Universidad Politécnica de Aguascalientes

Materia: Lenguajes y Autónomas.

Tarea: Ensayo Unidad 4.

Profesor: Dr. Christian José Correa Villalon.

Alumno: Juan Carlos Pedroza Hernández.

Matrícula: UP170132.

Grupo: ISEI007.

Fecha de Entrega: 17/04/2020

Contenido

[ Gramáticas y la jerarquía de CHOMSKY 2](#_Toc37263358)

[ Arboles de Derivación 4](#_Toc37263359)

# Gramáticas y la jerarquía de CHOMSKY

En 1956 Chomsky crea una manera para clasificar los lenguajes formales en cuatro categorías enumerados de 0 a 3 y los mecanismos organizados formalizadores como gramáticas formales, y expresiones y autómatas para reconocer cada tipo.

También se conoce bajo el nombre de Clasificación de Chomsky o Jerarquía matemática de los lenguajes.

Los autómatas son mecanismos formales que “realizan” derivaciones en gramáticas formales. La manera en que las realizan es mediamente el reconocimiento. Una palabra se genera en una gramática si y solo sí cumple con las condiciones terminales del autónomo. Por esto es que los autómatas son analizados los léxicos de las gramáticas que les corresponde

Se puede caracterizar qué tan compleja es una gramática formal. Noam Chomsky mostró que esta caracterización clasifica jerárquicamente a las gramáticas formales: Gramáticas en un nivel están incluidas en los siguientes niveles y la inclusión entre niveles es propia.

**La Jerarquía de Chomsky tiene cuatro niveles:**

**Gramáticas 0 (sin restricciones):**que incluye a todas las gramáticas formales. Estas gramáticas generan todos los lenguajes capaces de ser reconocidos por una máquina de Turing. Los lenguajes son conocidos como lenguajes recursivamente enumerarles.

**Gramáticas 1** **(gramáticas sensibles al contexto):** generan los lenguajes sensibles al contexto. Estas gramáticas tienen reglas de la forma con A un no terminal y “α, β y” γ cadenas de terminales y no terminales. Las cadenas α y β pueden ser vacías, pero “γ” no puede serlo.

La regla está permitida si S no aparece en la parte derecha de ninguna regla. Los lenguajes descritos por estas gramáticas son exactamente todos aquellos lenguajes reconocidos por una máquina de Turing determinista cuya cinta de memoria está acotada por un cierto número entero de veces sobre la longitud de entrada, también conocidas como autómatas linealmente acotados.

**Gramáticas 2** **(gramáticas libres del contexto):** generan los lenguajes independientes del contexto. Las reglas son de la forma con A un no terminal y γ una cadena de terminales y no terminales. Estos lenguajes son aquellos que pueden ser reconocidos por un autómata con pila.

**Gramáticas 3** **(gramáticas regulares):** generan los lenguajes regulares. Estas gramáticas se restringen a aquellas reglas que tienen en la parte izquierda un no terminal, y en la parte derecha un solo terminal, posiblemente seguido de un no terminal. La regla también está permitida si S no aparece en la parte derecha de ninguna regla. Estos lenguajes son aquellos que pueden ser aceptados por un autómata finito. También esta familia de lenguajes puede ser obtenidas por medio de expresiones regulares.

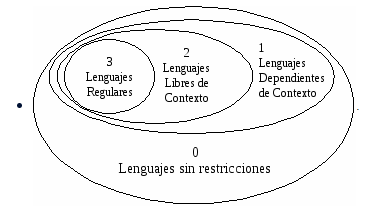


Ilustración 1: Lenguajes

La jerarquía de Chomsky no solo aporta un ordenamiento conjunto de los lenguajes, sino que proporciona un mecanismo de clasificación basado en características relacionadas con las formas gramaticales mínimas de cada lenguaje en particular, así como decide qué tipo de reconocedores mínimos servirían para determinar las cadenas válidas.

Tabla :gramáticas

|  |  |  |  |
| --- | --- | --- | --- |
| TIPO | NOMBRE | FORMA DE LAS PRODUCCIONES | TIPO DE AUTONOMAS O RECONOCEDORES |
| 0 | LRE | IRRESTRICTAS | Maquina de Turing |
| 1 | LDC |  | Autómatas Linealmente acotados |
| 2 | LLC |  | Autómata de pila |
| 3 | LR |  | Autómata finito Expresión regular |

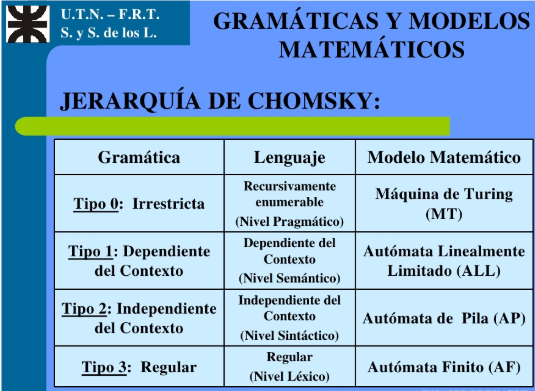


Ilustración : Jerarquía de Chomsky

# Arboles de Derivación

Un árbol de derivación (o *árbol sintáctico*) es una representación gráfica de cómo se deriva una forma sentencial a partir del símbolo no-terminal inicial.

Un árbol es un grafo dirigido acíclico en el cual cada nodo se conecta con un nodo distinguido, llamado nodo raíz por un único camino. Un nodo n 1 se dice descendiente de otro nodo n 2 si se puede llegar a n 1 a partir de n 1. El nodo raíz no es descendiente de ningún nodo, y los nodos que no tienen descendientes se denominan hojas. El resto de los nodos se denominan nodos interiores.

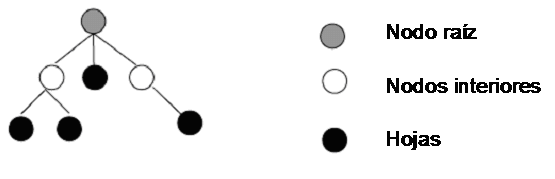


Ilustración : Árbol de derivación

Construcción de un árbol de derivación:

Cada nodo del árbol va a contener un símbolo.

En el nodo raíz se pone el símbolo inicial S.

Se efectúa una ramificación del árbol por cada producción que se aplique: Si a la variable de un nodo, A, se le aplica una determinada regla A ->α, entonces para cada símbolo que aparezca en α se añade un hijo con el símbolo correspondiente, situados en el orden de izquierda a derecha.

Este proceso se repite para todo paso de la derivación. Si la parte derecha es una cadena vacía, entonces se añade un solo hijo, etiquetado con ε. En cada momento, leyendo los nodos de izquierda a derecha se lee la palabra generada.

Ejemplo 1:

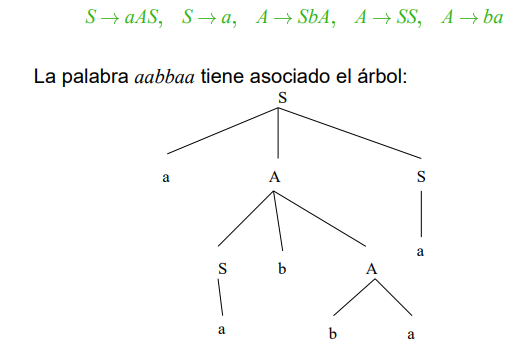


Ilustración : Ejemplo de creación de árbol de derivación

Un árbol de derivación puede proceder de dos cadenas de derivación distintas:

* Se llama derivación por la izquierda asociada a un árbol a aquella en la que siempre se deriva primero la primera variable (más a la izquierda) que aparece en la palabra.
* Se llama derivación por la derecha asociada a un árbol a aquella en la que siempre se deriva primero la última variable (más a la derecha) que aparece en la palabra.

Ejemplo 2:

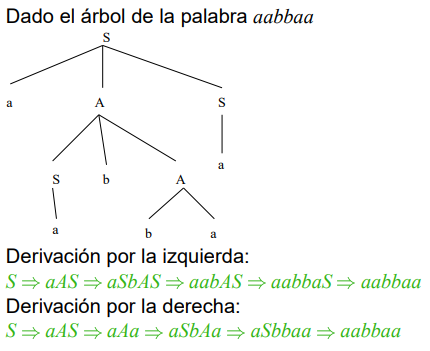


Ilustración : Tipos de derivación.

Gramática ambigua:

Una gramática se dice ambigua si existe una palabra con dos árboles de derivación distintos.

Ejemplo: La gramática S ->AA, A-> aSa, A ->a es ambigua, ya que la palabra a5 tiene los dos árboles siguientes.

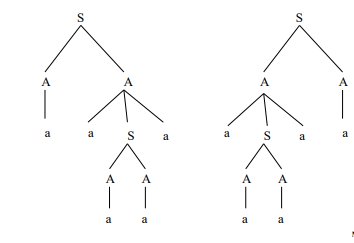


Ilustración : Gramática Ambigua.

Lenguaje inherentemente ambiguo:

Un lenguaje de tipo 2 es inherentemente ambiguo si toda gramática que lo genera es ambigua.

Ejemplo: El lenguaje generado por la gramática anterior S->AA, A->aSa, A-> a, no es inherentemente ambiguo. Este lenguaje es {a2+3i : i ≥ 0} y puede ser generado por la gramática:

S->aa, S ->aaU, U ->aaaU, U ->aaa, que no es ambigua. Único árbol para a5:

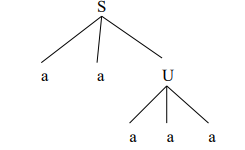


Ilustración : Lenguaje inherentemente ambiguo