Clasificación de gramáticas.

Contenido

- Introducción
- Gramáticas tipo 3: Regulares
- Gramáticas tipo 2: Independientes o libres del contexto
- Gramáticas tipo 1: Dependientes del contexto
- Gramáticas tipo 0: Sin restricciones
- Ejercicios

Introducción.

- Retomando la definición formal de gramáticas:

Se llama gramática formal definida sobre un alfabeto $\sum$ a una tupla de la forma:

$$G = \{ \sum_T, \sum_N, S, P \}$$

donde :

$\sum_T$ es el alfabeto de símbolos terminales

$\sum_N$ es el alfabeto de símbolos no terminales

(aparecen en los ejemplos encerrados entre **<>**)

S es el símbolo inicial de la gramática

P es un conjunto de producciones gramaticales

Introducción. Clasificación de gramáticas.

- En 1956, Noam Chomsky definió cuatro tipos de gramáticas formales que se diferencian en los tipos (reglas) de producción de la gramática: **jerarquía de Chomsky.**
- Esta clasificación describe las gramáticas desde los tipos más generales a los más específicos, dependiendo de la forma de las reglas de la gramática.
- Esta clasificación permite introducir, al mismo tiempo, una clasificación en los lenguajes que las gramáticas generan, y también, una clasificación en los autómatas que reconocen los lenguajes generados por las gramáticas.

Introducción. Jerarquía de Chomsky

Gramáticas	Lenguajes	Máquinas	Problemas
			No computables
Tipo 0 (G. sin restricción)	Recursivamente enumerables	Máquinas de Turing	Computables
Tipo 1 (G. sensible al contexto)	Lenguajes sensibles al contexto	Autómatas lineales acotados	
Tipo 2 (G. Libre de contexto)	Lenguajes independientes del contexto	Autómatas de pila	
Tipo 3 (G. Regular)	Lenguajes regulares	Autómatas finitos deterministas	Expresiones regulares

Gramática tipo 3. Gramática regular

- **Gramática regular derecha** (gramática lineal derecha) es una gramática formal $G = (\Sigma_N, \Sigma_T, S, P)$ tal que todas las reglas de producción en P son de la siguiente forma:
  - $B \rightarrow a$ - donde B es no terminal en Σ_N y a es terminal in Σ_T.
  - $B \rightarrow aC$ - donde B y C son no terminales en Σ_N y a es terminal en Σ_T.
  - $B \rightarrow \varepsilon$ - donde B es no terminal en Σ_N y ε denota la cadena vacía i.e. La cadena de longitud 0.
- **Gramática regular izquierda**(gramática lineal izquierda) es una gramática formal $G = (\Sigma_N, \Sigma_T, S, P)$ tal que todas las reglas de producción en P son de la siguiente forma:
  - $B \rightarrow a$ - donde B es no terminal en Σ_N y a es terminal in Σ_T.
  - $B \rightarrow Ca$ - donde B y C son no terminales en Σ_N y a es terminal en Σ_T.
  - $B \rightarrow \varepsilon$ - donde B es no terminal en Σ_N y ε denota la cadena vacía i.e. La cadena de longitud 0.

Gramática tipo 3. Gramática regular

- Ejemplos:

G_1 = ({A, B}, { 0, 1}, A, { **A → B1** | 1, **B → A0**})

Gramática lineal izquierda que describe el lenguaje:

L_1 = { 1, 101, 10101, ... } = $\{1(01)^n \mid n = 0, 1, 2, \ldots\}$

G_2 = ({ A, B}, {0, 1}, A, { **A → 1B** | 1, **B → 0A**})

Gramática lineal derecha que genera el mismo lenguaje que la gramática anterior.

Gramática tipo 3. Gramática regular

- Los lenguajes representados por este tipo de gramáticas se denominan lenguajes regulares.
- Los lenguajes regulares se utilizan para definir **estructura léxica** de los lenguajes de programación. Definen la sintaxis de los identificadores, números, cadenas y otros elementos básicos del lenguaje.
- Estos lenguajes pueden ser reconocidos por los autómatas finitos.
- Las gramáticas tipo 3 son las más restrictivas.

Gramática tipo 2. Independiente o libre del contexto

- Generan los lenguajes libres de contexto. Están definidas por reglas de la forma:

  $A \rightarrow \gamma$ - donde A es no terminal en $\sum_N$ y γ es una cadena de terminales y no terminales (incluye λ).

- Se considera libre del contexto cuando sus reglas de producción pueden ser aplicadas sin importar el contexto del símbolo no terminal A. No importa qué símbolos lo rodeen, el símbolo no terminal del lado izquierdo A puede ser siempre remplazado por el lado derecho γ . Esto es lo que distingue a ésta gramática de la gramática sensitiva o dependiente del contexto.
- Estos lenguajes pueden ser reconocidos por los Autómatas de Pila.

Gramática tipo 2. Independiente o libre del contexto

- Ejemplo:

Sea la gramática $G = (\{S\}, \{a, b\}, S, \{ S \rightarrow aSb \mid ab\})$.

La derivación de la palabra aaabbb será:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$$

Puede verse que el lenguaje definido por esta gramática es $\{a^n b^n \mid n=1, 2, ...\}$

Gramática tipo 2. Independiente o libre del contexto

- Los lenguajes independientes de contexto constituyen la base teórica para la sintaxis de la mayoría de los lenguajes de programación. Definen la sintaxis de las declaraciones, las proposiciones, las expresiones, etc.

Gramática tipo 1. Dependiente del contexto

- Generan los lenguajes dependientes de contexto. Contienen reglas de producción de la forma:

$$\alpha\, A\, \beta \rightarrow \alpha\, \gamma\, \beta$$

- **Donde:**
  - ***A*** es no terminal en Σ_N.
  - α, β y γ son cadenas de terminales y no terminales.
  - α y β pueden ser vacíos, pero γ tendrá que ser distinto del vacío.

Gramática tipo 1. Dependiente del contexto

- Se denominan gramáticas dependientes del contexto, debido a que **A** puede ser sustituida por γ siempre y cuando esté acompañada de α por la izquierda y de β por la derecha (**las partes izquierda y derecha deben tener una parte común**)

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

Solo se admite como regla compresora **S::= λ**

Estos lenguajes pueden ser reconocidos por autómatas lineales acotados (Maquina de Turing Determinista).

Gramática tipo 1. Ejemplo

- G= ({ a, b, c }, { S, B, C }, S, P)

  P = {

  $S \to aSBC \mid abC$

  $CB \to BC$

  $bB \to bb$

  $bC \to bc$

  $cC \to cc$

  }

  Derivación de aaabbbccc:

  $S \to aSBC \to aaSBCBC \to aaabCBCBC \to aaabBCCBC \to aaabBCBCC \to$

  $\to aaabBBCCC \to aaabbBCCC \to aaabbbCCC \to aaabbbcCC \to$

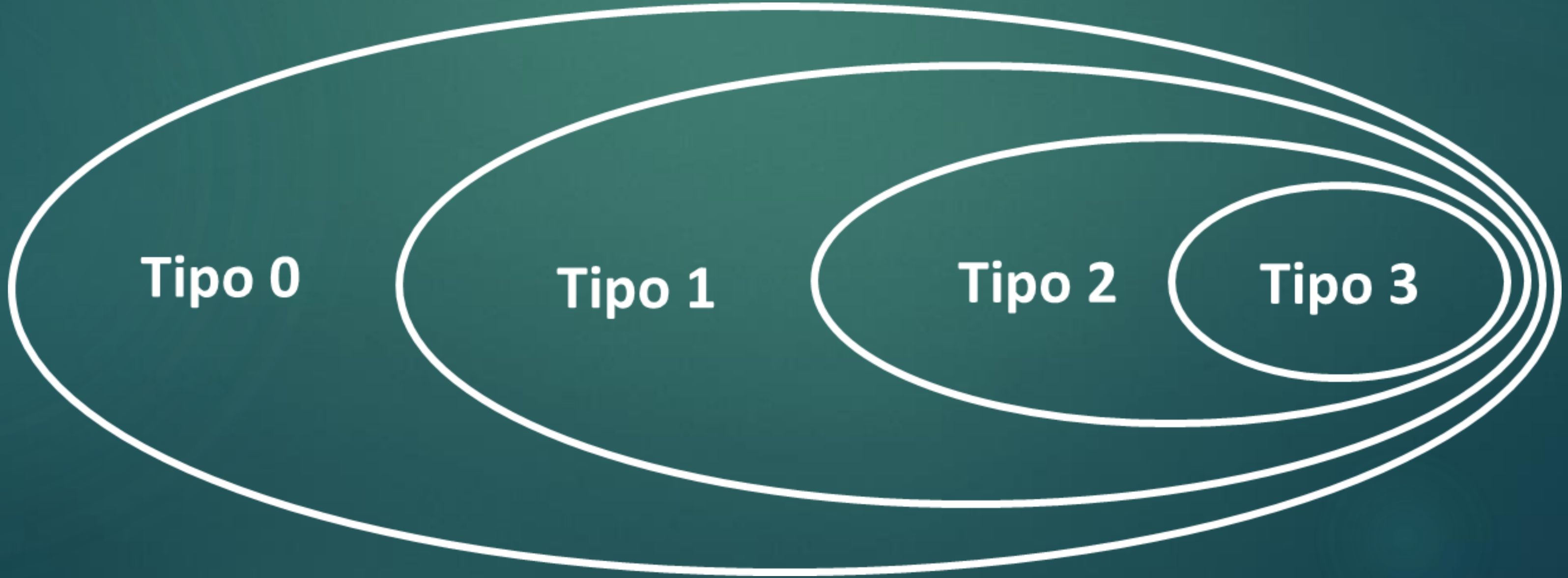
  $\to aaabbbccC \to aaabbbccc$

Gramática tipo 0. Sin restricciones

- Incluye todas las gramáticas formales.
- Es la gramática más general, a la que pertenece la semántica de los lenguajes naturales y artificiales.
- En la parte izquierda tiene que haber al menos un símbolo no terminal. **Respecto a las partes derechas de sus producciones no hay restricción alguna**.
- Estos lenguajes son todos los lenguajes que pueden ser reconocidos por una *máquina de Turing*.

Clasificación de las gramáticas

- Todo lenguaje de tipo 3 es de tipo 2, todo lenguaje de tipo 2 es de tipo 1, y todo lenguaje de tipo 1 es de tipo 0.
- Se dice que un lenguaje es de tipo k [k = 0, k = 1, k = 2, k = 3] cuando existe una gramática de tipo k que genera ese lenguaje.



Para cada caso: determinar el tipo de gramática, realizar derivaciones, determinar el lenguaje o expresión regular.

G=({a,b},{S}, S, P}
P: {
S→aSa
S→bSb
S→ε
}
	G=({a,b,c},{S,A,B, C},S,P}
	P: {
	S→aA
	A→aA | aB
	B→bB | bC
	C→cC | c
	}
		G=({a,b},{S,A}, S, P}
		P: {
		S→bSbb | A
		A→aA | ε
		}
			G=({a,b,c},{S,B,W, X}, S, P}
			P: {
			S→abc | aSBc
			cB→wB
			WB→WX
			WX→BX
			BX → Bc
			bB → bb
			}
				G=({a,b},{S,A,B,C, D}, S, P}
				P: {
				S→CD
				C→aCA | bCB
				AD→aD
				BD→bD
				Aa→aA
				Ab→bA
				Ba→aB
				Bb→bB
				C→λ
				D→λ
				}