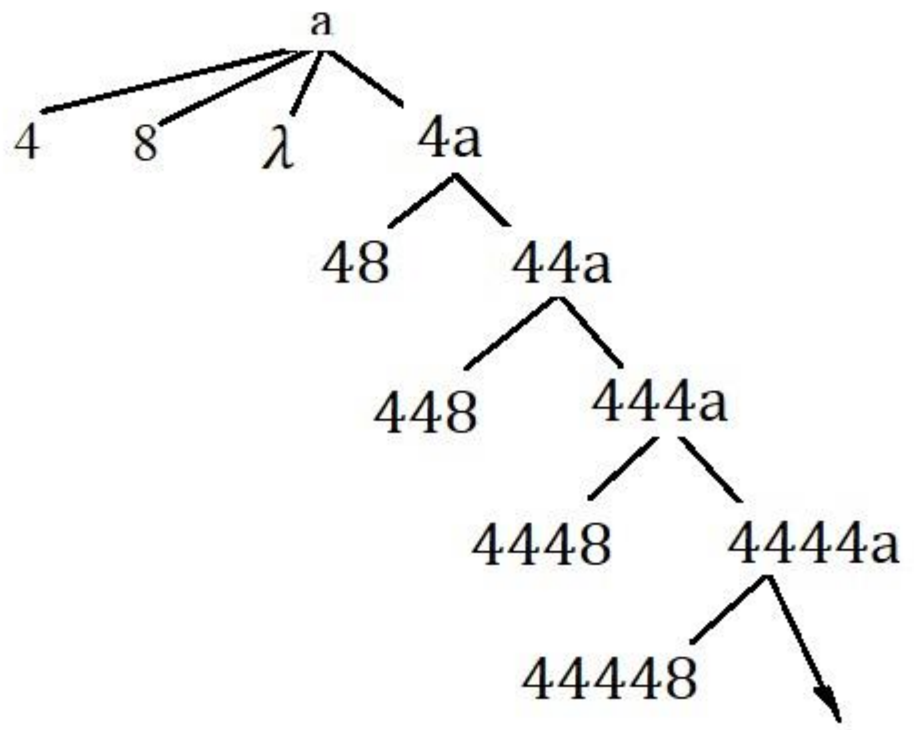
Ejemplo: Sea G = ({ s, a}; (4, 8); P; s) siendo

P: $\begin{cases} s \to 4/8/\lambda/4a \\ a \to 8 \,/\, 4a \end{cases}$

Una palabra del lenguaje L(G) generado por la gramática G es

$s \to 4a \to 44a \to 444a \to 4448$

Una forma práctica de representar las derivaciones son los árboles de derivación donde la raíz es el símbolo inicial.



$$L(G)=\left\{w\in V^{*}/\, w=\lambda \vee w=4 \vee w=4^{n}8,\ \ con\ n\geq 0\right\}$$

Gramática de Tipo 0 o Sin Restricciones

Es la gramática G = (Vn; Vt; P; s) que para cualquier producción $x \rightarrow y$, en la parte izquierda tiene que haber al menos un símbolo no terminal, respecto de la parte derecha no hay ningún tipo de restricción.

$x \in \{V_n \cup V_t\}^+$ e $y \in \{V_n \cup V_t\}^*$

Ejemplo: G = ({ t, s }; { 0, 1, 2 }; P; s) con P dada por:

$$P: \begin{cases} s \rightarrow 0s1 \\ s1 \rightarrow 1t \\ 01t \rightarrow 2 \end{cases}$$

Gramáticas de Tipo 1 o Sensibles al Contexto

Es la gramática $G = (V_n; V_t; P; s)$ que para cualquier producción $x \rightarrow y$, la longitud de x es menor o igual a la longitud de y, es decir long $x \leq$ long de y.

$x \in \{V_n \cup V_t\}^+$ e $y \in \{V_n \cup V_t\}^*$

No genera la palabra nula.

Ejemplo: $G = (\{ a, b, S \}; \{ 0, 1, 2 \}; P; s)$ con P dada por

$$P : \begin{cases} s \rightarrow 012 \,/\, 0a12 \\ a1 \rightarrow 1a \\ a2 \rightarrow b122 \\ 1b \rightarrow b1 \\ 0b \rightarrow 00 \,/\, 00a \end{cases}$$

Gramáticas de Tipo 2 o Independiente del Contexto

Es la gramática G = (V_n; V_t; P; s) que para cualquier producción $x \rightarrow y$, en la parte izquierda puede tener un solo símbolo no terminal y en la parte derecha tiene uno o más símbolos terminales o no terminales.

$x \in V_n$ e $y \in \{V_n \cup V_t\}^*$

Ejemplo: G = ({ z, y, S }; { 0, 1 }; P; s) con P dada por

$$P : \begin{cases} s \rightarrow zy \,/\, yz \\ z \rightarrow 1 \\ y \rightarrow 0 \end{cases}$$

Gramáticas de Tipo 3 o Regulares

Es la gramática $G = (V_n; V_t; P; s)$ que para cualquier producción $x \rightarrow y$, la parte izquierda es un único símbolo no terminal y la parte derecha es la concatenación de 2 símbolos siendo uno de ellos terminal o es un único símbolo terminal o la palabra nula.

Ejemplo: $G = (\{ z, y, S \}; \{ 0, 1 \}; P; s)$ con P dada por:

$$P: \begin{cases} s \rightarrow 1y \,/\, 1z \\ y \rightarrow 0 \\ z \rightarrow 0 \,/\, 1 \end{cases}$$

otra podría ser

$$P: \begin{cases} s \rightarrow 1z \,/\, 1y \\ y \rightarrow 0y \,/\, \lambda \\ z \rightarrow \lambda \,/\, 0 \end{cases}$$

Observaciones

- Si en las producciones $x \to y$, y es concatenación de un elemento no terminal con uno terminal, la gramática se dice regular a derecha.
- Si en las producciones $x \to y$, y es concatenación de un elemento terminal con uno no terminal, la gramática se dice regular a izquierda.

Ejemplos

1) La gramática $G = (\{ s, t, z \}, \{ 0, 1 \}, P; s)$ con P dada por:

$$P: \begin{cases} s \to t1 \,/\, z1 \\ t \to t0 \,/\, z \\ z \to 0 \end{cases}$$

es regular a derecha

2) La gramática $G = (\{ s, t, z \}, \{ 0, 1 \}, P; s)$ con P dada por:

$$P: \begin{cases} s \to 1t \,/\, 1z \\ t \to 0t \,/\, z \\ z \to 0 \end{cases}$$

es regular a izquierda

Lenguaje Regular

Un lenguaje es regular si existe una gramática regular que lo genere.

Un lenguaje regular sobre un alfabeto V se define recursivamente de la siguiente forma:

- $\varnothing$ es un lenguaje regular.
- $\{ \lambda \}$ es un lenguaje regular.
- Si V es un alfabeto y $a \in V$ entonces $\{ a \}$ es un lenguaje regular.
- Si L_1 y L_2 son dos lenguajes regulares entonces:

  $L_1 \cup L_2$, L_1L_2, L_1^* son lenguajes regulares.

Ejemplos: Sea el alfabeto $V = \{ 0, 1 \}$

1) $L = \{ 1^n 0^m / n \geq 1 y\ m \geq 1 \}$ lenguaje que contiene palabras que comienzan con una secuencia de unos y terminan con una secuencia de ceros.

2) $L = \{ 1 (0 1)^n / n \geq 0 \}$ lenguaje que contiene palabras que comienzan con un uno seguido por la secuencia 01 repetida cualquier número de veces o ninguna.

Gramática Tipo 0

- ✓Sin restricciones
- ✓Genera lenguajes recursivamente numerables

Gramática Tipo I

- ✓Dependientes del contexto (se tiene en cuenta lo que viene antes y después del símbolo que se sustituye).
- ✓Genera lenguajes sensibles (o dependientes) al contexto
- ✓Las producciones son del tipo: x → y donde x, y Є {V_n U V_t}
- ✓Nunca genera a la palabra nula

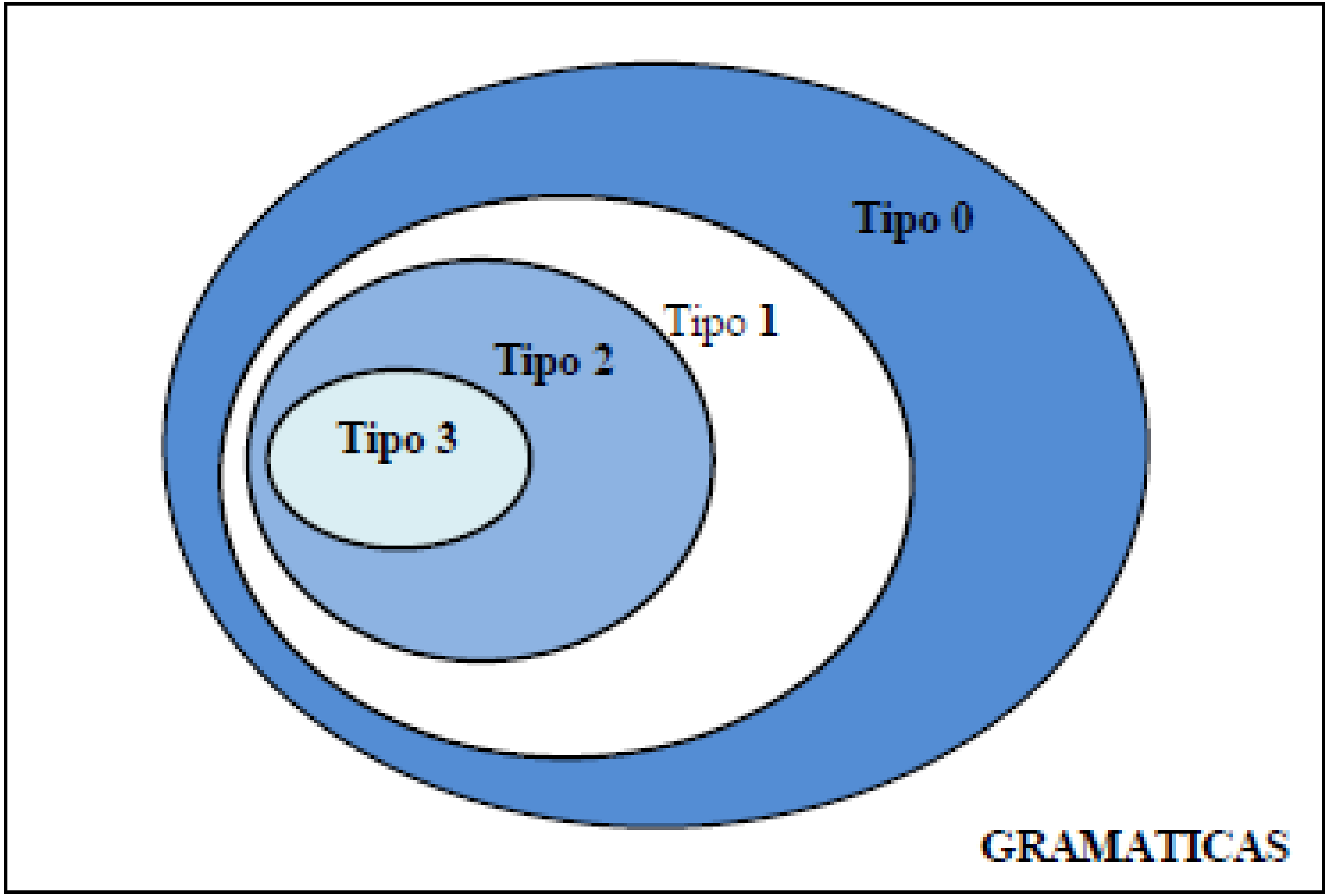
≤

Gramática Tipo II

- ✓Libres de contexto
- ✓Genera lenguajes libres al contexto
- ✓El lado izquierdo debe consistir en un sólo no terminal
- ✓No hay restricciones al lado derecho

Gramática Tipo III

- ✓Regulares
- ✓Genera lenguajes regulares
- ✓El lado izquierdo debe consistir en un sólo no terminal
- ✓El lado derecho debe ser un terminal seguido de un no terminal, o un sólo terminal o la cadena vacía


Tipo 0
Tipo 1
Tipo 2
Tipo 3
GRAMATICAS

AUTÓMATAS

Los autómatas son entes o máquinas abstractas que prueban la pertenencia o no de cada cadena de símbolos sobre un vocabulario dado a un cierto lenguaje. En el área de los intérpretes, compiladores, traductores y procesadores, es de notable importancia la simulación de procesos encargados del tratamiento de la información.

La información se codifica en cadenas de símbolos y un autómata es una máquina formal, es decir sin componentes físicas, que manipula cadenas de símbolos que lee en su entrada generando cadenas de símbolos en su salida.

AUTÓMATAS, LENGUAJES Y GRAMÁTICAS. LA JERARQUÍA DE CHOMSKY

Dado que las gramáticas proporcionan las reglas utilizadas en la generación de las cadenas de un lenguaje, se puede establecer una conexión entre las clases de lenguajes generados por ciertos tipos de gramáticas y las clases de lenguajes reconocibles por ciertas máquinas.

Tipo de Autómata	Lenguaje que procesa	Gramática de que genera
Autómatas finitos	Lenguajes regulares	Gramáticas regulares (Tipo 3)
Autómatas de Pila	Lenguajes libres de contexto	Gramáticas libres de contextos (Tipo 2)
Autómatas Linealmente acotados	Lenguajes dependientes del contexto	Gramáticas dependientes del contexto (Tipo 1)
Máquina de Turing	Lenguajes estructurados por frases	Gramáticas sin restricciones (Tipo 0)

Autómatas Finitos

Un autómata finito o máquina de estado finito es una herramienta que se utiliza para reconocer un determinado lenguaje regular.

Es un modelo matemático de un sistema que recibe una palabra y determina si pertenece o no al lenguaje que reconoce.

Cada autómata finito reconoce un único lenguaje regular.

Para un lenguaje regular puede haber muchos autómatas que lo reconozcan.

Definición

El autómata finito se representa con una 5-upla $A = (Q; V; \delta; q_0; F)$ donde:

- $Q = \{ q_0, q_1, q_2, ..., q_n \}$ es el conjunto finito de *estados* de la máquina.
- V es el alfabeto de entrada.
- $\delta: Q \times V \rightarrow Q$ es la *relación de transición*.
- $q_0 \in Q$ y se dice *estado inicial*.
- $\emptyset \neq F \subseteq Q$ es el conjunto de *estados finales*.

Tabla de transición: Sirve para representar la función transición

δ	Vocabulario de entrada. Elementos de V
Estados. Elementos de Q	Estados siguientes. Elementos de Q

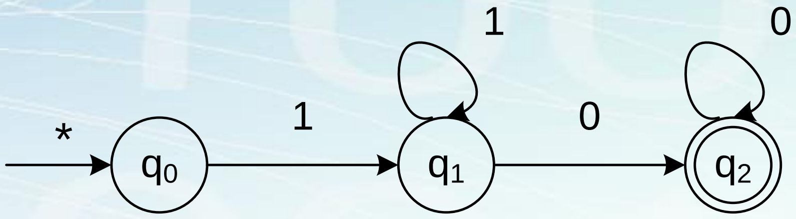
Diagrama:

Una manera de representar un autómata finito es mediante un diagrama (dígrafo) que permite visualizar el funcionamiento.

- ✓ Hay un nodo para cada estado de Q.
- ✓ El nodo correspondiente al estado inicial q_0, tendrá una flecha sin origen.
- ✓ Los nodos correspondientes a los estados de aceptación están marcados con un doble círculo. Los que no pertenecen a F tienen un círculo simple.

Ejemplo: $A = \{ \{ q_0, q_1, q_2 \}, \{ 0, 1 \}, \delta, q_0, \{ q_2 \} \}$

δ	0	1
q_0	-	q_1
q_1	q_2	q_1
q_2	q_2	-



$L = \{ 11^n 00^m \ / \ n \geq 0, m \geq 0 \}$

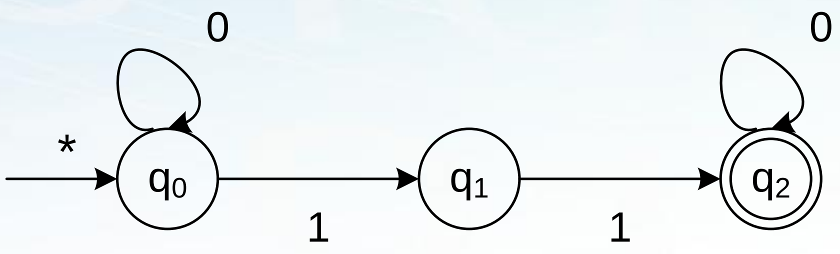
E.R. = 11* 00*

Autómatas Finitos Determinísticos (A.F.D.)

- Ninguna arista está etiquetada con λ.
- La relación de transición es una función.
- Un A.F.D. es completo si cada estado tiene una transición por cada letra del alfabeto.
- $\forall\, q \in Q, \forall\, a \in V \Rightarrow |\, \delta\,(q; a)\,| \leq 1$

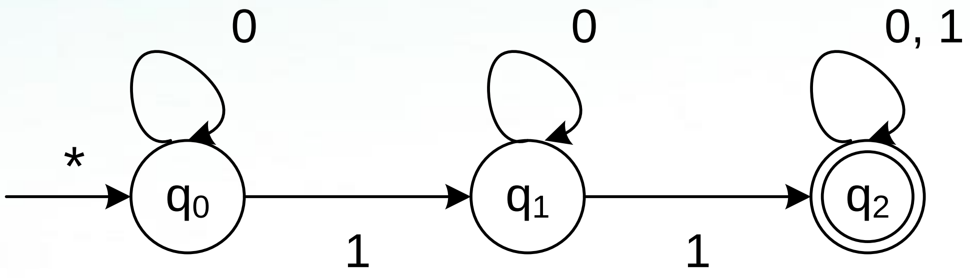
Ejemplos:

E.R= 0* 110*



A.F.D.

E.R= 0* 10* 1 (0 $\vee$ 1)*



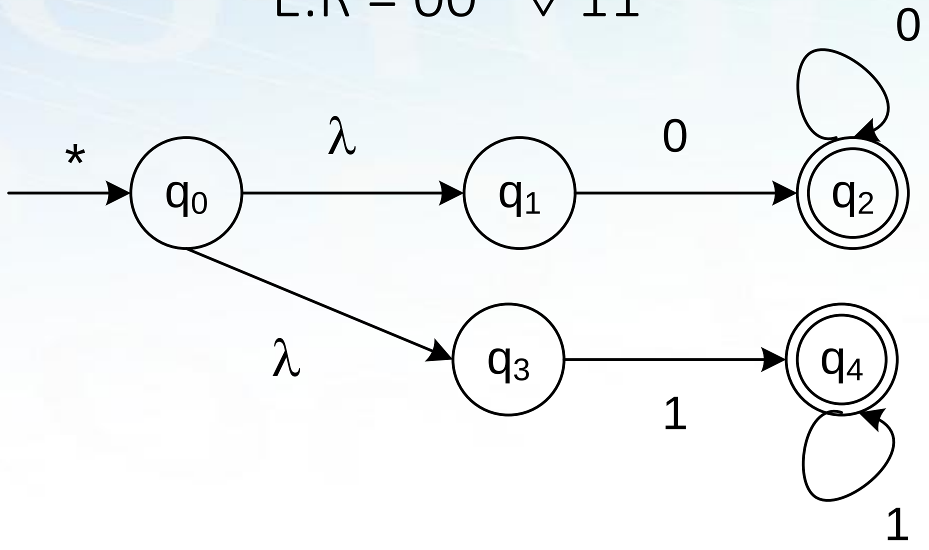
A.F.D. completo

Autómatas Finitos No Determinísticos (A.F.N.)

- Un autómata finito no determinístico es un autómata finito que puede realizar transiciones por la palabra λ.
- Una transición por la palabra λ, es un cambio de estado sin la intervención de ningún carácter de la palabra en estudio.
- Cada estado puede tener, en el diagrama de transición, más de una arista etiquetada con cada letra del alfabeto V.

Ejemplos:

E.R = $00^* \vee 11^*$



E.R = $0^*\ 0\ 1^*\ 0\ (\ 1^* \vee 1\ 1^*\ 0\ )^*$