

Introducción a la Lógica y la Computación

Parte III: Lenguajes y Autómatas

Clase del 4 de Noviembre de 2015

Lenguajes Formales y Teoría de Autómatas

- ▶ Hay fuertes relaciones entre la Teoría de Lenguajes Formales y la Teoría de Máquinas Abstractas.
- ▶ Ya vimos en la materia algunas relaciones (equivalencia entre Expresiones Regulares y Autómatas).
- ▶ Hay otros lenguajes formales muy útiles (sobre todo para los lenguajes de programación!): [Gramáticas](#).
- ▶ Vamos a estudiar gramáticas, las cosas que podemos expresar con ellas, y relaciones con otros formalismos (lenguajes y/o autómatas).

Lingüística y Ciencias de la Computación

- ▶ Dos “tipos” de lenguajes: **naturales** (inglés, español, etc.) y **formales** (matemática, lógica,...).
- ▶ Lenguajes Naturales: más flexibles, es más difícil describirlos.
- ▶ Lenguajes Formales: poseen reglas de sintaxis y semántica más rígidas.
- ▶ En la década del '50, Chomsky a través de la lingüística brinda valiosas herramientas para el estudio de lenguajes formales (y para las ciencias de la computación).
- ▶ Jerarquía de Chomsky: clasificación de los lenguajes de acuerdo a ciertas restricciones (las veremos más adelante).

Gramáticas y Lenguajes Naturales

$\langle oracion \rangle \rightarrow \langle sujeto \rangle \langle predicado \rangle$
 $\langle sujeto \rangle \rightarrow \langle sujeto \rangle \langle adjetivo \rangle$
 $\langle sujeto \rangle \rightarrow el\ perro$
 $\langle adjetivo \rangle \rightarrow pequeño$
 $\langle adjetivo \rangle \rightarrow grande$
 $\langle adjetivo \rangle \rightarrow bueno$
 $\langle predicado \rangle \rightarrow \langle verbo \rangle \langle adjetivo \rangle$
 $\langle verbo \rangle \rightarrow es$

Gramáticas y Lenguajes Naturales - Ejemplo

Supongamos que queremos derivar la frase “*El perro bueno es grande*”:

$\langle oracion \rangle \Rightarrow \langle sujeto \rangle \langle predicado \rangle$
 $\Rightarrow \langle sujeto \rangle \langle adjetivo \rangle \langle predicado \rangle$
 $\Rightarrow el\ perro \langle adjetivo \rangle \langle predicado \rangle$
 $\Rightarrow el\ perro \langle adjetivo \rangle \langle verbo \rangle \langle adjetivo \rangle$
 $\Rightarrow el\ perro\ bueno \langle verbo \rangle \langle adjetivo \rangle$
 $\Rightarrow el\ perro\ bueno\ es \langle adjetivo \rangle$
 $\Rightarrow el\ perro\ bueno\ es\ grande$

Sin embargo, con esta gramática podemos derivar también la frase: “*El perro grande es pequeño*”.

Definiciones

- ▶ Una gramática es un objeto formal para especificar, de manera finita, el conjunto de cadenas que constituyen un lenguaje.
- ▶ Primordialmente usamos las gramáticas para generar las distintas cadenas del lenguaje que define aplicando de manera sucesiva las reglas de producción que la conforman.
- ▶ Una gramática define la estructura de las frases y de las palabras de un lenguaje.

Definición

Una gramática es una 4-upla $\langle V, T, P, S \rangle$ tal que:

- ▶ V es un conjunto finito, de elementos llamados variables o *no terminales* o categorías sintácticas.
- ▶ T es un conjunto finito de elementos llamados *terminales*. $V \cap T = \emptyset$.
- ▶ S es una variable especial llamada *símbolo inicial*.
- ▶ P es un conjunto finito cuyos elementos son llamados *producciones*, tal que cada producción es de la forma $\alpha \rightarrow \beta$, $\alpha, \beta \in (V \cup T)^*$.

Ejemplo de Gramática

- ▶ $V = \{\langle oracion \rangle, \langle sujeto \rangle, \langle adjetivo \rangle, \langle predicado \rangle, \langle verbo \rangle\}$
- ▶ $T = \{a, b, c, d, e, \dots, z\}$
- ▶ $S = \langle oracion \rangle$
- ▶ $P = \{\langle oracion \rangle \rightarrow \langle sujeto \rangle \langle predicado \rangle, \langle sujeto \rangle \rightarrow \langle sujeto \rangle \langle adjetivo \rangle, \langle sujeto \rangle \rightarrow el\ perro, \langle adjetivo \rangle \rightarrow pequeño, \langle adjetivo \rangle \rightarrow grande, \langle adjetivo \rangle \rightarrow bueno, \langle predicado \rangle \rightarrow \langle verbo \rangle \langle adjetivo \rangle, \langle verbo \rangle \rightarrow es\}$

- ▶ Una gramática produce un único lenguaje, pero el mismo lenguaje puede ser definido por muchas gramáticas.
- ▶ Podríamos clasificar las gramáticas según el lenguaje que generan.
- ▶ Es decir dos gramáticas pertenecerán a la misma clase si generan el mismo lenguaje.
- ▶ **Problema!** No es **computable**.

*No existe **ningún** algoritmo que dadas dos gramáticas decida si las dos definen el mismo lenguaje.*

- ▶ Las gramáticas se clasifican en base a la **forma** de las mismas, y no de acuerdo a la naturaleza del lenguaje que generan.
- ▶ Es decir, se pueden caracterizar en función de la forma de sus **producciones** y así capturar su complejidad.
- ▶ Noam Chomsky logró clasificarlas de esta manera demostrando además que es una clasificación jerárquica – gramáticas de un nivel están incluidas en los siguiente.

- ▶ **Tipo 0:** Gramáticas sin restricciones.
- ▶ **Tipo 1:** Gramáticas sensibles al contexto.
- ▶ **Tipo 2:** Gramáticas independientes del contexto.
- ▶ **Tipo 3:** Gramáticas regulares.

- ▶ **Gramáticas de Tipo 0:** Las producciones tienen la forma

$$\alpha \rightarrow \beta, \text{ donde } \alpha \in (V \cup T)^+, \beta \in (V \cup T)^*.$$

Notar que para estas producciones no se establece ningún tipo de restricción respecto a su forma.

- ▶ **Gramáticas de Tipo 1:** Las reglas de producción de estas gramáticas tienen la forma

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2, \text{ donde } \alpha_1, \alpha_2 \in (V \cup T)^*, A \in V, \beta \in (V \cup T)^+.$$

En otras palabras, sólo se permite la sustitución del símbolo no terminal A por la cadena β cuando el símbolo A aparezca en el contexto indicado por α_1 y α_2 .

- ▶ **Gramáticas de Tipo 2:** también denominadas gramáticas Libres de Contexto se caracterizan porque su conjunto de producciones se ajustan al siguiente esquema

$$A \rightarrow \alpha, \text{ donde } A \in V, \alpha \in (V \cup T)^*.$$

A diferencia de las gramáticas de tipo 1 resulta que el símbolo A siempre se puede sustituir por la cadena α independientemente del contexto en el que aparezca el símbolo A .

- ▶ **Gramáticas de Tipo 3:** o gramáticas Regulares, son aquellas cuyas producciones tienen la forma

$A \rightarrow xB$, donde $A, B \in V, x \in T^*$. Además, B puede o no aparecer.

Hay dos tipos de gramáticas regulares:

- ▶ **Lineales a la derecha**, con producciones de la forma

$$A \rightarrow xB$$

$$A \rightarrow x$$

donde $A, B \in V, x \in T^*$.

- ▶ **Lineales a la izquierda**, con producciones de la forma

$$A \rightarrow Bx$$

$$A \rightarrow x$$

donde $A, B \in V, x \in T^*$.

Gramáticas Regulares (RG)

- ▶ Los lenguajes generados por una gramática regular, son lenguajes regulares.
- ▶ Es decir, dada una RG G , el lenguaje $L(G)$ es un lenguaje regular.
- ▶ Hay equivalencia con otros formalismos vistos antes: DFA, NFA, ER.
- ▶ Vamos a estudiar un poco las gramáticas regulares.

RG - Ejemplo

Encontremos una gramática que genere el lenguaje asociado a la expresión regular a^*b^* . Es decir, $L = \{a^n b^m \mid 0 \leq n, m\}$. La RG G tal que $L(G) = L$ es

$$\begin{aligned} S &\rightarrow aS \mid aT \mid bT \mid \epsilon \\ T &\rightarrow bT \mid \epsilon. \end{aligned}$$

Gramáticas Regulares y Autómatas

Proposición

Sea $G = \langle V, T, P, S \rangle$ una gramática regular, entonces existe $M = (Q, \Sigma, \delta, q_0, F)$ un NFA (sin movimientos ϵ) tal que $L(G) = L(M)$.

Explícitamente:

- ▶ $Q = V$,
- ▶ $\Sigma = T$
- ▶ $q_0 = S$,
- ▶ $B \in \delta(A, a)$ sii $A \rightarrow aB$,
- ▶ $F = \{A \in V : A \rightarrow \epsilon\}$.

Teorema

Sea r una expresión regular sobre un alfabeto Σ , entonces existe una gramática regular G con símbolos terminales en Σ tal que $L(r) = L(G)$.

Explícitamente:

1. $L(\emptyset) = \emptyset = L(G)$, donde G es una gramática sin producciones.
2. $L(\epsilon) = \{\epsilon\} = L(G)$, donde G tiene una única producción $S \rightarrow \epsilon$.
3. Para cada $a \in \Sigma$, $L(a) = \{a\} = L(G)$, donde G tiene producciones $S \rightarrow aU$ y $U \rightarrow \epsilon$.

Supongamos que r y r' son expresiones regulares tales que $L(r) = L(H)$, $L(r') = L(H')$ con $H = \langle V, \Sigma, P, S \rangle$ y $H' = \langle V', \Sigma, P', S' \rangle$. Supongamos además, sin pérdida de generalidad, que $V \cap V' = \emptyset$, es decir que no tienen variables comunes. Debido a la definición de gramática regular, el conjunto P es de la forma:

$$\{C_k \rightarrow c_k D_k\} \cup \{U_t \rightarrow \epsilon\},$$

para $k = 1, \dots, m, t = 1, \dots, s$. Entonces:

4. $L(r + r') = L(G)$, donde G tiene como variables $V_G = V \cup V' \cup \{S_G\}$, variable inicial S_G , y producciones

$$P_G = P \cup P' \cup \{S_G \rightarrow \alpha : \text{si } S \rightarrow \alpha \text{ ó } S' \rightarrow \alpha\}$$

5. $L(rr') = L(G)$, donde G tiene como variables $V_G = V \cup V'$, variable inicial S , y producciones $P_G = P^{(1)} \cup P'$, donde $P^{(1)}$ es igual a:

$$\{C_k \rightarrow c_k D_k\} \cup \{C_k \rightarrow c_k S' : \text{si } D_k \rightarrow \epsilon \in P\},$$

para $k = 1, \dots, m$.

6. $L(r^*) = L(G)$, donde G tiene como variables $V_G = V$, variable inicial S , y producciones

$$P_G = P \cup \{S \rightarrow \epsilon\} \cup \{C_k \rightarrow c_k S : \text{si } C_k \rightarrow \epsilon \in P\},$$

para $k = 1, \dots, m$.

Bibliografía

- ▶ *“Introducción a la Lógica y la Computación, Parte III: Lenguajes y Autómatas” (versión 2015)*. Raul Fervari y Ezequiel Orbe.
- ▶ Notas del Curso *“Autómatas y Lenguajes”*, de la Universidad Nacional de Río Cuarto. Francisco Bavera y Jorge Aguirre.
- ▶ *“Teoría de la Computación (Lenguajes Formales, Computabilidad y Complejidad)”*. Gonzalo Navarro.
- ▶ *“Introducción a la teoría de autómatas, lenguajes y computación”*. Jeffrey Ullman; John Hopcroft; Rajeev Motwani.