GRAMATICAS FORMALES

Tipos de Gramática

Una de sus muchas aportaciones fue la **jerarquía de Chomsky** para organizar las gramáticas formales, entendiéndose por gramática las reglas necesarias para definir la estructura de un lenguaje formal.

La jerarquía de Chomsky estructura las gramáticas en **cuatro niveles,** del más genérico y que incluye a los demás (tipo 0) al más específico (tipo 3):

Gramáticas tipo 0	Sin restricciones. Estas gramáticas tienen que tener en su parte izquierda al menos un símbolo no terminal (posteriormente veremos lo que significa, el concepto de **símbolo terminal y no terminal**).
Gramáticas tipo 1	Dependientes del contexto. Se las denomina **dependientes del contexto** porque hay que tener en cuenta los símbolos que vienen antes y después del que queremos sustituir (su contexto).
Gramáticas tipo 2	Independientes del contexto. Generan **lenguajes independientes del contexto** y se caracterizan porque en la parte izquierda de una producción **solo pueden tener un símbolo no terminal.**
Gramáticas tipo 3	Expresiones regulares. Estas son las gramáticas más restrictivas y generan lenguajes regulares. En su parte izquierda tienen solo un no terminal y en su parte derecha tienen solo un terminal.

Tipos de Gramática

- **Símbolos terminales ($\sum_T$):** representa al conjunto de palabras reservadas de un lenguaje de programación. Esto incluye los caracteres especiales y los operadores tanto aritméticos como lógicos. Ejemplo: en el lenguaje de programación C: $\sum_T$ = {main, {, }, #, include, if, then, define, int, float, char, double;, +, -, *, >, >=, |, ...}. Se denotan con letras minúsculas: Ej: **int, float, if.**
- **Símbolos no terminales ($\sum_N$):** representa el conjunto de variables que utilizamos en las producciones de las gramáticas como símbolos de transición. Esto incluye el símbolo inicial de la gramática (S) y las variables que utilicemos. Se denota con letras mayúsculas y veremos un ejemplo cuando expliquemos las producciones.
- **Símbolo inicial (S):** es un símbolo no terminal a partir del cual se obtienen todas las palabras del lenguaje y será el primer símbolo de la gramática G. También se le denomina el **axioma** de la gramática.
- **Conjunto de producciones (P):** es un conjunto de reglas que indica cómo se obtienen las palabras del lenguaje definido por G. Establece las relaciones entre los símbolos terminales y los no terminales.

Tipos de Gramática

- Esta Compuesta por 4 elementos.
  1. T: Símbolos terminales (Elementos que no generan nada)
  2. V: No terminales (Elementos del lado izquierdo de una producción, antes de la flecha "→"
  3. P: Conjunto de producciones (sentencias que se escriben en la gramática)
     - Cada regla o producción consta de:
       - Cabeza: variable
       - → = símbolo de producción
       - Cuerpo: cadena de 0 o mas símbolos terminales y/o variables.

     Es decir una regla tiene la forma Cabeza → Cuerpo , o por ejemplo: Aa →BA
  4. S: Símbolo inicial (primer elemento de la gramática)

Tipos de Gramática

- Una gramática es una estructura algebraica formada por cuatro elementos
- G={NT, T, S, P}
- NT es el conjunto de elementos NO TERMINALES
- T es el conjunto de elementos TERMINALES
- S es el SIMBOLO INICIAL de la gramática
- P es el conjunto de REGLAS DE PRODUCCION

G = (V , T , P , S)

V: No terminales

T: Símbolos terminales

P: Conjunto de producciones

S: Símbolo inicial

Tipos de Gramática

1. Las letras mayúsculas A,B,C,D,E y S denotan variables: y S es el símbolo de inicio.
2. Las letras minúsculas a,b,c,d,e,dígitos, y cadenas en letras negritas son terminales.
3. Las letras mayúsculas X,Y y Z denotan símbolos que pueden ser terminales o variables.
4. Las letras minúsculas u,v,w,x,y,z denotan cadenas terminales.
5. Las letras griegas α,β,γ denotan cadenas de variables terminales.

Gramática tipo 0

Características

- No tiene restricciones .
- Incluyen a todas las demás gramáticas formales.
- Las maquinas que lo aceptan son maquinas de Turing.

$$x \rightarrow y$$
$$x \in (NT/T)^{+}$$
$$y \in (NT/T)^{*}$$

Gramáticas de tipo 0

También llamadas *gramáticas no restringidas o gramáticas con estructura de frase.*

Las reglas de derivación son de la forma:

$$\alpha \rightarrow \beta$$

siendo $\alpha \in (VN \cup VT)^{+}$ $\beta \in (VN \cup VT)^{*}$, es decir la única restricción es que no puede haber reglas de la forma $\lambda \rightarrow \beta$ donde λ es la cadena vacía.

Ejemplo 1

Sea la gramática definida por G_1 = (*{S}, {0,1}, S,P) donde P={(S → 000S111),(0S1→ 01)}. Determinar el lenguaje que genera.*

Solución :
La única forma de generar sentencias es aplicando cualquier nº de veces la primera producción y terminando con la aplicación de la segunda, así se obtiene el lenguaje:

$$S \rightarrow 000S111 \rightarrow 000000S111111 \rightarrow \cdots \rightarrow 0^{(3n-1)}0S11^{(3n-1)} \rightarrow 0^{(3n)}1^{(3n)}$$

Por consiguiente el lenguaje que genera esta gramática es el conjunto infinito de instrucciones que se indica a continuación :

$$L(G_1) = \{0^{(3n)}1^{(3n)}/n \geq 1\}$$

Ejemplo 2

Si la 2ª producción de la gramática del ejemplo 1 fuese *S →01* el lenguaje sería :

$$L(G_2) = \{0^{(3n+1)}1^{(3n+1)}/n \geq 0\}$$

Ejemplo 3

Sea la gramática G_3 = ({*S*}, *{a,b}, S, P) donde P={(S → aSb), (S → ab)}. Determinar* el lenguaje que genera.

Solución :
Aplicando la primera producción n-1 veces, seguida por la aplicación de la segunda producción, se tiene que :

$$S \rightarrow aSb \rightarrow aaSbb \rightarrow a^3Sb^3 \rightarrow \cdots \rightarrow a^{(n-1)}Sb^{(n-1)} \rightarrow a^n b^n$$

El lenguaje generado :

$$L(G_3)=\{a^n\, b^n / n \geq 1\}$$

Ejemplo 4

Dada la gramática G_4 = *({S,A}, {a,b}, S, P) donde P={(S → abAS), (abA → baab),* (*S → a), (A → b)}. Determinar el lenguaje que genera.*

Solución :

Se generan sentencias del lenguaje aplicando las reglas hasta que se pueda
ver la forma general del lenguaje.

S →abAS →baabS →baaba
S →a
S →abAS →abbS →abba
S →abAS →abAabAS →· · ·→$(abA)^nS$ *→* $(abb)^na$
S →abAS →abAabAS →· · ·→$(abA)^nS$ *→* $(baab)^na$
S →abAS →abAabAS →
S →abAS →abAabAS →baababba
S →abAS →abAabAS →abAabAabAS →baababbbaaba

$L(G_4)$ = {cadenas que contienen *abb y baab intercambiándose y reproduciéndose* cualquier número de veces, y terminando siempre con el símbolo *a}*

Se puede observar que la forma de expresar este lenguaje no es simple, y surge la necesidad de tener una herramienta que permita describir los lenguajes de otra forma.

Ejemplo 5

Sea la gramática G_5 = (*{S,A,B}, {a,b}, S, P) donde las producciones P son :*

S →aB	***A →bAA***
S →bA	***B →b***
A →a	***B→bS***
A →aS	***B →aBB***

Determinar el lenguaje que genera.

Solución :

Se generan algunas instrucciones.

S →aB →ab
S →bA →ba
S →aB →abS →abbA →abba
S →bA →bbAA →bbaa
S →aB →abS →abaB →ababS →ababaB →ababab

L(G_5) = {cadenas que tienen igual nº de *a que de b}*

Ejemplo 6

Sea la gramática G_6 = (VN, VT, S, P) donde :
VN = { <número> , <dígito> }
VT = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
S = <número>
Las reglas de producción P son :
<número> ::= <dígito> <número>
<número> ::= <dígito>
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
Determinar el lenguaje que genera.

Solución :
Acontinuación se muestran algunas sentencias del lenguaje generado por esta gramática.

< número >→< dígito >< número >→ 7 < número >→ 72
< número >→< dígito >→ 7
< número >→< dígito >→ 0
< número >→< dígito >< número >→< dígito >< dígito >< número >→· · ·→235

L(G_6) = {conjunto de los números naturales en base diez}.

Ejemplo 7

Sea la gramática G_7 = ({A,S}, {a,b}, S, P) *donde las reglas de producción son :*

$S \rightarrow aS$

$S \rightarrow aA$

$A \rightarrow bA$

$A \rightarrow b$

Determinar el lenguaje que genera esta gramática.

Solución :

Se muestran algunas sentencias del lenguaje generado por la gramática.

$S \rightarrow aS \rightarrow aaA \rightarrow aab$

$S \rightarrow aA \rightarrow ab$

$S \rightarrow aS \rightarrow aaS \rightarrow aaaS \rightarrow \cdots \rightarrow anS \rightarrow a^n aA \rightarrow a^{n+1}b$

$S \rightarrow aA \rightarrow abA \rightarrow abbA \rightarrow abbbA \rightarrow \cdots \rightarrow ab^n A \rightarrow ab^{n+1}$

El lenguaje generado se puede definir con la siguiente expresión regular

$$\mathbf{L(G_7) = a\, a^*\, b\, b^*}$$

Gramática tipo 1

Características

- Generan los lenguajes sensibles al contexto.
- Las maquinas que los aceptan son autómatas linealmente acotados.

$$\alpha \rightarrow \beta\,;\ |\alpha| \leq |\beta|$$

$$\alpha = z_1 x\, z_2$$

$$\beta = z_1 y\, z_2$$

$$z_{1,}\, z_2 \in T^*$$

$$x \in NT$$

$$y \in (NT/T)^+$$

Tipos de Gramáticas Jerarquía de Chomsky

Gramáticas tipo 1

Las reglas de producción de esta gramática tienen la forma

xAy ::= xvy

donde **x**, **y** $\epsilon \sum^*$, **v** $\epsilon \sum^+$ y **A** ha de ser un símbolo no terminal.
(**A** puede transformarse en **v** sólo si aparece en el contexto definido por **x** e **y**)

• Ya que ***v*** no puede ser la palabra vacía, se deduce de aquí que este tipo de gramáticas no pueden tener reglas compresoras. Se admite una excepción en la regla **S ::= *λ*** (siendo **S** el axioma de la gramática). Como consecuencia se tiene que la palabra vacía pertenece al lenguaje generado por la gramática sólo si contiene esta regla.

• Los lenguajes generados por este tipo de gramáticas se denominan "**dependientes del contexto".**

Tipos de Gramáticas Jerarquía de Chomsky

Gramáticas tipo 1

Evidentemente todas las gramáticas de tipo 1 son también de tipo 0, y así, todos los lenguajes dependientes de contexto serán también lenguajes sin restricciones.

- **Ejemplo1 :**

G = ({S, B, C}, {a, b, c}, S,P), donde P es:

$S \rightarrow aSBc \mid aBC$
$bB \rightarrow bb$
$bC \rightarrow bc$
$CB \rightarrow BC$
$cC \rightarrow cc$
$aB \rightarrow ab$

Ejemplo 2

La gramática G = (*{S,A,B}, {a,b}, S, P) cuyas producciones P son:*

S →aB
S →bA
A →a
A →aS
A →bAA
B →b
B→bS
B →aBB

Ejemplo 3

La gramática G = (VN, VT, S, P) donde VN = { <número> , <dígito> }; VT = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }; S = <número> y las reglas de producción P son:

<número> ::= <dígito> <número>
<número> ::= <dígito>
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Ejemplo 4

La gramática G = ({*a,b*}*, {A,S}, S, P) y sus producciones P son:*

S →aS
S →aA
A →bA
A →b

Ejemplo 5

La gramática definida como G = ({*S*}*, {a,b}, S, P) donde P son las siguientes*
producciones : **?**

S →aaaaSbbbb
aSb →ab

La producción ***aSb →ab*** no es del tipo 1, pues se sustituye *S por vacío en el contexto a...b.*
Sin embargo si esta producción fuera ***S →ab*** *o* ***aSb →abb****,* entonces sería de tipo 1.

Ejemplo 6

La gramática G = ({*S,A}, {a,b}, S, P) con las producciones P siguientes :*

S →abAS
abA →baab
S →a
A →b

No es del tipo 1, ya que la producción ***abA →baab*** no es sensible al contexto. Lo sería si fuese ***abA →abab****.*

Propiedades de las gramáticas de tipo 1

Propiedad de no decrecimiento

Las cadenas que se obtienen en cualquier derivación de una gramática de tipo 1 son de longitud no decreciente, es decir :

$$\alpha \rightarrow \beta \Rightarrow |\beta| \geq |\alpha|$$

y que se puede enunciar como *la longitud de la parte derecha de la producción es mayor o igual a la de la parte izquierda.*

La **demostración** es inmediata. Si se define una producción de un lenguaje tipo 1 como :

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

siendo $\gamma \in (VN \cup VT)^{+}$, es decir **γ** nunca puede ser la cadena vacía, lo que implica que $|\gamma| \geq 1$ y como $|A|$ como mínimo vale 1, queda demostrada la propiedad :

$$|\alpha A \beta| \leq |\alpha \gamma \beta|$$

Propiedad de sensibilidad al contexto

También se puede demostrar *que si todas* las reglas de una gramática cumplen la condición de **no decrecimiento**, se puede hallar una gramática equivalente con las producciones sensibles al contexto. Esta segunda propiedad combinada con la primera hace que se pueda intercambiar la característica de no decrecimiento
con la definición.

Ejemplo 7

Sea la gramática G = ({*S,B,C}, {a,b,c}, S, P) donde P son las producciones :*

S →aSBC
S →aBC
CB →BC
bB →bb
bC →bc
cC →cc
aB →ab

La gramática anterior no es de tipo 1 según la definición dada, ya que la regla
CB → BC no respeta el contexto. Sin embargo puede apreciarse que todas las reglas de esta gramática son no decrecientes, por lo tanto es posible encontrar una gramática equivalente que genere el mismo lenguaje. Se puede sustituir la regla *CB → BC por :*

$CB \rightarrow XB$
$XB \rightarrow XY$
$XY \rightarrow BY$
$BY \rightarrow BC$

Puede observarse que ambas gramáticas son equivalentes y que generan el lenguaje :

$$\mathbf{L(G) = \{ \, a^n \, b^n \, c^n \, / \, n \geq 1 \, \}}$$

Gramática tipo 2

Características

- Generan los lenguajes independientes del contexto.
- Maquinas que lo aceptan son autómata de pila.

$$x \rightarrow y$$
$$x \in NT$$
$$y \in (NT/T)^*$$

Gramáticas tipo 2

- Las reglas de estas gramáticas se ajustan al siguiente esquema:

$$\mathbf{A ::= v}$$

donde $\mathbf{v} \in \Sigma^*$, y $\mathbf{A} \in \Sigma_N$ En concreto **v** puede ser $\boldsymbol{\lambda}$.

- Para toda gramática de tipo 2 existe una gramática equivalente desprovista de reglas de la forma A ::= $\boldsymbol{\lambda}$, que generará el mismo lenguaje que la de partida, excepto la palabra vacía. Si se le añade a la segunda gramática la regla S ::= $\boldsymbol{\lambda}$, las gramáticas generarán el mismo lenguaje.

- Por lo tanto, se pueden definir las gramáticas de tipo 2 de una forma más restringida, en el que las reglas de producción tendrán la siguiente forma
A ::= v donde $\mathbf{v} \in \Sigma^+$**, y A** $\in \Sigma_N$ **.** Además podrán contener la regla
S ::= $\boldsymbol{\lambda}$

- Los lenguajes generados por este tipo de gramáticas se denominan **independientes de contexto,** ya que la conversión de **A** en **v** puede realizarse independientemente del contexto en que aparezca **A**.

Gramáticas tipo 2

- La mayor parte de los lenguajes de programación de ordenadores pueden describirse mediante gramáticas de este tipo.

- Ejemplo : sea la gramática G = ({a, b}, {S}, S, { S ::= aSb | ab}).

Es una gramática de tipo 2. La derivación de la palabra aaabbb será:

S → aSb → aaSbb → aaabbb

Puede verse que el lenguaje definido por esta gramática es $\{a^n b^n \mid n=1, 2, ...\}$

- Un mismo lenguaje puede generarse por muchas gramáticas diferentes. Sin embargo, una gramática determinada describe siempre un lenguaje único.

Ejemplo 1

La gramática G = (*{S,A,B}, {a,b}, S, P) cuyas producciones P se muestran a continuación* es de tipo 2.

S →aB	***A →bAA***
S →bA	***B →b***
A →a	***B→bS***
A →aS	***B →aBB***

Ejemplo 2

La gramática G = (VN, VT, S, P) donde VN = { <número> , <dígito> }; VT = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }; S = <número> y las reglas de producción P que se muestran a continuación es de tipo 2.

<número> ::= <dígito> <número>
<número> ::= <dígito>
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Gramática tipo 3

Características

- También llamadas de contexto regular.
- Maquinas que lo aceptan son autómata finito, determinista o no determinista.

$$\alpha \rightarrow \beta$$

$$\alpha \in NT$$

$$\beta \in \begin{cases} aB \\ Ba \\ b \end{cases}$$

$$B \in NT$$

$$a \in T^{+}$$

$$b \in T^{*}$$

Gramáticas tipo 3

Estas gramáticas se clasifican en los dos grupos siguientes:

• **Gramáticas lineales por la izquierda,** cuyas reglas de producción pueden tener una de las formas siguientes:

A ::= a

A ::= Va

S ::= $\boldsymbol{\lambda}$

donde **a** $\in \sum_T$, **A**, V $\in \sum_N$, y S es el axioma de la gramática.

• **Gramáticas lineales por la derecha,** cuyas reglas de producción tendrán la forma:

A ::= a

A ::= aV

S ::= $\boldsymbol{\lambda}$

donde a $\in \sum_T$, A, V $\in \sum_N$, y S es el axioma de la gramática.

• Los lenguajes representados por este tipo de gramáticas se denominan **lenguajes regulares.**

Gramáticas tipo 3

- G1 = ({ 0, 1}, {A, B}, A, { A ::= B1 | 1, B ::= A0})

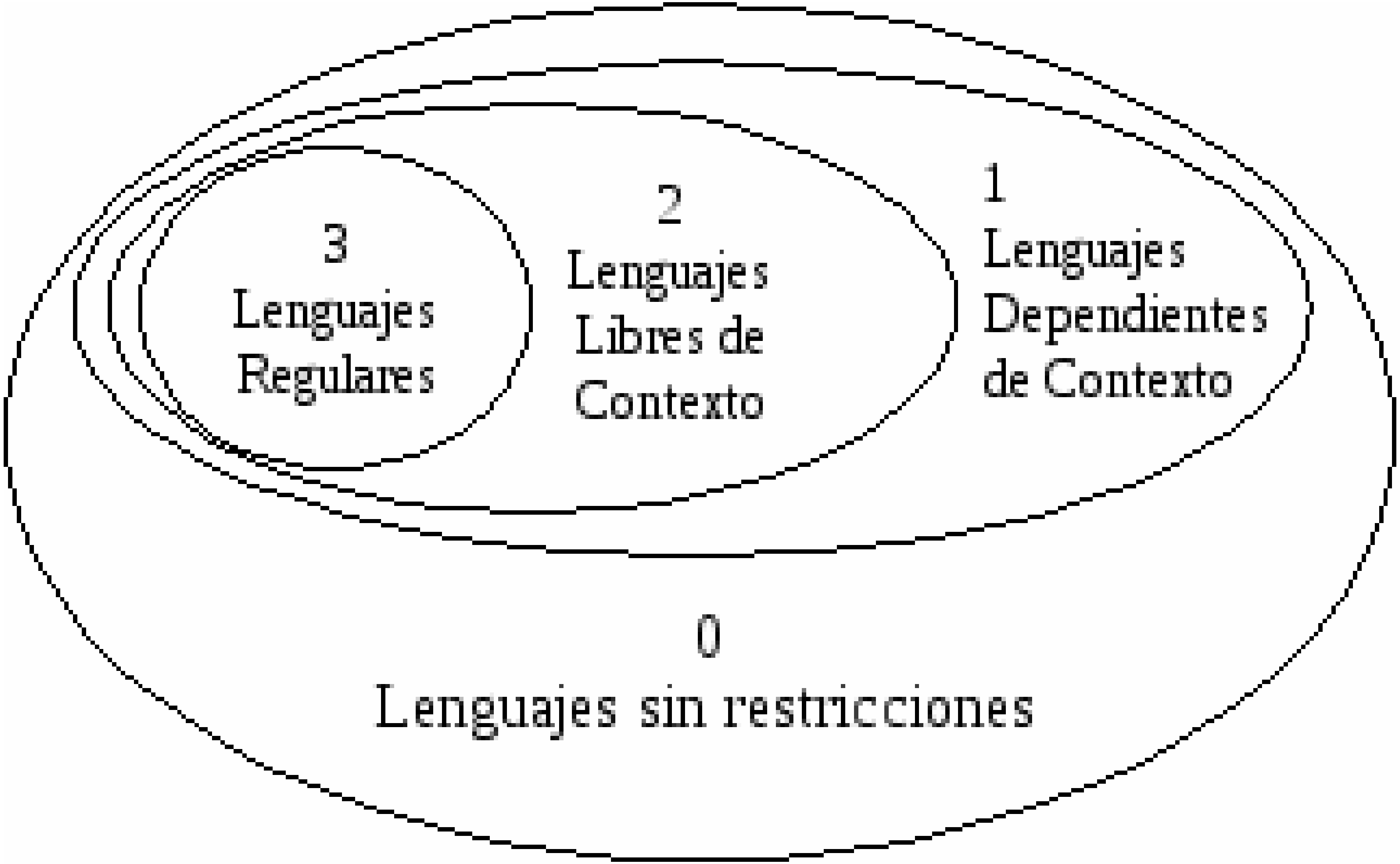
Gramática lineal por la izquierda que describe el lenguaje:

$$L1 = \{ 1, 101, 10101, \ldots \} = \{1(01)^n \mid n = 0, 1, 2, \ldots\}$$

- G2 = ({ 0, 1}, {A, B}, A, { A ::= 1B | 1, B ::= 0A})

Gramática lineal por la derecha que genera el mismo lenguaje que la gramática anterior.

EN RESUMEN


3
Lenguajes Regulares
2
Lenguajes Libres de Contexto
1
Lenguajes Dependientes de Contexto
0
Lenguajes sin restricciones

EN RESUMEN

<table>
<tr><th>Gramática</th><th>Tipo 3
Lenguajes Regulares</th><th>Tipo 2
Libres de Contexto
(Context Free)</th><th>Tipo 1
Dependiente del contexto</th><th>Tipo 0
Sin restricciones</th></tr>
<tr><td>Lenguaje</td><td>$\{a^n, n \geq 1\}$</td><td>$\{a^n\, b^n\, , n \geq 1\}$</td><td>$\{a^n\, b^n\, c^n\, , n \geq 1\}$</td><td rowspan="4">Todas las anteriores

Maquina de Turing</td></tr>
<tr><td>Reglas</td><td>A = a A</td><td>S = a S b</td><td>C c = V c c</td></tr>
<tr><td>Autómata</td><td>Autómata Finito Determinístico
AFD</td><td>Autómata a Pila
AP</td><td>Autómata Linealmente Acotado</td></tr>
<tr><td>Análisis</td><td>Análisis Léxico</td><td>Análisis Sintáctico</td><td>Tabla de Símbolos</td></tr>
</table>

TALLER INVESTIGACION 30%
SEGUNDO CORTE

Autómata Finito Determinístico AFD
Autómata a Pila AP
Autómata Linealmente Acotado
Maquina de Turing

Para cada uno de los conceptos descritos realizar lo siguiente:

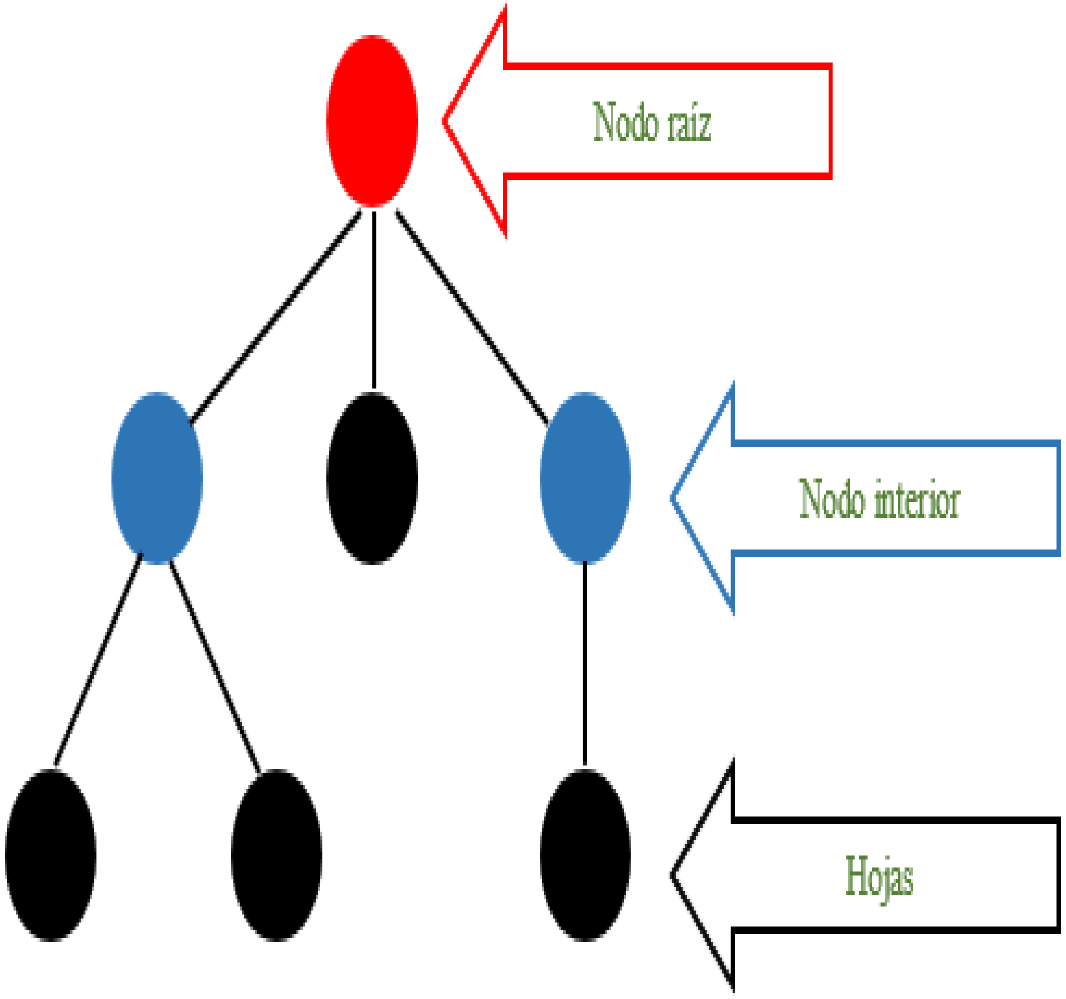
1. Definición
2. Representación
3. Ejemplos
4. Simuladores de aplicación

Árboles de derivación

Derivaciones utilizando una gramática

Derivación

- Aplicación de las producciones de una gramática para obtener una cadena de terminales.
- Consiste en sustituir la variable de la cabeza por el cuerpo de la producción.

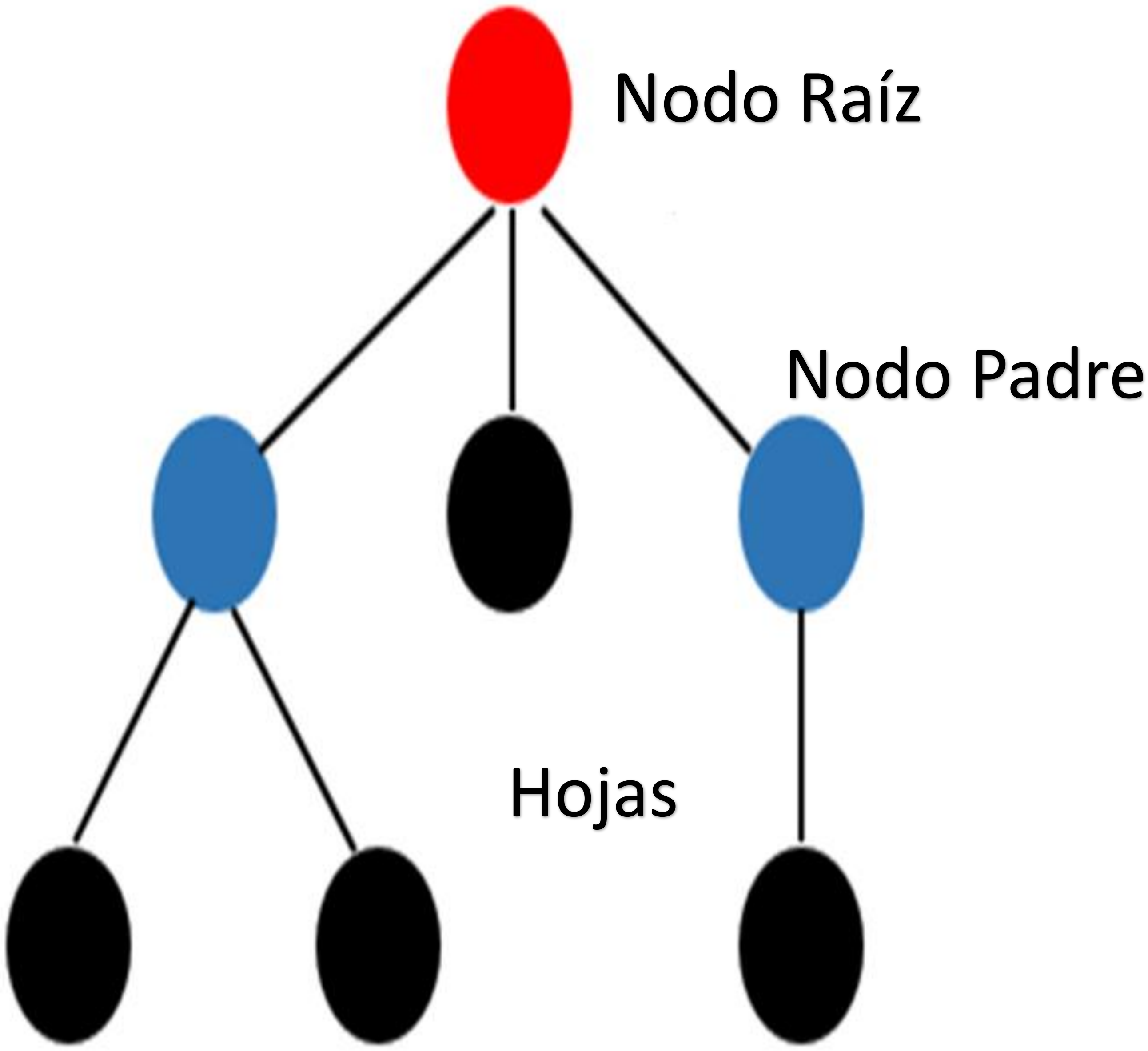


Estructura

- Un árbol es un conjunto de puntos, llamados nodos; unidos por líneas llamados arcos.

Satisfacen ciertas propiedades

- Hay un único nodo distinguido, llamado raíz.
- Todo nodo “c” excepto el nodo raíz esta conectado con un arco a otro nodo “k”, llamado padre de c. El padre de un nodo, se dibuja por encima de un nodo.
- Todos los nodos están conectados al nodo raíz mediante un único camino.
- Los nodos que no tienen hijos se denominan hojas, el resto de los nodos se denominan nodos interiores.



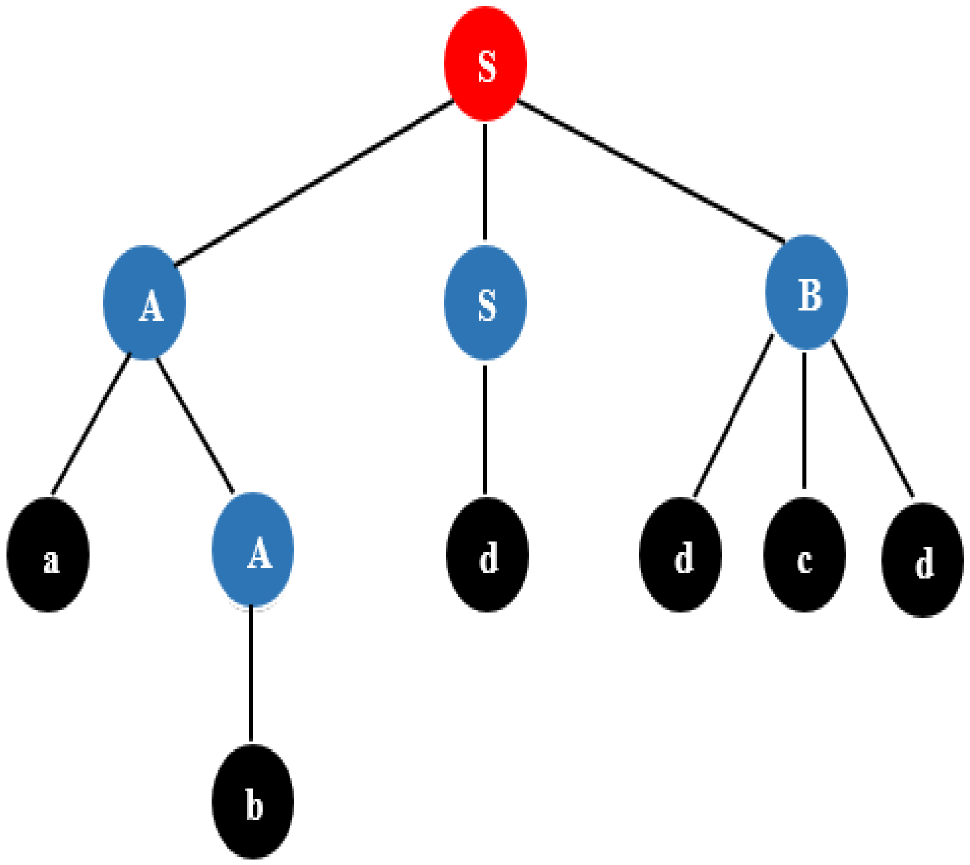
Tipos de derivación

Derivación a la izquierda.

Derivación a derecha.

Ejemplo:

1. S → ASB
2. A → b
3. aaA → aaBB
4. S → d
5. A → aA
6. B → dcd



Ejemplo

1. S → ASB
2. A → b
3. aaA → aaBB
4. S → d
5. A → aA
6. B → dcd

Derivación a la izquierda

$S \xrightarrow{1} ASB \xrightarrow{2} aASB \xrightarrow{4} abSB \rightarrow$

$abdB \xrightarrow{5} \xrightarrow{6} abddcd$

Derivación a la derecha

$S \xrightarrow{1} ASB \xrightarrow{6} ASdcd \xrightarrow{4} Addcd \rightarrow$

$\xrightarrow{5} aAddcd \xrightarrow{2} abddcd$

GRAMATICAS FORMALES

EJERCICIOS PRACTICOS JFLAP

Realizar en JFLAP

1. Sea la gramática G = (*{S}, {(,)}, S, P) donde las reglas de producción son :*

$$S \rightarrow SS$$

$$S \rightarrow (S)$$

$$S \rightarrow ()$$

a) Una lista de 5 palabras que son aceptadas

(())(())

(())(())(())(())

b) Una lista de 5 palabras que son rechazadas

c) Dar una descripción del lenguaje que representan

d) Indica el tipo de gramática en la jerarquía de Chomsky

TIPO 2

Realizar en JFLAP

1. Sea la gramática G = ({*S,A*}, *{a,b,c,d,e)}, S, P) donde las reglas de producción son :*

$$S \rightarrow Ac$$

$$S \rightarrow Ad$$

$$S \rightarrow Ae$$

$$A \rightarrow ab$$

S->Ae->abe

a) Una lista de 5 palabras que son aceptadas abc, abd, abe
b) Una lista de 5 palabras que son rechazadas: abb, abdcd, abbbb abeee, abde
c) Dar una descripción del lenguaje que representan
d) Indica el tipo de gramática en la jerarquía de Chomsky

Realizar en JFLAP

1. Sea la gramática G = (*{S}, {1,0}, S, P) donde las reglas de producción son :*

$$S \rightarrow \lambda$$

$$S \rightarrow 0$$

$$S \rightarrow 1$$

$$S \rightarrow 0S0$$

$$S \rightarrow 1S1$$

a) Una lista de 5 palabras que son aceptadas

b) Una lista de 5 palabras que son rechazadas

c) Dar una descripción del lenguaje que representan

d) Indica el tipo de gramática en la jerarquía de Chomsky

Realizar en JFLAP

1. Sea la gramática G = ({*S,B*}, {*a,b,c*}, *S, P*) *donde las reglas de producción son :*

$$S \rightarrow abc$$

$$S \rightarrow aBSc$$

$$Ba \rightarrow aB$$

$$Bb \rightarrow bb$$

S->aBSc->aBabcc

aaBbcc->aabbcc

aaabbbccc

abc

a) Una lista de 5 palabras que son aceptadas
b) Una lista de 5 palabras que son rechazadas
c) Dar una descripción del lenguaje que representan
d) Indica el tipo de gramática en la jerarquía de Chomsky

Realizar en JFLAP

1. Sea la gramática G = (*{S}, {x,y,z,+,-,*,/,(,)}, S, P) donde las reglas de producción son :*

$S \rightarrow x$

$S \rightarrow y$

$S \rightarrow z$

$S \rightarrow S+S$

$S \rightarrow S-S$

$S \rightarrow S*S$

$S \rightarrow S/S$

$S \rightarrow (S)$

S->S+S->S+S+S->x+y+z

S->S+S->S+S+S+S-

>x+y+z+(s)

x+y+z+(z)

a) Una lista de 5 palabras que son aceptadas
b) Una lista de 5 palabras que son rechazadas
c) Dar una descripción del lenguaje que representan
d) Indica el tipo de gramática en la jerarquía de Chomsky

Realizar en JFLAP

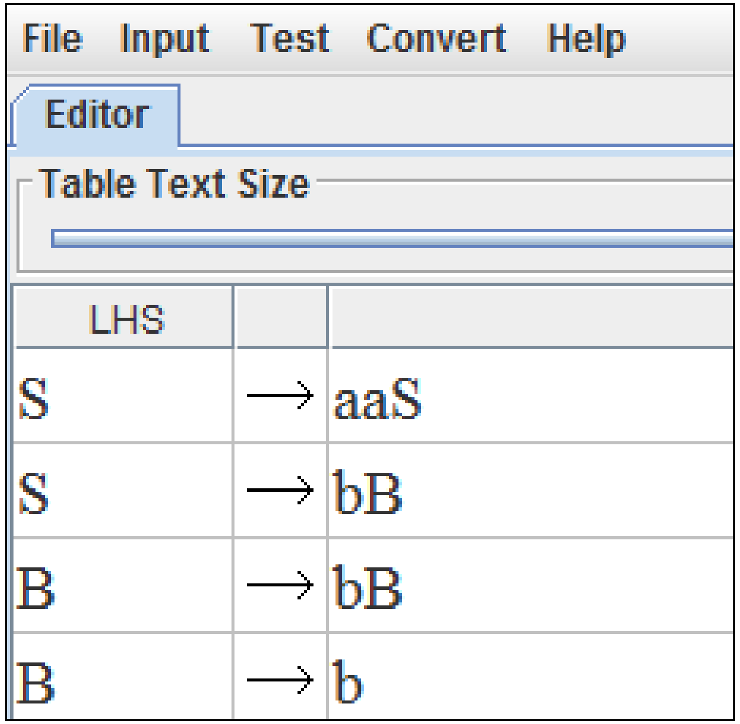
Escribe la siguiente gramática en el editor de JFLAP, y para cada una de ellas realice:

a) Una lista de 5 palabras que son aceptadas

b) Una lista de 5 palabras que son rechazadas

c) Dar una descripción del lenguaje que representan

d) Indica el tipo de gramática en la jerarquía de Chomsky



Realizar en JFLAP

Escribe la siguiente gramática en el editor de JFLAP, y para cada una de ellas realice:

a) Una lista de 5 palabras que son aceptadas
b) Una lista de 5 palabras que son rechazadas
c) Dar una descripción del lenguaje que representan
d) Indica el tipo de gramática en la jerarquía de Chomsky

Editor

Table Text Size

LHS		
S	$\rightarrow$	bA
A	$\rightarrow$	abaA
A	$\rightarrow$	abaB
B	$\rightarrow$	b
B	$\rightarrow$	λ

Realizar en JFLAP

Escribe la siguiente gramática en el editor de JFLAP, y para cada una de ellas realice:

a) Una lista de 5 palabras que son aceptadas

b) Una lista de 5 palabras que son rechazadas

c) Dar una descripción del lenguaje que representan

d) Indica el tipo de gramática en la jerarquía de Chomsky

Editor

Table Text Size

LHS		
S	$\rightarrow$	aaS
S	$\rightarrow$	abaA
S	$\rightarrow$	bA
S	$\rightarrow$	abbaB
A	$\rightarrow$	aaA
A	$\rightarrow$	abaB
A	$\rightarrow$	bB
B	$\rightarrow$	aaB
B	$\rightarrow$	λ

Realizar en JFLAP

1. Sea la gramática G = (*{S}, {a,b}, S, P) donde las reglas de producción son :*

$$S \rightarrow abS$$
$$S \rightarrow Sab$$
$$S \rightarrow aSb$$
$$S \rightarrow SS$$
$$S \rightarrow \lambda$$

Determinar el lenguaje que genera esta gramática.
Ejecuta el derivador de fuerza bruta de JFLAP para esta palabra. Proporciona dos árboles de derivación diferentes para esta palabra y anota el orden de producciones empleado.

Realizar en JFLAP

1. Sea la gramática G = ({A,B},{1,0},S,P) *donde las reglas de producción son :*

$$A \rightarrow 1B1$$
$$A \rightarrow 11$$
$$1B1 \rightarrow 101$$
$$1B1 \rightarrow 111$$

a) Una lista de 5 palabras que son aceptadas
b) Una lista de 5 palabras que son rechazadas
c) Dar una descripción del lenguaje que representan
d) Indica el tipo de gramática en la jerarquía de Chomsky

Realizar en JFLAP

Crear una gramática G reconozca todas la cadenas siguientes

aba

abba

abbbba

abbbbbba

abbbbbbbba

abbbbbbbbbbbbbbbba

Realizar en JFLAP

Crear una gramática G que genere números reales, con "." para separar la parte entera y la decimal. Comprobar qué secuencias de las siguientes son palabras del lenguaje L(G) y determinar el motivo. Anotar las observaciones

Palabras que debe aceptar	**0**	**00**	**00.1**	**3.1416**	**3445**	**1.23**	**9.45**

Realizar en JFLAP

1. Sea la gramática G = ({S,A,B,C},{a,b},S,P) *donde las reglas de producción son :*

$$S \rightarrow AC$$
$$C \rightarrow AB$$
$$B \rightarrow CC$$
$$A \rightarrow a$$
$$C \rightarrow \mathrm{b}$$
$$S \rightarrow SS$$
$$A \rightarrow \mathrm{AA}$$

a) Una lista de 5 palabras que son aceptadas
b) Una lista de 5 palabras que son rechazadas
c) Dar una descripción del lenguaje que representan
d) Indica el tipo de gramática en la jerarquía de Chomsky

Realizar en JFLAP

Las producción de la anterior gramática son de las forma:

$$a^n bb$$

1. Ejecuta el brute force parser para n= 2,...,8 y apunta el número de nodos generados. ¿Qué observación puedes hacer acerca del número de nodos generados? ¿Y del tiempo que ha tardado? ¿Sabes el motivo?

2. Elimina la producción S →SS y ejecuta de nuevo el brute force parser para n=2,...,8 y apunta el número de nodos generados. Compara los resultados con los del apartado (a). ¿Sabes cuál es el motivo de la diferencia?

3. Añadiéndolo al cambio anterior, cambia la producción A→AA a A→aA. Ejecuta el brute force parser para n=2,...,8 y apunta el número de nodos. Compara los resultados con los de los apartados (a) y (b). ¿Sabes cuál es el motivo de las diferencias?