Lenguajes formales y gramáticas

Pablo Verdes

LCC

4 de mayo de 2017

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 mucho más interesante que la sintaxis, y también más dificil 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 'buenas' 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 (lenguajes de tipo 3)
  2. Libres de contexto (lenguajes de tipo 2)
  3. Sensibles al contexto (lenguajes de tipo 1)
  4. Estructuradas por frases (lenguajes 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.

Lenguajes formales: conceptos básicos

- Un alfabeto es un conjunto finito y no vacío.
  Sus elementos reciben el nombre de símbolos, letras o caracteres.
  Notación: $\Sigma = \{a_1, a_2, \ldots, a_k\}$.

  Ejemplos:

  - $\Sigma = \{0, 1\}$, el alfabeto binario
  - $\Sigma = \{a, b, c, \ldots, z\}$, el conjunto de todas las letras minúsculas

- Una cadena o palabra sobre un alfabeto Σ es una secuencia finita de símbolos de Σ. Más precisamente, se la define como una función

  $$u : \{1, 2, \ldots, n\} \to \Sigma$$

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

  Se agrega la palabra especial $u : \{\} \to \Sigma$, llamada nula o vacía, considerada de longitud 0. Se la denota con λ o también ϵ.

Lenguajes formales: conceptos básicos

- Dada la palabra
$$u : \{1, 2, \ldots, n\} \to \Sigma$$
de longitud $n \geq 1$, $u(i)$ es su i-ésima letra.

  Por simplicidad de notación, escribiremos la cadena u como
$$u = u_1 u_2 \ldots u_n$$
donde cada $u_i \in \Sigma$.

  Ejemplos:

  - Si $\Sigma = \{a, b\}$ y la palabra $u : \{1, 2, 3\} \to \Sigma$ 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 $\Sigma = \{a, b, \ldots, z\}$

Lenguajes formales: conceptos básicos

- **Definiciones:**

$$\begin{aligned} \Sigma^n &= \{u \mid u : \{1, \ldots, n\} \to \Sigma\} && n \geq 1 \\ \Sigma^0 &= \{\lambda\} && n = 0 \end{aligned}$$

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

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

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

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

- Observemos que:
  - si $\Sigma = \emptyset$, resulta $\emptyset^* = \{\lambda\}$
  - si $\Sigma \neq \emptyset$, resulta Σ^* infinito numerable

Lenguajes formales: conceptos básicos

- Dado un alfabeto Σ y dos palabras

$$u : \{1, \ldots, m\} \to \Sigma \quad \text{y} \quad v : \{1, \ldots, n\} \to \Sigma,$$

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

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

En particular, $u\lambda = \lambda u = u$.

- 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. $u^n u = u u^n \qquad \forall n \geq 0$

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

$$\lambda^R = \lambda$$
$$(ua)^R = a u^R$$

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

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 Σ y dos lenguajes L_1 y L_2 sobre Σ,

- 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 $\overline{L} = \Sigma^* - L$
- 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 Σ y un lenguaje L sobre Σ,
  - 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

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

mientras que L^+ es la unión infinita

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

Operaciones sobre lenguajes

$$
\begin{aligned}
L^* &= \{\lambda\} \cup L \cup L^2 \cup \ldots \cup L^n \cup \ldots \\
L^+ &= \qquad\quad L \cup L^2 \cup \ldots \cup L^n \cup \ldots
\end{aligned}
$$

- 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

  - $\Sigma = \{$Julieta, eres, muy, hermosa$\}$
  - Reglas de producción:

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

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

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

    $\langle m \rangle \rightarrow \lambda \mid$ 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, σ) 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 $\lambda \to \beta$.
  - $(N \cup T)^*$ es el conjunto de cadenas sobre $N \cup T$.
  - A una producción de la forma (α, β) la notaremos $\alpha \to \beta$.
  - Una regla del tipo $\alpha \to \lambda$ recibe el nombre de **regla** λ.
- $\sigma \in N$ 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:

$$\sigma \to a\sigma bb$$
$$\sigma \to abb$$

  - σ es el símbolo inicial

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

$$\sigma \Rightarrow abb$$
$$\sigma \Rightarrow a\sigma bb \Rightarrow aabbbb$$
$$\sigma \Rightarrow a\sigma bb \Rightarrow aa\sigma bbbb \Rightarrow aaabbbbbb$$
$$\sigma \Rightarrow \ldots \Rightarrow a^n b^{2n}$$

Gramáticas

- **Ejemplo 2:**

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

donde el conjunto P consiste de las reglas de producción:

$$\sigma \to A\sigma B \tag{1}$$
$$A \to b \tag{2}$$
$$aaA \to aaBB \tag{3}$$
$$\sigma \to d \tag{4}$$
$$A \to aA \tag{5}$$
$$B \to dcd \tag{6}$$

- Ejemplo de generación de una palabra:

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

Gramáticas

- **Ejemplo 3:**

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

donde:

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

$$\sigma \to 000\sigma 111 \tag{1}$$

$$0\sigma 1 \to 01 \tag{2}$$

- ¿Qué palabras genera G_3?

$$\sigma \Rightarrow^{(1)} 000\sigma 111 \Rightarrow^{(2)} 000111$$

$$\sigma \Rightarrow^{(1)} 000\sigma 111 \Rightarrow^{(1)} (000)^2\sigma(111)^2 \Rightarrow^{(2)} (000)^2(111)^2$$

$$\sigma \Rightarrow^{(1,\ n\ veces)} (000)^n\sigma(111)^n \Rightarrow^{(2)} (000)^n(111)^n = 0^{3n}1^{3n} \quad n \geq 1$$

Gramáticas

- **Ejemplo 4:**

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

donde el conjunto P consiste de las reglas de producción:

$$\sigma \to a \mid aA \mid aB \tag{1}$$

$$A \to a \mid aA \mid bB \tag{2}$$

$$B \to b \mid bB \tag{3}$$

- ¿Qué tipo de palabras genera G_4?

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

Gramáticas

- **Ejemplo 5:**

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

donde el conjunto P consiste de las reglas de producción:

$$\sigma \to aA \tag{1}$$
$$A \to b \mid bB \tag{2}$$
$$B \to aA \tag{3}$$

- ¿Qué conjunto de palabras genera G_5?

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

Gramáticas

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

- Si $\alpha \to \beta$ 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 $\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_{n-1} \Rightarrow \alpha_n$, entonces diremos que α_n deriva de α_1, o que α_1 produce α_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: hemos visto que
  - $L(G_1) = \{a^n b^{2n} \mid n \geq 1\}$
  - $L(G_3) = \{0^{3n} 1^{3n} \mid n \geq 1\}$
  - $L(G_4) = \{a^m b^n \mid m > 0 \wedge n \geq 0\}$
  - $L(G_5) = \{(ab)^n \mid n > 0\}$

- 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:

$$A \to a$$

$$A \to aB$$

$$A \to \lambda$$

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:

$$A \to a$$
$$A \to Ba$$
$$A \to \lambda$$

- Por ejemplo, reglas de la forma:

$$yW \to x$$
$$X \to xZy$$
$$YX \to WvZ$$

  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:

$$S \to yX$$
$$X \to xxX$$
$$X \to yY$$
$$Y \to \lambda$$

- **Ejercicio:** Convierta la siguiente gramática regular en otra gramática regular que no contenga reglas que terminen en un solo terminal:

$$S \to xS$$
$$S \to y$$
$$X \to z$$

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:

$$A \to \delta$$

  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 $xNy \to xzy$ 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 δ en $A \rightarrow \delta$, podría ocurrir que δ contenga más de un no terminal, como en $A \rightarrow zXYz$.
- 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:

    $$S \to aSb \mid \lambda$$

    genera el lenguaje

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

  - La gramática con producciones:

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

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

    Ejemplo:

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

- **Ejercicio:** Dar una gramática 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:

$$\alpha A\beta \to \alpha\delta\beta$$

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

- Adicionalmente, se permite la regla $\sigma \to \lambda$, donde σ es el símbolo inicial, siempre que σ no aparezca en el lado derecho de otra producción. Sirve para agregar λ al lenguaje.

- Obs.:
  - La producción $abA \to baab$ no es sensible al contexto.
    Lo sería si fuese $abA \to abab$.
  - La producción $aSb \to ab$ 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 $\alpha \to \beta$ 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:

$$\alpha \to \delta$$

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 $\lambda \to \delta$.

Relación entre gramáticas

- Llamando

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

se tiene que

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

- Obs. que las inclusiones son en sentido estricto. ¿Qué quiere decir esto?

Tipos de lenguajes

- **Definición:** Diremos que un lenguaje L es de tipo i $(i = 0, 1, 2, 3)$ sii:
  1. existe una gramática G de tipo i tal que $L(G) = L$, y
  2. no existe una gramática G' de tipo j con $j > i$ tal que $L(G') = L$.

- **Ejemplos:**
  - $L = \{w \in \{a, b\}^* \mid \exists k \in \mathbb{N}_0$ tal que $N_a(w) = 3k\}$ es generado por la gramática $G = (\{\sigma, U, D\}, \{a, b\}, P, \sigma)$ con producciones P:

$$\sigma \to aU \mid b\sigma \mid \lambda$$
$$U \to aD \mid bU$$
$$D \to a\sigma \mid bD$$

    G es de tipo 3 $\Rightarrow$ L es regular

  - Vimos que $L' = \{a^n b^n \mid n \in \mathbb{N}_0\}$ es generado por la gramática con producciones $S \to aSb \mid \lambda$, que es de tipo 2. ¿Existe una gramática G de tipo 3 (regular) tal que $L(G) = L'$? Más adelante veremos que no. Por lo tanto, L' es de tipo 2.