

# Gramática libre de contexto

En lingüística e informática, una **gramática libre de contexto** (o **de contexto libre**) es una gramática formal en la que cada regla de producción es de la forma:

$$V \rightarrow w$$

Donde  $V$  es un símbolo no terminal y  $w$  es una cadena de terminales y/o no terminales. El término *libre de contexto* se refiere al hecho de que el no terminal  $V$  puede siempre ser sustituido por  $w$  sin tener en cuenta el contexto en el que ocurra. Un lenguaje formal es libre de contexto si hay una gramática libre de contexto que lo genera.

Las gramáticas libres de contexto permiten describir la mayoría de los lenguajes de programación, de hecho, la sintaxis de la mayoría de lenguajes de programación está definida mediante gramáticas libres de contexto. Por otro lado, estas gramáticas son suficientemente simples como para permitir el diseño de eficientes algoritmos de análisis sintáctico que, para una cadena de caracteres dada determinen cómo puede ser generada desde la gramática. Los analizadores LL y LR tratan restringidos subconjuntos de gramáticas libres de contexto.

La notación más frecuentemente utilizada para expresar gramáticas libres de contexto es la forma Backus-Naur.

## Definición formal

Así como cualquier gramática formal, una gramática libre de contexto puede ser definida mediante la 4-tupla:

$$G = (V_t, V_n, P, S)$$

donde

- $V_t$  es un conjunto finito de terminales
- $V_n$  es un conjunto finito de no terminales
- $P$  es un conjunto finito de producciones
- $S \in V_n$  el denominado Símbolo Inicial
- los elementos de  $P$  son de la forma

$$V_n \longrightarrow (V_t \cup V_n)^*$$

## Ejemplos

### Ejemplo 1

Una simple gramática libre de contexto es

$$S \rightarrow aSb \mid \epsilon$$

donde  $\mid$  es un *o lógico* y es usado para separar múltiples opciones para el mismo no terminal,  $\epsilon$  indica una cadena vacía. Esta gramática genera el lenguaje no regular  $\{a^n b^n : n \geq 0\}$ .

### Ejemplo 2

Aquí hay una gramática libre de contexto para expresiones enteras algebraicas sintácticamente correctas sobre las variables  $x$ ,  $y$  y  $z$ :

$$S \rightarrow x \mid y \mid z \mid S + S \mid S - S \mid S * S \mid S / S \mid (S)$$

Generaría, por ejemplo, la cadena  $(x + y) * x - z * y / (x + x)$

### Ejemplo 3

Una gramática libre de contexto para un lenguaje consistente en todas las cadenas que se pueden formar con las letras  $a$  y  $b$ , habiendo un número diferente de una que de otra, sería:

$$S \rightarrow U \mid V$$

$$U \rightarrow TaU \mid TaT$$

$$V \rightarrow TbV \mid TbT$$

$$T \rightarrow aTbT \mid bTaT \mid \epsilon$$

$T$  genera todas las cadenas con la misma cantidad de letras  $a$  que  $b$ ,  $U$  genera todas las cadenas con más letras  $a$ , y  $V$  todas las cadenas con más letras  $b$ .

### Ejemplo 4

Otro ejemplo para un lenguaje es  $\{a^n b^m c^{m+n} : n \geq 0, m \geq 0\}$ . No es un lenguaje regular, pero puede ser generado por la siguiente gramática libre de contexto.

$$S \rightarrow aSc \mid B$$

$$B \rightarrow bBc \mid \epsilon$$

### Otros ejemplos

Las gramáticas libres de contexto no están limitadas a lenguajes matemáticos formales.

La gramática de Lojban, un lenguaje artificial hablado con gran capacidad expresiva, es también libre de contexto y no ambiguo.

El lingüista indio Pánini (siglo IV a. C.) describió el sánscrito usando una gramática libre de contexto en su texto *Astadhiai*.

Recientemente se ha sugerido que una clase de poesía tamil llamada Venpa utiliza principalmente una gramática libre de contexto.

## Derivaciones y árboles sintácticos

Existen básicamente dos formas de describir cómo en una cierta gramática una cadena puede ser derivada desde el símbolo inicial. La forma más simple es listar las cadenas de símbolos consecutivas, comenzando por el símbolo inicial y finalizando con la cadena y las reglas que han sido aplicadas. Si introducimos estrategias como reemplazar siempre el no terminal de más a la izquierda primero, entonces la lista de reglas aplicadas es suficiente. A esto se le llama **derivación por la izquierda**. Por ejemplo, si tomamos la siguiente gramática:

$$(1) S \rightarrow S + S$$

$$(2) S \rightarrow 1$$

y la cadena " $1 + 1 + 1$ ", su derivación a la izquierda está en la lista  $[(1) (1) (2) (2) (2)]$ . Análogamente, la **derivación por la derecha** se define como la lista que obtenemos si siempre reemplazamos primero el no terminal de más a la derecha. En ese caso, la lista de reglas aplicadas para la derivación de la cadena con la gramática anterior sería la  $[(1) (2) (1) (2) (2)]$ .

La distinción entre derivación por la izquierda y por la derecha es importante porque en la mayoría de analizadores, la transformación de la entrada es definida dando una parte de código para cada producción que es ejecutada cuando la regla es aplicada. De modo que es importante saber qué derivación aplica el analizador, porque determina el orden en el que el código será ejecutado.

Una derivación también puede ser expresada mediante una estructura jerárquica sobre la cadena que está siendo derivada. Por ejemplo, la estructura de la derivación a la izquierda de la cadena " $1 + 1 + 1$ " con la gramática anterior

sería:

$$S \rightarrow S + S \quad (1)$$

$$S \rightarrow S + S + S \quad (1)$$

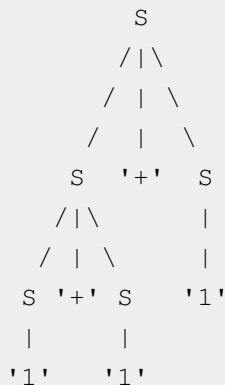
$$S \rightarrow 1 + S + S \quad (2)$$

$$S \rightarrow 1 + 1 + S \quad (2)$$

$$S \rightarrow 1 + 1 + 1 \quad (2)$$

$$\{\{1\}_S + \{1\}_S\}_S + \{1\}_S$$

donde  $\{\dots\}_S$  indica la subcadena reconocida como perteneciente a  $S$ . Esta jerarquía también se puede representar mediante un árbol sintáctico:



Este árbol es llamado árbol de sintaxis concreta de la cadena (ver también árbol de sintaxis abstracta). En este caso, las derivaciones por la izquierda y por la derecha presentadas definen la sintaxis del árbol; sin embargo, hay otra derivación (por la izquierda) de la misma cadena.

La derivación por la derecha:

$$S \rightarrow S + S \quad (1)$$

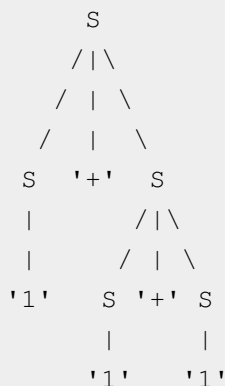
$$S \rightarrow 1 + S \quad (2)$$

$$S \rightarrow 1 + S + S \quad (1)$$

$$S \rightarrow 1 + 1 + S \quad (2)$$

$$S \rightarrow 1 + 1 + 1 \quad (2)$$

define el siguiente árbol sintáctico:



Si para una cadena del lenguaje de una gramática hay más de un árbol posible, entonces se dice que la gramática es ambigua. Normalmente estas gramáticas son más difíciles de analizar porque el analizador no puede decidir siempre que producción aplicar.

## Formas normales

Una gramática que no genera la cadena vacía puede ser transformada en una equivalente (que genera el mismo lenguaje) en forma normal de Chomsky o en forma normal de Greibach.

La simplicidad de las reglas en forma normal de Chomsky tiene implicaciones teóricas y prácticas. Por ejemplo, dada una gramática libre de contexto, se puede usar su forma normal para construir un algoritmo de coste polinomial que decida si una cadena forma parte del lenguaje definido por la gramática o no (algoritmo CYK).

## Problemas indecidibles

Algunas de las propiedades de las gramáticas, y en general, de los lenguajes libres del contexto son de naturaleza decidible, existiendo por lo tanto algoritmos de decisión para resolverlos. Sin embargo, al contrario que en el caso de los lenguajes regulares, existen problemas interesantes para los cuales se ha mostrado su naturaleza indecidible, y por lo tanto, se carece del correspondiente algoritmo.

Uno de los más sencillos es el de decidir si una gramática libre del contexto dada acepta el lenguaje de todas las posibles cadenas de símbolos. Este lenguaje viene a ser una reducción del problema de parada de una máquina de Turing con una entrada particular, y por lo tanto, un problema indecidible. La reducción usa el concepto de *historia computacional*, es decir, una cadena que describa el proceso de computación global de una máquina de Turing, esta cadena podría describirse mediante una gramática libre del contexto. Podemos construir, por tanto, una gramática libre del contexto que genere todas las cadenas no aceptadas por la máquina de Turing indicada.

Una consecuencia importante es que también es indecidible la comparación entre dos gramáticas libres del contexto para comprobar si el lenguaje generado coincide.

Por el contrario, el problema sencillo de determinar si dada una cadena es aceptada por una determinada gramática libre del contexto, sí que es decidible, y por lo tanto podrá escribirse el correspondiente algoritmo para decidirlo.

## Propiedades de los lenguajes libres de contexto

- Una de las definiciones alternativas y equivalentes de lenguaje libre de contexto emplea autómatas no deterministas: un lenguaje es libre de contexto si puede ser aceptado por ese autómata.
- Un lenguaje puede ser también modelado como un conjunto de todas las secuencias de terminales aceptadas por la gramática. Este modelo ayuda a entender las operaciones de conjuntos sobre lenguajes.
- La unión y concatenación de dos lenguajes libres de contexto es también libre de contexto. La intersección no tiene por qué serlo.
- El inverso de un lenguaje libre de contexto es también libre de contexto, pero el complemento no tiene por qué serlo.
- Los lenguajes regulares son libres de contexto porque pueden ser descritos mediante una gramática de libre contexto.
- La intersección de un lenguaje libre de contexto y un lenguaje regular es siempre libre de contexto.
- Existen Gramáticas sensibles al contexto que no son libres de contexto.
- Para demostrar que un lenguaje dado no es libre de contexto, se puede emplear el Lema del bombeo para lenguajes libres de contexto.
- El problema de determinar si una gramática sensible al contexto describe un lenguaje libre del contexto es indecidible.

# Fuentes y contribuyentes del artículo

**Gramática libre de contexto** *Fuente:* <http://es.wikipedia.org/w/index.php?oldid=57537443> *Contribuyentes:* Almorca, Andreasmperu, Beagle, Deleatur, Dortegaparrilla, Falopez, Fer31416, GermanX, Jacobodeharo, Kronoss, Magister Mathematicae, Manwë, Marciogutierrez, Rbonvall, Rosarino, Vaskop, 57 ediciones anónimas

## Licencia

---

Creative Commons Attribution-Share Alike 3.0 Unported  
[//creativecommons.org/licenses/by-sa/3.0/](https://creativecommons.org/licenses/by-sa/3.0/)

---