Autómatas y Lenguajes Formales 2017-1
Facultad de Ciencias UNAM
Nota de Clase 8

Favio E. Miranda Perea   A. Liliana Reyes Cabello   Lourdes González Huesca

28 de septiembre de 2016

Los **lenguajes libres de contexto** especializan o refinan a los lenguajes regulares, es decir se pueden describir cadenas más expresivas que en su conjunto establecen lenguajes más complejos como lenguajes de programación de alto nivel u otros lenguajes formales.

Estos lenguajes serán estudiados comenzando con las gramáticas que los definen para estudiar las características que los distinguen de los lenguajes regulares.

Hacemos énfasis en que la descripción de un lenguaje por medio de una gramática permite que la descripción y análisis del lenguaje sea claro.

1. Gramáticas Libres de Contexto

Definición 1 *Una* ***gramática es libre o independiente del contexto*** *si todas sus producciones son de la forma*

$$A \to \alpha$$

con $A \in V,\ \alpha \in (V \cup T)^{\star}$. *Se incluye a la regla* $S \to \varepsilon$.

Ejemplo: Las siguientes son gramáticas libres de contexto, en forma compacta están representadas por expresiones regulares:

- $L = a^{\star}$

$$S \to aS \mid \varepsilon$$

- $L = a^{\star}b^{\star}$

$$\begin{array}{rcl} S & \to & aS \mid bA \mid \varepsilon \\ A & \to & bA \mid b \mid \varepsilon \end{array}$$

- $L = 0^{+}1^{+}$

$$\begin{array}{rcl} S & \to & CU \\ C & \to & 0C \mid 0 \\ U & \to & 1U \mid 1 \end{array}$$

Ejemplo: Los siguientes ejemplos son lenguajes con una gramática libre de contexto que los genera:

- $L = \{a^n b a^m \mid n, m \geq 1\} = a^+ b a^+$

$$\begin{array}{rcl} S & \to & aS \mid aB \\ B & \to & bC \\ C & \to & aC \mid a \end{array}$$

- $L = \{a^n b^n \mid n \in \mathbb{N}\}$ (no es regular)

$$S \to aSb \mid \boldsymbol{\varepsilon}$$

- $L = \{w \in \{a, b\}^\star \mid w = w^R\}$ (no es regular)

$$S \to aSa \mid bSb \mid a \mid b \mid \boldsymbol{\varepsilon}$$

De los ejemplos anteriores podemos atestiguar que los lenguajes libres de contexto permiten expresar lenguajes más refinados como $a^n b^n$. También podemos observar que en particular toda gramática regular es libre de contexto.

1.1. Derivaciones y árboles

Una derivación formal usando gramáticas libres de contexto puede ser de dos tipos, por la izquierda o por la derecha, dependiendo de la forma en que se reescriben los símbolos no terminales.

Definición 2 *Una derivación $S \to^\star w$ es una* ***derivación por la izquierda*** *si en cada paso se reescribe la variable más a la izquierda en la palabra.*

Definición 3 *Una derivación $S \to^\star w$ es una* ***derivación por la derecha*** *si en cada paso se reescribe la variable más a la derecha en la palabra.*

Ejemplo: En la gramática

$$S \to aAs \mid a \qquad\qquad A \to SbA \mid SS \mid ba$$

1. Tenemos la siguiente derivación por la izquierda de la cadena *aabbaa*

$$\mathbf{S} \to a\mathbf{A}S \to a\mathbf{S}bAS \to aab\mathbf{A}S \to aabba\mathbf{S} \to aabbaa$$

2. Tenemos la siguiente derivación por la derecha de *aabbaa*

$$\mathbf{S} \to aA\mathbf{S} \to a\mathbf{A}a \to aSb\mathbf{A}a \to a\mathbf{S}bbaa \to aabbaa$$

Por otro lado también se puede verificar que una cadena es generada por una gramaática usando árboles de derivación o árboles sintácticos. Este mecanismo es muy útil para representar las derivaciones de gramáticas libres de contexto.

El uso de estas estructuras tienen beneficios tales como su utilización en compiladores, en específico para el análisis sintáctico de programas fuente o *parsing* y sirven de base para la generación de código.

1.2. Constucción de árbol es de derivación

Dada una gramática libre de contexto $G = \langle V, T, S, P \rangle$, un árbol de derivación en G se construye como sigue:

1. La raíz contiene al símbolo inicial S.

2. Cada nodo interior contiene una variable.

3. Cada hoja contiene un símbolo de $V \cup T \cup \{\varepsilon\}$.

4. Si un nodo interior contiene una variable A entonces sus hijos contienen símbolos (de izquierda a derecha) $a_1, \ldots, a_n$ si y sólo si $A \to a_1 a_2 \ldots a_n$ está en P.

5. La palabra generada se puede reconstriur al leer las hojas de izquierda a derecha.

2. Ambigüedad

Puede ser que dos derivaciones distintas tengan un mismo árbol de derivación y también puede suceder que haya más de un árbol de derivación para una cadena. Lo ideal es que cada cadena tenga sólo un árbol asociado. Si sucede lo anterior, implica que el lenguaje **no es ambiguo**. Desafortunadamente existen lenguajes ambiguos. Veamos una definición formal de este fenómeno:

Definición 4 *Una gramática se dice **ambigua** si existe una palabra w con dos o más árboles de derivación distintos. En general una palabra puede tener más de una derivación, pero* un sólo árbol *y en tal caso no hay ambigüedad.*

Algunas veces se puede suprimir la ambigüedad directamente. Sin embargo **no** hay un algoritmo para remover ambigüedad. Así también hay algunos lenguajes cuya ambigüedad es inevitable.

Ejemplo: Considere la siguiente gramática:

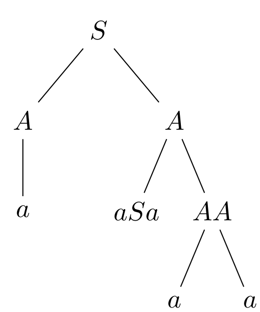
$$S \to AA \quad A \to aSa \mid a$$

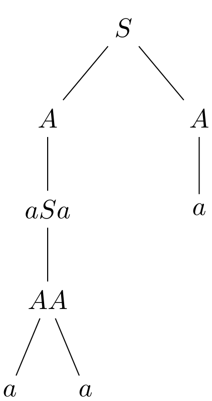
La palabra a^5 tiene las siguientes derivaciones diferentes y por la izquierda:

$$S \to AA \to aA \to aaSa \to aaAAa \to aaaAa \to aaaaa$$

$$S \to AA \to aSaA \to aAAaA \to aaAaA \to aaaaA \to aaaaa$$

Las dos derivaciones generan árboles distintos:





2.1. Lenguajes Ambiguos

Definición 5 *Un lenguaje L es ambiguo si existe una gramática ambigua G que genera a L. Un lenguaje es* ***inherentemente*** *ambiguo si todas las gramáticas que lo generan son ambiguas.*

Ejemplo: El lenguaje $L = \{a^{2+3i} \mid i \geq 0\}$ es ambiguo por ser generado por la siguiente gramática ambigua:

$$S \to AA \qquad\qquad A \to aSa \mid a$$

Sin embargo este lenguaje también es generado por una gramática no ambigua:

$$S \to aa \mid aaU \qquad\qquad U \to aaaU \mid aaa$$

En este caso la derivación de a^5 es:

$$S \to aaU \to aaaaa$$

y por lo tanto L no es un lenguaje inherentemente ambiguo.

Ejemplo: El lenguaje

$$L = \{a^n b^n c^m d^m \mid n, m \geq 1\} \cup \{a^n b^m c^m d^n \mid n, m \geq 1\}$$

es generado por la gramática:

$$\begin{array}{lll} S \to AB \mid C & A \to aAb \mid ab & B \to cBd \mid cd \\ C \to aCd \mid aDd & D \to bDc \mid bc & \end{array}$$

La cadena $aabbccdd$ tiene dos derivaciones por la izquierda:

$$S \to AB \to aAbB \to aabbB \to aabbcBd \to aabbccdd$$

$$S \to C \to aCd \to aaDdd \to aabDcdd \to aabbccdd$$

Como se comentó antes, la ambigüedad no puede ser eliminada con un método y en general, probar la ambigüedad inherente es complicado.

3. Propiedades de Cerradura

Estudiemos ahora a los lenguajes libres de contexto, la clase de estos lenguajes es cerrada bajo las siguientes operaciones:

- Unión: si L_1, L_2 son lenguajes libres del contexto entonces $L_1 \cup L_2$ es un lenguaje libre del contexto.
- Concatenación: si L_1, L_2 son lenguajes libres del contexto entonces $L_1 L_2$ es un lenguaje libre del contexto.
- Estrella de Kleene: si L_1 es un lenguaje libres del contexto entonces $L_1^\star$ es un lenguaje libre del contexto.

Veamos más a detalle estas operaciones:

- Cerradura bajo la unión
  Si $G_1 = \langle V_1, T, S_1, P_1\rangle$, $G_2 = \langle V_2, T, S_2, P_2\rangle$ son dos gramáticas libres de contexto donde $L_1 = L(G_1)$ y $L_2 = L(G_2)$ entonces $L_1 \cup L_2 = L(G)$ y G es la gramática

  $$G = \langle V_1 \cup V_2 \cup \{S\}, T, S, P\rangle$$

  y las reglas de producción están dadas por $P_1 \cup P_2$ más las producciones:

  $$S \to S_1 \qquad S \to S_2$$

- Cerradura bajo la concatenación
  Si $G_1 = \langle V_1, T, S_1, P_1\rangle$, $G_2 = \langle V_2, T, S_2, P_2\rangle$ son dos gramáticas libres de contexto donde $L_1 = L(G_1)$ y $L_2 = L(G_2)$ entonces $L_1 L_2 = L(G)$ donde G es la gramática

  $$G = \langle V_1 \cup V_2 \cup \{S\}, T, S, P\rangle$$

  y P está dado por $P_1 \cup P_2$ más la producción:

  $$S \to S_1 S_2$$

- Cerradura bajo la estrella de Kleene
  Si $G_1 = \langle V_1, T, S_1, P_1\rangle$ es una gramática libre de contexto con $L_1 = L(G_1)$ entonces $L_1^\star = L(G)$ donde G es la gramática

  $$G = \langle V_1 \cup \{S\}, T, S, P\rangle$$

  y P está dado por P_1 más la producciones:

  $$S \to S_1 S_1 \qquad S \to \varepsilon$$

La intuición puede traicionar y algunas propiedades de cerradura no son válidas. En general las siguientes propiedades **no son válidas** para lenguajes libres del contexto.

- Cerradura bajo la intersección.
- Cerradura bajo el complemento.
- Cerradura bajo la diferencia.

Estudiemos a detalle estas cerraduras:

- Intersección
  La intersección de dos lenguajes libres de contexto puede ser un lenguaje que **no** es libre del contexto.
  Por ejemplo considérense $L_1 = \{a^i b^i c^j \mid i, j \geq 1\}$ libre del contexto:

  $$S \to AB,\ A \to aAb \mid ab,\ B \to cC \mid c$$

  y $L_2 = \{a^i b^j c^j \mid i, j \geq 1\}$ también libre del contexto:

  $$S \to AB,\ A \to aA \mid a,\ B \to bBc \mid bc$$

  Pero $L_1 \cap L_2 = \{a^i b^i c^i \mid i \geq 1\}$ no es independiente del contexto.

- Complemento
  Si el complemento de un lenguaje libre de contexto L, $\bar{L}$ fuera también libre del contexto entonces la intersección también lo será pues:
  $$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$
- Diferencia
  Si la diferencia fuera un lenguaje libre de contexto, entonces también lo será el complemento pues:
  $$\overline{L} = \Sigma^\star - L$$