

# Lenguajes formales y gramáticas

Pablo Verdes

LCC

7 de mayo de 2018

# Generalidades y motivación

- En esta parte del curso queremos entender:
  - ▶ ¿Qué es un lenguaje? ¿Cómo se define?
  - ▶ ¿Cómo se pueden combinar distintos lenguajes?
  - ▶ ¿Cuál es la complejidad de un lenguaje?
- En el estudio de lenguajes existen, básicamente, dos enfoques:
  - ① el de **reconocimiento** ( $\rightarrow$  autómatas)
  - ② el **generativo** ( $\rightarrow$  gramáticas)
- Indep. del enfoque, se consideran típicamente dos aspectos:
  - ① **sintaxis**: cuáles son las palabras 'legales' en un lenguaje
  - ② **semántica**: cuál es el significado o interpretación de las palabras
- La semántica suele ser más interesante que la sintaxis, y también más difícil de estudiar. Aquí sólo estudiaremos sintaxis.

# Generalidades y motivación

- En el punto de vista de **reconocimiento**, el enfoque es el de una 'caja negra' (un **autómata**) que toma una palabra como entrada, hace ciertas operaciones y devuelve una de dos respuestas posibles, 'sí' o 'no':
  - ▶ **sí**: la palabra es aceptada por el autómata, lo cual significa que pertenece al lenguaje que queremos definir
  - ▶ **no**: la palabra es rechazada por el autómata, lo cual significa que no pertenece al lenguaje
- Esta caja negra puede realizar operaciones de manera **determinista** o **no determinista**. Esto último significa, en términos generales, que al autómata se le permite realizar diferentes cálculos en paralelo para luego ignorar algunos de ellos, dependiendo de los resultados.

# Generalidades y motivación

- El proceso de reconocimiento de un autómata no determinista es, en general, más complejo que el determinista.
- Sin embargo, a veces resultan equivalentes en cuanto a poder de reconocimiento de lenguajes. (Esto tiende a ocurrir para los casos extremos, cuando el tipo de autómata es o bien muy poco flexible o muy poderoso.)
- Tipos de autómatas (ordenados de manera creciente según su poder de reconocimiento):

Poder de reconoc.

- |                            |                             |              |
|----------------------------|-----------------------------|--------------|
| ① Aut. Finitos             | Det. (AFD) y No Det. (AFND) | $AFD = AFND$ |
| ② Aut. de Pila             | Det. (APD) y No Det. (APND) | $APD < APND$ |
| ③ Aut. linealmente acotado | (caso esp. de MT)           |              |
| ④ Máq. de Turing           | Det. (MTD) y No Det. (MTND) | $MTD = MTND$ |

# Generalidades y motivación

- En el punto de vista **generativo** nos interesan los formalismos que especifican un lenguaje en términos de reglas de construcción de palabras permitidas. El formalismo más común es el de **gramáticas formales**.
- Para ciertas clases de gramáticas  $G$ , es posible construir un autómata  $M_G$  que reconoce el lenguaje  $L(G)$  generado por  $G$ .
- Las gramáticas son, por naturaleza, **no deterministas**. Por tal motivo debemos considerar los autómatas no deterministas (aunque quisiéramos en principio evitarlos).
- Estudiaremos los siguientes tipos de gramáticas (llamada **jerarquía de Chomsky**) y sus correspondientes lenguajes:
  - 1 Regulares (o de tipo 3)
  - 2 Libres de contexto (o de tipo 2)
  - 3 Sensibles al contexto (o de tipo 1)
  - 4 Estructuradas por frases (o de tipo 0)

# Generalidades y motivación

- Estos tipos de gramáticas se corresponden exactamente con los tipos de autómatas antes vistos.
- Más aún, existen algoritmos para convertir gramáticas en sus correspondientes autómatas (y recíprocamente), aunque algunos de estos algoritmos no son prácticos.
- Construir un autómata que reconozca el lenguaje generado por una gramática es un problema práctico importante. Los casos más estudiados son los de las gramáticas regulares y las libres de contexto, pero el tema sigue abierto y se investiga activamente.
- Hay otras maneras de definir familias de lenguajes, por ej. la **clausura inductiva**. Consiste en definir un conjunto de lenguajes básicos (o atómicos) y algunas operaciones para combinarlos. La nueva familia de lenguajes se define como el conjunto más pequeño que contiene a los básicos y que es cerrado bajo las operaciones dadas. Este el enfoque empleado por las Expresiones Regulares.

# Lenguajes formales: conceptos básicos

- Un **alfabeto** es un conjunto finito y no vacío.  
Sus elementos reciben el nombre de **símbolos**, **letras** o **caracteres**.  
Notación:  $\Sigma = \{a_1, a_2, \dots, a_k\}$ .  
Ejemplos:
  - ▶  $\Sigma = \{0, 1\}$ , el alfabeto binario
  - ▶  $\Sigma = \{a, b, c, \dots, z\}$ , el conjunto de todas las letras minúsculas
- Una **cadena** o **palabra** sobre un alfabeto  $\Sigma$  es una secuencia finita de símbolos de  $\Sigma$ . Más precisamente, se la define como una función

$$u : \{1, 2, \dots, n\} \rightarrow \Sigma$$

El entero positivo  $n$  es la **longitud** de la palabra, que se denota también  $|w|$ .

Se agrega la palabra especial  $u : \{\} \rightarrow \Sigma$ , llamada **nula** o **vacía**, considerada de longitud 0. Se la denota con  $\lambda$  o también  $\epsilon$ .

# Lenguajes formales: conceptos básicos

- Dada la palabra

$$u : \{1, 2, \dots, n\} \rightarrow \Sigma$$

de longitud  $n \geq 1$ ,  $u(i)$  es su  $i$ -ésima letra.

Por simplicidad de notación, escribiremos la cadena  $u$  como

$$u = u_1 u_2 \dots u_n$$

donde cada  $u_i \in \Sigma$ .

Ejemplos:

- ▶ Si  $\Sigma = \{a, b\}$  y la palabra  $u : \{1, 2, 3\} \rightarrow \Sigma$  está definida por  $u(1) = a$ ,  $u(2) = b$  y  $u(3) = a$ , escribiremos  $u = aba$ .
- ▶ 0, 00, 010, 110, 010101110101 son palabras del alfabeto  $\Sigma = \{0, 1\}$
- ▶ *ta*, *pppipi*, *cuchunchoglu* son cadenas del alfabeto  $\Sigma = \{a, b, \dots, z\}$



# Lenguajes formales: conceptos básicos

- Definiciones:

$$\Sigma^n = \{u \mid u : \{1, \dots, n\} \rightarrow \Sigma\} \quad n \geq 1$$

$$\Sigma^0 = \{\lambda\} \quad n = 0$$

$$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n \quad \Sigma^+ = \bigcup_{n \geq 1} \Sigma^n$$

- $\Sigma^*$  es el conjunto de todas las palabras sobre un alfabeto  $\Sigma$ , incluida la palabra vacía.

Ejemplo: si  $\Sigma = \{a, b\}$ , tenemos

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, aba, baa, abb, bab, bba, bbb, \dots\}$$

- Observemos que:

- ▶ si  $\Sigma = \emptyset$ , resulta  $\emptyset^* = \{\lambda\}$
- ▶ si  $\Sigma \neq \emptyset$ , resulta  $\Sigma^*$  infinito numerable

# Lenguajes formales: conceptos básicos

- Dado un alfabeto  $\Sigma$  y dos palabras

$$u : \{1, \dots, m\} \rightarrow \Sigma \quad \text{y} \quad v : \{1, \dots, n\} \rightarrow \Sigma,$$

definimos la **concatenación** de  $u$  y  $v$  (denotada  $uv$ ) como la palabra  $uv : \{1, \dots, m+n\} \rightarrow \Sigma$  tal que

$$uv(i) = \begin{cases} u(i) & \text{si } 1 \leq i \leq m \\ v(i-m) & \text{si } m+1 \leq i \leq m+n \end{cases}$$

Se define también  $u\lambda = \lambda u = u$ .

- Es fácil verificar que  $u(vw) = (uv)w$  pero  $uv \neq vu$ .
- Dada una cadena  $u \in \Sigma^*$ , definimos su **potencia**  $u^n$  como:

$$u^n = \begin{cases} \lambda & \text{si } n = 0 \\ u^{n-1}u & \text{si } n \geq 1 \end{cases}$$

# Lenguajes formales: conceptos básicos

- Ejercicio: probar que

①  $u^1 = u$

②  $u^n u = u u^n \quad \forall n \geq 0$

- Dada una cadena  $u \in \Sigma^*$ , definimos inductivamente la **cadena reversa**  $u^R$  como:

$$\begin{aligned}\lambda^R &= \lambda \\ (ua)^R &= au^R\end{aligned}$$

donde  $a \in \Sigma$  y  $u \in \Sigma^*$ .

- Diremos que:
  - ▶  $v$  es **subcadena** de  $s$  si existen  $u, w \in \Sigma^*$  tales que  $s = uvw$ .
  - ▶  $v$  es **prefijo** de  $s$  si  $u = \lambda$ .
  - ▶  $v$  es **sufijo** de  $s$  si  $w = \lambda$ .
  - ▶  $v$  es subcadena (o prefijo, sufijo) **propia** de  $s$  si  $v$  es subcadena (o prefijo, sufijo) de  $s$  y además  $v \neq s$ .

# Lenguajes formales: definición

- Un **lenguaje** sobre un alfabeto  $\Sigma$  es un subconjunto de  $\Sigma^*$ .
- Dado un alfabeto  $\Sigma$ , llamaremos  $\mathcal{L}$  al conjunto de todos los lenguajes sobre  $\Sigma$ . Es decir,  $\mathcal{L} = \mathcal{P}(\Sigma^*)$ .
- Ejemplos:
  - ▶  $\Sigma = \{a, b, c\}$ ,  $L_1 = \{a, ab, bbca, ac\}$
  - ▶  $\Sigma = \{a, b, c\}$ ,  $L_2 = \{a^i bc^i \mid i \in \mathbb{N}_0\}$
  - ▶  $\Sigma = \{a, b, c\}$ ,  $L_3 = \{a^p \mid p \in \mathbb{N}_0 \wedge p \text{ es número primo}\}$
  - ▶  $L_4 = \{w \mid w \text{ es palabra del idioma español}\}$
  - ▶  $L_5 = \{\lambda\}$
  - ▶  $L_6 = \{\}$
  - ▶  $iL_5 = L_6?$
- Si  $\Sigma \neq \emptyset$ , entonces:
  - ▶  $\Sigma^*$  es infinito numerable:  $\text{card}(\Sigma^*) = \aleph_0$
  - ▶  $\mathcal{L} = \mathcal{P}(\Sigma^*)$  es infinito no numerable:  $\text{card}(\mathcal{L}) = \aleph_1$

# Operaciones sobre lenguajes

Una manera de construir lenguajes más complejos a partir de otros más simples es combinarlos a través de ciertas operaciones.

Dados un alfabeto  $\Sigma$  y dos lenguajes  $L_1$  y  $L_2$  sobre  $\Sigma$ ,

- la **unión**  $L_1 \cup L_2$  es el lenguaje

$$L_1 \cup L_2 = \{w \in \Sigma^* \mid w \in L_1 \vee w \in L_2\}$$

- la **intersección**  $L_1 \cap L_2$  es el lenguaje

$$L_1 \cap L_2 = \{w \in \Sigma^* \mid w \in L_1 \wedge w \in L_2\}$$

- la **diferencia**  $L_1 - L_2$  es el lenguaje

$$L_1 - L_2 = \{w \in \Sigma^* \mid w \in L_1 \wedge w \notin L_2\}$$

- el **complemento** de  $L$  es el lenguaje  $\bar{L} = \Sigma^* - L$

- la **concatenación**  $L_1 L_2$  es el lenguaje

$$L_1 L_2 = \{w \in \Sigma^* \mid \exists u \in L_1, \exists v \in L_2, w = uv\}$$

# Operaciones sobre lenguajes

- Dado un lenguaje  $L$ , definimos inductivamente su **potencia**  $L^n$  como:

$$L^0 = \{\lambda\}$$
$$L^{n+1} = L^n L$$

- (*Ejercicio*) Es fácil verificar que:

- ▶  $L\emptyset = \emptyset$
- ▶  $\emptyset L = \emptyset$
- ▶  $L\{\lambda\} = L$
- ▶  $\{\lambda\}L = L$
- ▶  $(L_1 \cup \{\lambda\})L_2 = L_1L_2 \cup L_2$
- ▶  $L_1(L_2 \cup \{\lambda\}) = L_1L_2 \cup L_1$
- ▶  $L^n L = LL^n$

pero en general  $L_1L_2 \neq L_2L_1$ .

- Las operaciones que hemos introducido hasta aquí (excepto la de complemento) producen lenguajes finitos si los de partida son finitos. Este no es el caso para las siguientes dos operaciones.

# Operaciones sobre lenguajes

- Dados un alfabeto  $\Sigma$  y un lenguaje  $L$  sobre  $\Sigma$ ,

- ▶ la **\*-clausura de Kleene** de  $L$  es el lenguaje

$$L^* = \bigcup_{n \geq 0} L^n$$

- ▶ la **+clausura de Kleene** de  $L$  es el lenguaje

$$L^+ = \bigcup_{n \geq 1} L^n$$

- Es decir,  $L^*$  es la unión infinita

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots \cup L^n \cup \dots$$

mientras que  $L^+$  es la unión infinita

$$L^+ = L^1 \cup L^2 \cup \dots \cup L^n \cup \dots$$

# Operaciones sobre lenguajes

$$L^* = \{\lambda\} \cup L \cup L^2 \cup \dots \cup L^n \cup \dots$$

$$L^+ = L \cup L^2 \cup \dots \cup L^n \cup \dots$$

- Dado que  $L^1 = L$ , vemos que tanto  $L^*$  como  $L^+$  contienen a  $L$ .
- Dado que  $L^0 = \{\lambda\}$ , vemos que  $L^* = L^+ \cup \{\lambda\}$ .
- Por lo tanto,  $\lambda \in L^*$ . Sin embargo, si  $\lambda \notin L$  entonces  $\lambda \notin L^+$ .
- Se puede probar que:
  - ▶  $\emptyset^* = \{\lambda\}$
  - ▶  $L^+ = L^*L$
  - ▶  $(L^*)^* = L^*$
  - ▶  $L^*L^* = L^*$



# Gramáticas

- Las gramáticas proveen un **mecanismo generador** de lenguajes.

## Ejemplo: El lenguaje de Romeo

- ▶  $\Sigma = \{\text{Julieta, eres, muy, hermosa}\}$

- ▶ Reglas de producción:

$\langle \text{frase} \rangle \rightarrow \langle \text{sujeto} \rangle \langle \text{predicado} \rangle$

$\langle \text{sujeto} \rangle \rightarrow \text{Julieta}$

$\langle \text{predicado} \rangle \rightarrow \text{eres } \langle m \rangle \text{ hermosa}$

$\langle m \rangle \rightarrow \lambda \mid \text{muy } \langle m \rangle$

- Una gramática posee los siguientes símbolos:

- ▶ **inicial**,
- ▶ **no terminales** (son estructuras intermedias), y
- ▶ **terminales** (son elementos del alfabeto).

# Gramáticas

**Definición:** Una **gramática** es una tupla  $(N, T, P, \sigma)$  donde:

- $N$  es un conjunto finito de símbolos llamados **no terminales**
- $T$  es un conjunto finito de símbolos, llamados **terminales** o **alfabeto**, tal que  $N \cap T = \emptyset$
- $P$  es un conjunto finito de **reglas de producción**, donde

$$P \subseteq ((N \cup T)^* - T^*) \times (N \cup T)^*$$

Obs.:

- ▶  $(N \cup T)^* - T^*$  es el conjunto de cadenas de no terminales y terminales que contienen al menos un no terminal. Dado que  $\lambda \notin (N \cup T)^* - T^*$ , no puede haber reglas del tipo  $\lambda \rightarrow \beta$ .
- ▶  $(N \cup T)^*$  es el conjunto de cadenas sobre  $N \cup T$ .
- ▶ A una producción de la forma  $(\alpha, \beta)$  la notaremos  $\alpha \rightarrow \beta$ .
- ▶ Una regla del tipo  $\alpha \rightarrow \lambda$  recibe el nombre de **regla**  $\lambda$ .

- $\sigma \in N$  es el **símbolo inicial**

# Gramáticas

## • Ejemplo 1:

$$G_1 = (N, T, P, \sigma)$$

donde:

- ▶  $N = \{\sigma\}$
- ▶  $T = \{a, b\}$
- ▶  $P$  está dado por las reglas de producción:

$$\sigma \rightarrow a\sigma bb$$

$$\sigma \rightarrow abb$$

- ▶  $\sigma$  es el símbolo inicial

- ¿Qué palabras genera  $G_1$ ? Aplicando las reglas, veamos algunos ejemplos:

$$\sigma \Rightarrow abb$$

$$\sigma \Rightarrow a\sigma bb \Rightarrow aabb$$

$$\sigma \Rightarrow a\sigma bb \Rightarrow aa\sigma bbbb \Rightarrow aaabbbbb$$

$$\sigma \Rightarrow \dots \Rightarrow a^n b^{2^n}$$

# Gramáticas

- Ejemplo 2:

$$G_2 = (\{\sigma, A, B\}, \{a, b, c, d\}, P, \sigma)$$

donde el conjunto  $P$  consiste de las reglas de producción:

$$\sigma \rightarrow A\sigma B \quad (1)$$

$$A \rightarrow b \quad (2)$$

$$\underline{aaA} \rightarrow aaBB \quad (3)$$

$$\sigma \rightarrow d \quad (4)$$

$$A \rightarrow aA \quad (5)$$

$$B \rightarrow dcd \quad (6)$$

- Ejemplo de generación de una palabra:

$$\sigma \Rightarrow^{(1)} A\sigma B \Rightarrow^{(5)} aA\sigma B \Rightarrow^{(2)} ab\sigma B \Rightarrow^{(4)} abdB \Rightarrow^{(6)} abddcd$$

## • Ejemplo 3:

$$G_3 = (N, T, P, \sigma)$$

donde:

- ▶  $N = \{\sigma\}$
- ▶  $T = \{0, 1\}$
- ▶  $P$  está dado por las reglas de producción:

$$\sigma \rightarrow \underline{000\sigma111} \quad (1)$$

$$0\sigma1 \rightarrow 01 \quad (2)$$

## • ¿Qué palabras genera $G_3$ ?

$$\begin{aligned} \sigma &\Rightarrow^{(1)} 000\sigma111 \Rightarrow^{(2)} 000111 \\ \sigma &\Rightarrow^{(1)} 000\sigma111 \Rightarrow^{(1)} (000)^2\sigma(111)^2 \Rightarrow^{(2)} (000)^2(111)^2 \\ \sigma &\Rightarrow^{(1, n \text{ veces})} (000)^n\sigma(111)^n \Rightarrow^{(2)} (000)^n(111)^n = 0^{3n}1^{3n} \quad n \geq 1 \end{aligned}$$

Hacerlo

(N U T) en la parte derecha p

- Ejemplo 4:

$$G_4 = (\{\sigma, A, B\}, \{a, b\}, P, \sigma)$$

donde el conjunto  $P$  consiste de las reglas de producción:

$$\sigma \rightarrow a \mid aA \mid aB \quad (1)$$

$$A \rightarrow a \mid aA \mid bB \quad (2)$$

$$B \rightarrow b \mid bB \quad (3)$$

- ¿Qué tipo de palabras genera  $G_4$ ?

$$\{a^m b^n \mid m > 0 \wedge n \geq 0\}$$

Parte izquierda, deb

- Ejemplo 5:

$$G_5 = (\{\sigma, A, B\}, \{a, b\}, P, \sigma)$$

donde el conjunto  $P$  consiste de las reglas de producción:

$$\sigma \rightarrow aA \quad (1)$$

$$A \rightarrow b \mid bB \quad (2)$$

$$B \rightarrow aA \quad (3)$$

- ¿Qué conjunto de palabras genera  $G_5$ ?

$$\{(ab)^n \mid n > 0\}$$

# Gramáticas

**Definiciones:** Sea  $G = (N, T, P, \sigma)$ .

- Si  $\alpha \rightarrow \beta$  es una regla de producción (es decir,  $(\alpha, \beta) \in P$ ), y  $x\alpha y$  es una cadena (es decir,  $x\alpha y \in (N \cup T)^* - T^*$ ), entonces diremos que  $x\beta y$  **deriva directamente** de  $x\alpha y$ , y escribiremos

$$x\alpha y \Rightarrow x\beta y$$

También diremos que  $x\alpha y$  **produce directamente**  $x\beta y$ .

- Sean  $\alpha_i \in (N \cup T)^* - T^*$ ,  $1 \leq i < n$ , y  $\alpha_n \in (N \cup T)^*$ . Si vale que  $\alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_{n-1} \Rightarrow \alpha_n$ , entonces diremos que  $\alpha_n$  **deriva** de  $\alpha_1$ , o que  $\alpha_1$  **produce**  $\alpha_n$ , y lo notaremos

$$\alpha_1 \Rightarrow^* \alpha_n$$

En el Ejemplo 2 vimos que  $\sigma \Rightarrow^* abddcd$ .



# Gramáticas

- Definimos el **lenguaje generado por una gramática**  $G = (N, T, P, \sigma)$  como:

$$L(G) = \{\alpha \in T^* \mid \sigma \Rightarrow^* \alpha\}$$

- Ejemplos:

- ▶  $L(G_1) = \{a^n b^{2n} \mid n \geq 1\}$
- ▶  $L(G_3) = \{0^{3n} 1^{3n} \mid n \geq 1\}$
- ▶  $L(G_4) = \{a^m b^n \mid m > 0 \wedge n \geq 0\}$
- ▶  $L(G_5) = \{(ab)^n \mid n > 0\}$

- Ejercicio: definir  $G$  tal que

- 1  $L(G) = \{ab, ba, abba, abbaab\}$
- 2  $L(G) = \{a^n ccb^n \mid n \geq 0\}$
- 3  $L(G) = \{a, b\}^*$

# Tipos de gramáticas y lenguajes

- Las gramáticas pueden ser:
  - 1 Regulares (o de tipo 3)
  - 2 Libres de contexto (tipo 2)
  - 3 Sensibles al contexto (tipo 1)
  - 4 Estructuradas por frases (tipo 0)
- A esta clasificación se la conoce con el nombre de [Jerarquía de Chomsky](#).

# Gramáticas regulares (tipo 3)

- Las gramáticas regulares o de tipo 3, también llamadas lineales, pueden ser clasificadas como derechas o izquierdas.
- Las reglas de producción de una gramática regular derecha se adhieren a las siguientes restricciones:
  - El lado izquierdo debe consistir en un solo no terminal.
  - El lado derecho está formado por un símbolo terminal, que puede estar seguido (o no) por un símbolo no terminal, o la cadena vacía.
- Es decir, las producciones de una gramática regular derecha pueden tener la forma:

$$A \rightarrow a$$

$$A \rightarrow aB$$

$$A \rightarrow \lambda$$

donde  $A, B \in N$  y  $a \in T$ .

## Gramáticas regulares (tipo 3)

- Alternativamente, en una **gramática regular izquierda** las reglas de producción son de la forma:

$$A \rightarrow a$$

$$A \rightarrow Ba$$

$$A \rightarrow \lambda$$

- Por ejemplo, reglas de la forma:

$$yW \rightarrow x$$

$$X \rightarrow xZy$$

$$YX \rightarrow WvZ$$

no están permitidas en una gramática regular. (¿Por qué?)

- Se puede convertir toda gramática regular derecha en izquierda y recíprocamente.

## Gramáticas regulares (tipo 3)

- **Ejercicio:** Sin modificar el lenguaje que genera, transforme la siguiente gramática de manera que resulte regular:

$$S \rightarrow yX$$

$$X \rightarrow \text{xx}X$$

$$X \rightarrow yY$$

$$Y \rightarrow \lambda$$

$$S \rightarrow yZZ \rightarrow xX$$

- **Ejercicio:** Convierta la siguiente gramática regular en otra gramática regular que no contenga reglas cuyo lado derecho consista de un solo símbolo terminal:

$$S \rightarrow xZZ \rightarrow y \mid \lambda$$

$$S \rightarrow xS$$

$$S \rightarrow y$$

$$X \rightarrow z$$

$$S \rightarrow xZZ \rightarrow yAA$$

# Gramáticas independientes del contexto (tipo 2)

- Las reglas de producción de una gramática independiente del contexto o libre de contexto se adhieren a las siguientes restricciones:

Como en las regulares, el lado izquierdo debe consistir en un solo no terminal.

- ▶ A diferencia de las regulares, no hay restricciones sobre la forma del lado derecho.

- Es decir, las reglas de producción tienen la forma:

$$A \rightarrow \delta$$

donde  $A \in N$  y  $\delta \in (N \cup T)^*$ .

- El término *independiente del contexto* hace referencia a que, debido a que en el lado izquierdo el no terminal aparece solo, la regla se puede aplicar sin importar el contexto en que aparezca dicho no terminal.
- En contraste, la regla  $xNy \rightarrow xzy$  permite reemplazar el no terminal  $N$  por  $z$  sólo cuando se encuentre en el contexto de  $x$  e  $y$ .

## Gramáticas independientes del contexto (tipo 2)

- Dado que no hay restricciones sobre la forma de  $\delta$  en  $A \rightarrow \delta$ , podría ocurrir que  $\delta$  contenga más de un no terminal, como en  $A \rightarrow zXYZz$ .
- Pero ¿cuál de los no terminales reemplazamos en el paso siguiente?
- El enfoque más común es el de la **derivación por la izquierda**, que consiste en reemplazar el no terminal situado más a la izquierda.
- Análogamente, se podría aplicar derivación por la derecha, o seguir algún otro patrón.
- Resulta ser que el orden en que se apliquen las reglas no afecta la determinación de si una cadena puede ser generada por la gramática o no.
- Esto es consecuencia de que *si existe una derivación que genera una cadena, entonces también existe una derivación por izquierda que la genera.*

# Gramáticas independientes del contexto (tipo 2)

- **Ejemplos:**

- ▶ La gramática con producciones:

$$S \rightarrow aSb \mid \lambda$$

genera el lenguaje

$$\{a^n b^n \mid n \geq 0\}.$$

- ▶ La gramática con producciones:

$$S \rightarrow x \mid y \mid z \mid (S + S) \mid (S - S) \mid (S \times S) \mid S/S$$

genera expresiones algebraicas sintácticamente correctas sobre las variables  $x, y, z$ .

Ejemplo:

$$(((x + y)/z - (x \times z))/(y - z))$$

- **Ejercicio:** Dar una gramática de tipo 2 que genere

$$\{a^n b^m c^{m+n} \mid n \geq 0, m \geq 0\}.$$



# Gramáticas dependientes del contexto (tipo 1)

- Las reglas de producción de una gramática dependiente del contexto o sensible al contexto son del tipo:

$$\alpha A \beta \rightarrow \alpha \delta \beta$$

donde  $A \in N$ ,  $\alpha, \beta \in (N \cup T)^*$  y  $\delta \in (N \cup T)^+$ .

- Adicionalmente, se permite la regla  $\sigma \rightarrow \lambda$ , donde  $\sigma$  es el símbolo inicial, siempre que  $\sigma$  no aparezca en el lado derecho de otra producción. Sirve para agregar  $\lambda$  al lenguaje.
- Obs.:
  - La producción  $abA \rightarrow baab$  no es sensible al contexto. Lo sería si fuese  $abA \rightarrow abab$ .
  - La producción  $aSb \rightarrow ab$  no es de tipo 1, pues  $\delta = \lambda \notin (N \cup T)^+$ .

# Gramáticas dependientes del contexto (tipo 1)

- Se pueden caracterizar como el conjunto de gramáticas de longitud **no decreciente**.
- Se dice que una gramática es de **longitud no decreciente** cuando sus reglas de producción  $\alpha \rightarrow \beta$  satisfacen  $|\alpha| \leq |\beta|$ .
- Es fácil probar que si  $G$  es sensible al contexto entonces es de longitud no decreciente:

$$|\alpha A \beta| \leq |\alpha \delta \beta|$$

pues  $\delta \in (N \cup T)^+$  y por lo tanto  $\delta \neq \lambda$ .

- Recíprocamente, se puede demostrar que si  **$G$  es de longitud no decreciente entonces existe una gramática sensible al contexto que genera el mismo lenguaje.**

# Gramáticas estructuradas por frases (tipo 0)

- Finalmente, en las gramáticas estructuradas por frases o irrestrictas no hay restricciones sobre la forma de las reglas de producción:

$$\alpha \rightarrow \delta$$

donde  $\alpha \in (N \cup T)^* - T^*$  y  $\delta \in (N \cup T)^*$ .

- Obs. que, dado que  $\lambda \notin (N \cup T)^* - T^*$ , no puede haber reglas del tipo  $\lambda \rightarrow \delta$ .

# Relación entre gramáticas

- Llamando

$$\mathcal{G}_i = \{G \mid G \text{ es de tipo } i\} \quad i = 0, 1, 2, 3$$

$$\mathcal{L}_i = \{L(G) \mid G \text{ es de tipo } i\} \quad i = 0, 1, 2, 3$$

se tienen las inclusiones

$$\mathcal{G}_3 \subset \mathcal{G}_2 \subset \mathcal{G}_1 \subset \mathcal{G}_0$$

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$$

- Observemos que las inclusiones son en sentido estricto. ¿Qué quiere decir esto?
- Más aún:

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0 \subset \mathcal{P}(T^*).$$

En otras palabras, existen lenguajes que no están en  $\mathcal{L}_0$ , es decir, para los cuales no existe gramática que los genere. A continuación lo demostraremos.

# Un lenguaje sin gramática

- $T^*$  es numerable:  $\text{card}(T^*) = \aleph_0$ . Podemos entonces denotar a sus palabras con los símbolos  $w_1, w_2, w_3, \dots$
- Veamos que el conjunto  $\mathcal{G}$ , de todas las gramáticas, también es numerable:  $\text{card}(\mathcal{G}) = \aleph_0$ .

Escribiremos a  $\mathcal{G}$  como u.n.c.n. En efecto, definiendo:

$$G_n = \{G = (N, T, P, \sigma) \mid \text{card}(N) = n\}$$

$$G_{n,t} = \{G_n = (N, T, P, \sigma) \mid \text{card}(T) = t\}$$

$$G_{n,t,p} = \{G_{n,t} = (N, T, P, \sigma) \mid \text{card}(P) = p\}$$

podemos escribir:

$$\mathcal{G} = \bigcup_{n=1}^{\infty} G_n = \bigcup_{n=1}^{\infty} \bigcup_{t=1}^{\infty} G_{n,t} = \bigcup_{n=1}^{\infty} \bigcup_{t=1}^{\infty} \bigcup_{p=1}^{\infty} G_{n,t,p}$$

- Podemos entonces denotar a las gramáticas con los símbolos  $G_1, G_2, G_3, \dots$

# Un lenguaje sin gramática

- Construimos la siguiente tabla, donde la entrada

$$(G_i, w_j) = 1 \text{ si } w_j \in L(G_i)$$

	$w_1$	$w_2$	$w_3$	$\dots$
$G_1$	0	1	1	$\dots$
$G_2$	1	1	0	$\dots$
$G_3$	1	0	0	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

- Ahora definimos los lenguajes

$$D = \{w_i \mid w_i \in L(G_i)\} = \{w_2, \dots\}$$

$$\overline{D} = \{w_i \mid w_i \notin L(G_i)\} = \{w_1, w_3, \dots\}$$

- ¿Existe una gramática que genere el lenguaje  $\overline{D}$ ? Veamos que no. Por el abs., sup. que  $\exists j \mid L(G_j) = \overline{D}$ . Pregunta: ¿ $w_j \in \overline{D}$ ?

$$\cdot \text{ si } w_j \in \overline{D} \Rightarrow w_j \notin L(G_j) = \overline{D}$$

$$\cdot \text{ si } w_j \notin \overline{D} \Rightarrow w_j \in L(G_j) = \overline{D}$$

Absurdo. Por lo tanto, no existe una gramática que genere  $\overline{D}$ .