

UNIVERSIDAD SAN PABLO DE GUATEMALA

Facultad de Ciencias Empresariales

Escuela de Ingeniería

Ingeniería en Sistemas y Ciencias de la Computación



GRAMÁTICAS FORMALES Y GRAMÁTICAS LIBRES DE CONTEXTO

Trabajo presentado en el curso de Compiladores

Impartido por el Ing. Marcos Alfredo Orozco De Paz

Yellsmy Lilibeth Toj García

2000579

Guatemala, 26 de septiembre de 2023.

GRAMÁTICAS FORMALES

Una gramática formal es una descripción precisa de la estructura de las sentencias que conforman un lenguaje, siendo este último, en el contexto de la programación, un lenguaje de programación. Se desglosa en cuatro componentes esenciales: el alfabeto de símbolos terminales (Σ_T), el alfabeto de símbolos no terminales (Σ_N), el símbolo inicial o axioma (S), y un conjunto de producciones (P). La notación de una gramática formal se expresa de la siguiente manera:

$$G = \{\Sigma_T, \Sigma_N, S, P\}$$

Donde:

- **Símbolos terminales (Σ_T):** Estos representan el conjunto de palabras reservadas y símbolos especiales dentro de un lenguaje de programación. Esto engloba caracteres especiales y operadores, tales como "main," "{," "}," "#," "include," "if," "then," "define," "int," "float," "char," "double," "+," "-", "*", ">," ">=," "|," y otros. Estos símbolos terminales se representan en minúsculas, por ejemplo: "int" o "float."
- **Símbolos no terminales (Σ_N):** Estos representan variables utilizadas en las producciones gramaticales como símbolos de transición. Incluyen el símbolo inicial de la gramática (S) y cualquier otra variable que se utilice en las producciones. Estos símbolos no terminales se representan en mayúsculas y ejemplificaremos esto al describir las producciones.
- **Símbolo inicial (S):** Este es un símbolo no terminal a partir del cual se derivan todas las palabras del lenguaje, y suele ser el primer símbolo de la gramática G. También se conoce como el axioma de la gramática.
- **Conjunto de producciones (P):** Este conjunto de reglas establece cómo se forman las palabras en el lenguaje definido por G, y establece las relaciones entre los símbolos terminales y los no terminales. Las producciones definen cómo se pueden combinar y reorganizar los símbolos no terminales y terminales para construir sentencias y expresiones válidas en el lenguaje.

En conjunto, una gramática formal proporciona un marco riguroso y preciso para definir la sintaxis de un lenguaje de programación, permitiendo a los programadores escribir código que cumpla con las reglas gramaticales del lenguaje y, así, ejecutar de manera adecuada las instrucciones dentro del sistema informático.

Tipos de Gramáticas Formales

Existen varios tipos de gramáticas formales, clasificadas en función de su capacidad expresiva y su complejidad. Los tipos más comunes son las gramáticas de tipo 0 (Turing completas), gramáticas de tipo 1 (sensibles al contexto), gramáticas de tipo 2 (contexto libre), y gramáticas de tipo 3 (regulares). Cada tipo tiene reglas y restricciones específicas, siendo las gramáticas de tipo 2 (contexto libre) las más utilizadas para describir la sintaxis de la mayoría de los lenguajes de programación.

GRAMÁTICAS LIBRES DE CONTEXTO

Las gramáticas libres del contexto, también conocidas como gramáticas de tipo 2 o gramáticas independientes del contexto, son una clase importante de gramáticas formales utilizadas para definir lenguajes independientes del contexto. Estos lenguajes son aquellos que pueden ser reconocidos por autómatas de pila, tanto determinísticos como no determinísticos.

Una gramática libre del contexto se define mediante una cuadrupla $G = (N, T, P, S)$, donde:

- N es un conjunto finito de símbolos no terminales.
- T es un conjunto finito de símbolos terminales, y la intersección de N y T es un conjunto vacío ($N \cap T = \emptyset$).
- P es un conjunto finito de producciones.
- S es el símbolo distinguido o axioma, que no pertenece al conjunto de símbolos no terminales ni terminales ($S \notin N \cup T$).

En una gramática libre del contexto, cada producción en P tiene la forma:

$$A \in N \cup \{S\}$$

$$A \rightarrow \omega$$

$$\omega \in (N \cup T)^* - \{\varepsilon\}$$

Esto significa que en el lado izquierdo de una producción puede aparecer el símbolo distinguido o cualquier símbolo no terminal, y en el lado derecho de una producción puede encontrarse cualquier cadena de símbolos terminales y/o no terminales de longitud mayor o igual a 1.

Es importante destacar que una gramática libre del contexto también puede incluir la producción $S \rightarrow \varepsilon$ si el lenguaje que se desea generar permite la cadena vacía. Esta producción permite representar la posibilidad de tener una cadena vacía en el lenguaje definido por la gramática.

Las gramáticas libres del contexto se escriben, frecuentemente, utilizando una notación conocida como BNF (Backus-Naur Form). BNF es la técnica más común para definir la sintaxis de los lenguajes de programación. En esta notación se deben seguir las siguientes convenciones:

- los no terminales se escriben entre paréntesis angulares $\langle \rangle$
- los terminales se representan con cadenas de caracteres sin paréntesis angulares
- el lado izquierdo de cada regla debe tener únicamente un no terminal (ya que es una gramática libre del contexto)
- el símbolo $::=$, que se lee “se define como” o “se reescribe como”, se utiliza en lugar de \rightarrow
- varias producciones del tipo

$$\langle A \rangle ::= \langle B1 \rangle$$

$$\langle A \rangle ::= \langle B2 \rangle$$

$$\langle A \rangle ::= \langle Bn \rangle$$

se pueden escribir como $\langle A \rangle ::= \langle B1 \rangle \square \langle B2 \rangle \square \dots \square \langle Bn \rangle$

En resumen, estas gramáticas se componen de símbolos terminales y no terminales, producciones que definen reglas de reescritura y un símbolo axioma que inicia la generación de cadenas válidas en el lenguaje.