



TECNOLÓGICO
NACIONAL DE MÉXICO®



“TECNOLÓGICO NACIONAL DE
MEXICO”

INSTITUTO TECNOLÓGICO DE
IZTAPALAPA

INTEGRANTES:

ISC-6AM

LENGUAJES Y AUTOMATAS I

M.C. ABIEL TOMÁS PARRA
HERNÁNDEZ

SEP 2020 / FEB 2021

ACTIVIDAD SEMANA 5



Cuanenemi Cuanalo Mario Alberto

Representación finita del lenguaje

Bueno ya una vez expuesto el tema en el video y checado información extra para la comprensión del tema nos lleva a Representación finita del lenguaje

Un lenguaje consiste de un grupo de cadenas de un alfabeto. Usualmente ciertas restricciones se aplican a las cadenas del lenguaje. Por ejemplo el lenguaje Español consiste de todas las cadenas de palabras que nosotros llamamos oraciones. No todas las combinaciones de palabras forman oraciones. De ahí que un lenguaje consiste de un subconjunto del conjunto de todas las posibles cadenas que se pueden formar del alfabeto.

Ejemplo: El lenguaje L de cadenas de el alfabeto $\{a,b\}$ en

donde cada cadena comienza con una a y tiene longitud par.

Las cadenas aa, ab, aaaa, abbb, abab, abbbbaaba forman parte

de ese lenguaje.

Palíndromos

Cadenas que se leen igual de izquierda a derecha y viceversa.

Un alfabeto

Es un conjunto finito de símbolos.

Un Lenguaje:

Conjunto de cadenas de símbolos tomados de algún alfabeto.

Lectura:

Sobre el alfabeto $\{0,1\}$ es seria la siguiente $\{1,0\}$.



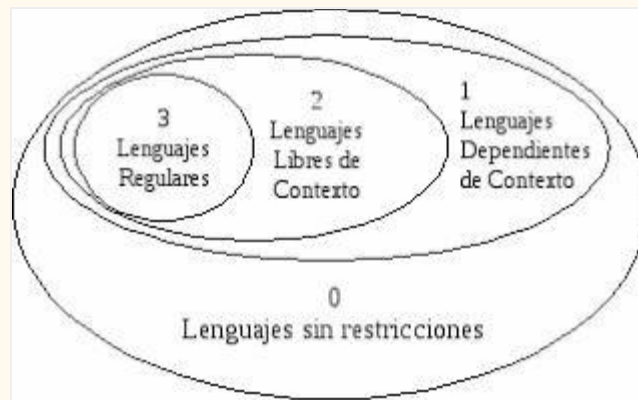
Expresiones Regulares

Los lenguajes aceptados por un autómata finito se describen con facilidad mediante expresiones simples llamadas expresiones regulares.

Sea S un alfabeto. La expresión regular sobre S y los conjuntos que denotan se definen de manera recursiva.

\emptyset es una expresión regular y denota al conjunto vacío. ϵ es una expresión regular y denota al conjunto $\{\epsilon\}$. Para cada $a \in S$, a es una expresión regular y denota al conjunto $\{a\}$

TIPOS DE GRAMÁTICA



Ø TIPO “0” O “No restringida o recursivamente enumerables”:

Incluye a todas las gramáticas formales. Estas gramáticas generan todos los lenguajes de ser reconocidos por una máquina de Turing. Y con este lenguaje se hacen los parsers de un compilador

CARACTERÍSTICAS: De este tipo es que del lado derecho de cada producción puede empezar con un símbolo terminal o con un no terminal y del lado izquierdo puedes



empezar con más de un símbolo no terminal.

RESTRICCIONES: Es que no tiene solamente que el del lado izquierdo debe haber por lo menos un símbolo no terminal

NOTA: “+” significa “*sin incluir la cadena vacía*” y significa “*incluyendo la cadena vacía*”. “|” significa “o”.

Estos lenguajes también son denominados “recursivamente enumerables”

$$\begin{aligned}x &\rightarrow y \\ x &\in (NT/T)^+ \\ y &\in (NT/T)^*\end{aligned}$$

Ø TIPO “1” O “Sensible al contexto”:

Estos tipos de lenguajes se resuelven mediante autómatas lineales limitados. Con este tipo se hacen los parser (analizador sintáctico) de un compilador que transforma su entrada en un árbol de derivación.

El analizador sintáctico convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada

CARACTERÍSTICAS: Del lado derecho de cada producción puede empezar con un símbolo terminal o con un no terminal y del lado izquierdo puede empezar con más de un símbolo no terminal.

RESTRICCIONES: el número de no terminales del lado izquierdo de la producción debe ser menor o igual al número de símbolos del lado derecho

NOTA: Los lenguajes regulares y los libres de contexto también se puede resolver mediante autómatas lineales limitados

$$\begin{aligned}\alpha &\rightarrow \beta ; |\alpha| \leq |\beta| \\ \alpha &= z_1 x z_2 \\ \beta &= z_1 y z_2 \\ z_1, z_2 &\in T^* \\ x &\in NT \\ y &\in (NT/T)^+\end{aligned}$$

Ø TIPO “2” O “Libres o Independientes de contexto”:

Estos tipos de lenguajes se resuelven mediante autómatas descendentes y con este tipo de lenguaje se programa los parser en un compilador, permiten describir la mayoría de los lenguajes de programación, de hecho, la sintaxis de la mayoría de los lenguajes de programación está definida mediante gramáticas libres de contexto.



CARACTERÍSTICAS: Del lado derecho de cada producción puede empezar con símbolo terminal o con un no terminal

Los lenguajes regulares también se pueden resolver mediante autómatas descendentes

$$\begin{aligned} x &\rightarrow y \\ x &\in NT \\ y &\in (NT/T)^* \end{aligned}$$

Ø TIPO “3” O “Lenguajes regulares”:

Estos tipos de lenguajes se resuelven mediante autómatas finitos y con este tipo de lenguaje se hacen los scanners. 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 y también esta familia de lenguajes pueden ser obtenidas por medio de expresiones regulares

CARACTERÍSTICAS: Del lado derecho de cada producción debe empezar con un símbolo termin al.

$$\begin{aligned} \alpha &\rightarrow \beta \\ \alpha &\in NT \\ \beta &\in \begin{cases} aB \\ Ba \\ b \end{cases} \\ B &\in NT \\ a &\in T^+ \\ b &\in T^* \end{aligned}$$

Árboles de Derivación

Un árbol de derivación permite mostrar gráficamente cómo se puede derivar cualquier cadena de un lenguaje a partir del símbolo distinguido de una gramática que genera ese lenguaje.



Un árbol es un conjunto de puntos, llamados nodos, unidos por líneas, llamadas arcos. Un arco conecta dos nodos distintos.

Esquema de un árbol de derivación

Propiedades de un árbol de derivación:

Sea $G = (N, T, S, P)$ una gramática libre de contexto, sea S una variable. Diremos que un árbol $TA = (N, E)$ etiquetado es un árbol de derivación asociado a G si verifica las propiedades siguientes:

- La raíz del árbol es un símbolo no terminal.
- Cada hoja corresponde a un símbolo terminal o λ .
- Cada nodo interior corresponde a un símbolo no terminal.

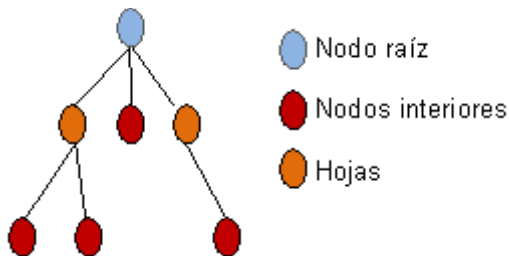
Para cada cadena del lenguaje generado por una gramática es posible construir (al menos) un árbol de derivación, en el cual cada hoja tiene como rótulo uno de los símbolos de la cadena.

Propiedades de un árbol de derivación.

Sea $G = (N, T, S, P)$ una gramática libre de contexto, sea S una variable. Diremos que un árbol $TA = (N, E)$ etiquetado es un árbol de derivación asociado a G si verifica las propiedades siguientes:

- La raíz del árbol es un símbolo no terminal
- cada hoja corresponde a un símbolo terminal o λ .
- cada nodo interior corresponde a un símbolo no terminal.

Para cada cadena del lenguaje generado por una gramática es posible construir (al menos) un árbol de derivación, en el cual cada hoja tiene como rótulo uno de los símbolos de la cadena.



Para cada cadena del lenguaje generado por una gramática es posible construir (al menos) un árbol de derivación, en el cual cada hoja tiene como rótulo uno de los símbolos de la cadena.

Si un nodo está etiquetado con una variable X y sus descendientes (leídos de izquierda a derecha) en el árbol son X_1, \dots, X_k , entonces hay una producción $X \rightarrow X_1 \dots X_k$ en G .

Sea $G = (N, T, S, P)$ una GLC. Un árbol es un árbol de derivación para G si:

1. Todo vértice tiene una etiqueta tomada de
2. La etiqueta de la raíz es el símbolo inicial S



3. Los vértices interiores tienen etiquetas de N
4. Si un nodo n tiene etiqueta A y n_1, n_2, \dots, n_k respectivamente son hijos del vértice n , ordenados de izquierda a derecha, con etiquetas x_1, x_2, \dots, x_k respectivamente, entonces: $A \rightarrow x_1 x_2 \dots x_k$ debe ser una producción en P
5. Si el vértice n tiene etiqueta λ , entonces n es una hoja y es el único hijo de su padre.

Árbol de derivación. Ejemplo

Sea $G = (N, T, S, P)$ una GLC con $P: S \rightarrow ab | aSb$

La derivación de la cadena $aaabbb$ será: y
el árbol de derivación:

