

## Práctica 4

Integrantes:

- |                                   |           |
|-----------------------------------|-----------|
| - Mayra Díaz Tramontana           | 201910147 |
| - Joaquín Elías Ramírez Gutiérrez | 201910277 |

Carpeta con los archivos:

1. Diseñe gramáticas para:

- a. El conjunto de cadenas binarias palíndromas (que se lean igual en ambas direcciones)

$$S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon$$

- b. El conjunto de cadenas binarias con igual cantidad de 0's y 1's

$$S \rightarrow SS \mid 1S0 \mid 0S1 \mid \varepsilon$$

Otra posibilidad:

$$S \rightarrow 1S0S \mid 0S1S \mid \varepsilon$$

2. Describa el lenguaje definido por las siguientes gramáticas, desarrolle la derivación por la izquierda, por la derecha, e indique si es una gramática ambigua:

- a.  $S \rightarrow S+S \mid SS \mid (S) \mid S^* \mid a$ , cadena:  $(a+a)^*a$

i. **m.i:**  $S \rightarrow SS \rightarrow S^*S \rightarrow (S)^*S \rightarrow (S+S)^*S \rightarrow (a+S)^*S \rightarrow (a+a)^*S \rightarrow (a+a)^*a$

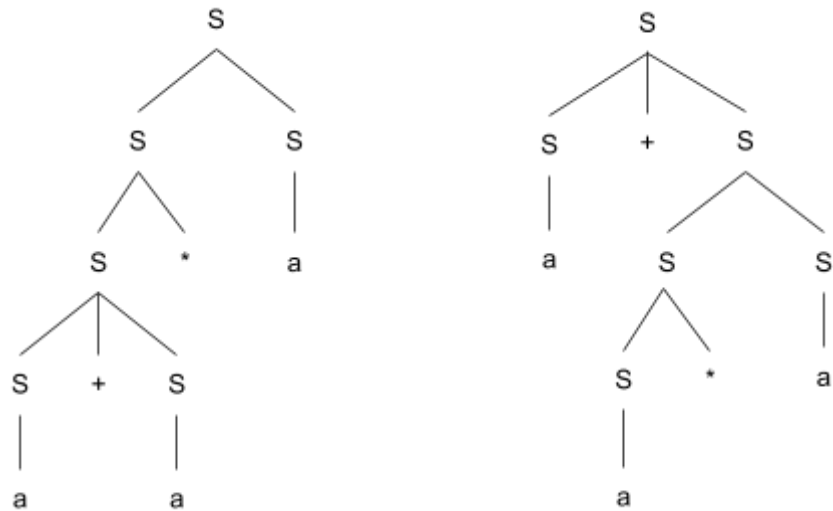
ii. **m.d:**  $S \rightarrow SS \rightarrow Sa \rightarrow S^*a \rightarrow (S)^*a \rightarrow (S+S)^*a \rightarrow (S+a)^*a \rightarrow (a+a)^*a$

iii. Sí es ambigua.

Dos árboles diferentes para la cadena  $a+a^*a$  y dos derivaciones m.i.

**m.i:**  $S \rightarrow SS \rightarrow S^*S \rightarrow S+S^*S \rightarrow a+S^*S \rightarrow a+a^*S \rightarrow a+a^*a$

**m.i:**  $S \rightarrow S+S \rightarrow a+S \rightarrow a+SS \rightarrow a+S^*S \rightarrow a+a^*S \rightarrow a+a^*a$



- iv. Cadenas de '+', '\*', 'a' y paréntesis simétricos, en las que '+' no puede ser ni el inicio ni el final, y '\*' no puede ser el inicio.

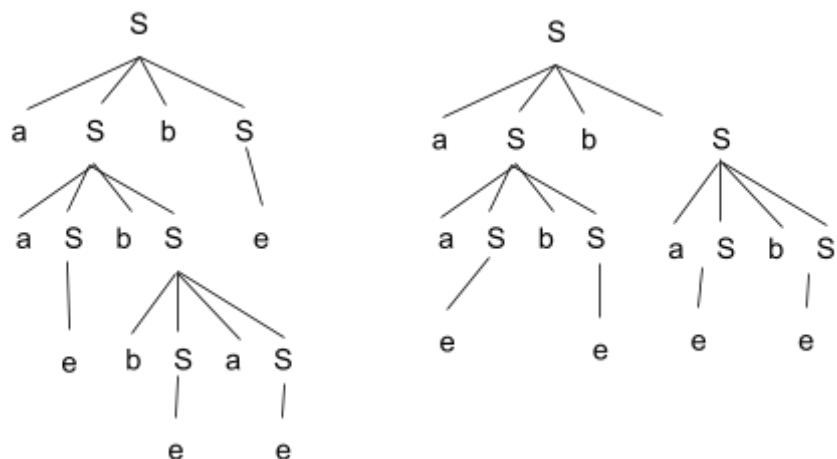
b.  $S \rightarrow aSbS \mid bSaS \mid \epsilon$  cadena: **aabbab**

- i. **m.i:**  $S \rightarrow aSbS \rightarrow aaSbSbS \rightarrow aabSbS \rightarrow aabbS \rightarrow aabbaSbS \rightarrow aabbabS \rightarrow aabbab$
- ii. **m.d:**  $S \rightarrow aSbS \rightarrow aSbaSbS \rightarrow aSbaSb \rightarrow aSbab \rightarrow aaSbSbab \rightarrow aaSbbab \rightarrow aabbab$
- iii. Sí es ambigua.

Dos árboles diferentes para la cadena aabbab y dos derivaciones m.i.

**m.i:**  $S \rightarrow aSbS \rightarrow aaSbSbS \rightarrow aabSbS \rightarrow aabbS \rightarrow aabbaSbS \rightarrow aabbabS \rightarrow aabbab$

**m.i:**  $S \rightarrow aSbS \rightarrow aaSbSbS \rightarrow aabSbS \rightarrow aabbSaSbS \rightarrow aabbaSbS \rightarrow aabbabS \rightarrow aabbab$



[illegible]
$$\mathbf{m.i: S \rightarrow aSbS \rightarrow abS \rightarrow abbSaS \rightarrow abbaS \rightarrow abbaS} \\ \rightarrow abbaaSbS \rightarrow abbaabS \rightarrow abbaab$$

c.  $S \rightarrow (L) \mid a$   
 $L \rightarrow L, S \mid S$       cadena:  $((a, a), a, (a))$

i. **m.i:**  $S \rightarrow (L) \rightarrow (L, S) \rightarrow (L, S, S) \rightarrow (S, S, S) \rightarrow ((L), S, S) \rightarrow ((L, S), S, S) \rightarrow ((S, S), S, S) \rightarrow ((a, S), S, S) \rightarrow ((a, a), S, S) \rightarrow ((a, a), a, S) \rightarrow ((a, a), a, (L)) \rightarrow ((a, a), a, (S)) \rightarrow ((a, a), a, (a))$

ii. **m.d:**  $S \rightarrow (L) \rightarrow (L, S) \rightarrow (L, (L)) \rightarrow (L, (S)) \rightarrow (L, (a)) \rightarrow (L, S, (a)) \rightarrow (L, a, (a)) \rightarrow (S, a, (a)) \rightarrow ((L), a, (a)) \rightarrow ((L, S), a, (a)) \rightarrow ((L, a), a, (a)) \rightarrow ((S, a), a, (a)) \rightarrow ((a, a), a, (a))$

iii. No es ambigua. Para probarlo se tendrían que usar otros métodos más formales, como inducción matemática.

- iv. Cadenas en forma de tuplas de Python, separadas por comas y paréntesis.

3. Dada la gramática:

$S \rightarrow f(P)$

$P \rightarrow x$

$P \rightarrow P, x$

$P \rightarrow x, P$

a. Describa el lenguaje definido por ella.

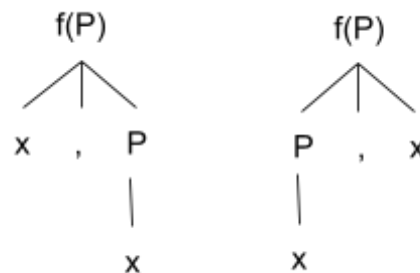
Definiciones de funciones de una a más variables.

b. Construya una derivación para  $f(x, x, x)$

**mi:**  $S \rightarrow f(P) \rightarrow f(P, x) \rightarrow$   
 $f(x, P, x) \rightarrow f(x, x, x)$

c. ¿Es una gramática ambigua? Argumente con un ejemplo.

Sí es ambigua, dos árboles de derivación para  $f(x, x)$ :



4. Muestre con un ejemplo que la siguiente gramática es ambigua

$S \rightarrow a$

$S \rightarrow (L)$

$L \rightarrow S$

$L \rightarrow L * L$

$L \rightarrow L\#$

**m.i:**  $S \rightarrow (L) \rightarrow (L * L) \rightarrow (L\# * L) \rightarrow (L * L\# * L) \rightarrow (S * L\# * L) \rightarrow$   
 $(a * L\# * L) \rightarrow (a * S\# * L) \rightarrow (a * a\# * L) \rightarrow (a * a\# * S) \rightarrow (a * a\# * a)$

**m.i:**  $S \rightarrow (L) \rightarrow (L * L) \rightarrow (L * L * L) \rightarrow (S * L * L) \rightarrow (a * L * L) \rightarrow$   
 $(a * L\# * L) \rightarrow (a * S\# * L) \rightarrow (a * a\# * L) \rightarrow (a * a\# * S) \rightarrow (a * a\# * a)$

Sí es ambigua, porque existen dos derivaciones m.i. para la misma cadena (a#a#a)

5. Construya la versión no-ambigua de la siguiente gramática

$\text{exp} \rightarrow \text{exp} + \text{exp}$

$\text{exp} \rightarrow \text{exp} - \text{exp}$

$\text{exp} \rightarrow \text{exp} * \text{exp}$

$\text{exp} \rightarrow \text{exp} / \text{exp}$

$\text{exp} \rightarrow \text{num}$

$\text{exp} \rightarrow (\text{exp})$

construya el árbol sintáctico para  $2 + 3 * 4$

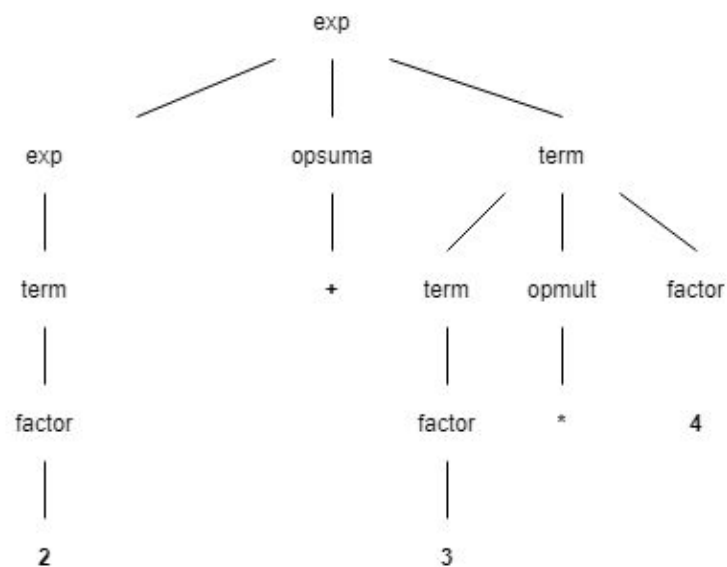
$\text{exp} \rightarrow \text{exp opsuma term} \mid \text{term}$

$\text{opsuma} \rightarrow + \mid -$

$\text{term} \rightarrow \text{term opmult factor} \mid \text{factor}$

$\text{opmult} \rightarrow * \mid /$

$\text{factor} \rightarrow (\text{exp}) \mid \text{num}$



6. Dada la gramática

$$A \rightarrow (A)A \mid \epsilon$$

escriba un pseudocódigo para el análisis sintáctico de esta gramática mediante el método descendente recursivo

```
procedure A:
begin
    case token of
        (:    match(());
        A;
        match());
        A;
         $\epsilon$ : match( $\epsilon$ );

    else error;

    end case;
end A;
```