

Una gramática libre de contexto es una especificación para la estructura sintáctica de un lenguaje de programación. Una especificación es muy similar a la especificación de la estructura léxica de un lenguaje utilizando expresiones regulares, excepto que una gramática libre de contexto involucra reglas de recursividad. Un ejemplo serían las expresiones aritméticas:

Exp → exp op exp | (exp) | numero

Op → + | - | * | /

Comparacion con respecto a la notación de una expresión regular

En las reglas de una expresión regular básica contamos con tres operaciones fundamentales: la selección, representada por el símbolo “|”, que permite elegir entre varias alternativas; la concatenación; y la repetición, indicada mediante el asterisco *, que expresa que un elemento puede aparecer cero o más veces. Además, las expresiones regulares utilizan el signo de igualdad (=) para definir un nombre que representa una expresión más amplia o compleja. Esta asignación facilita la reutilización de patrones y hace que la descripción de estructuras léxicas sea más clara y manejable, ya que permite referirse a un patrón mediante un identificador en lugar de reescribirlo cada vez.

Especificacion de las reglas de una gramática libre de contexto

Al igual que las expresiones regulares, las reglas gramaticales se definen sobre un alfabeto o conjunto de símbolos. En las expresiones regulares se hace uso de caracteres individuales y las gramáticas libre de contexto donde los símbolos son tokens, que representan cadenas de caracteres mas complejas como palabras reservadas, operadores, identificadores o números.

En las reglas BNF se compone de nombre de la estructura, símbolo de definición y la secuencia de símbolos o estructuras con alternativas separadas por “|”.

Ejemplo:

Exp → Exp Op Exp | (Exp) | Numero

Op → + | - | *

Se hace uso de paréntesis para agrupar opciones, como en las expresiones regulares. Ademas en lugar de usar “|”, es posible escribir varias reglas con el mismo nombre.

Exp → Exp Op Exp

Exp → (Exp)

Exp → Numero

Derivaciones y el lenguaje definido por una gramática

Una derivación es el proceso mediante el cual, empezando desde un símbolo inicial, se reemplazan no terminales usando las reglas de la gramática hasta obtener una cadena compuesta solo por terminales, el conjunto de todas las cadenas que se pueden generar por derivación son los lenguaje definido por gramática.

Las no terminales (como Exp) pueden seguir generándose, los terminales (+,(,if) son los símbolos finales. Las gramáticas pueden incluir producciones epsilon que representan cadenas vacías y permiten definir elementos opcionales o estructuras repetitivas sin contenido. Por ultimo la recursión puede ser por la izquierda o por la derecha y es equivalente a la repetición que en expresiones regulares se expresa mediante * o +.

Ejemplos:

Sentencia if

sentencia → sent-if

Sent-if → “if” “(“ exp “)” sentencia | “if” “(“ exp “)” sentencia “else” sentencia