Tema 3: Gramáticas Formales

Informática Teórica I

Teoría de Gramáticas Formales. Bibliografía

M. Alfonseca, J. Sancho y M. Martínez. "Teoría de Lenguajes, Gramáticas y Autómatas", R.A.E.C., Madrid, (1998).

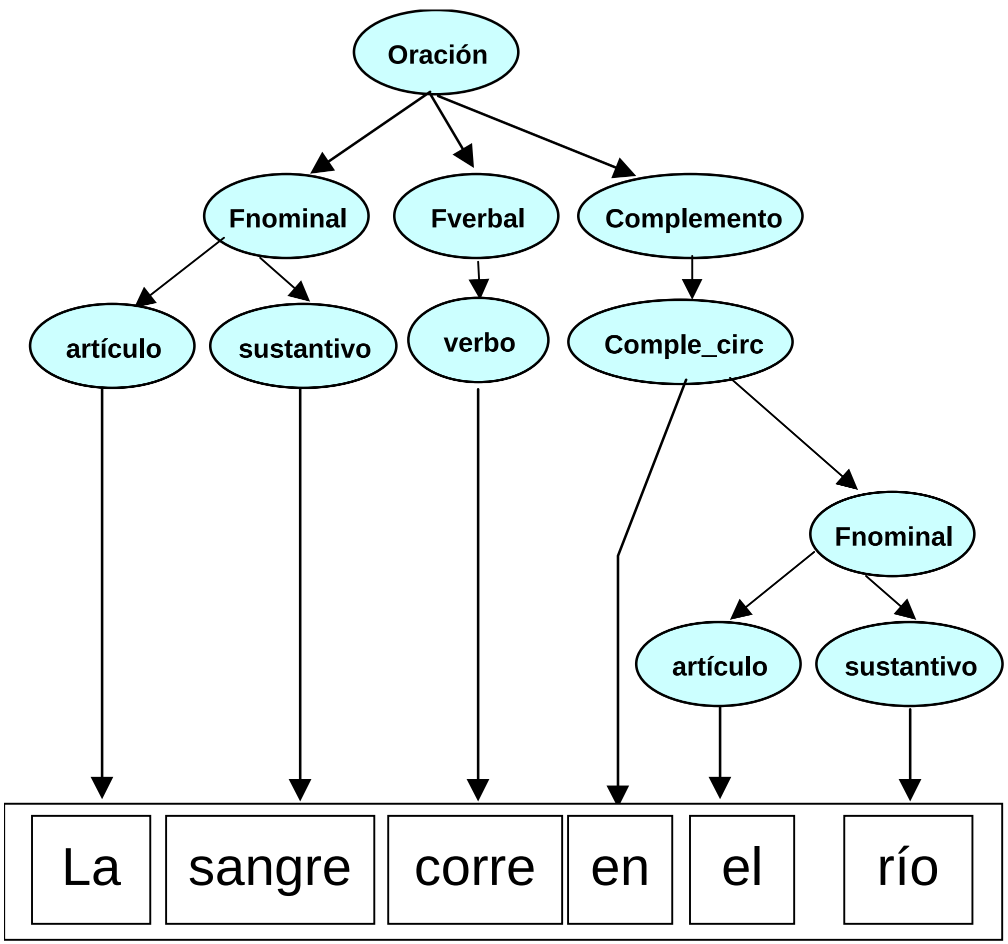
P. Isasi, P. Martínez y D. Borrajo. "Lenguajes, Gramáticas y Autómatas, un Enfoque Práctico". Addison-Wesley, (1997).

John E. Hopcroft y Jefrey D. Ullman. "Introduction to Automata Theory, Languages and Computation". Addison-Wesley, (1979).

F.J. Sanchis y C. Galán. "Compiladores: teoría y construcción". Paraninfo, (1986).

ramáticas Formales. Introducción

a Gramática del castellano como diagrama sintáctico



ramáticas Formales. Introducción

a Gramática del castellano en notación de Backus

:= <Fnominal> <Fverbal> | <Fnominal> <Fverbal><Complemento>

>::= <Sustantivo> | <NomPr> | <Artículo> <Sustantivo>

| <Artículo> <Sustantivo> <Adjetivo>

| <Artículo> <Sustantivo> <Adjetivo>

| < Fnominal >de < Fnominal >

= <Verb> | <Verb> <Adverbio>

ento>::=<ComDir> |<ComIn> |<ComCir> | <ComDir> <ComIn>

= <Fnominal> , <ComIn>::= a <Fnominal> | para <Fnominal> | ....

:= en <Fnominal> | desde <Fnominal> | cuando <Fnominal> | ....

ustantivo>, <Adjetivo>, <Adverbio>, <Artículo>, <NomPr>, <Verbo>, etc
valores palabras propias de estas categorías gramaticales

...amáticas Formales. Definición

...ormal" para especificar de manera finita el conjunto de cadenas de símbolos que constituyen un lenguaje

...es una cuádrupla: **(Σ_T, Σ_N, S, P),** Σ_T y Σ_N son alfabetos:

- **Σ_T:** alfabeto de símbolos terminales. Todas las cadenas del lenguaje representado por la G (L(G)) están formadas con símbolos de este alfabeto.
- **Σ_N:** Conjunto de símbolos auxiliares introducidos como elementos auxiliares para la definición de G pero que no figuran en las cadenas de L(G).
- **S:** axioma o símbolo destacado. Es un símbolo NT a partir del que se comienzan a aplicar las reglas de P.
- **P:** conjunto de reglas de producción: **u::=v** donde $u \in \Sigma^+$ y $v \in \Sigma^*$
  u = xAy tal que x, y $\in \Sigma^*$ y A $\in \Sigma_N$.

ramáticas Formales. Definición

ática formal:

e cumple entre los alfabetos de G:

$\Sigma_T \cup \Sigma_N$ = Alfabeto o vocabulario de G

$\Sigma_T \cap \Sigma_N = \phi$ (son disjuntos)

tación de Backus: Si u::=v y u::= w son dos reglas de producción
P, entonces se puede escribir u ::= v | w

denomina notación BNF: Forma Normal de Backus o Forma
rmal de Backus-Naur

jemplo: sea G = ({0,1,2,3,4,5,6,7,8,9}, {N,C}, N,P)

P = {N::= CN | C

C::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

ramáticas Formales. Definiciones

ma sentencial:

Sea una *G*, sea $x \in \Sigma^*$

x se denomina **forma sentencial de G** si se verifica:

S $*\rightarrow$ x

es decir, que existe una relación de Thue entre el axioma y *x*.

Si **x** $\in \Sigma_T$ se dice que **x** es una **Sentencia o instrucción del lenguaje descrito por G**

ramáticas Formales. Definiciones

epto de Lenguaje: asociado a una gramática

na G

lenguaje asociado a esa G:

lenguaje generado por esa G

lenguaje descrito por esa G

a $L(G) = \{x \mid S \xrightarrow{*} x \text{ AND } x \in \Sigma_T^*\}$

ir, al conjunto de todas las sentencias de la Gramática

ramáticas Formales. Definiciones

ir cuál es el lenguaje descrito por la G del ejemplo:

a G = ({0,1,2,3,4,5,6,7,8,9}, {N,C}, N,P)

= {N::= NC | C, C::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

ramáticas Formales. Definiciones: ramáticas equivalentes

ramáticas equivalentes:

Dos gramáticas G1 y G2 son equivalentes si describen /generan el mismo lenguaje:

$\boxed{G1 \approx G2}$

Si

$\boxed{L(G1) = L(G2)}$

ramáticas Formales. Definiciones: ases y asideros

ea *G*

ea $v = xuy$ una forma sentencial de G ($S \ast\!\rightarrow v$)

e dice que u es una **frase de la forma sentencial** v respecto del mbolo no terminal U ($U \in \Sigma_{NT}$) si se cumple:

$$S \ast\!\rightarrow x\,U\,y$$

$$U +\!\rightarrow u \qquad \textbf{derivación de longitud n}$$

i U es una forma sentencial de *G*, entonces todas las frases que erivan de U serán a su vez formas sentenciales de *G*

es **frase simple de** v si se cumple:

$$S \ast\!\rightarrow x\,U\,y$$

$$U \rightarrow u \qquad \textbf{derivación directa}$$

e llama **asidero, handle o pivote** a la frase simple más a la quierda de la forma sentencial

ramáticas Formales. Definiciones: ases y asideros. Ejemplo

o alfonseca pg 50:

la gramática que describe los números enteros positivos, mostrar que N no es una frase de N1. Encontrar todas las ses de N1. ¿Cuáles son frases simples? ¿Cuál es el asidero?

G = ({0,1,2,3,4,5,6,7,8,9}, {N,C}, N,P)

P = {N::= NC |C, C::= 0|1|2|3|4|5|6|7|8|9}

y en la forma sentencial 2C0?

ramáticas Formales. Definiciones: ases y asideros. Ejemplo

ercicio alfonseca pg 50:

v = xuy = N1

- **N**1 N es la frase
- N**1** 1 es la frase
- **N1** N1 es la frase

amáticas Formales. Definiciones: ecursividad

a *G*

a *G* se llama **recursiva en *U***, $U \in \Sigma_{NT}$, si se cumple:

$$U \overset{+}{\rightarrow} x\,U\,y$$

$x = \lambda$ $(U \overset{+}{\rightarrow} U\,y)$ se dice que *G* es **recursiva a izquierdas**

$y = \lambda$ $(U \overset{+}{\rightarrow} xU)$ se dice que G es **recursiva a derechas**

na regla de producción es recursiva si tiene la forma:

$$U ::= x\,U\,y$$

un lenguaje es infinito, la gramática que lo representa tiene que r recursiva

icios: alfonseca página 50 ejercicios 1, 2 y 3

rarquía de Chomsky

G= (Σ_T, Σ_N, S, P)
$\Sigma_T \cup \Sigma_N = \Sigma$, alfabeto gramática, $\Sigma_T \cap \Sigma_N = \phi$

Tipos: según la forma de sus reglas de derivación
G3⊂G2⊂G1⊂G0 Jerárquica restricciones en su conjunto P.

G0
o Restringidas
Estructura de Frase

u∈Σ^+
v∈Σ^*
tricción: **λ::= v ∉P**
tencial:
,y∈Σ*, A∈Σ_N
= {(0,1), (S), S, P}, donde:
→000S111), (0S1→01)}

ctura De Frases:
=xvy) ∈ P, donde:
Σ^*, A ∈ Σ_N, u∈Σ^+
xvy cuando v=λ,
xy reglas compresoras

- Los lenguajes que son representados por G0 se llaman lenguajes sin restricciones
- Chomsky 1959: todo L(G0) puede ser descrito por una G0 con estructura de frases. Ejemplo:

G= ({a,b},{A,B.C},A,P)
P = {A::= aABC / abC
CB::= BC
bB::=bb
bC::=b}

G' = {a,b},{A,B.C,X,Y},A,P')
P' = {A::= aABC / abC
CB::=XB, XB::=XY
XY::=BY, BY::=BC
bB::=bb
bC::=b}

rarquía de Chomsky

G= (Σ_T, Σ_N, S, P)
$\Sigma_T \cup \Sigma_N = \Sigma$, alfabeto gramática
$\Sigma_T \cap \Sigma_N = \phi$

Tipos: según la forma de sus reglas de derivación
3⊂G2⊂G1⊂G0 Jerárquica restricciones en su conjunto P.

G1
Sensibles al Contexto

y ::= xvy

$x,y \in \Sigma^*$, $A \in \Sigma_N$
$v \in \Sigma^+$ No permite reglas compresoras
Excepción: **(S::= λ)$\in$P**

6,7,8,9,), (<número>, <dígito>),<número>, P},
<dígito><número>,
<dígito>,
1|2|3|4|5|6|7|8|9)}

ontexto:
iar **A** por **v**, siempre en el contexto **x...y**

- Los lenguajes que son representados por G1 se llaman **lenguajes sensibles al contexto**
- $\lambda \in L(G1)$ Sii $(S::=\lambda) \in P$
- Ejemplo de G que no es G1:

G= ({a,b}, {S}, S, P)

P= {S::= aaaaSbbbb, aSb::= ab}

- Ejemplo de G que si es G1:

G= ({a,b}, {S}, S, P)

P= {S::= aaaaSbbbb, aSb::= abb}

rarquía de Chomsky

ad de NO DECRECIMIENTO de las G1:

s cadenas que se obtienen en cualquier derivación de una G1 son de gitud no decreciente, es decir:

$$\mathbf{u \rightarrow v \Rightarrow |v| \geq |u|}$$

ongitud de la parte decha de la producción es mayor o igual a la longitud la parte izquierda

- Demostración:

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

  donde $\gamma \in (\Sigma_T \cup \Sigma_{NT})^+$ por definición de G1 (no compresoras)

  es decir, $\gamma \neq \lambda$ siempre, lo que implica $|\gamma| \geq 1$

  y como $|A| = 1$ como mínimo, queda demostrado

rarquía de Chomsky

$G = (\Sigma_T, \Sigma_N, S, P)$
$\Sigma_T \cup \Sigma_N = \Sigma$, alfabeto gramática
$\Sigma_T \cap \Sigma_N = \phi$

Tipos: según la forma de sus reglas de derivación
G3⊂G2⊂G1⊂G0 Jerárquica restricciones en su conjunto P.

G2
de Contexto Libre

uajes que son representados por
man **lenguajes no sensibles al**
o de contexto libre

∃ l(G2') sin reglas A::= λ (A ≠ S)
G2). Ver algoritmo eliminación
O generativas.

A ::= v

$v \in \Sigma^*$
$A \in \Sigma_N$ puede aparecer $A ::= \lambda$

r ∈ P se caracterizan por tener un sólo símbolo NT en su parte izquierda

(S::= λ) ∈P y **(A::= λ) ∉P**
(algoritmo limpieza reglas NO generativas)

Ejemplo:
G= {(a,b), (S,A),S, P}
P={(S → aS, S → aA, A→ bA, A → b)}

Contexto Libre: se puede cambiar **A** por **v**, en cualquier contexto

rarquía de Chomsky

$G = (\Sigma_T, \Sigma_N, S, P)$
$\Sigma_T \cup \Sigma_N = \Sigma$, alfabeto gramática
$\Sigma_T \cap \Sigma_N = \phi$

Tipos: según la forma de sus reglas de derivación
$G3 \subset G2 \subset G1 \subset G0$ Jerárquica restricciones en su conjunto P.

G3
Gramáticas Regulares

G3 Lineales por la Izda.	**G3 Lineales por la Dcha.**
A::= a	**A::= a**
A::= Va	**A::= aV**
S::= λ	**S::= λ**

$a \in \Sigma_T$
$A,V \in \Sigma_N$ S es axioma

uajes que son representados por
man **lenguajes regulares**

$\exists$ l(G3') sin reglas A::= λ ($A \neq S$)
G3). Ver algoritmo eliminación
O generativas.

r $\in$ P un sólo símbolo NT en su parte izda y su parte dcha comienza por un T seguido o no de NT (al revés en lineal derecha)

Ejemplo:
G= {(a,b), (S,A),S, P},
P={(S $\rightarrow$ aS, S $\rightarrow$ aA, A$\rightarrow$ bA, A $\rightarrow$ b)}

rarquía de Chomsky. Ejemplos I

nplos de Gramáticas: Isasi, Martínez y Borrajo, páginas 16 a 19

= ({0,1}, {A,B,S},S, P), P={S::= A0, A0::= 1B1, 1A::= 0B0, B::=λ / 1 / 0}

= ({0,1}, {A,B},A, P), P={A::= 1B1 / 11, 1B1::= 101 / 111}

= ({0,1}, {A,B},A, P), P={A::= 1B1 / 11, B::= 0 / 1}

= ({0,1}, {A,B, C},A, P), P={A::= 1B, B::=1/ 0C / 1C, C::= 1)}

rarquía de Chomsky. Ejemplos I

nplos de Gramáticas: Isasi, Martínez y Borrajo, páginas 16 a 19

= ({0,1}, {A,B,S},S, P), P={S::= A0, A0::= 1B1, 1A::= 0B0, B::=λ / 1 / 0}

B::=λ es una regla compresora
A0::= 1B1 y 1A::= 0B0 no guardan el contexto
Se trata de G0 sin restricciones
L = {11, 101, 111}
¿Necesita un lenguaje tan sencillo de una gramática del mayor nivel en la jerarquía de Chomsky?

rarquía de Chomsky. Ejemplos I

nplos de Gramáticas: Isasi, Martínez y Borrajo, páginas 16 a 19

= ({0,1}, {A,B},A, P), P={A::= 1B1 / 11, 1B1::= 101 / 111}

No hay reglas compresoras
1B1::= 101 / 111 guardan el contexto
Se trata de G1 con estructura de frases
L1 = {11, 101, 111}
¿Necesita un lenguaje tan sencillo de una gramática del mayor nivel en la jerarquía de Chomsky?

rarquía de Chomsky. Ejemplos I

nplos de Gramáticas: Isasi, Martínez y Borrajo, páginas 16 a 19

= ({0,1}, {A,B},A, P), P={A::= 1B1 / 11, B::= 0 / 1}

o hay reglas compresoras

n solo símbolo a la izda

n solo símbolo a la izda

bertad a la dcha → G independiente del contexto G2

= {11, 101, 111}

Necesita un lenguaje tan sencillo de una gramática alto nivel en la rarquía de Chomsky?

rarquía de Chomsky. Ejemplos I

nplos de Gramáticas: Isasi, Martínez y Borrajo, páginas 16 a 19

= ({0,1}, {A,B, C},A, P), P={A::= 1B, B::= 1/ 0C / 1C, C::= 1)}

o hay reglas compresoras

n solo símbolo a la izda

n solo símbolo a la izda

n libertad a la dcha → G regular lineal derecha G3 LD

= {11, 101, 111}

Necesita un lenguaje tan sencillo de una gramática alto nivel en la rarquía de Chomsky? NO, es una G3 la que habría que construir

rarquía de Chomsky

Lenguaje se denomina de tipo i (i= 0, 1, 2, 3) si existe una tipo i tal que L= L(Gi)

rarquía: relación de inclusión:

$$\boxed{G3 \subset G2 \subset G1 \subset G0}$$

ramáticas Formales. Gramáticas Regulares

ATICAS EQUIVALENTES

lineal derecha con reglas del tipo A::= aS

una G3' lineal derecha equivalente sin reglas del tipo A::= aS

lineal derecha

una G3' lineal izquierda equivalente

ramáticas Formales. Gramáticas Regulares
ramáticas Equivalentes

lineal derecha con reglas del tipo A::= aS

na G3' lineal derecha equivalente sin reglas del tipo A::= aS

cedimiento de transformación

se añade un nuevo símbolo en el alfabeto Σ_N

$\forall$ S::= x, donde $x \in \Sigma^+$, se añade una regla B::= x

Se transforman las reglas A::= aS (que desaparecen) en reglas del tipo A::= aB.

Las reglas tipo S::= λ no se ven afectadas por este algoritmo.

ramáticas Formales. Gramáticas Regulares
ramáticas Equivalentes

ineal derecha ∃ una G3' lineal izquierda equivalente Procedimiento

nsformación

eación de un grafo dirigido:

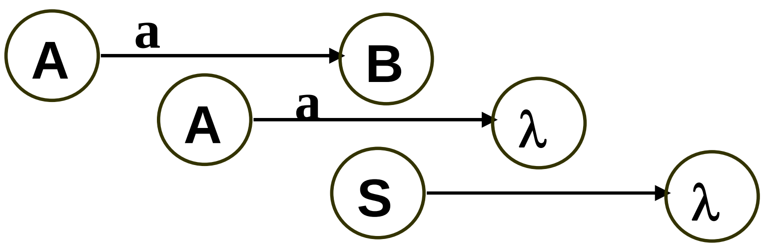
úmero de nodos = C(Σ_N) +1, cada nodo etiquetado con símbolos de Σ_N y

otro con λ

ada A::=aB $\in$P

ada A::=a $\in$P

S::= λ $\in$P



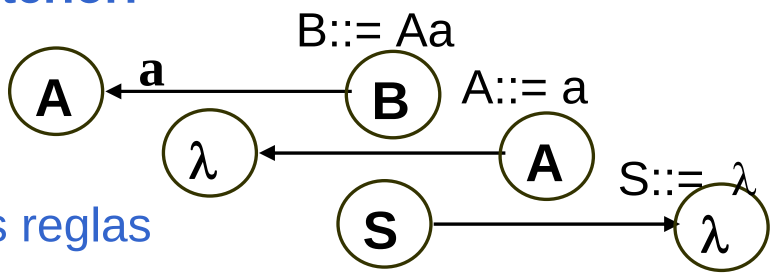
nsformación del grafo dirigido anterior:

cambiar etiquetas de S y λ

tir sentido de todos arcos

acer el camino y generar las nuevas reglas

pretar el grafo para obtener la G3LI equivalente



ramáticas Formales. Gramáticas Regulares
ramáticas Equivalentes

icio: alfonseca pg 58: Sea la G3 LD, G1 = ({0,1}, {S,B}, S, P),

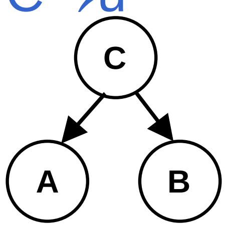
e

{S::= 1B /1, B::= 0S}, hallar la G3 LI equivalente.

mostrar que el lenguaje descrito por ambas es el mismo.

ramáticas Formales. Árboles de derivación

erivaciones de las G tipo 1, 2 y 3 les corresponde un árbol
ente, También llamado **árbol sintáctico o parse tree**
enta las producciones aplicadas durante la generación de una
, es decir, su estructura de acuerdo con la G
rbol <u>ordenado</u> y <u>etiquetado</u> que se construye:
raíz se denota por el axioma de G
a derivación directa se representa por un conjunto de ramas
e salen de un nodo dado (parte izquierda de la P)
licar una regla un símbolo de la parte izda queda sustituido por
palabra *u* de la parte dcha. Por cada uno de los símbolos de u
dibuja una rama que sale del NT a ese símbolo: pej sea u=AB y
la P= C→u



El símbolo más a la izda en la P queda más a la izda en el árbol

ramáticas Formales. Árboles de derivación

a G1, además, se debe conservar el contexto
cada rama:

nodo de partida se llama **padre** del nodo final
nodo final es **hijo** del nodo padre
os nodos hijos del mismo padre se llaman **hermanos**
n nodo es **ascendiente** de otro si es su padre o ascendiente de su
adre
n nodo es **descendiente** de otro si es su hijo o descendiente de
us hijos

rgo del proceso de construcción del árbol, los nodos finales de
paso sucesivo, leídos de izda a dcha dan la **forma sentencial**
da por la derivación representada por el árbol.

junto de las hojas del árbol (nodos denotados por símbolos
ales o λ) leídos de izda a dcha nos dan la **sentencia** generada por
vación

ramáticas Formales. Árboles de derivación

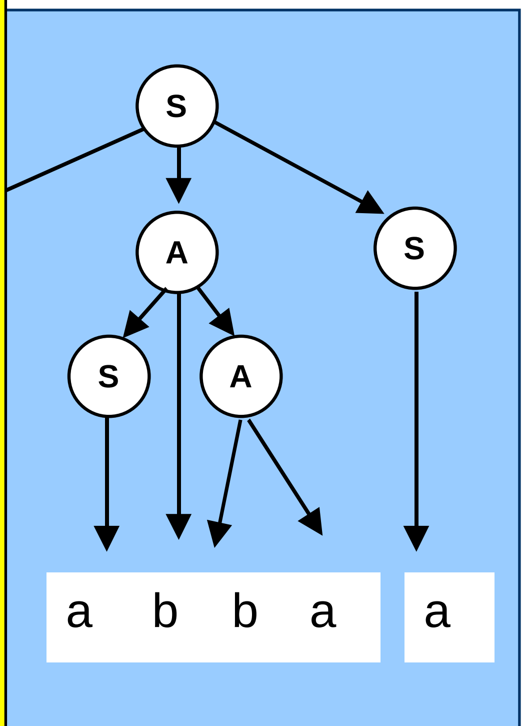
a la G = ({a,b}, {A,S}, S, {S::= aAS / a, A::= SbA / SS / ba}).

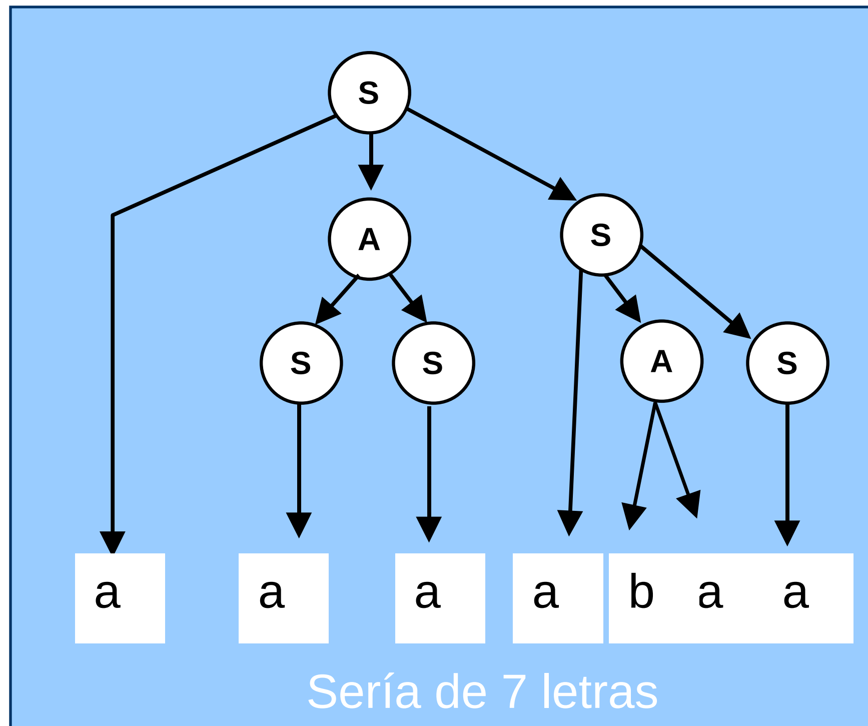
ar un árbol de derivación para una sentencia de 6 letras.

amáticas Formales. Árboles de derivación

a la G = ({a,b}, {A,S}, S, {S::= aAS / a, A::= SbA / SS / ba}).

ar un árbol de derivación para una sentencia de 6 letras.

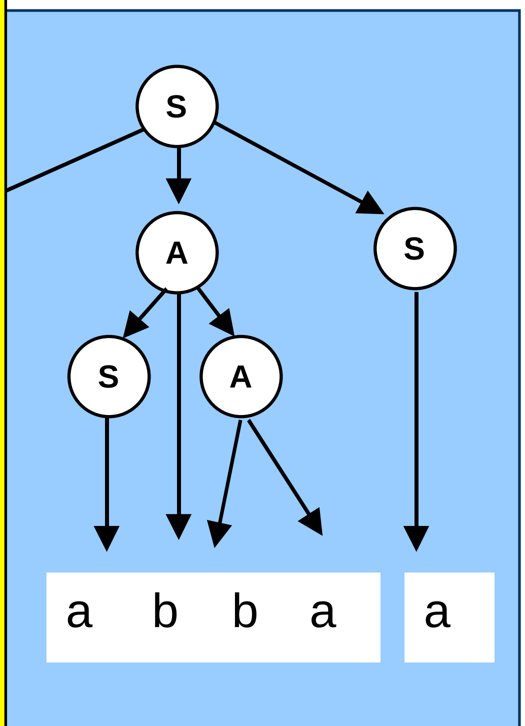


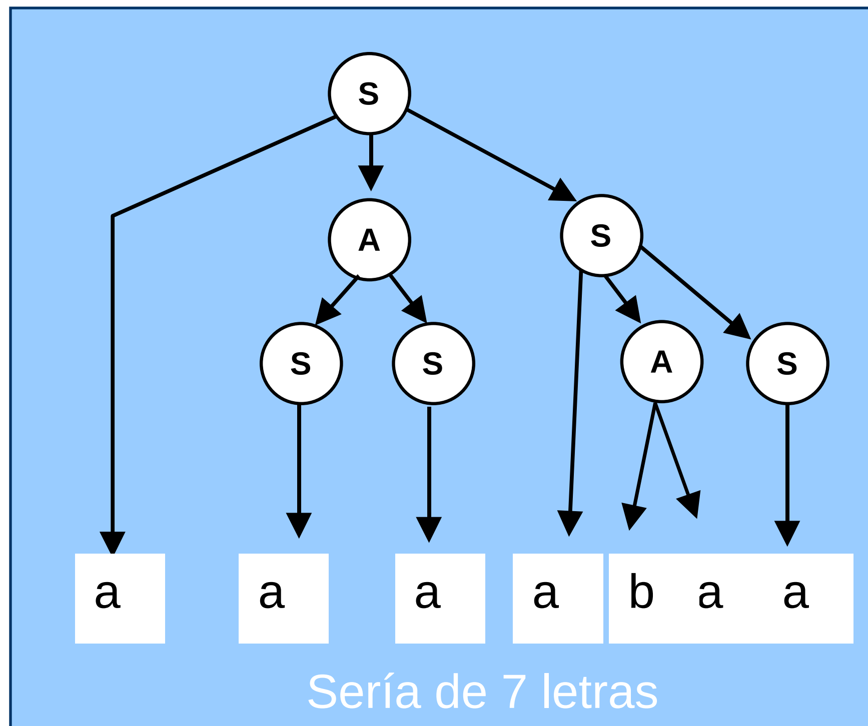


…amáticas Formales. Árboles de derivación

…a la G = ({a,b}, {A,S}, S, {S::= aAS / a, A::= SbA / SS / ba}).

…ar un árbol de derivación para una sentencia de 6 letras.





ramáticas Formales. Árboles de derivación

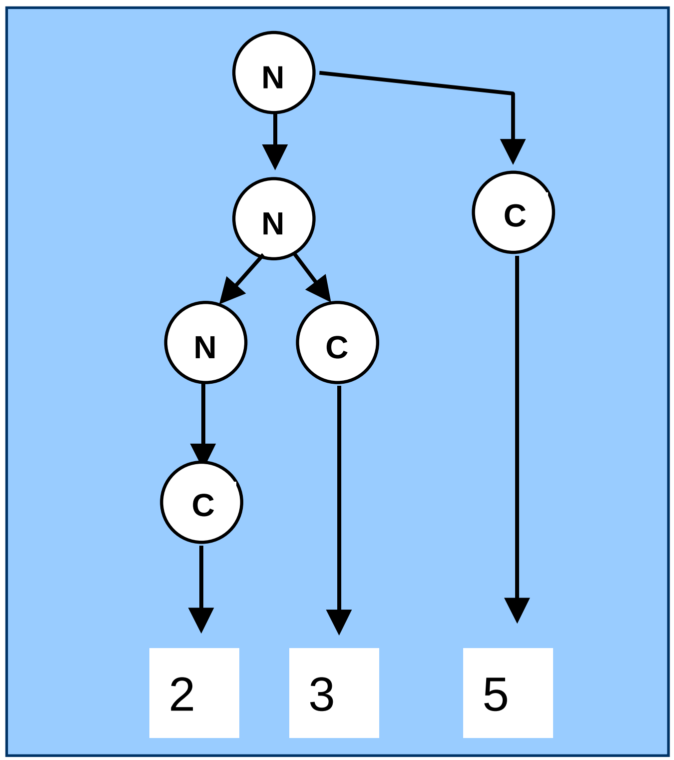
la G = ({0,1,2,3,4,5,6,7,8,9}, {N,C}, N, {N ::= NC / C, C ::= 0 /.../ 9})

r un árbol de derivación para N $\rightarrow$ 235

ramáticas Formales. Árboles de derivación

la G = ({0,1,2,3,4,5,6,7,8,9}, {N,C}, N, {N ::= NC / C, C ::= 0 /.../ 9})

r un árbol de derivación para N → 235



ramáticas Formales. Árboles de derivación

la G = ({0,1}, {A,B}, A, {A::= 1A / OB, B::= 0B / 0}) Una de cuyas aciones válidas es: A $\rightarrow$ 1A $\rightarrow$ 1OB $\rightarrow$ 100B $\rightarrow$ 1000

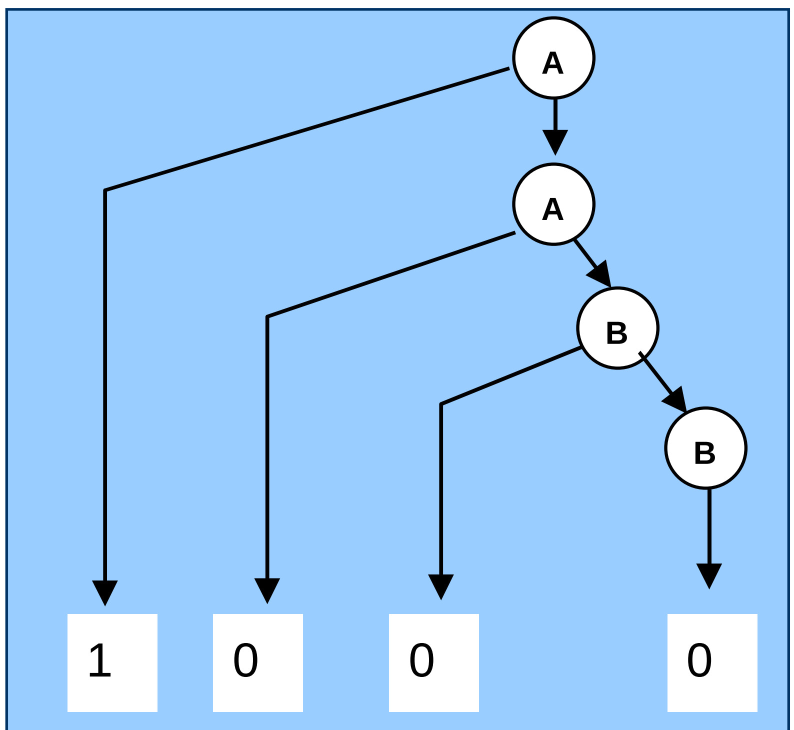
ar un árbol de derivación para A $\rightarrow$ 1000

ramáticas Formales. Árboles de derivación

la G = ({0,1}, {A,B}, A, {A::= 1A / OB, B::= 0B / 0}) Una de cuyas

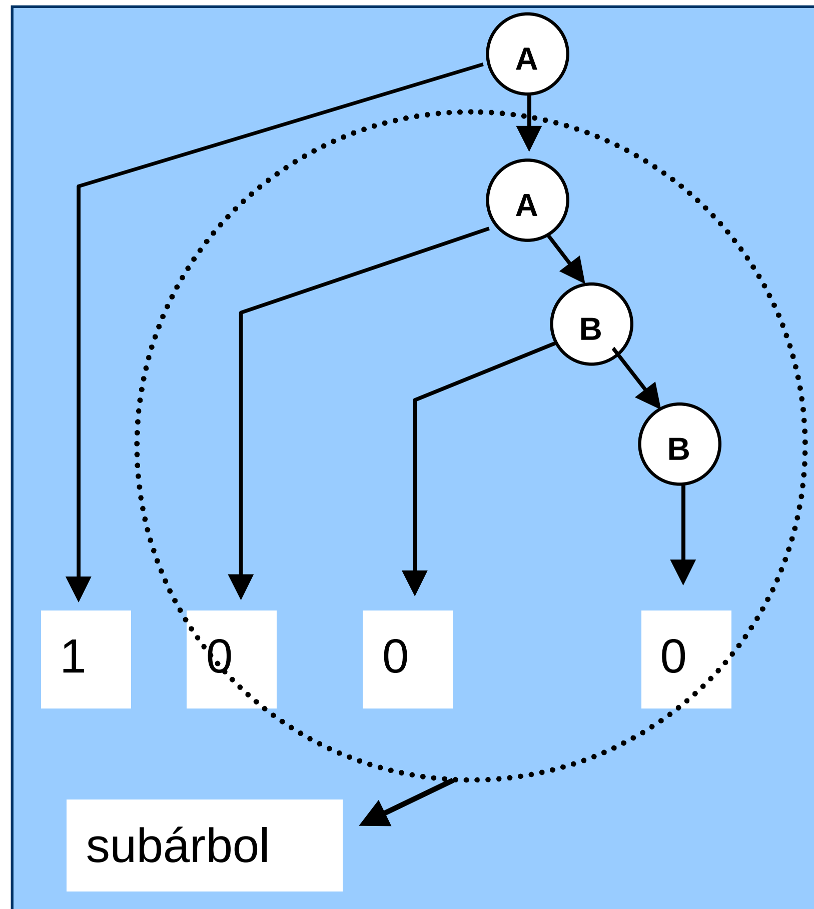
aciones válidas es: A → 1A → 1OB → 100B → 1000

ar un árbol de derivación para A → 1000



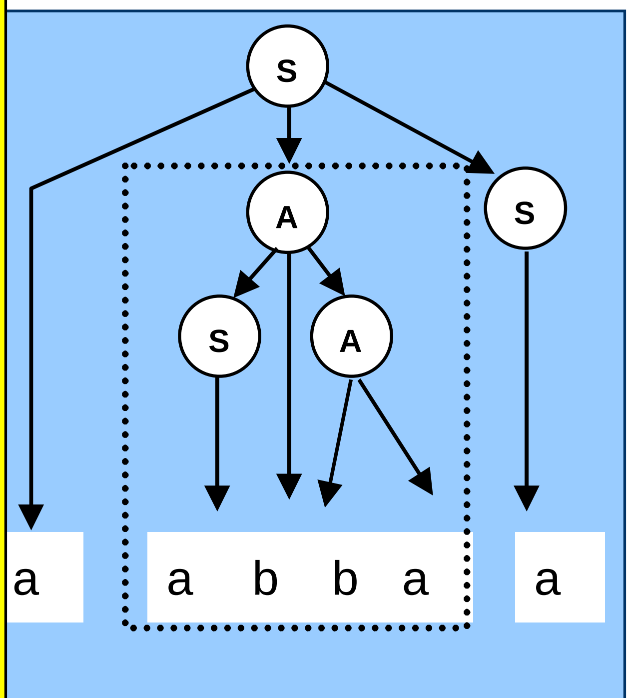
ramáticas Formales. Subárboles de derivación

un árbol A correspondiente
derivación, se llama
rbol de A al árbol cuya raíz
nodo cualquiera de A, cuyos
son todos los
ndientes de la raíz del
bol en A y cuyas ramas son
las que unen dichos nodos
si en A.



ramáticas Formales. Subárboles de derivación

rema: las hojas de un subárbol, leídas de izda a dcha, forman una e respecto al símbolo NT raíz del subárbol



abba es una frase de la forma sentencial aabbaa respecto del símbolo A

ramáticas Formales. Ambigüedad

o relacionado con el de árbol de derivación:

a sentencia puede obtenerse en una G por medio de dos o más les de derivación diferentes, **la sentencia es ambigua**

G es ambigua si contiene al menos una sentencia ambigua

que una G sea ambigua, es posible que el lenguaje que describe ea ambiguo [Floyd 1962] ⇒ es posible encontrar una G valente que no lo sea

en lenguajes para los que NO es posible encontrar G no iguas ⇒ Lenguajes Inherentemente Ambiguos [Gross 1964]

opiedad de ambigüedad es indecidible. Tan solo es posible ontrar condiciones suficientes que aseguren que una G es no igua

cidible: no existe un algoritmo que acepte una G y determine con eza y en un tiempo finito si una G es ambigua o no.

ramáticas Formales. Ambigüedad

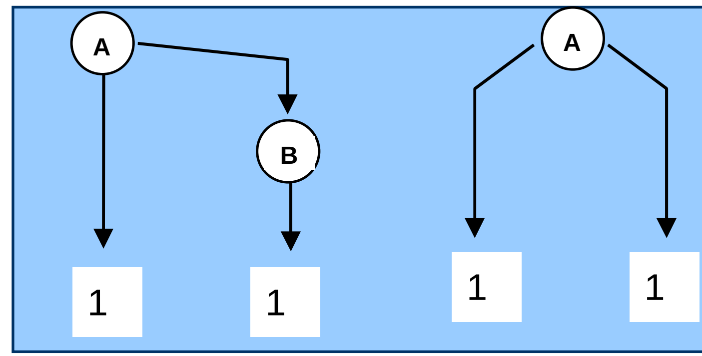
efiniciones se deduce que existen 3 niveles de ambigüedad:

ntencia: una sentencia es ambigua si puede obtenerse por

dio de dos o más árboles de

erivación diferentes

= ({1}, {A,B}, A, {A::= 1B / 11, B::= 1})



mática: es ambigua si contiene al menos una sentencia

bigua, ej: la G anterior

nguaje: un L es ambiguo si existe una G ambigua que lo

nera, ej: L = {11} es ambiguo, pero como todos los lenguajes

ambiguos, es un término que no da información.

ramáticas Formales. Ambigüedad

guajes Inherentemente Ambiguos: para los que NO es posible ntrar G no ambiguas ⇒ ejemplo $L = \{\{a^n b^m c^m d^n\} \cup \{a^n b^n c^m d^m\} \ / \ m,n \geq 1\}$

mplo: L = {11} no es inherentemente ambiguo

= ({1}, {A}, A, {A..=11})

ramáticas Formales. Ambigüedad

a G = ({a,b}, {A,B,S}, S, P)

= bA / aB

= bAA / a / aS

:= b / bS / aBB}

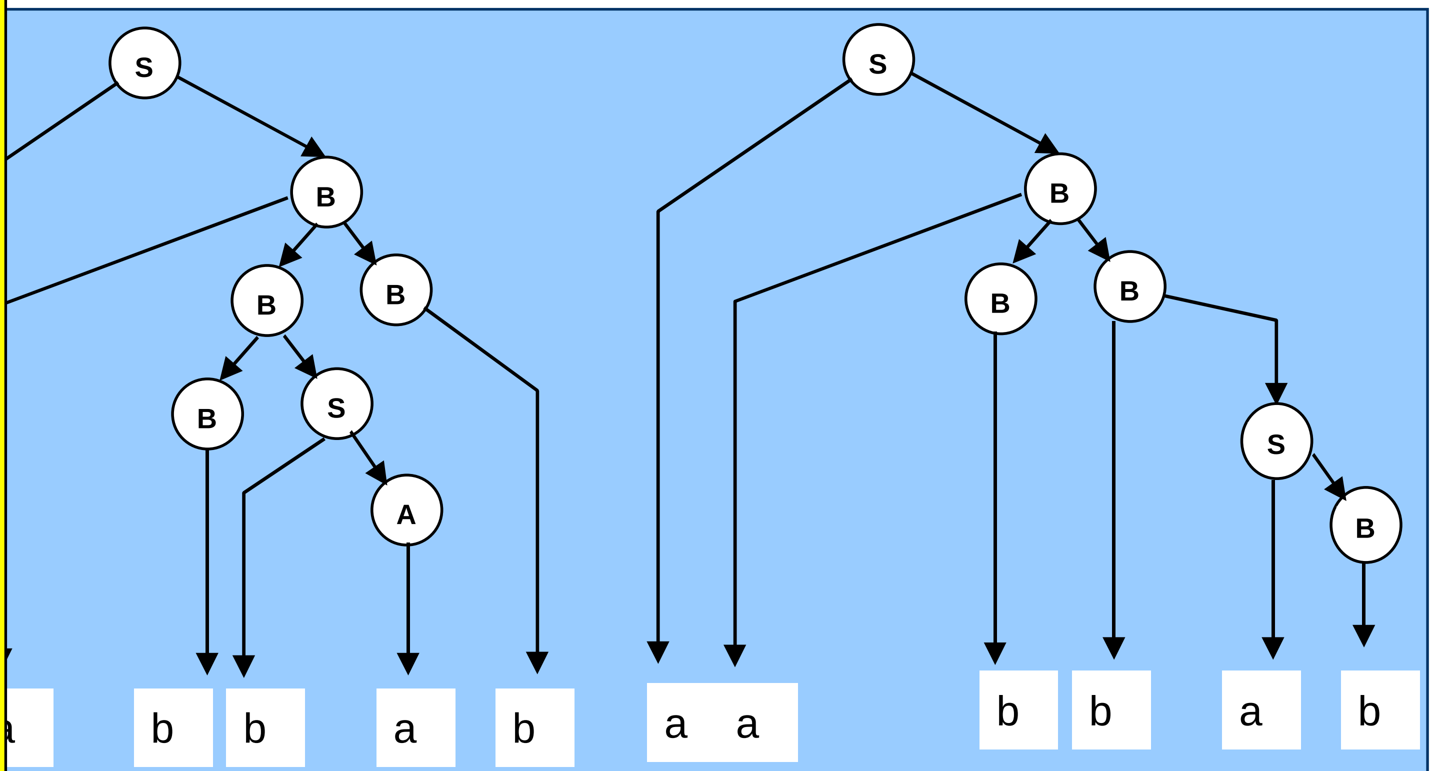
strar que es una G ambigua al serlo la sentencia “aabbab”

ramáticas Formales. Ambigüedad

a G = ({a,b}, {A,B,S}, S, P)

= bA / aB, A::= bAA / a / aS, B ::= b / bS / aBB}

strar que es una G ambigua al serlo la sentencia “aabbab”



ramáticas Formales.
ramáticas Independientes del Contexto

ramáticas de tipo 2 en la jerarquía de Chomsky.

a: tema 9

cia:

as empleadas en la definición de lenguajes de programación y en

mpilación de los mismos

pilador
- analizador → aceptador de la G
  - explorador (AF)
  - reconocedor (AP)
- generador de código → Traductor

ramáticas Formales.
ramáticas Independientes del Contexto

lenguajes generados por gramáticas del tipo 2 de la jerarquía omsky se les denomina:

Lenguajes independientes del contexto o lenguajes de contexto libre

Se representan como L(G2)

Existen algoritmos que permiten reconocer si un L(G2) es vacío, finito o infinito

ramáticas Formales.
ramáticas Independientes del Contexto

uaje, vacío o no?:

a G2, $m = C(\Sigma_{NT})$, **L(G2)** $\neq \phi$ si $\exists\, x \in L(G2)$ tal que x puede erarse con un árbol de derivación en el que todos los caminos en longitud $\leq m$

eneran todos los árboles de derivación con caminos $\leq m = C(\Sigma_{NT})$ ante el algoritmo:

onjunto de árboles con longitud 0 (un árbol con S como raíz y sin as)

partir del conjunto de árboles de longitud n, generamos el junto de longitud $n+1 < m+1$ aplicando al conjunto de partida una ducción que no haga duplicarse algún NT en el camino siderado

e aplica el paso b) recursivamente hasta que no puedan erarse más árboles con caminos de longitud $\leq m$. Al ser m y el nero de reglas de P finito $\Rightarrow$ el algoritmo termina

L(G2) = ϕ si ninguno de los árboles genera una sentencia

ramáticas Formales.
ramáticas Independientes del Contexto

plo de L(G2) = ϕ

G = ({a,b}, {A,B,C,S}, S, P)

$m \leq C(\Sigma_{NT}) = 4$

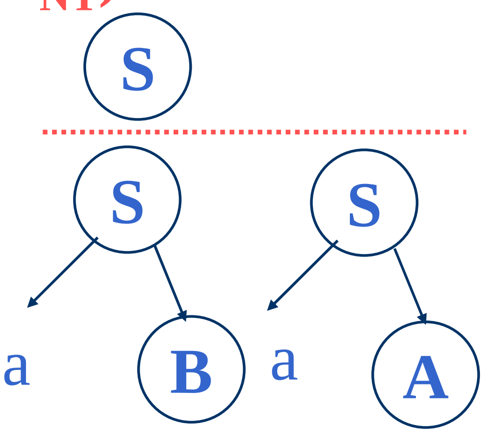
= {

S ::= aB / aA

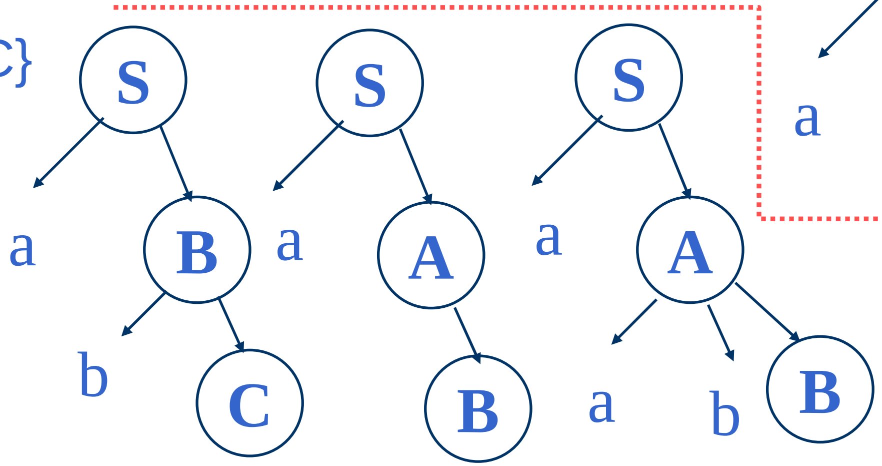
A ::= B / abB

B ::= bC}

1. m=0



2. m=1



m=2

m=3

No se generan sentencias y salen NT ya obtenidos
Lenguaje vacío

ramáticas Formales.
ramáticas Independientes del Contexto

G2) es no vacío, comprobar si **L(G2) =** $\propto$

onstruye un grafo cuyos nodos están etiquetados con los
olos de (Σ_{NT}) mediante el algoritmo:

a) si $\exists$ una producción A::= α B β, se crea un arco de A a B donde A,B $\in \Sigma_{NT}$ y $\alpha,\beta \in \Sigma^*$

b) si no existen ciclos en el grafo el L(G2) = finito

c) L(G2) = $\propto$ si existen ciclos accesibles desde el axioma que corresponden a derivaciones de la forma A $\rightarrow$+ α A β, donde /α/+ /β/ >0 (que no sean λ las dos a la vez).

G2) $\neq \propto$ **si no hay ciclos en el grafo**

amáticas Formales.
amáticas Independientes del Contexto

emplo de **L(G2) =** ∞

Sea G = ({a,b,c}, {A,B,C,S}, S, P)

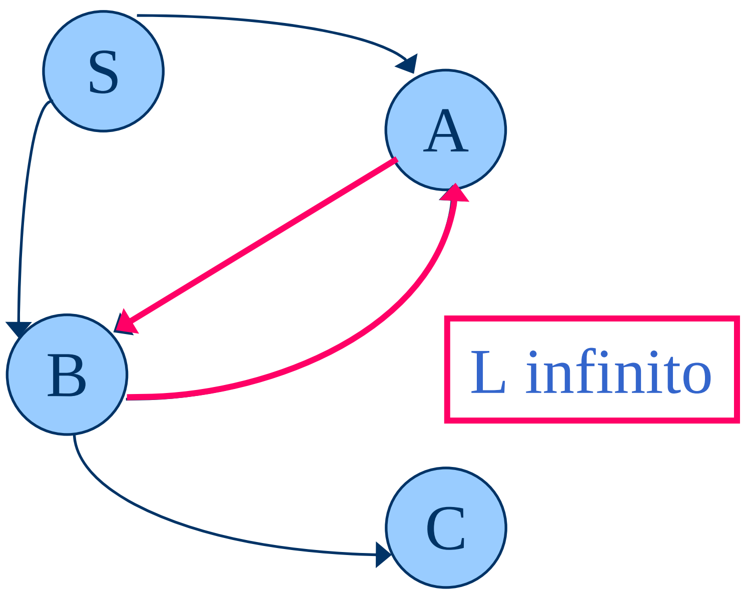
P = {

S ::= aB / aA

A ::= abB

B ::= bC / aA

C ::= c }



ramáticas Independientes del Contexto
ramáticas Bien Formadas

sformación de una G dada en otra equivalente cuyas reglas de ucción estén en un formato carente de imperfecciones:

Limpieza de Gramáticas

1. Reglas Innecesarias
2. Símbolos Inaccesibles
3. Reglas Supérfluas

Eliminación de símbolos no generativos

Eliminación de reglas no generativas

Eliminación de reglas de redenominación

ramáticas Independientes del Contexto
ramáticas Bien Formadas

a de Gramáticas

las Innecesarias: las reglas **A::= A $\in$P** son innecesarias y cen que G sea ambigua $\Rightarrow$ eliminarlas

bolos Inaccesibles: sea U::= x $\in$P, donde U$\in \Sigma_N \neq$ S y no rece en la parte derecha de ninguna otra regla de producción ice que U es inaccesible.

odo símbolo U$\in \Sigma_N$ no inaccesible debe cumplir S $\rightarrow$ * xUy.

Eliminación de símbolos inaccesibles:

a) matriz booleana (alfonseca cap. 3)

b) grafo análogo al de L(G2) = $\propto$. Los símbolos inaccesibles no serán alcanzables desde el axioma.

Se eliminan así como las reglas que los contengan

ramáticas Independientes del Contexto
ramáticas Bien Formadas

Limpieza de Gramáticas

2. Símbolos Inaccesibles: ejemplo:

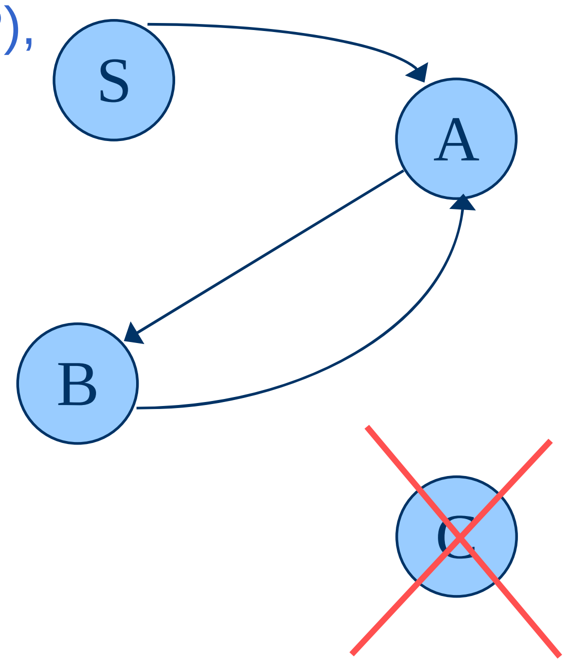
sea la G = ({a,b,c}, {S,A,B,~~C~~}, S, P),

donde P = {S::= aA

A ::= Bc

B::= bA

~~C::= c~~}



ramáticas Independientes del Contexto
ramáticas Bien Formadas

za de Gramáticas

eglas Supérfluas: son aquellas que no contribuyen a la rmación de palabras x $\in \Sigma_T^*$.

Todo símbolo no supérfluo debe cumplir U $\rightarrow$+t, tal que t $\in \Sigma_T^*$

algoritmo: es un algoritmo recursivo de marcado

a) marcar los NT para los que existe una regla U ::= x donde x $\in \Sigma^*$ (es una cadena de T o λ, o en pasadas sucesivas contiene NT marcados)

b) si todos los NT están marcados $\Rightarrow$ no existen símbolos supérfluos y fin

c) si la última vez que se pasó por el paso a) se marcó un NT, volver al paso a).

d) todo A $\in \Sigma_{NT}$ no marcado es supérfluo.

ramáticas Independientes del Contexto
ramáticas Bien Formadas

mpieza de Gramáticas

Reglas Supérfluas:

ejemplo: sea G= ({e,f}, {S,A,B,C,D}, S, P)

donde P= {

S::= Be

A::= Ae /e

B::= Ce / Af

C::= Cf

D::= f}

1ª pasada:

D::= f y A::= e

2ª pasada:

B::= Af

3ª pasada:

S::= Be

ramáticas Independientes del Contexto
ramáticas Bien Formadas

mpieza de Gramáticas

Reglas Supérfluas:

ejemplo: sea G= ({e,f}, {S,A,B,~~C~~,D}, S, P)

donde P= {

S::= Be

A::= Ae /e

B::= ~~C~~e / Af

C::~~=~~Cf

D::= f}

1ª pasada:

D::= f y A::= e

2ª pasada:

B::= Af

3ª pasada:

S::= Be

Sin marcar: C, que se elimina, así como sus reglas

ramáticas Independientes del Contexto
ramáticas Bien Formadas

iminación de símbolos no generativos:

Sea G2 = (Σ_T, Σ_N, S, P), $\forall$**A**$\in\Sigma_{\mathbf{N}}$ construiremos la gramática G(A), donde A es el axioma. Si **L(G(A))=** $\phi \Rightarrow$ **A es símbolo no generativo** y se puede eliminar, así como todas las reglas que lo contengan, obteniéndose otra G2 equivalente.

ramáticas Independientes del Contexto
ramáticas Bien Formadas

Eliminación de reglas no generativas: son $A ::= \lambda$ $(A \neq S)$

Si $\lambda \in$ **L(G)**: se añade $S ::= \lambda$ **y** $\forall\, A \in \Sigma_{Nt}$ $(A ::= \lambda$ $A \neq S)$ y $\forall$ regla de G de la forma $B ::= xAy$, añadiremos en P' una regla de la forma $B ::= xy$, excepto en el caso en que $x = y = \lambda$

...amáticas Independientes del Contexto
...ramáticas Bien Formadas

...ación de reglas de redenominación: son reglas del tipo A::=B

...tmo:

...G' equivalente contiene todas las reglas excepto las tipo

A::= B

...$\in \Sigma_{NT}$ | $A \rightarrow^* B$ en G y $\forall$ (B::= x) $\in$P donde $x \notin \Sigma_{NT} \Rightarrow$

P' = P + {A::= x}

ramáticas Independientes del Contexto
ramáticas Bien Formadas

= ({0,1}, {S,A,B,C},S, P),

donde P = {S::= AB / 0S1 / A / C,A ::= 0AB / B::= B1 / λ}

- **No hay innecesarias**

- **Inaccesibles**

ramáticas Independientes del Contexto
ramáticas Bien Formadas

= ({0,1}, {S,A,B,C},S, P),

donde P = {S::= AB / 0S1 / A / C,A ::= 0AB / B::= B1 / λ}

- **No hay innecesarias**
- **Inaccesibles**
- **R. Supérfluas y S. No generativos**
- **R. No generativas**
- **R. De redenominación**

ramáticas Independientes del Contexto
ramáticas Bien Formadas

= ({0,1}, {S,A,B,C},S, P),

donde P = {S::= AB / 0S1 / A / ~~C~~,A ::= 0AB / B::= B1 / λ}

- **No hay innecesarias**
- **Inaccesibles: no hay**
- **R. Supérfluas y S. No generativos:**
- **R. No generativas: se quedan así las**

P' = {S::= A / B / 0S1/λ, A::= 0AB/ 0A /0B /0, B::=B1/1**}**

- **R. De redenominación**

2:

A / B, se sustituyen por sus partes derechas:

$\boxed{\text{S::= 0AB/ 0A /0B /0/ B1/1/ 0S1}}$

ramáticas Independientes del Contexto
ramáticas Bien Formadas

= ({0,1}, {S,A,B,C},S, P),

donde P = {S::= AB / 0S1 / A / C,A ::= 0AB / B::= B1 / λ}

= ({0,1}, {S,A,B},S, P),

donde P = {S::= 0AB/ 0A /0B /0/ B1/1/ 0S1/ λ,

A::= 0AB/ 0A /0B /0, B::=B1/1}

Es la gramática bien formada equivalente

ramáticas Independientes del Contexto
ormas Normales

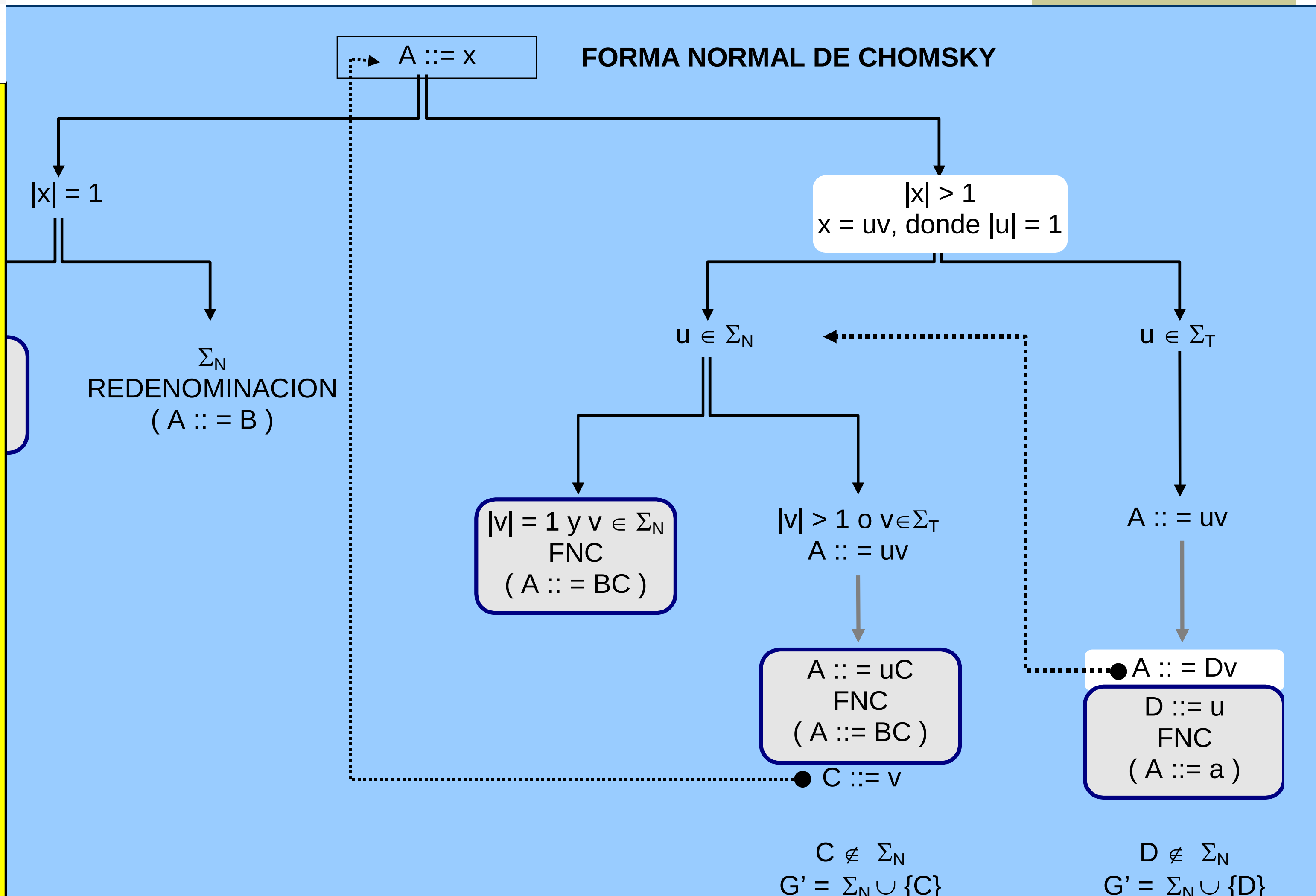
on notaciones que se aplican a las G2:

- Afectan a la forma de las reglas de producción

on dos las que se va a estudiar:

- Forma Normal de Chomsky
- Forma Normal de Greibach

ramáticas Independientes del Contexto
orma Normal de Chomsky



ramáticas Independientes del Contexto
orma Normal de Chomsky

ercicio: Alfonseca pg 212: sea la G = ({0,1}, {S,A,B},S, P),

onde P = {S::= AB / 0S1 / 0AB / 0B / 0A / 0 / B1 / 1 / λ

A ::= 0AB / 0B / 0A / 0

B::= B1 / 1}

Hallar la gramática en FNC equivalente

amáticas Independientes del Contexto
orma Normal de Greibach

s una notación muy interesante para algunos reconocimientos icos. En ella todas las reglas tienen la parte derecha comenzando terminal seguido opcionalmente de uno o varios NT

EMA: todo **L de contexto libre sin** λ puede ser generado por una la que todas las reglas sean de la forma:

$\mathbf{A \rightarrow a\alpha}$ donde $A \in \Sigma_{NT}$ $a \in \Sigma_T$ $\alpha \in \Sigma_{NT}^*$

Si $\lambda \in L$ habrá que añadir **S::=** λ

REMA: toda G2 puede reducirse a otra G2 equivalente sin reglas vas a izquierdas

ramáticas Independientes del Contexto
orma Normal de Greibach

: para transformar una G2 en su equivalente en forma normal de bach:

Eliminar la recursividad a izquierdas

Aplicar el algoritmo de transformación a FNG, verificando en cada paso que no aparezcan nuevas reglas recursivas a izquierdas y si aparecen, eliminándolas con el paso 1

ramáticas Independientes del Contexto
orma Normal de Greibach

nar la recursividad a izquierdas: en un paso (estudiar en varios

s, Isasi Martínez y Borrajo, pg 24):

sea G = ({α,β}, {A}, A, P), donde P = {A::= A α / β}

A → β

A → A α → β α

A → A α → A α α → β α α

A → A α → A α α → ... → β ..... α α

A::= β

A::= βX

X::= α X

X::= α

ramáticas Independientes del Contexto
orma Normal de Greibach

nar la recursividad a izquierdas, resumiendo, sería:

G = $(\{\alpha_1, \alpha_2, \beta_1, \beta_2\}, \{A\}, A, P)$, donde $P = \{A ::= A\,\alpha_1 / A\alpha_2 / \beta_1 / \beta_2\}$

daría:

$A ::= \beta_1 / \beta_2 / \beta_1 X / \beta_2 X$

$X ::= \alpha_1 / \alpha_2 / \alpha_1 X / \alpha_2 X$

·amáticas Independientes del Contexto
ɔrma Normal de Greibach

ɪrmación de G2 bien formada sin RI a FNG:

tablecer una relación de orden parcial en Σ_{NT}

: $\{A_1, A_2, ..., A_n\}$ basándose en: si Ai $\rightarrow$ Aj α, Ai precederá a Aj.

Cuando hay reglas “contradictorias” usar una de ellas para el orden y mirar el resto para ver que conviene más

clasifican las reglas en 3 grupos:

po 1: Ai $\rightarrow$ a α ,donde a $\in\Sigma_T$ y $\alpha \in\Sigma_{NT}^*$

po 2: Ai $\rightarrow$ Aj α donde Ai precede a Aj en el conjunto Σ_{NT} ordenado

po 3: Ak $\rightarrow$ Ai α donde Ai precede a Ak en el conjunto Σ_{NT} ordenado

se transforman las reglas de grupo 3 $\rightarrow$ grupo 2 $\rightarrow$ grupo 1: FNG

$\rightarrow$ Ai α se sustituye **Ai** por la parte dcha de todas las reglas que tienen **Ai** como parte izda

ɛr lo mismo con las de grupo 2

ramáticas Independientes del Contexto
orma Normal de Greibach

ormación de G2 bien formada sin RI a FNG:

uando todas las reglas son de grupo 1, la G está en FNG a falta **minar los símbolos terminales no situados en la cabecera de te derecha**

α **a** β **donde a**$\in\Sigma_T$ **y** $\alpha \neq \lambda$

α **B** β
a