Autómatas y Lenguajes Formales 2019-II

Homomorfismos, lenguajes de Dyck y el Teorema de Chomsky-Schuetzenberger, jerarquía de Chomsky

Dr. Favio Ezequiel Miranda Perea
favio@ciencias.unam.mx

Facultad de Ciencias UNAM[1]

3 de febrero de 2020

[1] Con el apoyo del proyecto PAPIME PE102117

Homomorfismos de lenguajes

- Sean Σ, Δ dos alfabetos. Una función $h : \Sigma^\star \to \Delta^\star$ es un homomorfismo si y sólo si
  - $h(\varepsilon) = \varepsilon$
  - $h(xy) = h(x)h(y)$

- Un homomorfismo h queda determinado de manera única por sus valores en Σ.

- Más aún, cualquier función $f : \Sigma \to \Delta^\star$ puede extenderse de manera única a un homomorfismo $h : \Sigma^\star \to \Delta^\star$ tal que si $x \in \Sigma$ entonces $h(x) = f(x)$. Este homomorfismo se define como:

$$h(s_1 s_2 \dots s_n) =_{def} f(s_1)f(s_2)\dots f(s_n)$$

  para cada $s_i \in \Sigma$

Imágen e imagen inversa de una función

Dados una función $f : A \to B$, $X \subseteq A$ y $Y \subseteq B$, definimos:

- La imagen de X bajo f, denotada $f[X]$, como

$$f[X] = \{f(x) \mid x \in A\}$$

- La imagen inversa de Y bajo f, denotada $f^{-1}[Y]$, como

$$f^{-1}[Y] = \{x \mid f(x) \in Y\}$$

- Obsérvese que $f[X] \subseteq B$ y $f^{-1}[Y] \subseteq A$
- Propiedad relevante: $f[X \cup Z] = f[X] \cup f[Z]$

Propiedades de cerradura

Homomorfismos de lenguajes

Sean $h : \Sigma^\star \to \Delta^\star, L \subseteq \Sigma^\star, S \subseteq \Delta^\star$.

- Si L es regular entonces $h[L]$ es regular. Es decir, la imagen de un lenguaje regular bajo un homomorfismo es regular.
- Si S es regular entonces $h^{-1}[S]$ es regular. Es decir, la imagen inversa de un lenguaje regular bajo un homomorfismo es regular.
- Si L es libre de contexto entonces $h[L]$ es libre de contexto. Es decir, la imagen de un lenguaje libre de contexto bajo un homomorfismo es libre de contexto.
- Propiedad de cerradura importante: Si R es regular y L es libre de contexto entonces $R \cap L$ es libre de contexto.

Homomorfismos de borrado

- Sean Σ, B dos alfabetos ajenos, es decir, $\Sigma \cap B = \varnothing$.
- Definimos $f : \Sigma \cup B \to \Sigma^\star$ como

$$
\begin{aligned}
f(s) &= \sigma \quad \text{si } s \in \Sigma \\
f(s) &= \varepsilon \quad \text{si } s \in B
\end{aligned}
$$

- f induce un único homomorfismo $h : (\Sigma \cup B)^\star \to \Sigma^\star$ de tal forma que $h(x) = y$ si y sólo si y se obtiene a partir de x al borrar todos los símbolos de B.
- A este homomorfismo lo denotamos con $erase_B$
- Ejemplo: Si $\Sigma = \{0, 1\}$ y $B = \{(,)\}$ entonces $erase_B$ es el homomorfismo que borra paréntesis, por ejemplo:

$$erase_B\Big(00(11)(0)(01)\Big) = 0011001$$

Gramática de borrado a partir de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto y $R \subseteq T$
- Definimos la gramática que borra a (símbolos de) R en G, denotada $Er_R(G)$, como

$$Er_R(G) = \langle V, T - R, S, \mathcal{P}' \rangle$$

- Donde $\mathcal{P}'$ se define como sigue:
  - Si $X \to v$ es una producción en $\mathcal{P}$, entonces $X \to erase_R(v)$ es una producción de $\mathcal{P}'$.
- Si G es una gramática libre de contexto y $R \subseteq T$, entonces

$$L(Er_R(G)) = erase_R[L(G)]$$

- Corolario: Si $L \subseteq \Sigma^{\star}$ es libre de contexto y $R \subseteq \Sigma$ entonces $erase_R[L]$ es libre de contexto.

Gramática separadora de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto en forma normal de Chomsky.
- Definimos la gramática separadora de G, denotada G_{sep} como

$$G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$$

- El conjunto de producciones $\mathcal{P}'$ se define como sigue:
  - Las producciones de $\mathcal{P}$ de la forma $A \to a$, están en $\mathcal{P}'$
  - Para cada producción en $\mathcal{P}$ de la forma $A_i \to B_i C_i$, $\mathcal{P}'$ tiene la siguiente producción

$$A_i \to p_i B_i \bar{p}_i C_i$$

- Donde $B = \{p_1, \bar{p}_1, \ldots, p_n, \bar{p}_n\}$ es un conjunto de símbolos nuevos, i.e., $B \cap T = \varnothing$
- B es esencialmente un conjunto de n juegos distintos de paréntesis.

Gramática separadora

Ejemplo

- Sea G dada por

$$S \to XY \quad S \to YX \quad Y \to ZZ \quad X \to a \quad Z \to a$$

- G es ambigua:

$$S \to XY \to aY \to aZZ \to aaZ \to aaa$$

$$S \to YX \to ZZX \to aZX \to aaX \to aaa$$

- G_{sep} está dada por

$$S \to (X)Y \quad S \to [Y]X \quad Y \to \{Z\}Z \quad X \to a \quad Z \to a$$

- G_{sep} separa y codifica las dos derivaciones en G:

$$S \to (X)Y \to (a)Y \to a\{Z\}Z \to (a)\{a\}Z \to (a)\{a\}a$$

$$S \to [Y]X \to [\{Z\}Z]X \to [\{a\}Z]X \to [\{a\}a]X \to [\{a\}a]a$$

Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \ldots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \varnothing$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
  - $B_1 = \{(,)\}$
  - $B_2 = \{(,), [,]\}$
  - $B_3 = \{(,), [,], \{, \}\}$
- Con $D_n(A)$ denotamos al lenguaje de cadenas de $A \cup B_n$ generadas por la siguiente gramática libre de contexto
  - Para toda $a \in A,\ S \to a$
  - Para toda $p_i \in P_n,\ \ S \to p_i S \bar{p}_i$
  - $S \to SS$
  - $S \to \varepsilon$

Lenguajes de Dyck
Propiedades

- $D_n(A)$ se llama el lenguaje (generalizado) de Dyck de orden n.
- $D_n(\varnothing)$ se llama el lenguaje (simple) de Dyck de orden n, denotado simplemente D_n.
- Los lenguajes de Dyck tienen las siguientes propiedades:
  - $A^\star \subseteq D_n(A)$
  - Si $x, y \in D_n(A)$ entonces $xy \in D_n(A)$
  - Si $x \in D_n(A)$ entonces $p_i x \bar{p}_i \in D_n(A)$
  - Si $x \in D_n(A)$ y $x \notin A^\star$ entonces existen $u \in A^\star$, $v, w \in D_n(A)$, $i \leq n$ tales que $x = up_i v\bar{p}_i w$

La gramática J

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática J como sigue:
- $J = \langle V, T \cup B, S, \mathcal{P}'' \rangle$ donde $\mathcal{P}''$ consta de las siguientes producciones:
  - Todas las producciones $V \to a$ de $\mathcal{P}'$
  - Si $A_i \to p_i B_i \bar{p}_i C_i$ está en $\mathcal{P}'$ entonces $\mathcal{P}''$ tiene a:
    - $A_i \to p_i B_i$
    - $V \to a\bar{p}_i C_i$, para cada producción $V \to a$ que esté en $\mathcal{P}'$
- Es claro que J es equivalente a una gramática lineal derecha.
- Por lo tanto $L(J)$ es un lenguaje regular.

Gramática J

Ejemplo

- G_{sep} está dada por

$$S \to (X)Y \qquad S \to [Y]X \qquad Y \to \{Z\}Z \qquad X \to a \qquad Z \to a$$

- J se construye como sigue:
  - $S \to (X)Y$ genera a : $S \to (X \qquad X \to a)Y \qquad Z \to a)Y$
  - $S \to [Y]X$ genera a : $S \to [Y \qquad X \to a]X \qquad Z \to a]X$
  - $Y \to \{Z\}Z$ genera a : $Y \to \{Z \qquad X \to a\}Z \qquad Z \to a\}Z$
- Así J es:

$$S \to (X \qquad X \to a)Y \qquad Z \to a)Y$$

$$S \to [Y \qquad X \to a]X \qquad Z \to a]X$$

$$Y \to \{Z \qquad X \to a\}Z \qquad Z \to a\}Z$$

$$X \to a \qquad Z \to a$$

Propiedades de G_{sep}

P1 $erase_B[L(G_{sep})] = L(G)$

P2 $L(G_{sep}) \subseteq D_n(T)$

P3 $L(G_{sep}) \subseteq L(J)$

P4 $L(J) \cap D_n(T) \subseteq L(G_{sep})$

P5 Si G está en FNC entonces existe R regular tal que

$$L(G_{sep}) = R \cap D_n(T)$$

Teorema de Chomsky-Schuetzenberger

Sea $L \subseteq T^{\star}$. Las siguientes condiciones son equivalentes:

I) L es un lenguaje libre de contexto

II) Existen un lenguaje regular R, un homomorfismo h y un número natural $n \in \mathbb{N}$ tales que

$$L = h[R \cap D_n(T)]$$

Es decir, todo lenguaje libre de contexto es imagen homomórfica de la intersección de un lenguaje regular y un lenguaje de Dyck.

Teorema de Chomsky-Schuetzenberger

Versión usual

- El teorema de Chomsky-Schuetzenberger suele enunciarse y demostrarse utilizando lenguajes de Dyck simples como sigue:

  Si L es un lenguaje libre de contexto entonces existen un lenguaje regular R, un homomorfismo h y un número natural $n > 0$ tales que:

  $$L = h[R \cap D_n]$$

- Este teorema resulta equivalente a la versión aquí presentada.

Lenguajes recursivamente enumerables o tipo 0

Jerarquía de Chomsky

Son aquellos lenguajes generados por una gramática cuyas producciones no tienen restricciones.
Tales gramáticas pueden incluir reglas de la forma

$$\alpha \to \varepsilon, \text{ donde } \alpha \in (V \cup T)^*, \ \alpha \neq \varepsilon$$

De manera que la gramática es capaz de borrar cadenas. Tales gramáticas se conocen como contraibles.
Ejemplo:

$$aS \to bSb, \ aSb \to \varepsilon, \ SbS \to bcS$$

Los lenguajes tipo 0 son equivalentes a las máquinas de Turing. Es decir, L es generado por una gramática tipo 0 si y sólo si L es reconocido por una máquina de Turing.

Lenguajes recursivamente enumerables o tipo 0

Jerarquía de Chomsky

- La siguiente es una gramática de tipo 0:

$$S \to AT \qquad A \to 0AU \qquad A \to 1AI \qquad U0 \to 0U$$

$$U1 \to 1U \qquad I0 \to 0I \qquad I1 \to 1I \qquad UT \to 0T$$

$$IT \to 1T \qquad A \to \varepsilon \qquad T \to \varepsilon$$

- La gramática es contraible debido a que se pueden borrar A y T.
- $L(G) = \{ww \mid w \in \{0,1\}^\star\}$

Lenguajes dependientes del contexto o tipo 1

Jerarquía de Chomsky

Son aquellos generados por gramáticas con todas sus producciones son de la forma

$$\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$$

con $\alpha_1, \alpha_2 \in (V \cup T)^\star,\ A \in V,\ \beta \neq \varepsilon$.
Con la posible excepción de la regla $S \to \varepsilon$, en cuyo caso se prohibe la presencia de S a la derecha de las producciones.
Por ejemplo la siguiente gramática dependiente del contexto genera al lenguaje $L = \{a^i b^i c^i \mid i \geq 0\}$

$$S \to A \quad A \to aABC \mid abC \quad CB \to BC$$

$$bB \to bb \quad bC \to bc \quad cC \to cc$$

Lenguajes dependientes del contexto o tipo 1

Jerarquía de Chomsky

- Estos lenguajes también se llaman sensibles al contexto y tienen cierta relevancia para la teoría de lenguajes de programación.
- Un lenguaje L es sensible al contexto si y sólo si L es reconocido por un autómata linealmente acotado.
- Un autómata linealmente acotado es una clase especial de Máquina de Turing que sólo permite el uso de un pedazo de la cinta cuyo tamaño depende de la longitud de la cadena de entrada.

Lenguajes libres del contexto o tipo 2

Jerarquía de Chomsky

- Son aquellos generados por gramáticas con todas sus producciones de la forma

$$A \to \alpha$$

  con $A \in V,\ \alpha \in (V \cup T)^\star$.

- Esta definición incluye a la regla $S \to \varepsilon$.
- La mayoría de las gramáticas para lenguajes de programación caen en esta categoría.
- Un lenguaje L es libre de contexto si y sólo si L es reconocido por un autómata de pila.

Lenguajes regulares o tipo 3

Jerarquía de Chomsky

Son aquellos generados por una gramática de una de las siguientes formas:

- Lineal por la derecha: todas las producciones de la forma

$$A \to aB \quad A \to a \quad A \to \varepsilon$$

- Lineas por la izquierda: todas las producciones de la forma

$$A \to Ba \quad A \to a \quad A \to \varepsilon$$

- No se permite mezclar ambos tipos de producciones.
- Tales gramáticas se conocen también como gramáticas regulares.
-
- Un lenguaje L es tipo 3 si y sólo si L es reconocido por un autómata finito.

Jerarquía de Chomsky

Observaciones

- Decimos que un lenguaje es de tipo *i* si y sólo si *i* es el índice mas grande tal que existe una gramática de tipo *i* que genera a *L*
- La jerarquía de gramáticas genera una jerarquía en los lenguajes generados:

$$\mathcal{L}_3 \subsetneq \mathcal{L}_2 \subsetneq \mathcal{L}_1 \subsetneq \mathcal{L}_0 \subsetneq \Sigma^\star$$

- $\{0^n 1^n \mid n \in \mathbb{N}\} \in \mathcal{L}_2 \setminus \mathcal{L}_3$
- $\{0^n 1^n 0^n \mid n \in \mathbb{N}\} \in \mathcal{L}_1 \setminus \mathcal{L}_2$
- Un ejemplo en $\mathcal{L}_0 \setminus \mathcal{L}_1$ puede obtenerse mediante un truco de diagonalización o bien mediante complicados resultados de la teoría de la complejidad
- El lenguaje diagonal $L_{\mathcal{D}}$ cumple $L_{\mathcal{D}} \in \Sigma^\star \setminus \mathcal{L}_0$