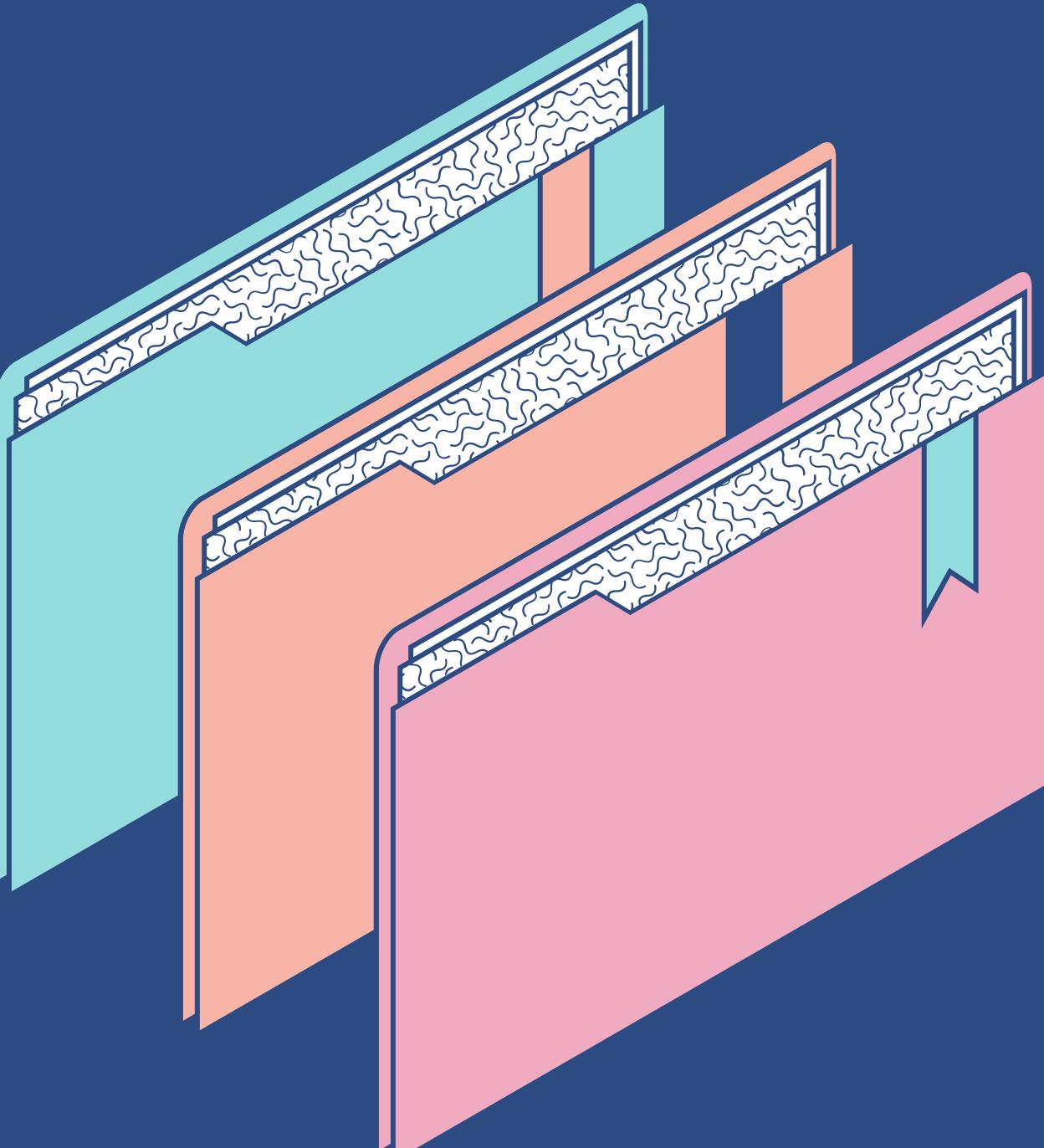




PRIMER SEMESTRE - 2023

Organización de Lenguajes y Compiladores 2

Clase 2 - Traducción dirigida por la sintaxis

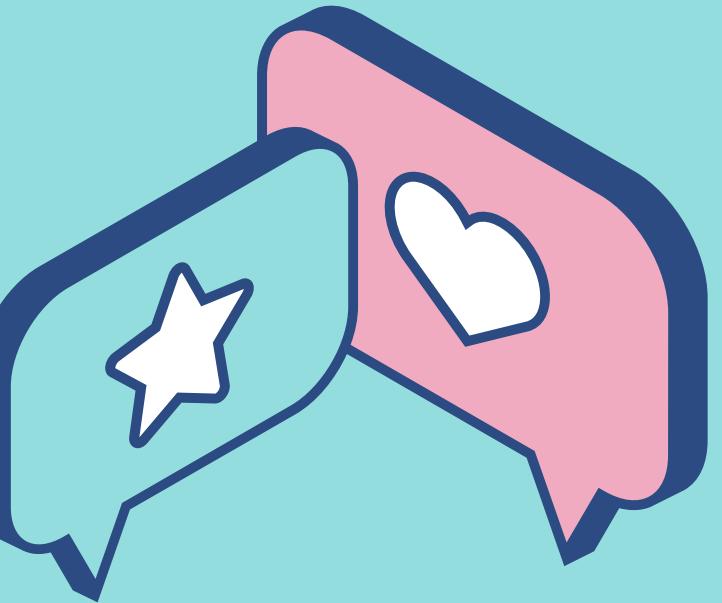


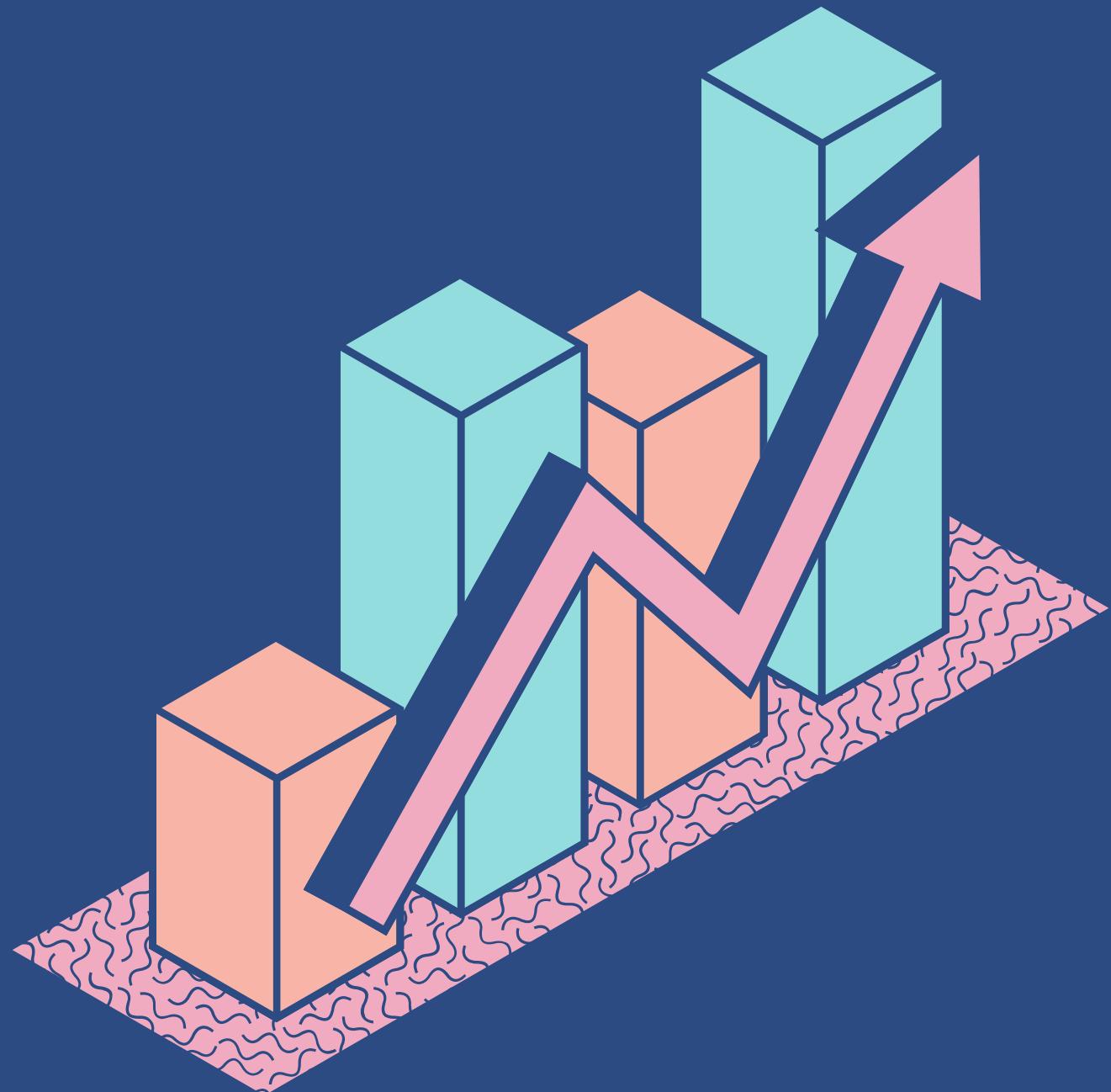
Agenda

TEMAS CLAVE QUE SE DEBATIRÁN
EN ESTA PRESENTACIÓN

- Anuncios
- Lista de asistencia
- Traducción dirigida por la sintaxis
- Ejemplo teórico
- Ejemplo práctico

Anuncios...





Lista de asistencia...



Traducción dirigida por la sintaxis

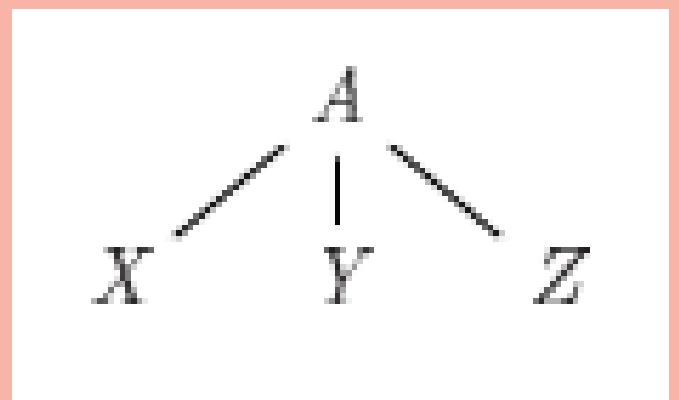
El análisis se organiza de acuerdo con la “sintaxis” del lenguaje que se va a compilar.

La sintaxis de un lenguaje de programación describe el formato apropiado de sus programas, mientras que la semántica del lenguaje define lo que sus programas significan.

Árbol de análisis sintáctico

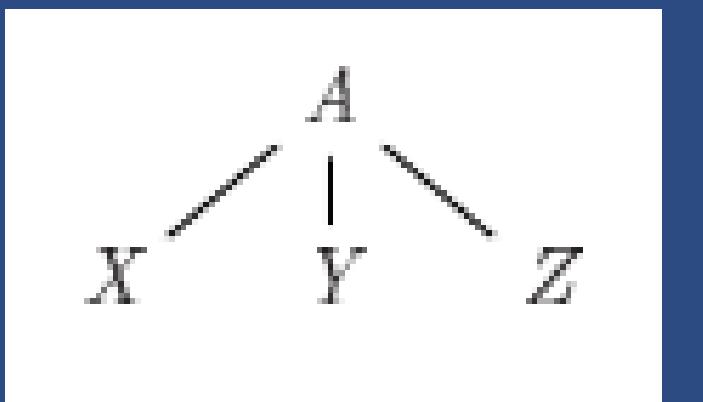
Un árbol de análisis sintáctico muestra, en forma gráfica, la manera en que el símbolo inicial de una gramática deriva a una cadena en el lenguaje.

Un árbol de análisis sintáctico o bien árbol sintáctico concreto es tambien llamado concrete syntax tree (CST)



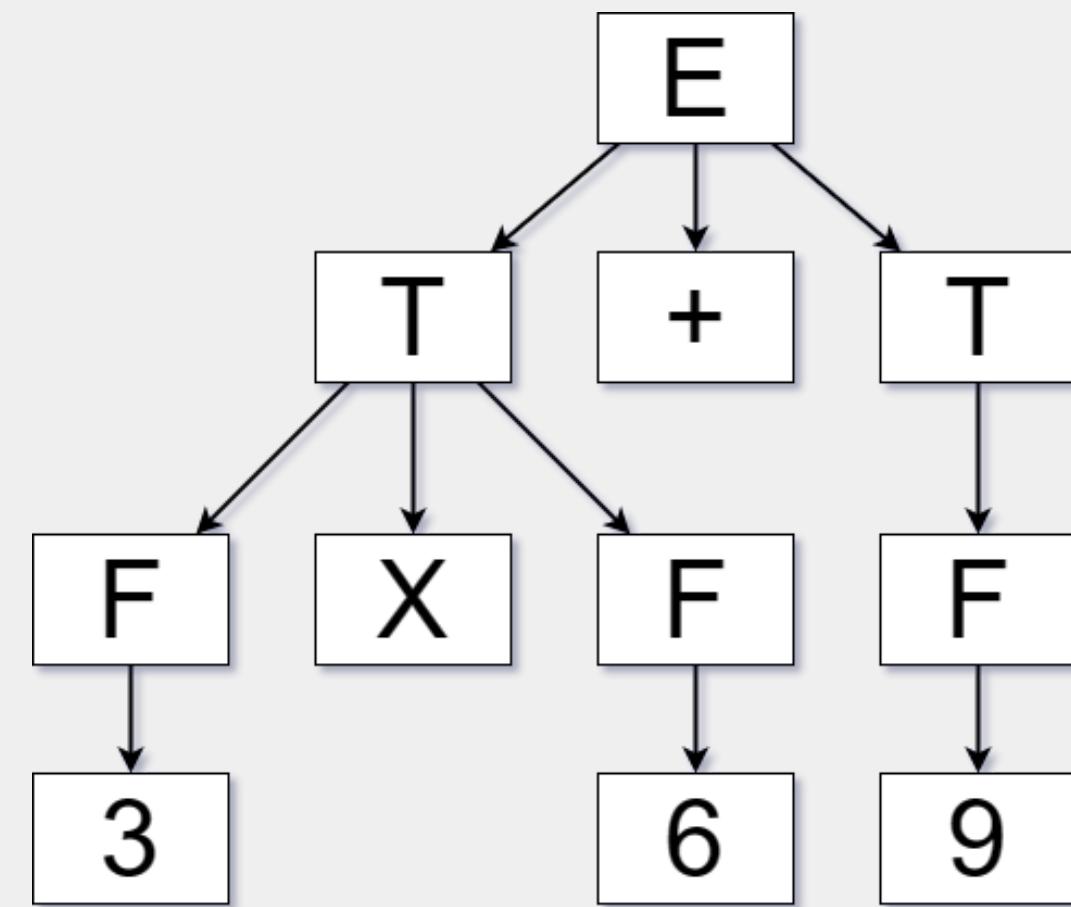
Árbol de sintaxis abstracta

Un árbol de análisis sintáctico es un CST y nos muestra exactamente como fue derivado nuestro texto. Sin embargo una estructura mas útil es el abstract syntax tree (AST) que retiene únicamente la estructura esencial de la entrada.

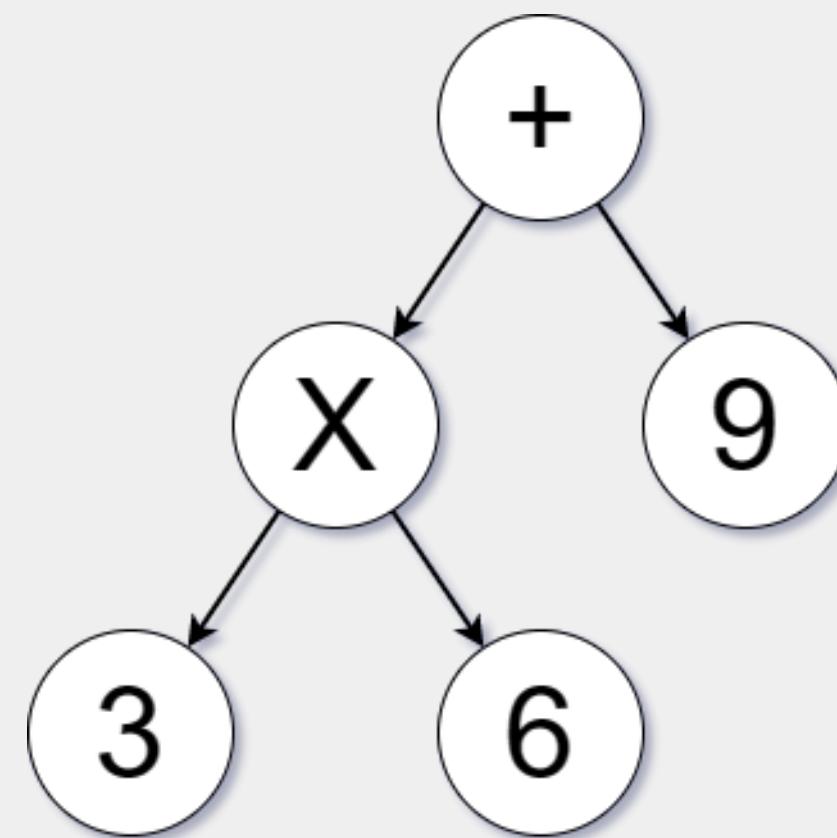




Árbol de análisis sintáctico (CST)



Árbol de sintaxis abstracta (AST)



GRAMATICAS LIBRES DEL CONTEXTO

$$G = (V, T, P, S)$$

Lenguaje Libre de Contexto

Conjunto de Variables / No terminales

Conjunto de Terminales

Conjunto de Producción

Símbolo de Inicio

ARTEAGA MEDRANDA LUIS FERNANDO

Recursividad por la izquierda

Se produce una producción “recursivas por la izquierda” cuando el símbolo más a la izquierda del cuerpo de la producción es el mismo que el no terminal en el encabezado de la producción:

$$\text{expr} \rightarrow \text{expr} + \text{term}$$

Podemos eliminar una producción recursiva por la izquierda, reescribiendo la producción problemática, por ejemplo:

$$A \rightarrow A\alpha \mid \beta$$

Podemos eliminar el efecto recursivo si reescribimos las producciones para A de la siguiente manera, usando un nuevo no terminal R:

$$\begin{array}{lcl} A & \rightarrow & \beta R \\ R & \rightarrow & \alpha R \mid \epsilon \end{array}$$



Definición dirigida por la sintaxis

Es una gramática libre de contexto de la mano de varios atributos y reglas. Los atributos se asocian a los símbolos gramaticales y las reglas a las producciones.

Producción

SUMA \rightarrow E + E



Reglas semánticas

SUMA.valor = E1.valor + E2.valor
SUMA.linea = E1.linea
SUMA.columna = E1.columna

ASIGN \rightarrow ID = E



ASIGN.id = ID.valor
ASIGN.valor = E.valor
ASIGN.linea = ID.linea
ASIGN.columna = ID.columna

Tipos de atributos

Existen dos tipos de atributos: Sintetizados y heredados.



Atributos sintetizados

El nodo N de un árbol de análisis sintáctico se determina usando los valores de los atributos del hijo de N y del mismo N



Atributos Heredados

El nodo N se define solamente en términos de los valores de los atributos del padre de N y de los hermanos.





Ejemplo teórico

Recursividad

$S \rightarrow E_n$

$E \rightarrow E + F$

$E \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow \text{digit}$

$$A \rightarrow A\alpha \mid \beta$$

$$\begin{array}{lcl} A & \rightarrow & \beta R \\ R & \rightarrow & \alpha R \mid \epsilon \end{array}$$



Recursividad

$S \rightarrow E_n$

$E \rightarrow E + F$

$E \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow \text{digit}$

$$A \rightarrow A\alpha \mid \beta$$

$$\begin{array}{lcl} A & \rightarrow & \beta R \\ R & \rightarrow & \alpha R \mid \epsilon \end{array}$$

$S \rightarrow E_n$

$A = E$

$\alpha = + F$

$\beta = F$

Recursividad

$S \rightarrow E_n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$

$$\frac{A \rightarrow A\alpha \mid \beta}{\begin{array}{l} A \rightarrow \beta R \\ R \rightarrow \alpha R \mid \epsilon \end{array}}$$

$S \rightarrow E_n$
 $E \rightarrow F \ E'$

$A = E$
 $\alpha = + F$
 $\beta = F$

Recursividad

$S \rightarrow E_n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$

$$\begin{array}{c} A \rightarrow A\alpha \mid \beta \\ \\ A \rightarrow \beta R \\ R \rightarrow \alpha R \mid \epsilon \end{array}$$

$S \rightarrow E_n$
 $E \rightarrow F E'$
 $E' \rightarrow + F E'$

$$\begin{array}{l} A = E \\ \alpha = + F \\ \beta = F \end{array}$$

Recursividad

$S \rightarrow E_n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$

$$\begin{array}{c} A \rightarrow A\alpha \mid \beta \\ \\ A \rightarrow \beta R \\ R \rightarrow \alpha R \mid \epsilon \end{array}$$

$S \rightarrow E_n$
 $E \rightarrow F E'$
 $E' \rightarrow + F E'$
 $E' \rightarrow \epsilon$

$A = E$
 $\alpha = + F$
 $\beta = F$

Recursividad

$S \rightarrow E_n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$

$$\begin{array}{c} A \rightarrow A\alpha \mid \beta \\ \\ A \rightarrow \beta R \\ R \rightarrow \alpha R \mid \epsilon \end{array}$$

$S \rightarrow E_n$
 $E \rightarrow F E'$
 $E' \rightarrow + F E'$
 $E' \rightarrow \epsilon$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$

$A = E$
 $\alpha = + F$
 $\beta = F$

AST

$$3 + 6 + (2 + 3)n$$

$S \rightarrow E^n$

$E \rightarrow E + F$

$E \rightarrow F$

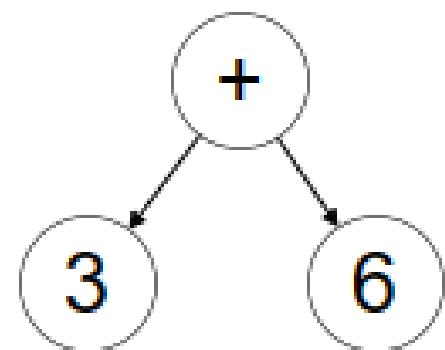
$F \rightarrow (E)$

$F \rightarrow \text{digit}$

AST

$$3 + 6 + (2 + 3)n$$

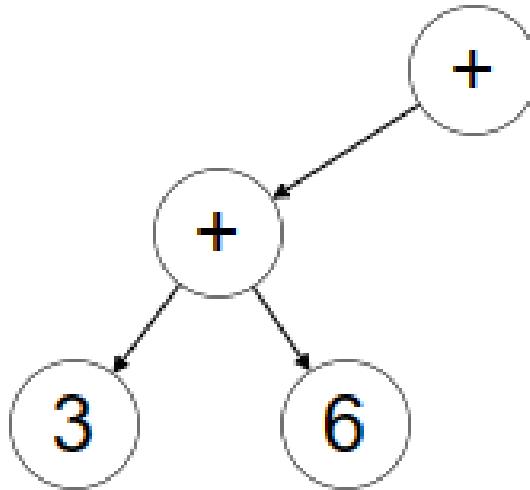
$S \rightarrow E^n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$



AST

$$3 + 6 + (2 + 3)n$$

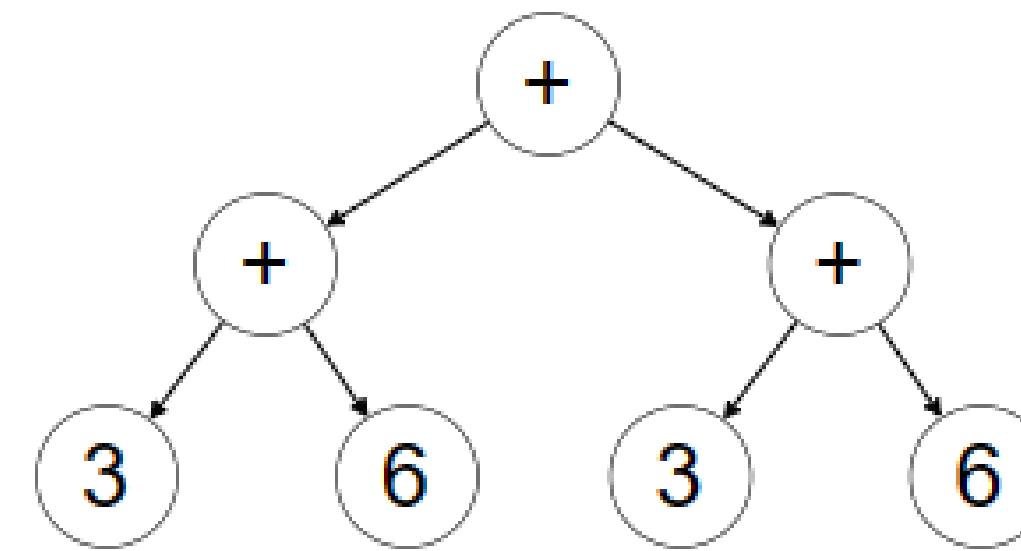
$S \rightarrow E^n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$



AST

$$3 + 6 + (2 + 3)n$$

$S \rightarrow E^n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$



CST

$$3 + 6 + (2 + 3)n$$

$S \rightarrow E^n$

$E \rightarrow E + F$

$E \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow \text{digit}$

CST

$$3 + 6 + (2 + 3)n$$

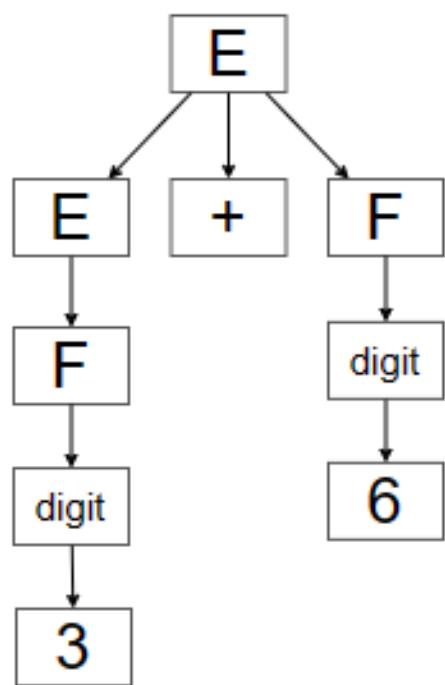
$S \rightarrow E^n$

$E \rightarrow E + F$

$E \rightarrow F$

$F \rightarrow (E)$

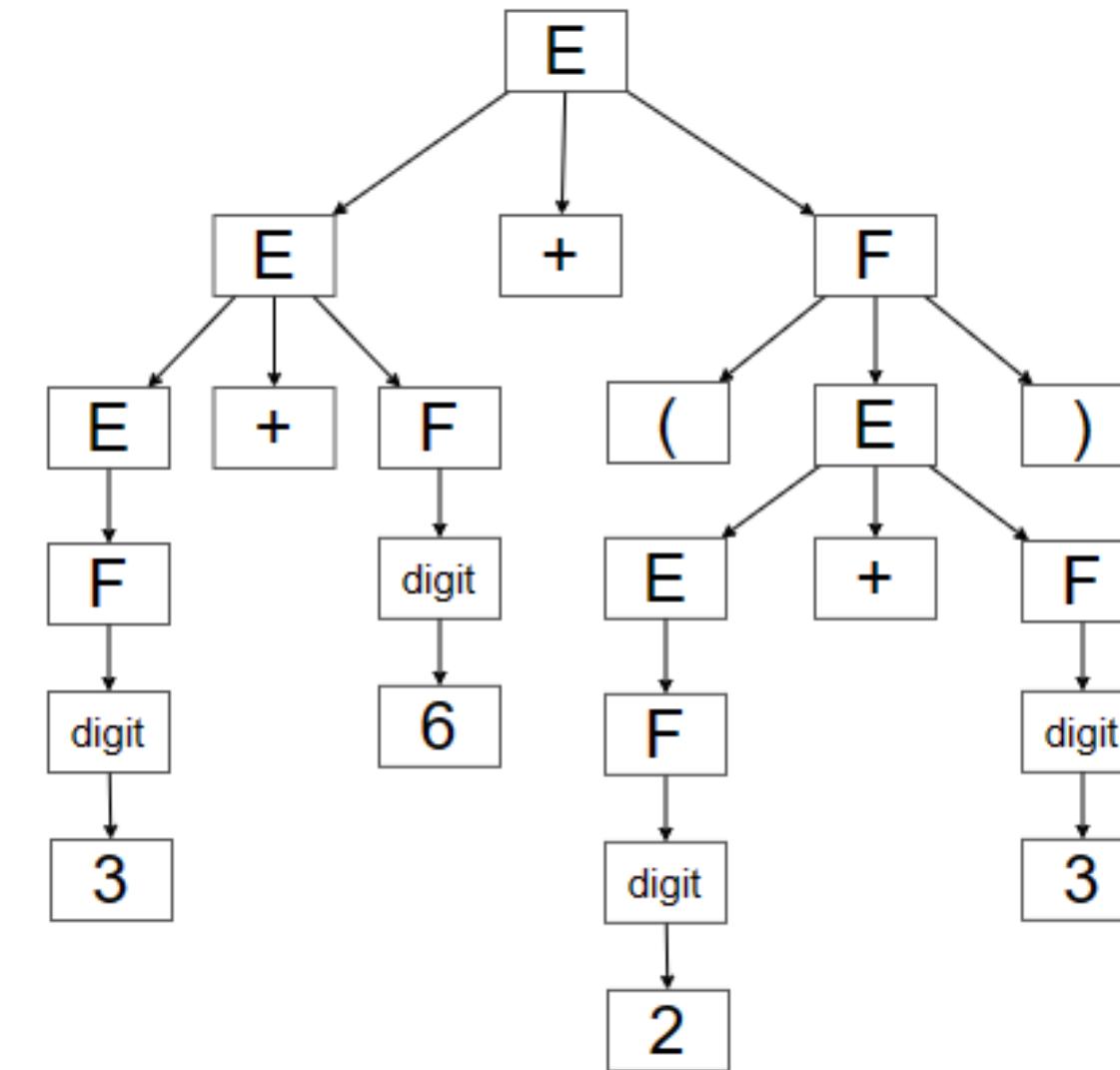
$F \rightarrow \text{digit}$



CST

$$3 + 6 + (2 + 3)n$$

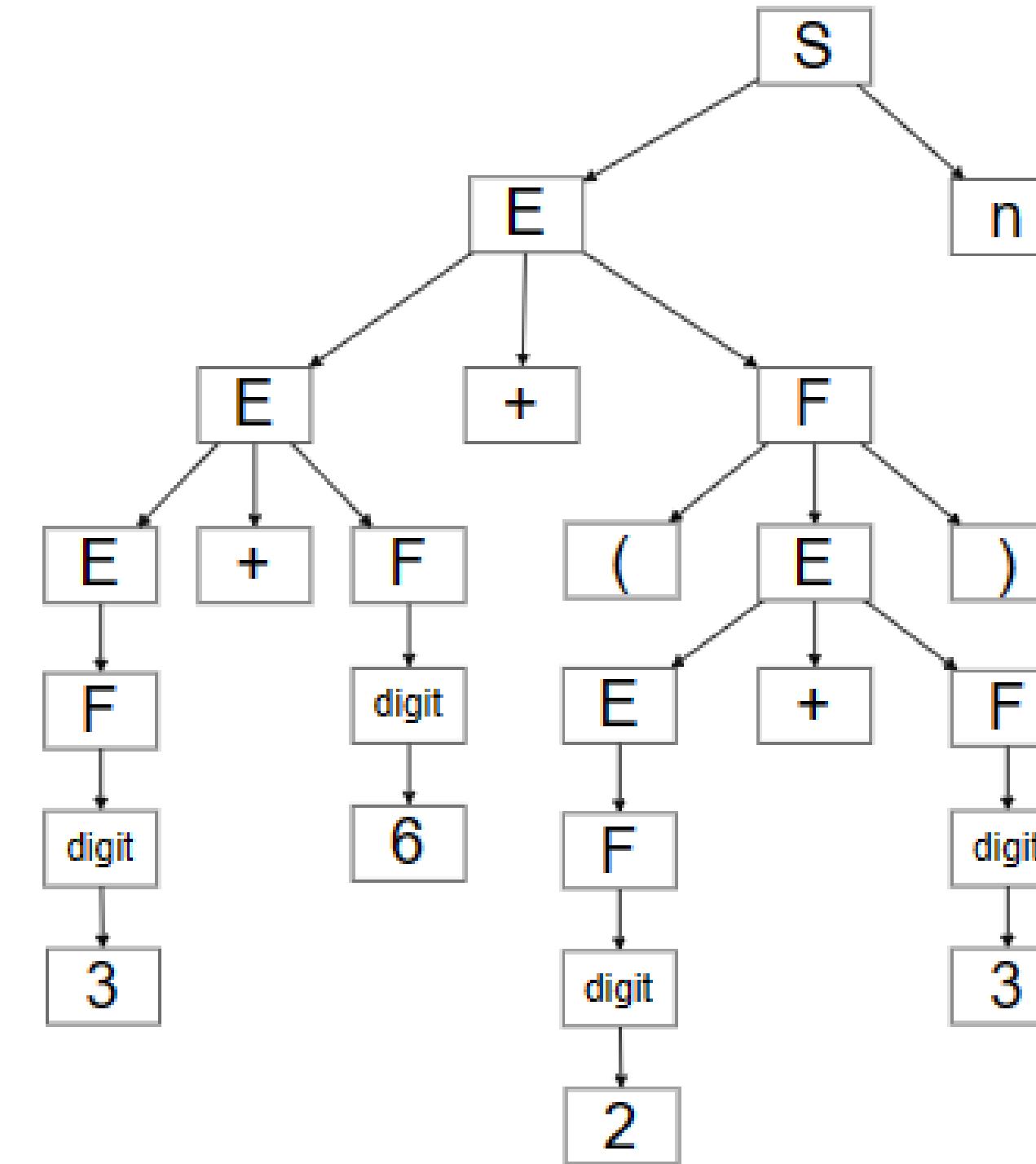
$S \rightarrow E^n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$



CST

$$3 + 6 + (2 + 3)n$$

$S \rightarrow E^n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$



Definición dirigida por la sintaxis

$$3 + 6 + (2 + 3)n \longrightarrow 14$$

$S \rightarrow E_n$
 $E \rightarrow E + F$
 $E \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow \text{digit}$

obj {
 obj.val
 obj.line
 obj.col
}

Definición dirigida por la sintaxis

Entrada: 3 + 6 + (2 + 3)n

Salida: 14

$S \rightarrow E_n$	
$E \rightarrow E + F$	
$E \rightarrow F$	
$F \rightarrow (E)$	
$F \rightarrow \text{digit}$	<code>F.val = parseInt(digit); F.line = digit.line; F.col = digit.col;</code>

Definición dirigida por la sintaxis

Entrada: $3 + 6 + (2 + 3)n$

Salida: 14

$S \rightarrow E_n$	
$E \rightarrow E + F$	
$E \rightarrow F$	
$F \rightarrow (E)$	$F.val = E.val;$ $F.line = E.line;$ $F.col = E.col;$
$F \rightarrow digit$	$F.val = parseInt(digit);$ $F.line = digit.line;$ $F.col = digit.col;$

Definición dirigida por la sintaxis

Entrada: $3 + 6 + (2 + 3)n$

Salida: 14

$S \rightarrow E_n$	
$E \rightarrow E + F$	
$E \rightarrow F$	$E.val = F.val;$ $E.line = F.line;$ $E.col = F.col;$
$F \rightarrow (E)$	$F.val = E.val;$ $F.line = E.line;$ $F.col = E.col;$
$F \rightarrow digit$	$F.val = parseInt(digit);$ $F.line = digit.line;$ $F.col = digit.col;$

Definición dirigida por la sintaxis

Entrada: $3 + 6 + (2 + 3)n$

Salida: 14

$S \rightarrow E_n$	
$E \rightarrow E + F$	$E.val = E1.val + F.val;$ $E.line = E1.line;$ $E.col = E1.col;$
$E \rightarrow F$	$E.val = F.val;$ $E.line = F.line;$ $E.col = F.col;$
$F \rightarrow (E)$	$F.val = E.val;$ $F.line = E.line;$ $F.col = E.col;$
$F \rightarrow digit$	$F.val = parseInt(digit);$ $F.line = digit.line;$ $F.col = digit.col;$

Definición dirigida por la sintaxis

Entrada: 3 + 6 + (2 + 3)n

Salida: 14

$S \rightarrow E_n$	<code>Print(E.val);</code>
$E \rightarrow E + F$	<code>E.val = E1.val + F.val; E.line = E1.line; E.col = E1.col;</code>
$E \rightarrow F$	<code>E.val = F.val; E.line = F.line; E.col = F.col;</code>
$F \rightarrow (E)$	<code>F.val = E.val; F.line = E.line; F.col = E.col;</code>
$F \rightarrow \text{digit}$	<code>F.val = parseInt(digit); F.line = digit.line; F.col = digit.col;</code>

Atributos sintetizados y heredados

$S \rightarrow E_n$

$E \rightarrow E + F$

$E \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow \text{digit}$

$3 + 6 + (2 + 3)n$

Ejemplo Práctico

```
double dblTemp;
bool again = true;

while (again) {
    iN = -1;
    again = false;
    getline(cin, sInput);
    system("cls");
    iLength = sInput.length();
    if (iLength < 4) {
        again = true;
        continue;
    } else if (sInput[iLength - 3] != '0') {
        again = true;
        continue;
    }
    while (++iN < iLength) {
        if (isdigit(sInput[iN])) {
            continue;
        } else if (iN == (iLength - 3)) {
```





PRIMER SEMESTRE - 2023

Gracias por su
atención..



Clase 1 - Traducción dirigida por la sintaxis