

# Introducción a la Lógica y la Computación

Mariana Badano   Héctor Gramaglia   Pedro Sánchez Terraf  
M. Clara Gorín   Matías Steinberg

FaMAF, 20 de noviembre de 2020



- 1 Autómatas para lenguajes libres de contexto
  - Ejemplo de computación para  $L_{01}$
- 2 Lenguajes sensibles al contexto
  - Lema de bombeo para libres de contexto
  - Ejemplo de lenguaje no libre de contexto
  - Gramáticas sensibles al contexto
- 3 Gramáticas irrestrictas
  - Computabilidad
  - Jerarquía de Chomsky

# Un vistazo a los PDA

Los lenguajes libres de contexto son exactamente los aceptados por los **autómatas a pila** (“*push-down automata*”, **PDA**).

# Un vistazo a los PDA

Los lenguajes libres de contexto son exactamente los aceptados por los **autómatas a pila** (“*push-down automata*”, **PDA**).

Los PDA tienen toda la estructura de un NFA pero además pueden guardar información en una pila (LIFO) sin límite de memoria.

# Un vistazo a los PDA

Los lenguajes libres de contexto son exactamente los aceptados por los **autómatas a pila** (“*push-down automata*”, **PDA**).

Los PDA tienen toda la estructura de un NFA pero además pueden guardar información en una pila (LIFO) sin límite de memoria. O sea, tienen memoria (además de “un core”) pero no es **RAM**.

# Un vistazo a los PDA

Los lenguajes libres de contexto son exactamente los aceptados por los **autómatas a pila** (“*push-down automata*”, **PDA**).

Los PDA tienen toda la estructura de un NFA pero además pueden guardar información en una pila (LIFO) sin límite de memoria. O sea, tienen memoria (además de “un core”) pero no es **RAM**.

A diferencia de los autómatas finitos, el rol de los PDA es comparativamente secundario (a nivel práctico y en cuanto a uso en demostraciones).

# Un vistazo a los PDA

Los lenguajes libres de contexto son exactamente los aceptados por los **autómatas a pila** (“*push-down automata*”, **PDA**).

Los PDA tienen toda la estructura de un NFA pero además pueden guardar información en una pila (LIFO) sin límite de memoria. O sea, tienen memoria (además de “un core”) pero no es **RAM**.

A diferencia de los autómatas finitos, el rol de los PDA es comparativamente secundario (a nivel práctico y en cuanto a uso en demostraciones).

## Consumir palabras

Los PDA dan un modelo de computación “aceptador” (*acceptor*), que va consumiendo una palabra y finalmente la acepta o no.

# Un vistazo a los PDA

Los lenguajes libres de contexto son exactamente los aceptados por los **autómatas a pila** (“*push-down automata*”, **PDA**).

Los PDA tienen toda la estructura de un NFA pero además pueden guardar información en una pila (LIFO) sin límite de memoria. O sea, tienen memoria (además de “un core”) pero no es **RAM**.

A diferencia de los autómatas finitos, el rol de los PDA es comparativamente secundario (a nivel práctico y en cuanto a uso en demostraciones).

## Consumir palabras

Los PDA dan un modelo de computación “aceptador” (*acceptor*), que va consumiendo una palabra y finalmente la acepta o no.

**Ejemplo:**  $G_{01} := (\{S\}, \{0, 1\}, \{S \rightarrow \epsilon; S \rightarrow 0S1\}, S)$ .

# Viejo Pumping Lemma

## Lema de bombeo

Sea  $L$  **regular**. Toda palabra  $\alpha \in L$  larga ( $\geq k$ ) tiene un prefijo  $\beta\gamma$  corto ( $\leq k$ ) con una parte ( $\gamma \neq \epsilon$ ) “repetible” (sin salirse de  $L$ ).

# Viejo Pumping Lemma

## Lema de bombeo

Sea  $L$  **regular**. Toda palabra  $\alpha \in L$  larga ( $\geq k$ ) tiene un prefijo  $\beta\gamma$  corto ( $\leq k$ ) con una parte ( $\gamma \neq \epsilon$ ) “repetible” (sin salirse de  $L$ ).

su contrarrecíproca,

Si **para todo**  $k \in \mathbb{N}$  **existe**  $\alpha \in L$  con  $|\alpha| \geq k$  tal que **para todas**  $\beta, \gamma, \delta$  tales que  $\alpha = \beta\gamma\delta$ ,  $\gamma \neq \epsilon$  y  $|\beta\gamma| \leq k$ , **existe**  $n > 0$  tal que  $\beta\gamma^n\delta \notin L$ , **entonces**  $L$  **no es regular**.



UNC  
Universidad  
Nacional  
de Córdoba



# Viejo Pumping Lemma

## Lema de bombeo

Sea  $L$  **regular**. Toda palabra  $\alpha \in L$  larga ( $\geq k$ ) tiene un prefijo  $\beta\gamma$  corto ( $\leq k$ ) con una parte ( $\gamma \neq \epsilon$ ) “repetible” (sin salirse de  $L$ ).

su contrarrecíproca,

Si **para todo**  $k \in \mathbb{N}$  **existe**  $\alpha \in L$  con  $|\alpha| \geq k$  tal que **para todas**  $\beta, \gamma, \delta$  tales que  $\alpha = \beta\gamma\delta$ ,  $\gamma \neq \epsilon$  y  $|\beta\gamma| \leq k$ , **existe**  $n > 0$  tal que  $\beta\gamma^n\delta \notin L$ , entonces  $L$  no es regular.

y la versión **estratégica**:

- 1 **Adversario** elige  $k \in \mathbb{N}$ .
- 2 **Nosotros** debemos dar  $\alpha \in L$  con  $|\alpha| \geq k$ .
- 3 **Adversario** descompone  $\alpha$  en tres pedazos  $\beta\gamma\delta$  (pero debe cumplir que  $|\beta\gamma| \leq k$  y que  $\gamma \neq \epsilon$ ).
- 4 **Nosotros** damos un  $n > 0$  y armamos  $\beta\gamma^n\delta \notin L$ .

# Nuevo Pumping Lemma

## Lema de bombeo para lenguajes libres de contexto

Sea  $L$  libre de contexto. Toda palabra  $\alpha \in L$  larga ( $\geq k$ ) tiene un segmento  $\eta\beta\gamma$  corto ( $\leq k$ ) con extremos ( $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ) "repetibles" (sin salirse de  $L$ ).

# Nuevo Pumping Lemma

## Lema de bombeo para lenguajes libres de contexto

Sea  $L$  libre de contexto. Toda palabra  $\alpha \in L$  larga ( $\geq k$ ) tiene un segmento  $\eta\beta\gamma$  corto ( $\leq k$ ) con extremos ( $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ) "repetibles" (sin salirse de  $L$ ).

Si para todo  $k \in \mathbb{N}$  existe  $\alpha \in L$  con  $|\alpha| \geq k$  tal que para todos  $\zeta, \eta, \beta, \gamma, \delta$  tales que  $\alpha = \zeta\eta\beta\gamma\delta$ ,  $|\eta\beta\gamma| \leq k$  y  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ , existe  $n > 0$  tal que  $\zeta\eta^n\beta\gamma^n\delta \notin L$ , entonces  $L$  no es libre de contexto.

# Nuevo Pumping Lemma

## Lema de bombeo para lenguajes libres de contexto

Sea  $L$  libre de contexto. Toda palabra  $\alpha \in L$  larga ( $\geq k$ ) tiene un segmento  $\eta\beta\gamma$  corto ( $\leq k$ ) con extremos ( $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ) "repetibles" (sin salirse de  $L$ ).

Si para todo  $k \in \mathbb{N}$  existe  $\alpha \in L$  con  $|\alpha| \geq k$  tal que para todos  $\zeta, \eta, \beta, \gamma, \delta$  tales que  $\alpha = \zeta\eta\beta\gamma\delta$ ,  $|\eta\beta\gamma| \leq k$  y  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ , existe  $n > 0$  tal que  $\zeta\eta^n\beta\gamma^n\delta \notin L$ , entonces  $L$  no es libre de contexto.

## Versión estratégica

- 1 Adversario elige  $k \in \mathbb{N}$ .
- 2 Nosotros debemos dar  $\alpha \in L$  con  $|\alpha| \geq k$ .
- 3 Adversario descompone  $\alpha$  en cinco pedazos  $\zeta\eta\beta\gamma\delta$  (pero debe cumplir que  $|\eta\beta\gamma| \leq k$  y que  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ).
- 4 Nosotros damos un  $n > 0$  y armamos  $\zeta\eta^n\beta\gamma^n\delta \notin L$ .

# Ejemplo de lenguaje no libre de contexto

- 1 **Adversario** elige  $k \in \mathbb{N}$ .
- 2 **Nosotros** debemos dar  $\alpha \in L$  con  $|\alpha| \geq k$ .
- 3 **Adversario** descompone  $\alpha$  en cinco pedazos  $\zeta\eta\beta\gamma\delta$  (pero debe cumplir que  $|\eta\beta\gamma| \leq k$  y que  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ).
- 4 **Nosotros** damos un  $n > 0$  y armamos  $\zeta\eta^n\beta\gamma^n\delta \notin L$ .

# Ejemplo de lenguaje no libre de contexto

- 1 **Adversario** elige  $k \in \mathbb{N}$ .
- 2 **Nosotros** debemos dar  $\alpha \in L$  con  $|\alpha| \geq k$ .
- 3 **Adversario** descompone  $\alpha$  en cinco pedazos  $\zeta\eta\beta\gamma\delta$  (pero debe cumplir que  $|\eta\beta\gamma| \leq k$  y que  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ).
- 4 **Nosotros** damos un  $n > 0$  y armamos  $\zeta\eta^n\beta\gamma^n\delta \notin L$ .

$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}$  no es libre de contexto.

## Estrategia

# Ejemplo de lenguaje no libre de contexto

- 1 **Adversario** elige  $k \in \mathbb{N}$ .
- 2 **Nosotros** debemos dar  $\alpha \in L$  con  $|\alpha| \geq k$ .
- 3 **Adversario** descompone  $\alpha$  en cinco pedazos  $\zeta\eta\beta\gamma\delta$  (pero debe cumplir que  $|\eta\beta\gamma| \leq k$  y que  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ).
- 4 **Nosotros** damos un  $n > 0$  y armamos  $\zeta\eta^n\beta\gamma^n\delta \notin L$ .

$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}$  no es libre de contexto.

## Estrategia

- Adversario elige  $k$ .

# Ejemplo de lenguaje no libre de contexto

- 1 **Adversario** elige  $k \in \mathbb{N}$ .
- 2 **Nosotros** debemos dar  $\alpha \in L$  con  $|\alpha| \geq k$ .
- 3 **Adversario** descompone  $\alpha$  en cinco pedazos  $\zeta\eta\beta\gamma\delta$  (pero debe cumplir que  $|\eta\beta\gamma| \leq k$  y que  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ).
- 4 **Nosotros** damos un  $n > 0$  y armamos  $\zeta\eta^n\beta\gamma^n\delta \notin L$ .

$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}$  no es libre de contexto.

## Estrategia

- Adversario elige  $k$ .
- Nosotros proponemos  $a^k b^k c^k \in L$ .

# Ejemplo de lenguaje no libre de contexto

- 1 **Adversario** elige  $k \in \mathbb{N}$ .
- 2 **Nosotros** debemos dar  $\alpha \in L$  con  $|\alpha| \geq k$ .
- 3 **Adversario** descompone  $\alpha$  en cinco pedazos  $\zeta\eta\beta\gamma\delta$  (pero debe cumplir que  $|\eta\beta\gamma| \leq k$  y que  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ).
- 4 **Nosotros** damos un  $n > 0$  y armamos  $\zeta\eta^n\beta\gamma^n\delta \notin L$ .

$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}$  no es libre de contexto.

## Estrategia

- Adversario elige  $k$ .
- Nosotros proponemos  $a^k b^k c^k \in L$ .
- Adversario descompone  $a^k b^k c^k = \zeta \eta \beta \gamma \delta$

# Ejemplo de lenguaje no libre de contexto

- 1 **Adversario** elige  $k \in \mathbb{N}$ .
- 2 **Nosotros** debemos dar  $\alpha \in L$  con  $|\alpha| \geq k$ .
- 3 **Adversario** descompone  $\alpha$  en cinco pedazos  $\zeta\eta\beta\gamma\delta$  (pero debe cumplir que  $|\eta\beta\gamma| \leq k$  y que  $\eta \neq \epsilon$  ó  $\gamma \neq \epsilon$ ).
- 4 **Nosotros** damos un  $n > 0$  y armamos  $\zeta\eta^n\beta\gamma^n\delta \notin L$ .

$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}$  no es libre de contexto.

## Estrategia

- Adversario elige  $k$ .
- Nosotros proponemos  $a^k b^k c^k \in L$ .
- Adversario descompone  $a^k b^k c^k = \zeta \eta \beta \gamma \delta$
- Nosotros jugamos  $n = 2$  y ganamos:  $\zeta \eta^2 \beta \gamma^2 \delta \notin L$ .

## Lenguaje sensible al contexto

$$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}.$$

## Lenguaje sensible al contexto

$$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}.$$

$$S \longrightarrow aSBC \mid aBC$$

$$CB \longrightarrow BC$$

$$aB \longrightarrow ab$$

$$bB \longrightarrow bb$$

$$bC \longrightarrow bc$$

$$cC \longrightarrow cc$$

## Lenguaje sensible al contexto

$$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}.$$

$$\begin{aligned} S &\longrightarrow aSBC \mid aBC \\ CB &\longrightarrow BC \\ aB &\longrightarrow ab \\ bB &\longrightarrow bb \\ bC &\longrightarrow bc \\ cC &\longrightarrow cc \end{aligned}$$

## Definición

Una **gramática sensible al contexto** tiene producciones de la forma  $\alpha A \beta \longrightarrow \alpha \gamma \beta$  con  $\gamma \neq \epsilon$ , y  $S \longrightarrow \epsilon$  si  $S$  no ocurre en ningún lado derecho.

## Lenguaje sensible al contexto

$$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}.$$

$$S \longrightarrow aSBC \mid aBC$$

$$CB \longrightarrow BC$$

$$aB \longrightarrow ab$$

$$bB \longrightarrow bb$$

$$bC \longrightarrow bc$$

$$cC \longrightarrow cc$$

## Definición

Una **gramática sensible al contexto** tiene producciones de la forma  $\alpha A \beta \longrightarrow \alpha \gamma \beta$  con  $\gamma \neq \epsilon$ , y  $S \longrightarrow \epsilon$  si  $S$  no ocurre en ningún lado derecho.

# Gramáticas sensibles al contexto (CSG)

## Lenguaje sensible al contexto

$$L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}, n \geq 1\}.$$

$$\begin{aligned} S &\longrightarrow aSBC \mid aBC \\ CB &\longrightarrow BC \\ aB &\longrightarrow ab \\ bB &\longrightarrow bb \\ bC &\longrightarrow bc \\ cC &\longrightarrow cc \end{aligned}$$

## Truco de Révész

$$\begin{aligned} CB &\longrightarrow CZ \\ CZ &\longrightarrow WZ \\ WZ &\longrightarrow WC \\ WC &\longrightarrow BC \end{aligned}$$

## Definición

Una **gramática sensible al contexto** tiene producciones de la forma  $\alpha A \beta \longrightarrow \alpha \gamma \beta$  con  $\gamma \neq \epsilon$ , y  $S \longrightarrow \epsilon$  si  $S$  no ocurre en ningún lado derecho.

La familia de los lenguajes sensibles al contexto es enormemente más vasta que la de los libres de contexto.

La familia de los lenguajes sensibles al contexto es enormemente más vasta que la de los libres de contexto.

- Equivalen problemas resolubles/decidibles usando “un core + RAM proporcional a la longitud de la entrada”.

Sin embargo, casi todas las CSG admiten una forma muy simple

La familia de los lenguajes sensibles al contexto es enormemente más vasta que la de los libres de contexto.

- Equivalen problemas resolubles/decidibles usando “un core + RAM proporcional a la longitud de la entrada”.

Sin embargo, casi todas las CSG admiten una forma muy simple:

## Teorema (Forma normal de Kuroda)

*Toda CSG que no derive  $\epsilon$  es (débilmente) equivalente a una con producciones de las formas  $AB \rightarrow CD$ ,  $A \rightarrow BC$  ó  $A \rightarrow a$ .*

## Teorema (Forma normal de Kuroda)

*Toda CSG que no derive  $\epsilon$  es (débilmente) equivalente a una con producciones de las formas  $AB \rightarrow CD$ ,  $A \rightarrow BC$  ó  $A \rightarrow a$ .*

## Teorema (Forma normal de Kuroda)

Toda CSG que no derive  $\epsilon$  es (débilmente) equivalente a una con producciones de las formas  $AB \rightarrow CD$ ,  $A \rightarrow BC$  ó  $A \rightarrow a$ .

$$S \rightarrow aSBC \mid aBC$$

$$CB \rightarrow BC$$

$$aB \rightarrow ab$$

$$bB \rightarrow bb$$

$$bC \rightarrow bc$$

$$cC \rightarrow cc$$

## Teorema (Forma normal de Kuroda)

Toda CSG que no derive  $\epsilon$  es (débilmente) equivalente a una con producciones de las formas  $AB \rightarrow CD$ ,  $A \rightarrow BC$  ó  $A \rightarrow a$ .

$$S \rightarrow aSBC \mid aBC$$

$$CB \rightarrow BC$$

$$aB \rightarrow ab \text{ etc } \dots$$

$$bB \rightarrow bb$$

$$bC \rightarrow bc$$

$$cC \rightarrow cc$$

## Teorema (Forma normal de Kuroda)

Toda CSG que no derive  $\epsilon$  es (débilmente) equivalente a una con producciones de las formas  $AB \rightarrow CD$ ,  $A \rightarrow BC$  ó  $A \rightarrow a$ .

$$\begin{aligned} S &\rightarrow aSBC \mid aBC \\ CB &\rightarrow BC \\ aB &\rightarrow ab \text{ etc } \dots \\ bB &\rightarrow bb \\ bC &\rightarrow bc \\ cC &\rightarrow cc \end{aligned}$$

$$\begin{aligned} S &\rightarrow AE \mid AD \\ A &\rightarrow a \\ D &\rightarrow BC \\ E &\rightarrow SD \end{aligned}$$

## Teorema (Forma normal de Kuroda)

Toda CSG que no derive  $\epsilon$  es (débilmente) equivalente a una con producciones de las formas  $AB \rightarrow CD$ ,  $A \rightarrow BC$  ó  $A \rightarrow a$ .

$$\begin{aligned} S &\rightarrow aSBC \mid aBC \\ CB &\rightarrow BC \\ aB &\rightarrow ab \text{ etc } \dots \\ bB &\rightarrow bb \\ bC &\rightarrow bc \\ cC &\rightarrow cc \end{aligned}$$

$$\begin{aligned} S &\rightarrow AE \mid AD \\ A &\rightarrow a \\ D &\rightarrow BC \\ E &\rightarrow SD \end{aligned}$$

## La maldita palabra vacía

¿Qué pasaría si permitiéramos las producciones  $A \rightarrow \epsilon$  en la forma normal de Kuroda?

## Definición

Una gramática es **irrestricta** o de **tipo 0** si tiene producciones arbitrarias:

$\alpha \rightarrow \beta$  con  $\alpha \neq \epsilon$ .

## Definición

Una gramática es **irrestricta** o de **tipo 0** si tiene producciones arbitrarias:  
 $\alpha \longrightarrow \beta$  con  $\alpha \neq \epsilon$ .

## Teorema (Kuroda para gramáticas tipo 0)

*Toda gramática es (débilmente) equivalente a una con producciones de las formas  $AB \longrightarrow CD$ ,  $A \longrightarrow BC$ ,  $A \longrightarrow a$  ó  $A \longrightarrow \epsilon$ .*

## Definición

Una gramática es **irrestricta** o de **tipo 0** si tiene producciones arbitrarias:  
 $\alpha \longrightarrow \beta$  con  $\alpha \neq \epsilon$ .

## Teorema (Kuroda para gramáticas tipo 0)

*Toda gramática es (débilmente) equivalente a una con producciones de las formas  $AB \longrightarrow CD$ ,  $A \longrightarrow BC$ ,  $A \longrightarrow a$  ó  $A \longrightarrow \epsilon$ .*

El salto de CSG a gramáticas tipo 0 es más grande que el de CFG a CSG

## Definición

Una gramática es **irrestricta** o de **tipo 0** si tiene producciones arbitrarias:  
 $\alpha \rightarrow \beta$  con  $\alpha \neq \epsilon$ .

## Teorema (Kuroda para gramáticas tipo 0)

*Toda gramática es (débilmente) equivalente a una con producciones de las formas  $AB \rightarrow CD$ ,  $A \rightarrow BC$ ,  $A \rightarrow a$  ó  $A \rightarrow \epsilon$ .*

El salto de CSG a gramáticas tipo 0 es más grande que el de CFG a CSG:

- Para cualquier lenguaje  $L$  listable por cualquier método computacional conocido (cantidad de cores **arbitraria**, con acceso a memoria y tiempo de cómputo **ilimitados**) existe  $G$  de tipo 0 tal que  $L(G) = L$ .

Sea  $L \subseteq \Sigma^*$ .

Sea  $L \subseteq \Sigma^*$ .

## Definición

- $L$  es **computable (decidible, recursivo)** si hay un algoritmo que responde correctamente a la pregunta “ $\alpha \in L$ ” para cada  $\alpha$ .

Sea  $L \subseteq \Sigma^*$ .

## Definición

- $L$  es **computable (decidible, recursivo)** si hay un **algoritmo** que responde correctamente a la pregunta “ $\alpha \in L$ ” para cada  $\alpha$ .
- $L$  es **listable (recursivamente enumerable)** si hay un algoritmo que eventualmente lista todos las cadenas en  $L$ .

Sea  $L \subseteq \Sigma^*$ .

## Definición

- $L$  es **computable (decidible, recursivo)** si hay un algoritmo que responde correctamente a la pregunta “ $\alpha \in L$ ” para cada  $\alpha$ .
- $L$  es **listable (recursivamente enumerable)** si hay un algoritmo que eventualmente lista todos las cadenas en  $L$ .

El único requerimiento sobre el algoritmo en el primer ítem es que *termine*, sin limitación sobre uso de memoria y tiempo de cómputo.

Sea  $L \subseteq \Sigma^*$ .

## Definición

- $L$  es **computable (decidible, recursivo)** si hay un algoritmo que responde correctamente a la pregunta “ $\alpha \in L$ ” para cada  $\alpha$ .
- $L$  es **listable (recursivamente enumerable)** si hay un algoritmo que eventualmente lista todas las cadenas en  $L$ .

El único requerimiento sobre el algoritmo en el primer ítem es que *termine*, sin limitación sobre uso de memoria y tiempo de cómputo.

## Teorema

- *Hay conjuntos listables que no son recursivos.*
- *Los lenguajes de las gramáticas irrestrictas son exactamente los recursivamente enumerables.*

Una clasificación de gramáticas y sus lenguajes.

Una clasificación de gramáticas y sus lenguajes.

- **Tipo 3:** Regulares.  $A \rightarrow aB, A \rightarrow \epsilon$ .

Una clasificación de gramáticas y sus lenguajes.

- **Tipo 3:** Regulares.  $A \rightarrow aB, A \rightarrow \epsilon$ .
- **Tipo 2:** Libres de contexto.  $A \rightarrow \alpha$ .

Una clasificación de gramáticas y sus lenguajes.

- **Tipo 3:** Regulares.  $A \rightarrow aB, A \rightarrow \epsilon$ .
- **Tipo 2:** Libres de contexto.  $A \rightarrow \alpha$ .
- **Tipo 1:** Sensibles al contexto.  $\alpha A \beta \rightarrow \alpha \gamma \beta$  ( $\gamma \neq \emptyset$ ),  
 $S \rightarrow \epsilon$  (si no está del lado derecho).

Una clasificación de gramáticas y sus lenguajes.

- **Tipo 3:** Regulares.  $A \rightarrow aB, A \rightarrow \epsilon$ .
- **Tipo 2:** Libres de contexto.  $A \rightarrow \alpha$ .
- **Tipo 1:** Sensibles al contexto.  $\alpha A \beta \rightarrow \alpha \gamma \beta$  ( $\gamma \neq \emptyset$ ),  
 $S \rightarrow \epsilon$  (si no está del lado derecho).
- **Tipo 0:** Irrestringidas  $\alpha \rightarrow \gamma$  ( $\alpha \neq \epsilon$ ).

# Fin de la Tercera Parte

