

# Teoría de la Computación 2025

Lab 05

10.septiembre.2025

1. Construir una gramática CFG  $G = (V, T, S, P)$  para los lenguajes que se indican a continuación. Para cada gramática, indicar explícitamente los siguientes:  $V$  el conjunto de variables,  $T$  el conjunto de símbolos terminales,  $S$  el símbolo inicial y  $P$  el conjunto de producciones o reglas de producción.

a)  $L(G) = \{0^m 1^n 2^n 3^m : m, n \geq 0\}$ .

b)  $L(G) = \{0^m 1^n 2^{m+n} : m, n \geq 0\}$ .

2. Construir una gramática CFG que permita escribir la notación de punto flotante. Esto es, que permita escribir cantidades como

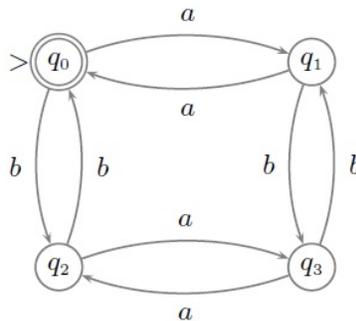
3.14, 0.5, 42. 6.022e23, 9.11E-31

y construir el árbol sintáctico para la expresión 6.022E23.

Observe que, en general, una expresión de punto flotante tiene la estructura

$$\text{Float} \longrightarrow \text{Int} \cdot (\text{Int} \mid \text{IntLetter} [\text{UnaryOp}] \text{Int})$$

3. (a) Considere el siguiente autómata AFD. ¿Cuál es el lenguaje  $L(M)$  generado?  
(b) Construya una gramática CFG que represente el mismo lenguaje  $L(M)$ .  
(c) Muestre que la cadena  $w = \text{abbabb}$  está en  $L(M)$ , mostrando una derivación para  $w$ , y grafique el árbol sintáctico de dicha derivación.



4. Considere la gramática  $G = (V, T, S, P)$ , con  $V = \{S\}$ ,  $T = \{a, 1, +\}$ ,  $S = S$ , y con reglas

$$S \longrightarrow S + S$$

$$S \longrightarrow 1$$

$$S \longrightarrow a$$

- (a) Para la cadena  $w = 1 + 1 + 1 + a$ , encuentre: una derivación *leftmost*, una derivación *rightmost*, y una derivación que no sea *leftmost* ni *rightmost*. Para cada una, dibuje el árbol sintáctico.  
(b) ¿Es  $G$  ambigua?

5. Determine si la siguiente gramática  $G$ , con reglas de producción

$$X \rightarrow X + X \mid X * X \mid X \mid a$$

es ambigua o no.

6. Remover la ambigüedad en las siguientes gramáticas CFG.

i)  $V = \{A\}$ ,  $T = \{a, (, )\}$ ,  $S = A$

$$A \rightarrow AA$$

$$A \rightarrow (A)$$

$$A \rightarrow a$$

ii)  $V = \{E\}$ ,  $T = \{+, *, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ,  $S = E$

$$E \rightarrow E + E$$

$$E \rightarrow E * E$$

$$E \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

iii)  $V = \{S, A, B\}$ ,  $T = \{a, b\}$ ,  $S = S$

$$S \rightarrow SS$$

$$S \rightarrow AB$$

$$A \rightarrow Aa$$

$$A \rightarrow a$$

$$B \rightarrow Bb$$

$$B \rightarrow b$$

7. Aplicar el algoritmos de reducción para simplificar las siguientes gramáticas

i)  $V = \{S, A, B, C\}$ ,  $T = \{a, b\}$ ,  $S = S$

$$S \rightarrow aS \mid A \mid C$$

$$A \rightarrow a$$

$$B \rightarrow aa$$

$$C \rightarrow aCb$$

ii)  $V = \{S, A, B\}$ ,  $T = \{a, b\}$ ,  $S = S$

$$S \rightarrow ASA \mid aB \mid b \mid a \mid SA \mid AS \mid S$$

$$A \rightarrow B \mid b$$

$$B \rightarrow b$$

8. Convertir las gramáticas del ejercicio 7 a su Forma Normal de Chomsky.

9. Investigar y explicar qué es un *parser* **Top-Down**. Explicar cómo la forma normal de Greibach ayuda en el diseño de un *parser* Top-Down.