# Autómatas y Lenguajes

3er curso
1er cuatrimestre

Alfonso Ortega: alfonso.ortega@uam.es

# UNIDAD 1: Modelos de cómputo y familias de lenguajes

## TEMA 2: Gramáticas independientes del contexto y autómatas a pila

## Introducción a las gramáticas formales, gramáticas independientes del contexto

# Introducción a las gramáticas

## Concepto intuitivo

- Los alumnos ya conocen las gramáticas formales, son las mismas que se estudiaron en el colegio para analizar sintácticamente las oraciones de la lengua castellana.
- Recuérdese que se podía considerar que:
  - Una **oración** está compuesta por **sujeto** y **predicado**.
  - El **sujeto** puede ser un **sintagma nominal**.
  - El **predicado** puede componerse de **verbo** y **sintagma nominal**.
  - Algunos **sintagmas nominales** son **Pedro**, **Juan**, **carne**, **leche** y **agua**.
  - Algunas formas válidas de **verbos** pueden ser **come** y **bebe**

# Introducción a las gramáticas

## Ejemplo introductorio

- Y se podría analizar la oración “Juan bebe leche” de la siguiente manera

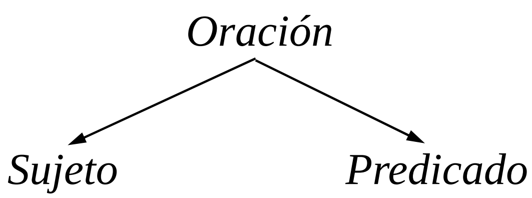
*Oración*



# Introducción a las gramáticas

## Ejemplo introductorio

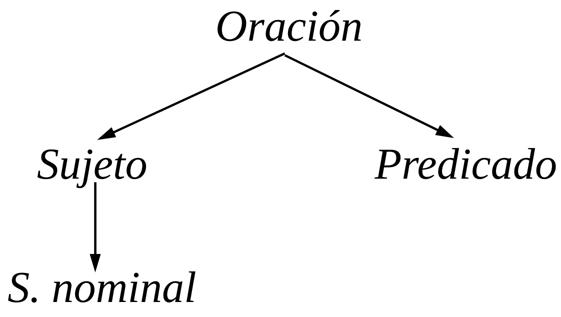
- Y se podría analizar la oración “Juan bebe leche” de la siguiente manera



# Introducción a las gramáticas

## Ejemplo introductorio

- Y se podría analizar la oración “Juan bebe leche” de la siguiente manera

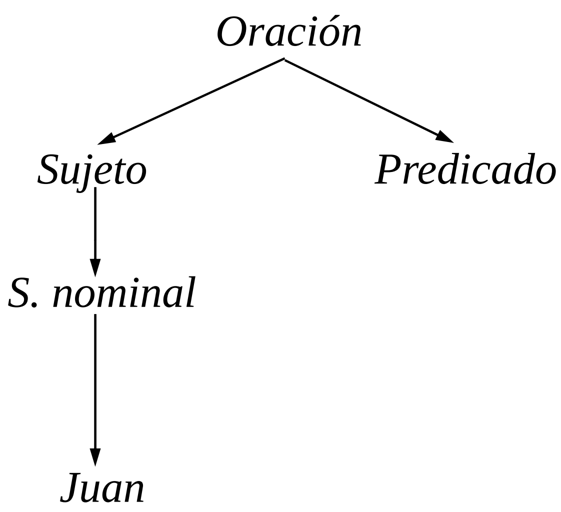





# Introducción a las gramáticas

## Ejemplo introductorio

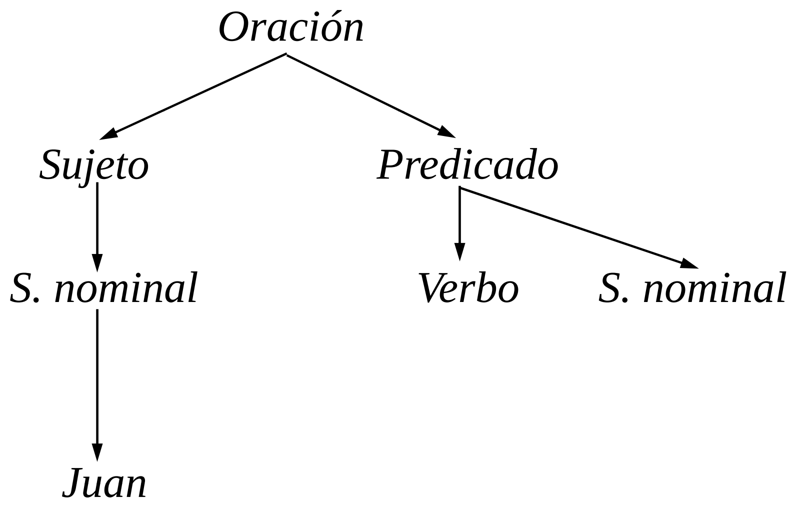
- Y se podría analizar la oración “Juan bebe leche” de la siguiente manera



# Introducción a las gramáticas

## Ejemplo introductorio

- Y se podría analizar la oración “Juan bebe leche” de la siguiente manera

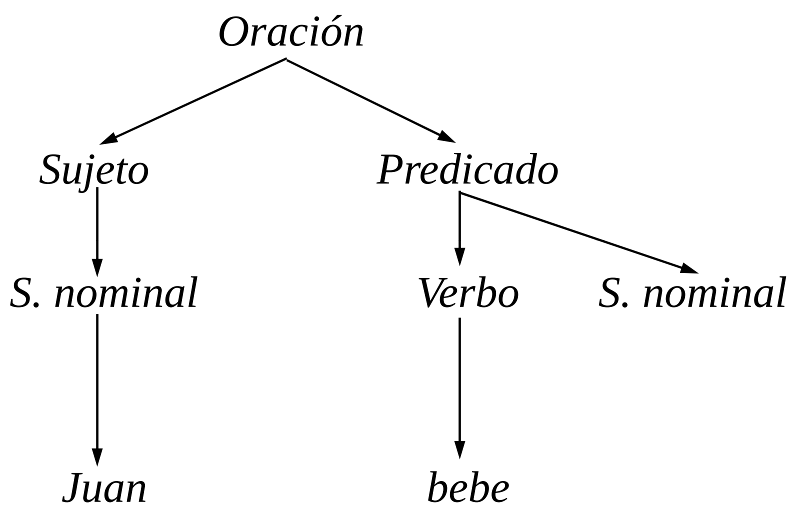





# Introducción a las gramáticas

## Ejemplo introductorio

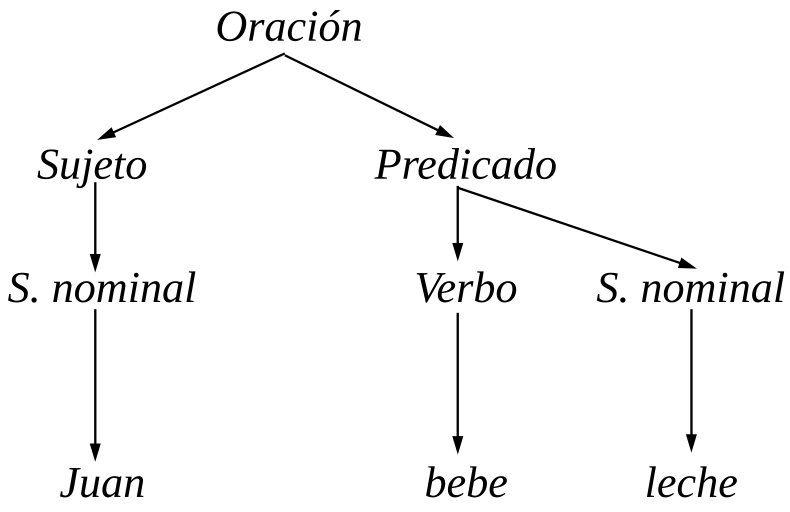
- Y se podría analizar la oración “Juan bebe leche” de la siguiente manera



# Introducción a las gramáticas

## Ejemplo introductorio

- Y se podría analizar la oración “Juan bebe leche” de la siguiente manera






# Derivaciones y otros conceptos

## Ejemplo introductorio

- Informalmente, cada uno de los pasos que ha llevado de *Oración* a *Juan bebe leche* se ha producido por medio de una **derivación** que ha cambiado uno de los componentes por alguna de las opciones permitidas para él:
  - *Oración* por *sujeto predicado*
  - *Sujeto* por *S. nominal*
  - *S. nominal* por *Juan*
  - *Predicado* por *verbo* y *S. nominal*
  - *Verbo* por *bebe*
  - *S. nominal* por *leche*
- Cada una de las expresiones que declara cómo puede cambiarse un componente por otros se llama **regla de producción**.
- La disposición gráfica de las derivaciones en la forma de la página anterior se llama **árbol de derivación**.

## Ejemplo introductorio

- En cada instante, las hojas del árbol leídas de izquierda a derecha componen lo que se conoce como **forma sentencial** y son
  - *Oración*
  - *sujeto predicado*
  - *S. nominal predicado*
  - *Juan predicado*
  - *Juan verbo s. nominal*
  - *Juan bebe s. nominal*
  - *Juan bebe leche*
- De ellas, solamente la última es una oración correcta castellana, obsérvese que está compuesta por componentes para los que no se han definido cambios, es decir, en términos de derivaciones, no tienen derivaciones definidas para ellos. Estas formas sentenciales son las **sentencias** del lenguaje que estamos considerando.





## Ejemplo introductorio

- En ese sentido, los símbolos para los que no se definen derivaciones posibles se llaman **terminales**, en las hojas del árbol donde haya uno de estos símbolos el proceso de derivación **termina**. En este ejemplo los terminales serían
  - *Pedro*
  - *Juan*
  - *agua*
  - *leche*
  - *carne*
  - *bebe*
  - *come*

# Derivaciones y otros conceptos

## Ejemplo introductorio

- En oposición a los terminales, aquellos componentes para los que haya definidas posibles derivaciones se llaman **no terminales**. En este ejemplo son los siguientes
  - *oración*
  - *sujeto*
  - *predicado*
  - *s. nominal*
  - *verbo*
- De ellos, sólo uno representa la estructura completa que se está analizando (*oración*) y debería ser la raíz de todos los árboles de derivación de la gramática. El no terminal que cumple estas condiciones se llaman **símbolo de inicio** o **axioma**.



# Gramática formal

## Definición

- Una gramática es una cuádrupla

$$G=(\Sigma_T,\Sigma_N,S,P)$$

- Donde:
  - $\Sigma_T$ es el **alfabeto de símbolos terminales**.
  - $\Sigma_N$ es el **alfabeto de símbolos no terminales**.
  - Y se cumple que $\Sigma_T \cap \Sigma_N = \Phi$, es decir, estos dos alfabetos no comparten ningún símbolo.
  - $S \in \Sigma_N$ es el **axioma** de la gramática.
  - $P$ es un **conjunto finito de reglas de producción** de la forma

$$u::=v \quad ,, u\in\Sigma^+, v\in\Sigma^*, u=xAy, \quad x,y\in\Sigma^*, \quad A\in\Sigma_N$$

  - Es decir,
    - en la parte izquierda puede haber una cadena de símbolos con al menos un no terminal
    - a la derecha puede haber una cadena de símbolos que puede ser vacía.

# Gramática formal

## Definición

- Observación:
  - Se suele considerar la unión de todos los símbolos de la gramática y llamarlo simplemente $\Sigma$, es decir, se cumple

$$\Sigma=\Sigma_T \cup \Sigma_N$$



# Gramática formal

## Ejemplo

- Observaciones:
  - Es frecuente considerar el alfabeto compuesto por todos los símbolos de la gramática y llamarlo simplemente $\Sigma$, es decir:

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

  - Donde $P$ se define de la siguiente manera:

$$\begin{aligned} P=\{&N::=CN\,, \\ &N::=C\,, \\ &C::=0\,, \\ &C::=1\,, \\ &C::=2\,, \\ &\ldots \\ &C::=9\} \end{aligned}$$

# Notación BNF

## Definición y ejemplo

- La notación BNF (***B***ackus ***N***aur ***F***orm) permite una representación más condensada para las reglas que comparten la parte izquierda separando todas las partes derechas con el símbolo |.
- La gramática del ejemplo anterior se podría representar con la notación BNF de la siguiente manera

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

  - Donde $P$ se define de la siguiente manera:

$$P=\{N::=CN \mid C,$$
$$C::=0 \mid 1 \mid 2 \mid \cdots \mid 9\}$$



# Derivación directa

## Definición

- Dado
  - un alfabeto $\Sigma$
  - P un conjunto de reglas sobre el alfabeto
  - dos palabras del alfabeto $v,w\in\Sigma^*$
- Se dice que $w$ **es derivación directa de** $v$ o que $v$ **produce directamente** $w$ o que $w$ **se reduce directamente a** $v$ mediante P y se representa como $v\Rightarrow w$ si se puede obtener una a partir de la otra aplicando la transformación de alguna regla de producción en algún punto de $v$. Es decir
  - $\exists\ x::=y \in \mathrm{P}$
  - $\exists\ z,u \in\Sigma^* \mid v=zxu,\ w=zyu$

# Derivación directa

## Ejemplos

- Si consideramos el alfabeto $\Sigma_{\tilde{n}}=\{a,b,c,...,z\}$
  - La producción $me::=ba$
  - Permite la derivación directa $camello \Rightarrow caballo$

- Si se considera la gramática anterior

  $G=(\Sigma_T=\{0,1,2,3,4,5,6,7,8,9\},\Sigma_N=\{N,C\},N,P=\{N::=CN \mid C,\ \ C::=0 \mid 1 \mid 2 \mid \cdots \mid 9\})$
  - Se pueden escribir las siguientes derivaciones directas

    $N \Rightarrow CN$

    $CN \Rightarrow CCN$

    $CCN \Rightarrow CCC$

    $CCC \Rightarrow 2CC$

    $2CC \Rightarrow 21C$

    $21C \Rightarrow 210$



# Derivación

## Definición

- Dado
  - un alfabeto $\Sigma$
  - un conjunto de reglas de producción para palabras de ese alfabeto $P$
  - dos palabras del alfabeto $v,w\in\Sigma^*$
- Se dice que ***w* es derivación de *v*** o que ***v* produce *w*** o que ***w* se reduce a *v*** y se representa como $v\Rightarrow+w$ si hay una cadena finita de ($n$) derivaciones directas que lleve desde $v$ a $w$.
- A esa secuencia se la llama **derivación de longitud *n***.

$$v\Rightarrow+w \ \Leftrightarrow\text{(def.)}\ \exists\ u_0, u_1,..., u_n \in\Sigma^* \ ,,\ n>0 \ \mid$$
$$v=u_0 \ , \ u_i\Rightarrow u_{i+1} \ \forall i \ , \ w=u_n$$

- Observe
  - que $\Rightarrow+$ se puede considerar el cierre transitivo de la relación $\Rightarrow$.
  - entonces $\Rightarrow*$ considera, además, la coincidencia entre $v$ y $w$ (reflexivo).

$$v\Rightarrow*w \ \Leftrightarrow\text{(def.)}\ v \Rightarrow +w \vee v=w$$

# Derivación

## Ejemplos

- Si consideramos el alfabeto $\Sigma_{\tilde{n}}=\{a,b,c,...,z\}$, la producción $me::=ba$ permite la derivación

$$camello \Rightarrow^{+} caballo$$

- Si se considera la gramática

$$G=(\Sigma_T=\{0,1,2,3,4,5,6,7,8,9\},\Sigma_N=\{N,C\},N,P=\{N::=CN \mid C,\ C::=0 \mid 1 \mid 2 \mid \cdots \mid 9\})$$

  - Se pueden escribir las siguientes derivaciones

$$N \Rightarrow^{+} 210$$

$$N \Rightarrow^{*} N$$



# Lenguaje generado por una gramática

## Concepto y definición

- Dada una gramática $G=(\Sigma_T,\Sigma_N,S,P)$
- Se define el **lenguaje generado por la gramática** y se representa como $L(G)$ al conjunto de todas las cadenas de símbolos terminales que se puedan derivar a partir del axioma. Es decir, es el conjunto de todas las sentencias de $G$.
- Formalmente

$$L(G)=\{x \mid S \Rightarrow^{*} x \wedge x \in \Sigma_T^{*}\}$$

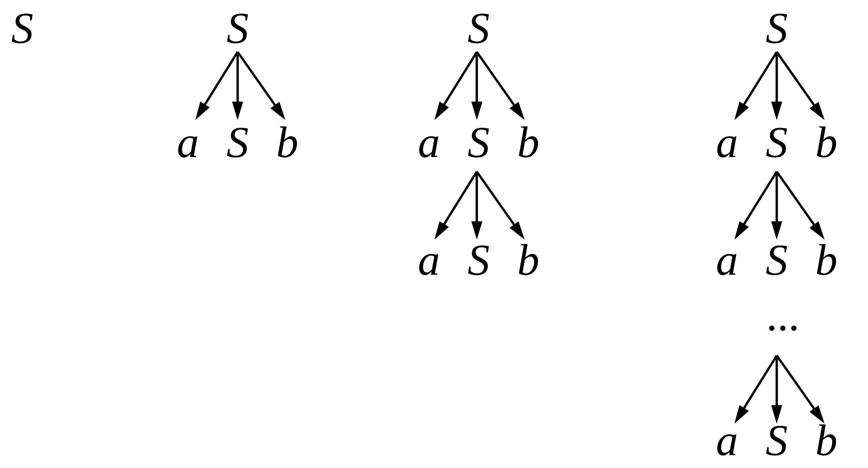
## Lenguaje generado por una gramática

### Ejemplos

- Puede comprobarse que la siguiente gramática genera el lenguaje $\{a^n b^n \mid n \geq 0\}$

$$G=(\Sigma_T=\{a,b\},\Sigma_N=\{S\},S,P=\{S::=aSb \mid \lambda\})$$

$\lambda$     $ab$     $aabb$     $aaa...bbb$






## Gramáticas equivalentes

### Definición

- Se dice que dos gramáticas son equivalentes sin generan el mismo lenguaje
- Formalmente
  - Dadas dos gramáticas $G$ y $G'$

    son equivalentes $\Leftrightarrow L(G) = L(G')$

# Gramáticas equivalentes

## Ejemplos

- El alumno ya habrá observado que es posible definir muchas gramáticas distintas para el mismo lenguaje. Todas ellas son equivalentes.
- Se puede demostrar que hay infinitas gramáticas equivalentes a una dada.
- Por ejemplo, dada una gramática

$$G=(\Sigma_T,\Sigma_N,S,P)$$

  - La gramática

$$G'=(\Sigma_T,\Sigma_N'=\{Z\},Z,P\cup\{Z::=S\}) \;,,\; Z\notin\Sigma_N$$

  - Es claramente equivalente a $G$.



# Derivaciones más a la izquierda y más a la derecha

## Descripción intuitiva

- Dentro de las posibles derivaciones que una gramática proporciona resulta de interés teórico estudiar las que cumplen algunas condiciones.

## Derivaciones más a la izquierda y más a la derecha

### Definición

- Una derivación

$$S \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow x$$

se dice que es **más a la izquierda** (**derecha**) si en cada uno de los pasos o derivaciones directas se ha aplicado una producción cuya parte izquierda es el símbolo no terminal situado más a la **izquierda** (**derecha**) en la forma sentencial anterior



## Derivaciones más a la izquierda y más a la derecha

### Ejemplos

- En el ejemplo de la gramática que genera números naturales

$$G=(\Sigma_T=\{0,1,2,3,4,5,6,7,8,9\},\Sigma_N=\{N,C\},N,P=\{N::=CN \mid C,\ \ C::=0 \mid 1 \mid 2 \mid \cdots \mid 9\})$$

- La derivación del número 210 a partir de $N$
  - Se podría conseguir con la siguiente derivación más a la izquierda:

$$N \Rightarrow \underline{\mathbf{C}}N \Rightarrow 2\underline{\mathbf{N}} \Rightarrow 2\underline{\mathbf{C}}N \Rightarrow 21\underline{\mathbf{N}} \Rightarrow 21\underline{\mathbf{C}} \Rightarrow 210$$

  - Y con la siguiente derivación más a la derecha:

$$N \Rightarrow C\underline{\mathbf{N}} \Rightarrow CC\underline{\mathbf{N}} \Rightarrow CC\underline{\mathbf{C}} \Rightarrow C\underline{\mathbf{C}}0 \Rightarrow \underline{\mathbf{C}}10 \Rightarrow 210$$

# Árbol de derivación

## Definición

- El árbol de derivación es una disposición gráfica de las derivaciones que pueden realizarse con una gramática.
- La definición que se va a proporcionar es válida para las gramáticas que cumplen la condición de que sus reglas sólo sustituyen un símbolo (algunas definiciones de algunos tipos de gramática que todavía no se han estudiado pueden permitir que una derivación cambie más de un símbolo).



# Árbol de derivación

## Definición

- El proceso de construcción es el que puede deducirse de forma natural del primer ejemplo sobre una gramática para algunas frases castellanas:
  - El árbol tiene los nodos etiquetados con símbolos de la gramática
  - La raíz del árbol está etiquetada con el axioma.
  - Un nodo del árbol etiquetado con un símbolo no terminal puede tener hijos creados de la siguiente manera:
    - Cualquiera de las reglas que tengan como parte izquierda el no terminal del nodo padre puede aplicarse a él.
    - Se generarán tantos hijos como símbolos haya en la parte derecha de la regla etiquetados con esos símbolos y se disponen de izquierda a derecha respetando el orden en el que aparecen en la parte derecha de la regla.

# Árbol de derivación

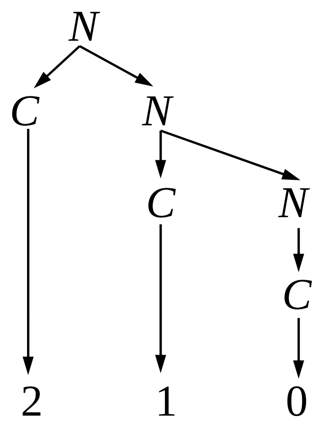
## Ejemplo

- Puede comprobarse que a las dos derivaciones siguientes

$$N \Rightarrow \underline{\mathbf{C}}N \Rightarrow 2\underline{\mathbf{N}} \Rightarrow 2\underline{\mathbf{C}}N \Rightarrow 21\underline{\mathbf{N}} \Rightarrow 21\underline{\mathbf{C}} \Rightarrow 210$$

$$N \Rightarrow C\underline{\mathbf{N}} \Rightarrow CC\underline{\mathbf{N}} \Rightarrow CC\underline{\mathbf{C}} \Rightarrow C\underline{\mathbf{C}}0 \Rightarrow \underline{\mathbf{C}}10 \Rightarrow 210$$

- Les corresponde el siguiente árbol de derivación:






# Ambigüedad en las gramáticas

## Gramáticas ambiguas: concepto informal

- Obsérvese que, a pesar de que las derivaciones son distintas, el árbol de derivación es el mismo.
- Esto no es general. Hay situaciones en las que se pueden encontrar árboles de derivación distintos para la misma sentencia.
- A esta circunstancia le corresponde el concepto de ambigüedad.

# Ambigüedad en las gramáticas

## Gramáticas ambiguas: definición formal

- Una gramática

$$G=(\Sigma_T,\Sigma_N,S,P)$$

Se dice que **es ambigua** si existe en ella una sentencia para la que existen dos árboles de derivación distintos



# Ambigüedad en las gramáticas

## Gramáticas ambiguas: definición formal

- Gracias a los tipos especiales de derivaciones puede ofrecerse la siguiente definición de gramática ambigua

- Una gramática

$$G=(\Sigma_T,\Sigma_N,S,P)$$

Se dice que **es ambigua** si existe en ella una sentencia que puede obtenerse a partir del axioma mediante dos derivaciones más a la izquierda (derecha) distintas

# Ambigüedad en las gramáticas

## Derivaciones vs. Árboles de derivación

- Como se puede deducir de las anteriores páginas, la relación entre árboles y derivaciones no es de uno a uno
  - Es posible que algunas gramáticas presenten múltiples derivaciones distintas a las que corresponde el mismo árbol.
- Esto está causado por la posibilidad de elegir, en cada derivación directa de una derivación, uno cualquiera de todos los símbolos no terminales de la forma sentencial considerada en ese momento.



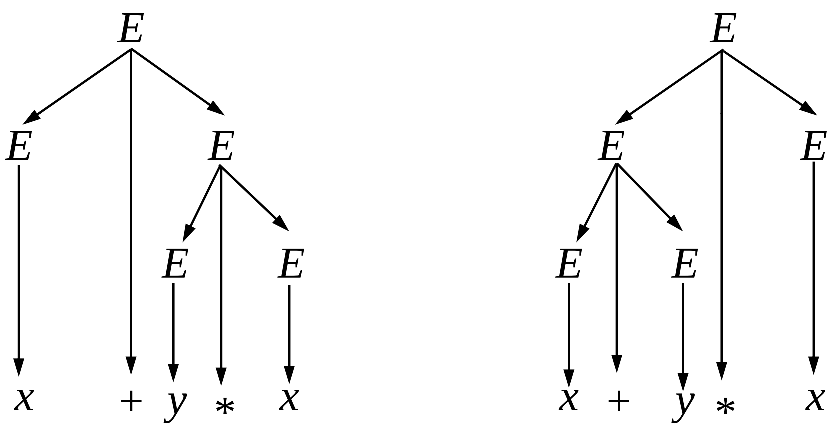
# Ambigüedad en las gramáticas

## Gramáticas ambiguas: ejemplos

- Considérese la siguiente gramática

$$G=(\Sigma_T=\{x,y,+,*\},\Sigma_N=\{E\},E,P=\{E::=E+E \mid E*E \mid x \mid y\})$$

- Capaz de generar los siguientes dos árboles de derivación distintos para la misma sentencia: $x+y*x$



# Gramáticas independientes del contexto

## Concepto informal: independencia del contexto

- Informalmente, se conoce como gramáticas independientes del contexto a aquellas en las que la transformación de los símbolos no terminales no depende de los símbolos que lo acompañen en la forma sentencial en la que se aplique su regla de producción.
- Así se justifica la siguiente definición formal.



# Gramáticas independientes del contexto

## Definición formal

- La condición que cumplen las gramáticas independientes del contexto tiene que ver con la forma de sus reglas de producción.
- Las gramáticas independientes del contexto se definen como aquellas gramáticas

$$G=(\Sigma_T,\Sigma_N,S,P)$$

- En las que las reglas de producción tienen la siguiente forma

$$A::=v \;\; ,, \; A\in\Sigma_N \;\wedge\; v\in (\Sigma_T\cup\Sigma_N)^*$$

- Esta definición presenta la dificultad de permitir reglas $\lambda$ para símbolos no terminales distintos del axioma (ya que $v\in (\Sigma_T\cup\Sigma_N)^*$ podría ser que $v=\lambda$)

# Gramáticas independientes del contexto

## Definición formal

- Esto no es siempre deseable, sin embargo, se puede demostrar que si el lenguaje generado por una gramática contiene $\lambda$ puede obtenerse una gramática equivalente en el que la única regla $\lambda$ sea la del axioma

$$S::=\lambda$$

- Lo que permite la siguiente definición más restringida, cómoda y equivalente de la condición que las reglas de una gramática independiente del contexto debe cumplir

$$A::=v \quad ,, A\in\Sigma_N \;\wedge v\in (\Sigma_T\cup\Sigma_N)^+$$

- Adicionalmente puede ocurrir

$$S::=\lambda \;\in P$$



# Gramáticas bien formadas

## Introducción a ciertas propiedades

- El objetivo fundamental de las gic está ligado a la definíción de los lenguajes de programación de alto nivel y al diseño de sus compiladores e intérpretes.
- Para facilitar su tratamiento (diseño de algoritmos de análisis, etc...) a lo largo del tiempo se han ido idenficando ciertas propiedades que es interesante que cumplan.
- En ese sentido, se considera que las gramáticas se pueden normalizar (hasta que estén “bien formadas”), cuando se consiguen versiones equivalentes que cumplen algunas propiedades.
- Aunque no son objeto de este curso sí resulta interesante mencionar al menos los conceptos asociados con esas propiedades.
  - Reglas lambda y símbolos anulables; símbolos útiles e inútiles (generativos, alcanzables); reglas de redenominación o unitarias
- Puesto que, para determinados propósitos, es necesario someter a las gramáticas que se están diseñando a algoritmos de normalización como
  - Eliminación de símbolos anulables
  - Eliminación de símbolos inútiles
  - Eliminación de reglas unitarias
  - Normalizaciones de Chomsky y Greibach

# Gramáticas independientes del contexto

## Ejemplo 1

- Diseñar una gramática independiente del contexto que genere el siguiente lenguaje

$$L=\{ww^{-1} \mid w\in\{a,b\}^*\}$$

  - Puede realizarse la siguiente definición recursiva de las cadenas del lenguaje
    1. La cadena más corta es $\lambda$.
    2. Dada una cadena del lenguaje ($S$), para obtener otra hay que asegurarse de que por delante y por detrás de ella se tienen los mismos símbolos, lo mínimo son $a$ y $a$ o $b$ y $b$.
  - Lo que puede conseguirse con las siguientes reglas
    1. $S::=\lambda$.
    2. $S::=aSa$
       $S::=bSb$
  - Por lo que la gramática quedaría definida de la siguiente forma

$$G=(\Sigma_T=\{a,\, b\},\Sigma_N=\{S\},S,P)$$

    - Donde P contiene las reglas de producción de los puntos 1 y 2



# Gramáticas independientes del contexto

## Ejemplo 2

- Diseñar una gramática independiente del contexto que genere el siguiente lenguaje

$$L=\{w\in\{a,b\}^* \mid \#_a(w)=\#_b(w)\}$$

  - La primera idea podía consistir en darse cuenta de que la estructura de las palabras del lenguaje permite afirmar que:
    1. La palabra más pequeña es $\lambda$
    2. Dada una palabra del lenguaje (la representaremos mediante el axioma $S$), se puede obtener otra añadiendo siempre un símbolo $a$ y otro $b$.
  - La primera opción se puede traducir en la regla de producción $S::=\lambda$.
  - Para la segunda, puesto que en la parte derecha hay que distribuir $a$, $b$ y $S$, cabe la duda sobre cómo hacerlo:
    - Como el orden en el que aparezcan los símbolos $a$ y $b$ no importa, tendremos que disponerlos de las dos formas posibles

$$ab \quad \text{y} \quad ba$$

    - Para la colocación del axioma observemos que si se opta por una solución como

$$S::=Sab \quad \text{y } S::=Sba$$

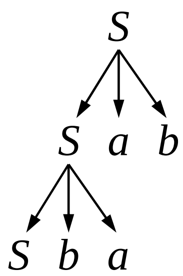
# Gramáticas independientes del contexto

## Ejemplo 2

- Habrá alguna estructura de palabras válidas no representable, por ejemplo

$aaa...bbb$

- Ya que si se observan las derivaciones de esas reglas:
    - Mientras se use la primera se obtienen al final de la cadena grupos $ab$ ($(ab)^n$)
    - Cuando se use la segunda se generan grupos ba ($(ba)^n$)
    - Parece que el lenguaje generado por esas reglas sería $(ab+ba)(ab+ba)$... (sólo es obligatoria una aparición de $(ab+ba)$ )
    - En cualquier caso no será posible aceptar cadenas con la estructura $aaa...bbb$ ya que, como mucho, pueden generarse 2 juntas como muestra el árbol





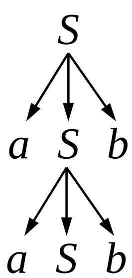

# Gramáticas independientes del contexto

## Ejemplo 2

- La razón puede consistir en que es necesario poder generar todas las maneras posibles en las que se puede añadir un símbolo $a$ y una $b$ a una cadena que pertenece al lenguaje:

| | |
|---|---|
| $S::=Sab$ | $S::=Sba$ |
| $S::=aSb$ | $S::=bSa$ |
| $S::=abS$ | $S::=baS$ |

- Obsérvese:
    - Que ahora sí se puede generar esa estructura



# Gramáticas independientes del contexto

## Ejemplo 2

- Lamentablemente, este lenguaje en apariencia simple, no puede abordarse sólo con esta aproximación.
- Considere cadenas un poco más "enrevesadas" como aquellas que comienzan y terminan con más de un símbolo repetido del mismo tipo, como por ejemplo

$$aabbbbaa$$

- La única manera de obtener dos símbolos $a$ ($aa$) al principio sería mediante el uso de la regla $S::=aSb$ pero no podrían usarse ya que el final de la cadena no es correcto (termina por $a$ y no por $b$)
- En casos así, se podría pensar que tal vez se ha abordado con una propuesta demasiado simple una estructura que en realidad es más compleja.
- Una buena alternativa en estos casos es empezar con probar primero con estructuras sencillas, como son la secuencia. ¿Se subsanaría el problema si consideramos que nuestras cadenas son realmente secuencias de las que conseguimos generar con las reglas propuestas hasta el momento?



# Gramáticas independientes del contexto

## Ejemplo 2

- Podríamos conseguir esto con las reglas que ya conocemos para secuencias de elementos (secuencias de dígitos en los números, por ejemplo)
- Para ello podríamos introducir un nuevo axioma (observe que en este caso la secuencia es opcional, ya que podemos tener en el lenguaje la cadena vacía)

$$X::=SX \mid \lambda$$

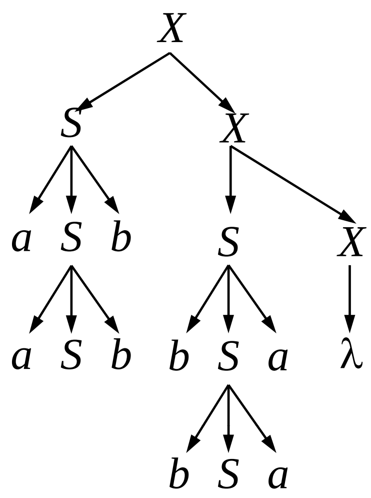
- Dada la estructura de las cadenas, cada uno de los elementos de la secuencia tienen que ser capaces de reproducir cualquier posible estructura de las cadenas correctas. Eso se puede conseguir con la siguiente regla

$$S::=X$$

# Gramáticas independientes del contexto

## Ejemplo 2

- Es fácil comprobar que ahora sí podríamos generar la cadena problemática *aabbbbaa* que podríamos considerar como la concatenación de dos subcadenas *aabb bbaa* que pueden ser generadas con las reglas anteriores






# Gramáticas independientes del contexto

## Ejemplo 2

- En principio, no aparecen más cadenas no generables por lo que podemos dar por buena esta estructura.
- Llegado a este punto podría considerarse si se puede simplificar un poco la gramática, aunque en general, a lo largo de este curso no nos importará que las gramáticas resultantes sean no ambiguas o que cumplan alguna otra condición que suele ser recomendable para diferentes usos posteriores, sí podríamos reflexionar un poco al respecto de cómo simplificar ésta
- Obsérvese que la secuencia podría generarse también con una regla más compleja con menos no terminales. (Queda fuera del ámbito de este curso pero la opción de secuencia $X::=SX \mid \lambda;\ S::=X$ corresponde con el nivel más simple de gramáticas, el nivel regular y la siguiente propuesta no)

$$S::=SS$$

- No es una demostración pero observe la siguiente derivación

$$X \Rightarrow S\mathbf{X} \Rightarrow SS$$

- La regla que permite recuperar la $X$ desde $S$ está implícita en $S::=SS$

# Gramáticas independientes del contexto

## Ejemplo 2

- Y puede observarse también que, si consideramos sólo la posición central del axioma $S::=Sab$ $S::=Sba$ podríamos considerar redundantes el resto.

$$S \Rightarrow S\mathbf{S} \Rightarrow Sa\mathbf{S}b \Rightarrow Sab$$

$$S \Rightarrow S\mathbf{S} \Rightarrow Sb\mathbf{S}a \Rightarrow Sba$$

- En resumen, podría demostrarse que las cadenas de nuestro lenguaje siempre pueden considerarse una secuencia (pueden dividirse en otras dos y a su vez seguir esa división de manera recursiva) de cadenas que tienen el mismo número de símbolos $a$ y $b$ dispuestas de manera simétrica como lo hacen las reglas $S::=aSb$ $S::=bSa$
- Con lo que podría obtenerse una versión más sencilla de la gramática como se muestra

$$G=(\Sigma_T=\{a, b\},\Sigma_N=\{S\},S,P)$$

- Donde $P$ contiene las siguientes reglas de producción

$$S::=\lambda$$
$$S::=aSb$$
$$S::=bSa$$
$$S::=SS$$



# Gramáticas independientes del contexto

## Ejemplo 3

- Diseñar una gramática independiente del contexto que genere el siguiente lenguaje

$$L=\{a^n b^n | \, n\geq 0\}$$

  - Obsérvese que las cadenas de este lenguaje pueden definirse recursivamente de la siguiente manera:
    1. La más corta es $\lambda$.
    2. Dada otra palabra del lenguaje ($S$), para conseguir una nueva se puede añadir por delante de ella una $a$ y por detrás una $b$.
  - Lo que podría conseguirse con las siguientes reglas
    1. $S::=\lambda$.
    2. $S::=aSb$
  - Por lo que la gramática quedaría definida de la siguiente forma

$$G=(\Sigma_T=\{a, b\},\Sigma_N=\{S\},S,P)$$

- Donde P contiene las reglas de producción de los puntos 1 y 2.

# Gramáticas independientes del contexto

## Ejemplo 3 bis

- Diseñar una gramática independiente del contexto que genere el siguiente lenguaje

$$L=\{a^n b^n |\ n>0\}$$

  - En este caso sólo hay que cambiar la definición de la cadena más corta
    1. La más corta es *ab*.
  - Por lo que la gramática quedaría definida de la siguiente forma

$$G=(\Sigma_T=\{a, b\},\Sigma_N=\{S\},S,P)$$

    - Donde P contiene las siguientes reglas de producción

$$S::=ab$$

$$S::=aSb$$