

Manual Técnico

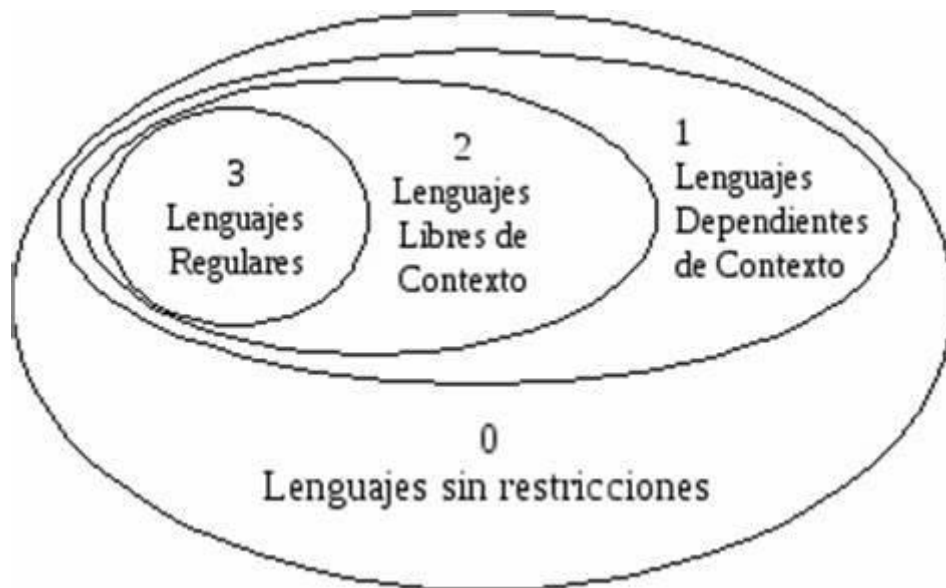
Gramáticas tipo 2

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.

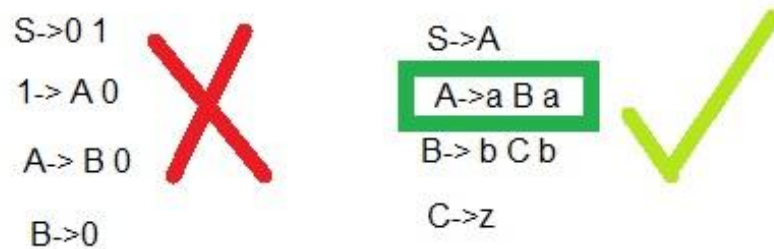
Las gramáticas libres de contexto permiten describir la mayoría de los lenguajes de programación, de hecho, la sintaxis de la mayoría de lenguajes de programación está definida mediante gramáticas libres de contexto. Por otro lado, estas gramáticas son suficientemente simples como para permitir el diseño 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



¿Cómo se identifica una gramática tipo 2?

La teoría dice que para que una gramática de tipo 2 se diferencie de una gramática de tipo 3, sus producciones pueden traer más de un terminal o un no terminal, mientras que en las tipo 3 solo puede traer un terminal o un no terminal o ambas pero no más.

Por lo que para diferenciar las gramáticas el programa toma la producción más grande y si el tamaño de esta es mayor a 2, la gramática es aceptada, de lo contrario se toma como una gramática de tipo 3 y es rechazada.



¿Cómo se Guardan las Gramáticas?

Para el almacenamiento de las gramáticas se utilizaron listas enlazadas las cuales guardan la siguiente información: Inicio, terminales, no terminales, nombre y producciones

Lista simplemente enlazada.



¿Cómo se identifican los componentes de las Gramáticas?

Para identificar las gramáticas, primero a la cadena se le hace un Split() y se separa por los saltos de línea. Y a la primera posición de ese arreglo se le asigna un valor.

0 = Nombre

El siguiente valor contiene lo que son los terminales, No Terminales y donde inicia la producción separados por “;” por lo que a esa línea se le aplicara un Split(“;”), y se le vuelve a asignar un valor por lo que quedaría de la siguiente manera.

0= No Terminales

1= Terminales

2 = Inicial

Para el resto de la cadena original se le toma como las producciones, para que al final al momento de mostrarlo quede de la siguiente manera:

```
Nombre: Grm1
No Terminales = {S,A,B,C}
Terminales={a,b,z}
Inicial={S}
Produccion:
S->A
A->a|A|a
A->B
B->b|B|b
B->C
C->z
*
```