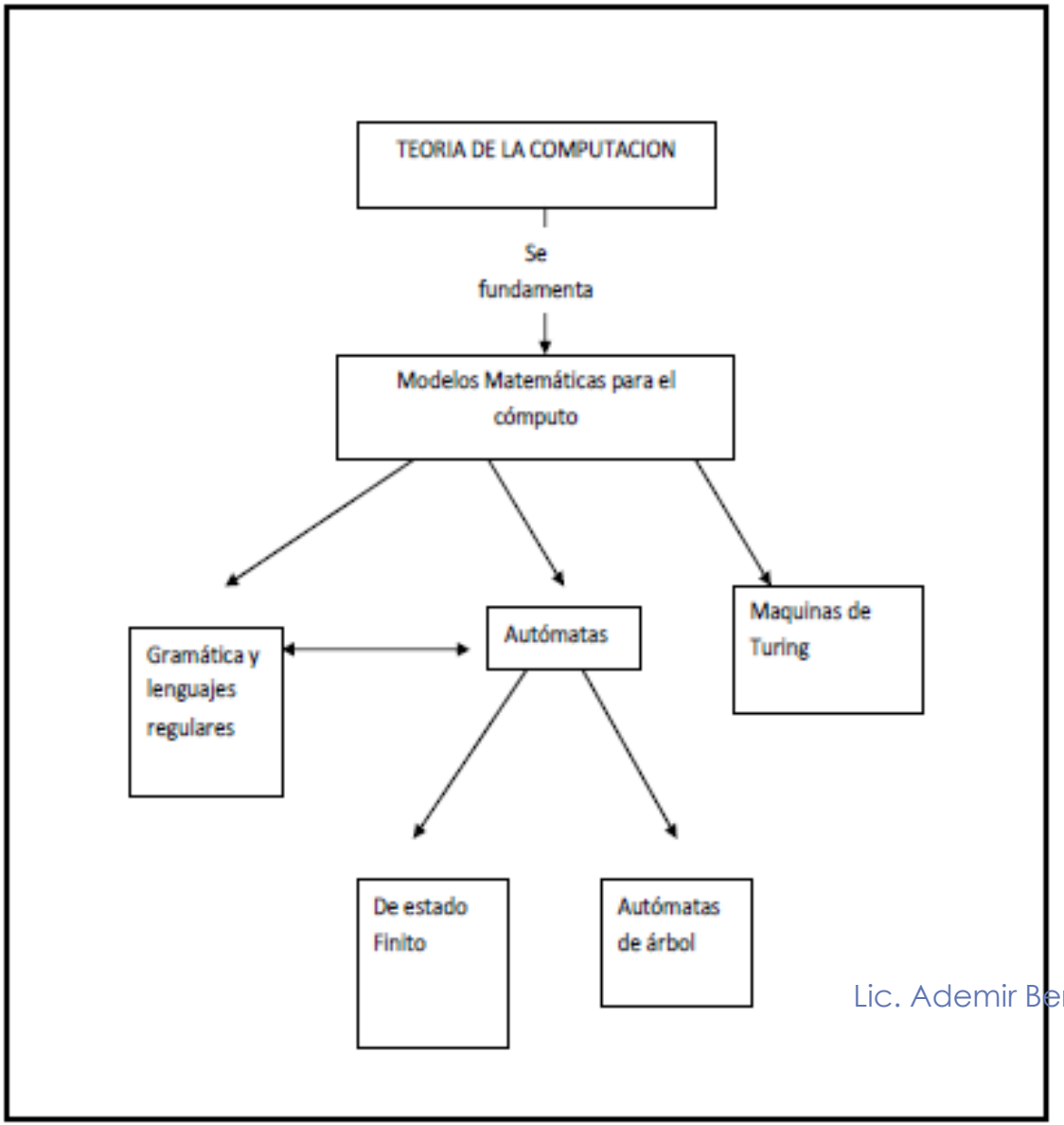
Definición formal de gramática

Lic. Ademir Bermúdez

ESTRUCTURA CONCEPTUAL



Gramáticas Generativas

1.3.3. Gramáticas Generativas

Desde un punto de vista matemático una gramática se define de la siguiente forma:

Definición 16 *Una gramática generativa es un cuadrupla* (V,T,P,S) *en la que*

- V *es un alfabeto, llamado de variables o símbolos no terminales. Sus elementos se suelen representar con letras mayúsculas.*
- T *es un alfabeto, llamado de símbolos terminales. Sus elementos se suelen representar con letras minúsculas.*
- P *es un conjunto de pares* (α,β)*, llamados reglas de producción, donde* $\alpha,\beta \in (V \cup T)^*$ *y* α *contiene, al menos un símbolo de* V*.*

  El par (α,β) *se suele representar como* $\alpha \rightarrow \beta$*.*
- S *es un elemento de* V*, llamado símbolo de partida.*

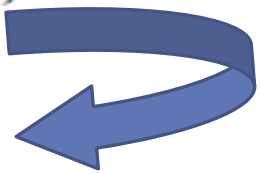
La razón de notar los elementos del alfabeto V con letras mayúsculas es para no confundirlos con los símbolos terminales. Las cadenas del alfabeto $(V \cup T)$ se notan con letras griegas para no confundirlas con las cadenas del alfabeto T, que seguirán notándose como de costumbre: $u, v, x \ldots$.

Ejemplo

Sea la gramática (V;T;P;S) dada por los siguientes elementos

- $V = \{E\}$
- $T = \{+, *, (,), a, b, c\}$

- $V = \{E\}$
- $T = \{+, *, (,), a, b, c\}$



- *P está compuesto por las siguientes reglas de producción*

$$E \to E + E, \quad E \to E * E, \quad E \to (E),$$
$$E \to a, \quad E \to b, \quad E \to c$$

- $S = E$

Una gramática se usa para generar las distintas palabras de un determinado lenguaje. Esta generación se hace mediante una aplicación sucesiva de reglas de producción comenzando por

el símbolo de partida S. Las siguientes definiciones expresan esta idea de forma más rigurosa.

Definición 17 *Dada una gramática* $G = (V, T, P, S)$ *y dos palabras* $\alpha, \beta \in (V \cup T)^*$, *decimos que* β *es derivable a partir de* α *en un paso* $(\alpha \Longrightarrow \beta)$ *si y solo si existe una producción* $\gamma \rightarrow \varphi$ *tal que*

- γ *es una subcadena de* α.
- β *se puede obtener a partir de* α, *cambiando la subcadena* γ *for* φ.

Haciendo referencia a la gramática del ejemplo anterior, tenemos las siguientes derivaciones

$$E \Longrightarrow E+E \Longrightarrow (E)+E \Longrightarrow (E)+(E) \Longrightarrow (E*E)+(E) \Longrightarrow (E*E)+(E*E)$$

Definición 18 *Dada una gramática* $G = (V, T, P, S)$ *y dos palabras* $\alpha, \beta \in (V \cup T)^*$, *decimos que* β *es derivable de* α $(\alpha \overset{*}{\Longrightarrow} \beta)$, *si y solo si existe una sucesión de palabras* $\gamma_1, \ldots, \gamma_n$ $(n \geq 1)$ *tales que*

$$\alpha = \gamma_1 \Longrightarrow \gamma_2 \Longrightarrow \ldots \Longrightarrow \gamma_n = \beta$$

Ejemplo:

En el caso anterior podemos decir que (E E)+(E *E) es derivable a partir de E:*

$$E \overset{*}{\Longrightarrow} (E * E) + (E * E)$$

Definición 19 *Se llama lenguaje generado por una gramática* $G = (V, T, P, S)$ *al conjunto de cadenas formadas por símbolos terminales y que son derivables a partir del símbolo de partida. Es decir,*

$$L(G) = \{u \in T^* \mid S \overset{*}{\Longrightarrow} u\}$$

Ejemplo

*En el caso de la gramática de los ejemplos anteriores (E *E)+(E*E) no pertenece al lenguaje generado por G, ya que hay símbolos que no son terminales. Sin embargo, (a+c)*(a+b) si pertenece a L(G), ya que se puede comprobar que es derivable a partir de E (símbolo de partida) y solo tiene símbolos terminales.*

Si en una gramática comenzamos a hacer derivaciones a partir del símbolo original *S*, dicha derivación acabará cuando solo queden símbolos terminales, en cuyo caso la palabra resultante pertenece a *L*(*G*), o cuando queden variables pero no se pueda aplicar ninguna regla de producción, en cuyo caso dicha derivación no puede llevar a ninguna palabra de *L*(*G*). En general, cuando estemos haciendo una derivación puede haber más de una regla de producción aplicable en cada momento.

- Cuando no sea importante distinguir si la derivación de una palabra en una gramática se haya realizado en uno o varios pasos, entonces eliminaremos la del símbolo de derivación.
- Así, escribiremos

$$\alpha \Longrightarrow \beta, \text{ en lugar de } \alpha \overset{*}{\Longrightarrow} \beta.$$

Ejemplo

Sea G = (V;T;P;S) una gramática, donde

V = {S;A;B} ; T = {a;b}, las reglas de producción son

$$S \to aB, \quad S \to bA, \quad A \to a, \quad A \to aS,$$
$$A \to bAA, \quad B \to b, \quad B \to bS, \quad B \to aBB$$

y el símbolo de partida es S.

Esta gramática genera el lenguaje

$$L(G) = \{u \mid u \in \{a,b\}^+ \text{ y } N_a(u) = N_b(u)\}$$

donde $N_a(u)$ y $N_b(u)$ son el número de apariciones de símbolos a y b, en u, respectivamente.

Esto es fácil de ver interpretando que,

. *S genera (o produce) palabras con igual número de a que de b.*

. *A genera palabras con una a de más.*

. *B produce palabras con una b de más.*

. *S genera palabras con igual número de a que de b.*

Hay que demostrar que todas las palabras del lenguaje tienen el mismo número de a que de b, hay que probar que todas las palabras generadas cumplen esta condición y que todas las palabras que cumplen esta condición son generadas.

Para lo primero basta con considerar el siguiente razonamiento. Supongamos $N_{a+A}(\alpha)$ *y* $N_{b+B}(\alpha)$ *que son el número de a + el número de A en* α *y el número de b + el número de B en* α*, respectivamente. Entonces,*

- *Cuando se empieza a generar una palabra, comenzamos con S y tenemos la igualdad* $N_{a+A}(S) = N_{b+B}(S) = 0$.
- *También se puede comprobar que si* α' *se obtiene de* α *en un paso de derivación y* $N_{a+A}(\alpha) = N_{b+B}(\alpha)$, *entonces* $N_{a+A}(\alpha') = N_{b+B}(\alpha')$
- *Si la condición de igualdad de* N_{a+A} *y* N_{b+B} *se verifica al principio y, si se verifica antes de un paso, entonces se verifica después de aplicarlo, necesariamente se verifica al final de la derivación. Si hemos derivado u, entonces* $N_{a+A}(u) = N_{b+B}(u)$.
- *Como u no tiene variables, entonces* $N_{a+A}(u) = N_a(u)$ *y* $N_{b+B}(u) = N_b(u)$, *por lo tanto,* $N_a(u) = N_b(u)$, *es decir si u es generada contiene el mismo número de a que de b.*

Para demostrar que todas las palabras del lenguaje son generadas por la gramática, damos el siguiente algoritmo que en n pasos es capaz de generar una palabra de n símbolos. El algoritmo genera las palabras por la izquierda obteniendo, en cada paso, un nuevo símbolo de la palabra a generar.

- *Para generar una a*
  - *Si a último símbolo de la palabra, aplicar* $A \rightarrow a$
  - *Si no es el último símbolo*
    - *Si la primera variable es S aplicar* $S \rightarrow aB$
    - *Si la primera variable es B aplicar* $B \rightarrow aBB$
    - *Si la primera variable es A*
      - *Si haya más variables aplicar* $A \rightarrow a$
      - *Si no hay más, aplicar* $A \rightarrow aS$

- *Para generar una b*
  - *Si b último símbolo de la palabra, aplicar* $B \rightarrow b$
  - *Si no es el último símbolo*
    - *Si la primera variable es S aplicar* $S \rightarrow bA$
    - *Si la primera variable es A aplicar* $A \rightarrow bAA$
    - *Si la primera variable es B*
      - *Si haya más variables aplicar* $B \rightarrow b$
      - *Si no hay más, aplicar* $B \rightarrow bS$

Las condiciones que garantizan que todas las palabras son generadas mediante este algoritmo son las siguientes:

- *Las palabras generadas tienen primero símbolos terminales y después variables.*
- *Se genera un símbolo de la palabra en cada paso de derivación Las variables que aparecen en la palabra pueden ser:*
- *Una cadena de A (si hemos generado más b que a)*
- *Una cadena de B (si hemos generado más a que b)*
- *Una S si hemos generado las mismas a que b*

Antes de generar el último símbolo tendremos como variables:

Una A si tenemos que generar a

Una B si tenemos que generar b

Entonces aplicamos la primera opción para generar los símbolos y la palabra queda generada.

Ejemplo 13 *Sea* $G = (\{S,X,Y\},\{a,b,c\},P,S)$ *donde* P *tiene las reglas,*

$$S \to abc \qquad S \to aXbc \qquad Xb \to bX \qquad Xc \to Ybcc$$
$$bY \to Yb \qquad aY \to aaX \qquad aY \to aa$$

Esta gramática genera el lenguaje: $\{a^n b^n c^n \mid n = 1,2,\ldots\}$.

Para ver esto observemos que S en un paso, puede generar abc ó aXbc. Así que $abc \in L(G)$. *A partir de aXbc solo se puede relizar la siguiente sucesión de derivaciones,*

$$aXbc \Longrightarrow abXc \Longrightarrow abYbcc \Longrightarrow aYbbcc$$

En este momento podemos aplicar dos reglas:

- $aY \to aa$, *en cuyo caso producimos* $aabbcc = a^2b^2c^2 \in L(G)$
- $aY \to aaX$, *en cuyo caso producimos* $aaXbbcc$

A partir de aaXbbcc, se puede comprobar que necesariamente llegamos a $a^2Yb^3c^3$. *Aquí podemos aplicar otra vez las dos reglas de antes, produciendo* $a^3b^3c^3$ *ó* $a^3Xb^3c^3$. *Así, mediante un proceso de inducción, se puede llegar a demostrar que las únicas palabras de símbolos terminales que se pueden llegar a demostrar son* $a^n b^n c^n, n \geq 1$.

Jerarquía de Chomsky

De acuerdo con lo que hemos visto, toda gramática genera un único lenguaje, pero distintas gramáticas pueden generar el mismo lenguaje. Podríamos pensar en clásica las gramáticas por el lenguaje que generan, por este motivo hacemos la siguiente definición.

Definición 20 *Dos gramáticas se dicen débilmente equivalentes si generan el mismo lenguaje*.

Sin embargo, al hacer esta clasificación nos encontramos con que el problema de saber si dos gramáticas generan el mismo lenguaje es indecidible. No existe ningún algoritmo que acepte como entrada dos gramáticas y nos diga (la salida del algoritmo) si generan o no el mismo lenguaje.

De esta forma, tenemos que pensar en clasificaciones basadas en la forma de la gramática, más que en la naturaleza del lenguaje que generan. La siguiente clasificación se conoce como jerarquía de Chomsky y sigue esta dirección.

Definición 21 *Una gramática se dice que es de*

- **Tipo 0** *Cualquier gramática. Sin restricciones.*
- **Tipo 1** *Si todas las producciones tienen la forma*

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$$

donde $\alpha_1, \alpha_2, \beta \in (V \cup T)^*, A \in V$, *y* $\beta \neq \varepsilon$, *excepto posiblemente la regla* $S \rightarrow \varepsilon$, *en cuyo caso S no aparece a la derecha de las reglas.*

- **Tipo 2** *Si cualquier producción tiene la forma*

$$A \rightarrow \alpha$$

dode $A \in V, \alpha \in (V \cup T)^*$.

- **Tipo 3** *Si toda regla tiene la forma*

$$A \rightarrow uB \text{ ó } A \rightarrow u$$

donde $u \in T^*$ *y* $A, B \in V$

Definición 22 *Un lenguaje se dice que es de tipo* $i(i = 0, 1, 2, 3)$ *si y solo si es generado por una gramática de tipo i. La clase o familia de lenguajes de tipo i se denota por* $\mathcal{L}_i$.

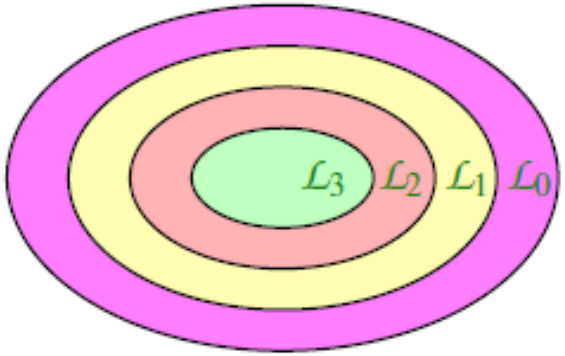


Figura 1.1: Estructura de las clases de lenguajes

Se puede demostar que $\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0$.

Las gramáticas de tipo 0 se llaman también gramáticas con estructura de frase, por su origen linguístico. Los lenguajes aceptados por las gramáticas de tipo 0 son los *recursivamente enumerables*.

Las gramáticas de tipo 1 se denominan dependientes del contexto. Los lenguajes aceptados por estas gramáticas son los lenguajes *dependientes del contexto*.

Las gramáticas de tipo 2, así como los lenguajes generados, se llaman independientes del contexto.

Las gramáticas de tipo 3 se denominan regulares o de estado finito. Los lenguajes aceptados por estas gramáticas se denominan *conjuntos regulares*.

Los homorfismos son útiles para demostrar teoremas.

Teorema 1 *Para toda gramática $G = (V,T,P,S)$ podemos dar otra gramática $G' = (V',T,P',S)$ que genere el mismo lenguaje y tal que en la parte izquierda de las reglas solo aparezcan variables.*

Demostración

Si la gramática es de tipo 3 ó 2 no hay nada que demostrar.

Si la gramática es de tipo 0 ó 1, entonces para cada $a_i \in T$ introducimos una variable $A_i \notin V$. Entonces hacemos $V' = V \cup \{A_1, \ldots, A_k\}$, donde k es el número de símbolos terminales.

Ahora P' estará formado por las reglas de P donde, en todas ellas, se cambia a_i por A_i. Aparte de ello añadimos una regla $A_i \to a_i$ para cada $a_i \in$T.

Podemos ver que $L(G) \subseteq L(G')$. En efecto, si derivamos $u = a_{i_1} \ldots a_{i_n} \in L(G)$, entonces usando las reglas correspondientes, podemos derivar $A_{i_1} \ldots A_{i_n}$ en G'. Como en G' tenemos las reglas $A_i \to a_i$, entonces podemos derivar $u \in L(G')$.

Para demostrar la inclusión inversa: $L(G') \subseteq L(G)$, definimos un homomorfismo h de $(V' \cup T)^*$ en $(V \cup T)^*$ de la siguiente forma,

1. $h(A_i) = a_i, \quad i = 1, \ldots, k$
2. $h(x) = x, \quad \forall x \in V \cup T$

Ahora se puede demostrar que este homorfismo transforma las reglas: si $\alpha \to \beta \in P'$, entonces $h(\alpha) \to h(\beta)$ es una producción de P ó $h(\alpha) = h(\beta)$.

Partiendo de esto se puede demostrar que si $\alpha \Longrightarrow \beta$ entonces $h(\alpha) \Longrightarrow h(\beta)$. En particular, como consecuencia, si $S \Longrightarrow u, \quad u \in T^*$, entonces $h(S) = S \Longrightarrow h(u) = u$. Es decir, si $u \in L(G')$ entonces $u \in L(G)$. Con lo que definitivamente $L(G) = L(G')$. ■

Ejercicios

1. *Demostrar que la gramática*

$$G = (\{S\}, \{a, b\}, \{S \to \varepsilon, S \to aSb\}, S)$$

genera el lenguaje

$$L = \{a^i b^i \mid i = 0, 1, 2\}$$

Solución:

Si seguimos este procedimiento, nos encontramos que podemos ir generando todas las palabras de la forma $a^i b^i$, y siempre nos queda la palabra $a^i S b^i$ para seguir generando las palabras de mayor longitud.

Por otra parte, estas son las únicas palabras que se pueden generar.

2. *Encontrar el lenguaje generado por la gramática* $G = (\{A,B,S\},\{a,b\},P,S)$ *donde* P *contiene las siguientes producciones*

$$S \to aAB \qquad bB \to a \qquad Ab \to SBb$$
$$Aa \to SaB \qquad B \to SA \qquad B \to ab$$

Solución:

El resultado es el Lenguaje vacío*: nunca se puede llegar a generar una palabra con símbolos terminales. Siempre que se sustituye S aparece A, y siempre que se sustituye A aparece S.*

3. *Encontrar una gramática libre del contexto para generar cada uno de los siguientes lenguajes*

a) $L = \{a^i b^j \mid i, j \in \mathbf{N}, i \leq j\}$

Solución:

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

$$S \rightarrow Sb$$

b) $L = \{a^i b^j a^j b^i \mid i, j \in \boldsymbol{N}\}$

Solución:

$$S \rightarrow aSb$$

$$S \rightarrow B, \quad B \rightarrow bBa, \quad B \rightarrow \varepsilon$$

c) $L = \{a^i b^i a^j b^j \mid i, j \in \mathbf{N}\}$

Solución:

Podemos generar $\{a^i b^i \mid i \in \mathbf{N}\}$ *con:*

$$S_1 \rightarrow aS_1b, \quad S_1 \rightarrow \varepsilon$$

El lenguaje L se puede generar añadiendo:

$$S \rightarrow S_1 S_1$$

siendo S el símbolo inicial.

d) $L = \{a^i b^i \mid i \in \boldsymbol{N}\} \cup \{b^i a^i \mid i \in \boldsymbol{N}\}$

Solución:

Podemos generar $\{a^i b^i \mid i \in \boldsymbol{N}\}$ *con:*

$$S_1 \to aS_1b, \quad S_1 \to \varepsilon$$

y $\{b^i a^i \mid i \in \boldsymbol{N}\}$ *con*

$$S_2 \to bS_2a, \quad S_2 \to \varepsilon$$

El lenguaje L se puede generar añadiendo:

$$S \to S_1, \quad S \to S_2$$

siendo S el símbolo inicial.

e) $L = \{uu^{-1} \mid u \in \{a,b\}^*\}$

Solución:

$$S \to aSa, \quad S \to bSb, \quad S \to \varepsilon$$

f) $L = \{a^i b^j c^{i+j} \mid i, j \in \boldsymbol{N}\}$

Solución:

$$S \to aSc, \quad S \to B,$$

$$B \to bBc, \quad B \to \varepsilon$$