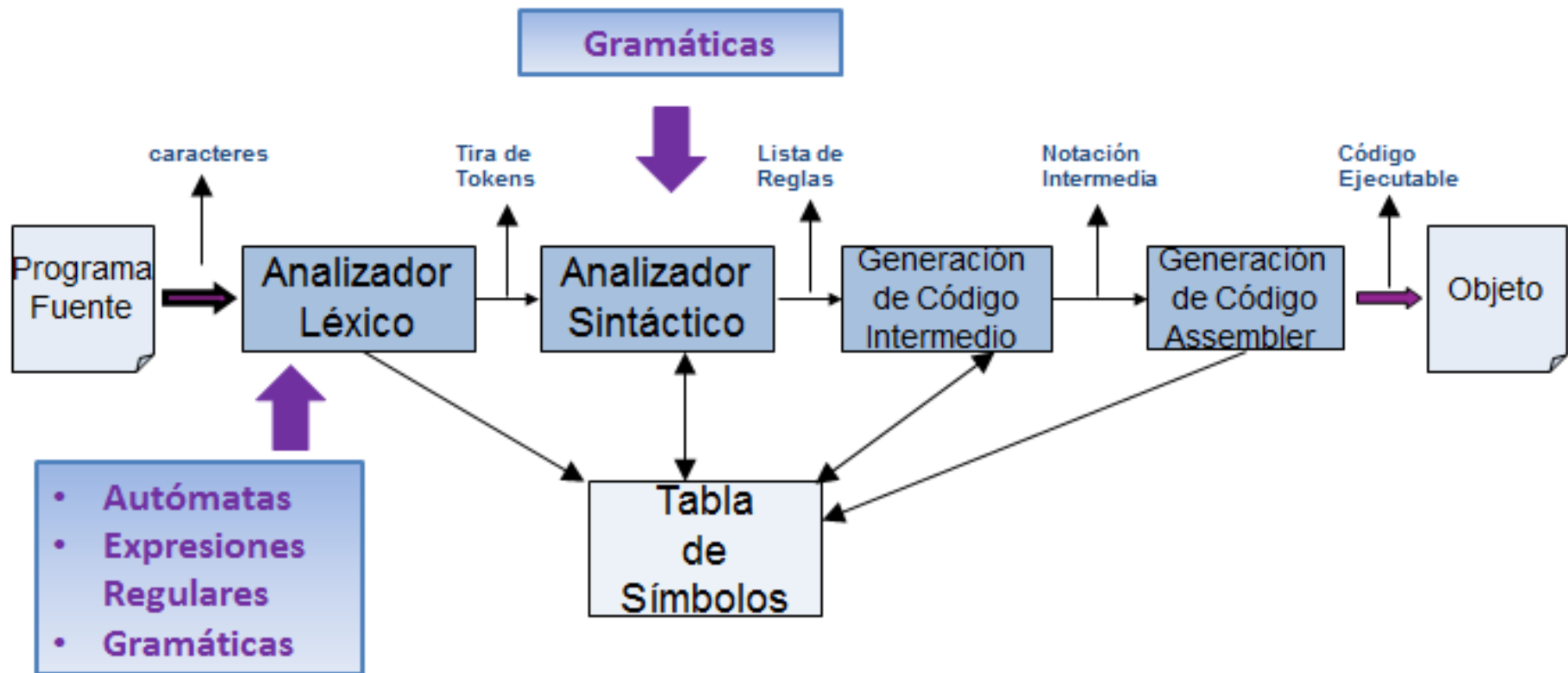


Lenguajes y Compiladores

Analizador Sintáctico
Gramáticas



Lenguajes de Programación



Analizador Sintáctico

Analiza sintácticamente el
programa de usuario

Analizador Sintáctico

- En general el analizador sintáctico es una función que lee el programa del usuario intentando cumplir las reglas gramaticales del lenguaje
- Su función es encontrar un conjunto de reglas que formen el programa de usuario
- Si no lo logra habrá un error de sintaxis

Analizador Sintáctico

El analizador sintáctico se modela manualmente o con **gramáticas libres de contexto (reglas)**.

Existen herramientas como YACC, BISON, JCUP que se ocupan de armar la función, a las que sólo es necesario escribirles las reglas correspondientes y un analizador léxico.

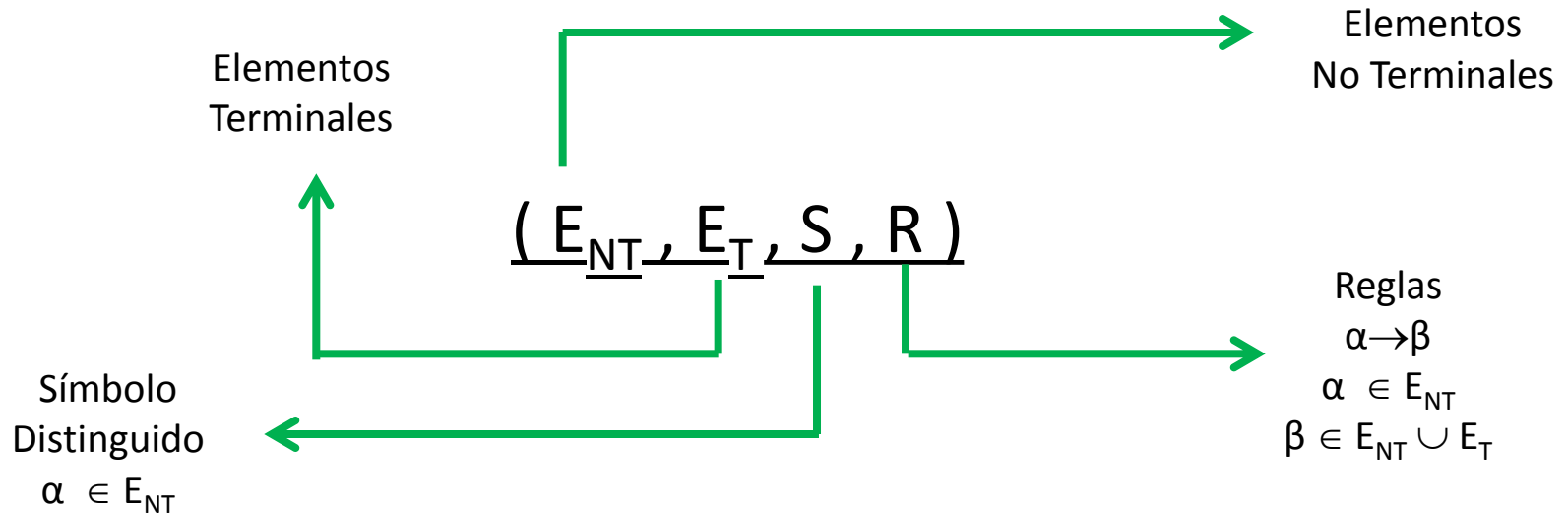
El resultado que entregan es una lista de reglas

Es necesario conocer el método de análisis que utilizan estas herramientas

Sintaxis

La **sintaxis** de un LP se describe a través de gramáticas libres de contexto.

Las gramáticas se conforman de cuatro elementos



- Los ***elementos no terminales*** se definen utilizando los elementos terminales a través de ***reglas*** gramaticales
- Los ***elementos terminales*** no se definen.
- Se define un ***elemento distinguido*** o “***Start symbol***” que pertenece al conjunto de no terminales.
- Las ***reglas*** son declarativas (no importa el orden en que fueron definidas) y pueden ser recursivas.

Sintaxis

$\langle \text{programa} \rangle \rightarrow \langle \text{bloque} \rangle$

$\langle \text{bloque} \rangle \rightarrow \langle \text{sent} \rangle$ **regla recursiva**

$\langle \text{bloque} \rangle \rightarrow \langle \text{bloque} \rangle \langle \text{sent} \rangle$

$\langle \text{sent} \rangle \rightarrow \text{WRITE CTE_S}$

(E_{NT}, E_T, S, R)

$E_{NT} = \{ \langle \text{programa} \rangle, \langle \text{bloque} \rangle \langle \text{sent} \rangle, \}$

$E_T = \{ \text{WRITE}, \text{CTE_S} \}$

$S = \{ \langle \text{programa} \rangle \}$

R

WRITE y CTE_S son tokens definidos en el Analizador Léxico

Sintaxis

Rekursividad en las Reglas

- Las reglas pueden ser recursivas a izquierda

<bloque> → <bloque> <sent>

- Las reglas pueden ser recursivas a derecha

<bloque> → <sent> <bloque>

Notación

	Gramáticas	BNF
E_{NT}	Letras Mayúsculas	Entre < >
E_T	Letras Minúsculas	Directamente
S	-	-
R	$\alpha \rightarrow \beta$	$\langle \alpha \rangle ::= \beta$

Sintaxis

Estructurar de lo más general a lo particular

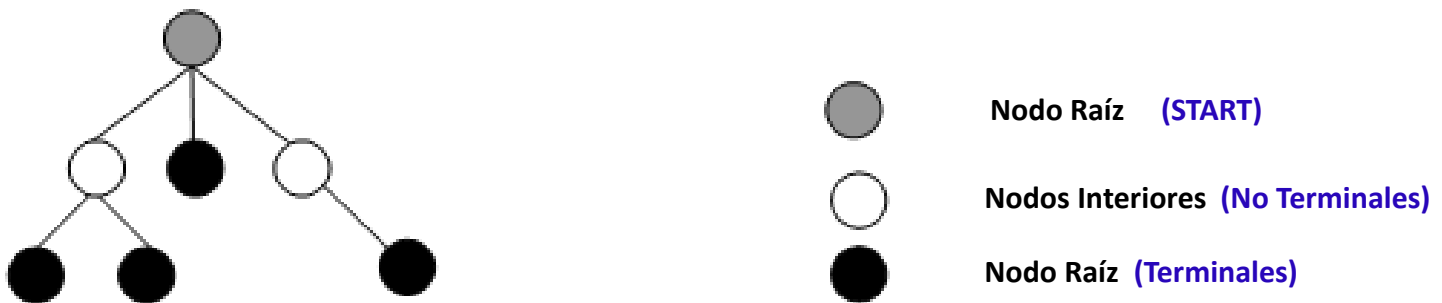
<programa> ::= <sentencia>
<programa> ::= <programa> <sentencia>
<sentencia> ::= <asignación>
<sentencia> ::= <iteración>
<sentencia> ::= <selección>
<asignación> ::= id := <expresión>
<selección> ::= if <condición> then <programa> else <programa>
<selección> ::= if <condición> then <programa>
<iteración> ::= while <condición> begin <programa> end
<condición> ::= <comparación>
<condición> ::= <condición> and <comparación>
<condición> ::= <condición> or <comparación>
<comparación> ::= <expresión> <comparador> <expresión>
<comparador> ::= >= | <= | < | > | ==
<expresión> ::= <expresión> + <termino> | <expresión> - <termino> |
<termino>
<termino> ::= <termino> * <factor> | <termino> / <factor> | <factor>
<factor> ::= (<expresión>) | id | cte

Árbol de parsing o Árbol de derivación

Permite mostrar gráficamente cómo se puede derivar cualquier sentencia de un lenguaje a partir de la gramática que represente a dicho lenguaje.

Un **árbol** es un grafo dirigido a cíclico con las siguientes propiedades:

- El nodo raíz está rotulado con el símbolo distinguido de la gramática
- Cada hoja corresponde a un símbolo terminal o un símbolo no terminal
- Cada nodo interior corresponde a un símbolo no terminal



Lectura de Izquierda a Derecha de Nodos Raíz (Terminales)

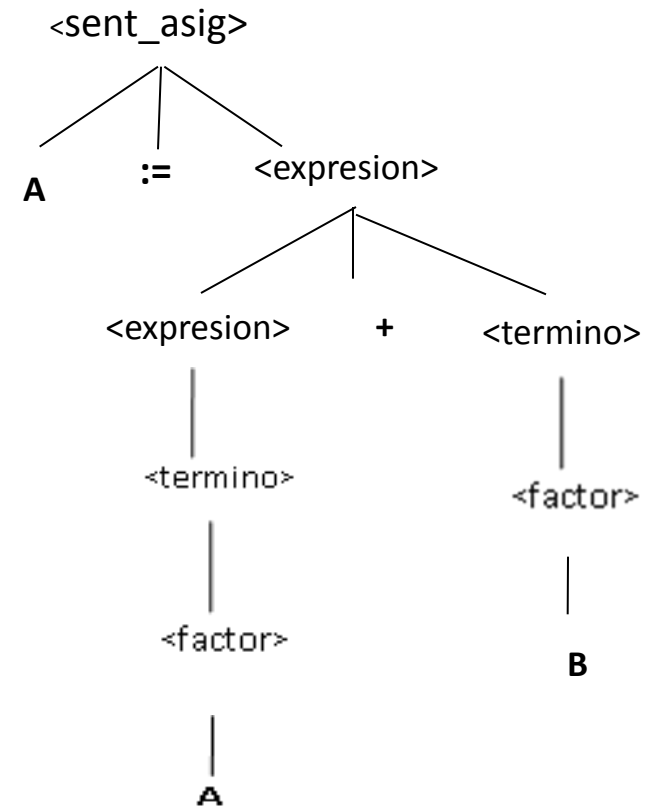
Árbol de parsing o Árbol de derivación

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

Árbol de parsing o Árbol de derivación

Sentencia $A := A + B$

$\langle \text{sent_asig} \rangle ::= \text{VAR} := \langle \text{expresión} \rangle$
 $\langle \text{expresión} \rangle ::= \langle \text{expresión} \rangle + \langle \text{término} \rangle$
 $\langle \text{expresión} \rangle ::= \langle \text{expresión} \rangle - \langle \text{término} \rangle$
 $\langle \text{expresión} \rangle ::= \langle \text{termino} \rangle$
 $\langle \text{término} \rangle ::= \langle \text{término} \rangle * \langle \text{factor} \rangle$
 $\langle \text{término} \rangle ::= \langle \text{término} \rangle / \langle \text{factor} \rangle$
 $\langle \text{término} \rangle ::= \langle \text{factor} \rangle$
 $\langle \text{factor} \rangle ::= (\langle \text{expresión} \rangle) \mid \text{VAR} \mid \text{NUM}$



Ambigüedad

Quiero
Reconocer

id + id * cte

$\langle \text{expresion} \rangle ::= \langle \text{expresion} \rangle + \langle \text{expresion} \rangle$

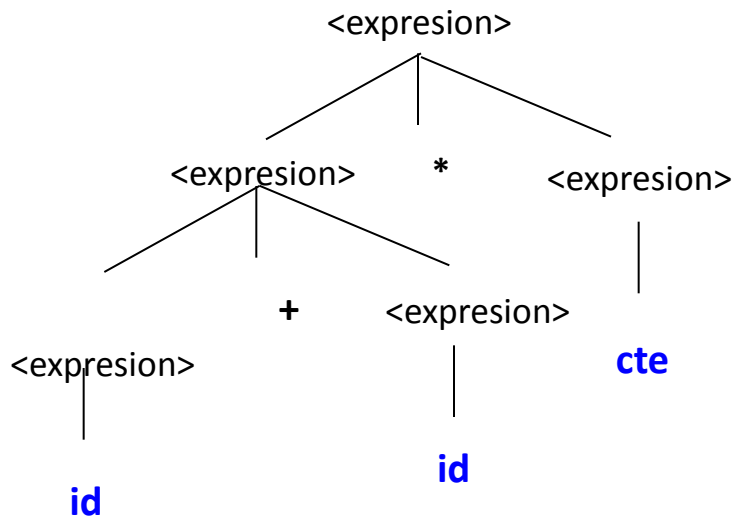
$\langle \text{expresion} \rangle ::= \langle \text{expresion} \rangle - \langle \text{expresion} \rangle$

$\langle \text{expresion} \rangle ::= \langle \text{expresion} \rangle * \langle \text{expresion} \rangle$

$\langle \text{expresion} \rangle ::= \langle \text{expresion} \rangle / \langle \text{expresion} \rangle$

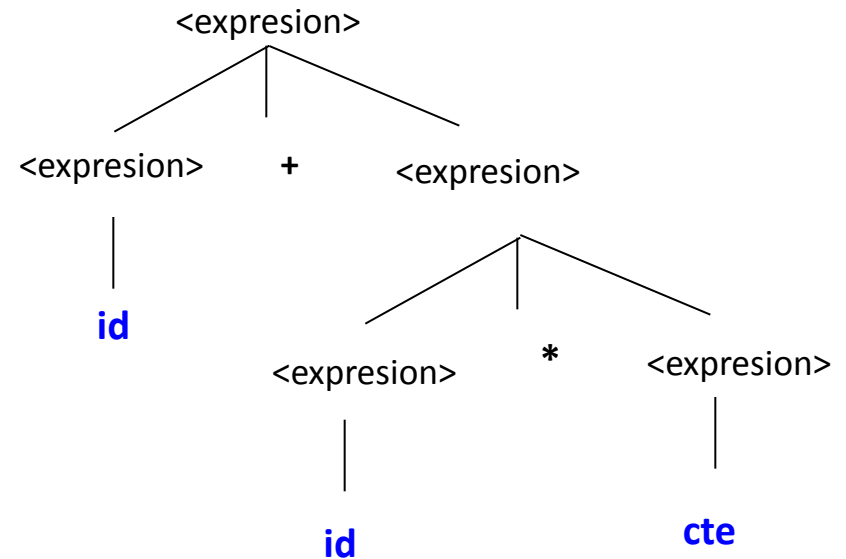
$\langle \text{expresion} \rangle ::= \text{cte}$

$\langle \text{expresion} \rangle ::= \text{id}$



$a + b * 2 = 22$

suponiendo $a=4$ $b=7$



$a + b * 2 = 18$

- Una gramática es ambigua si existen dos árboles de derivación para la misma sentencia, hilera o programa.

SOLUCIÓN

Cambiar las reglas de la gramática

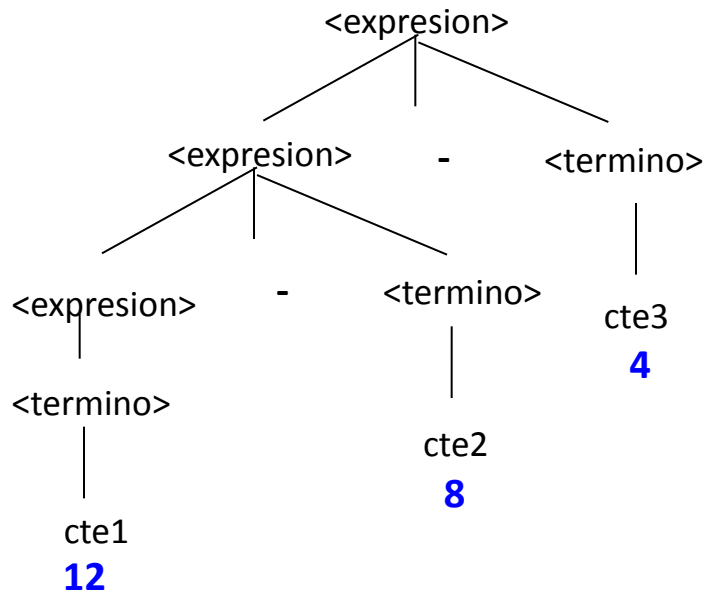
Asociatividad

Quiero Hacer

12 - 8 - 4

Gramática 1

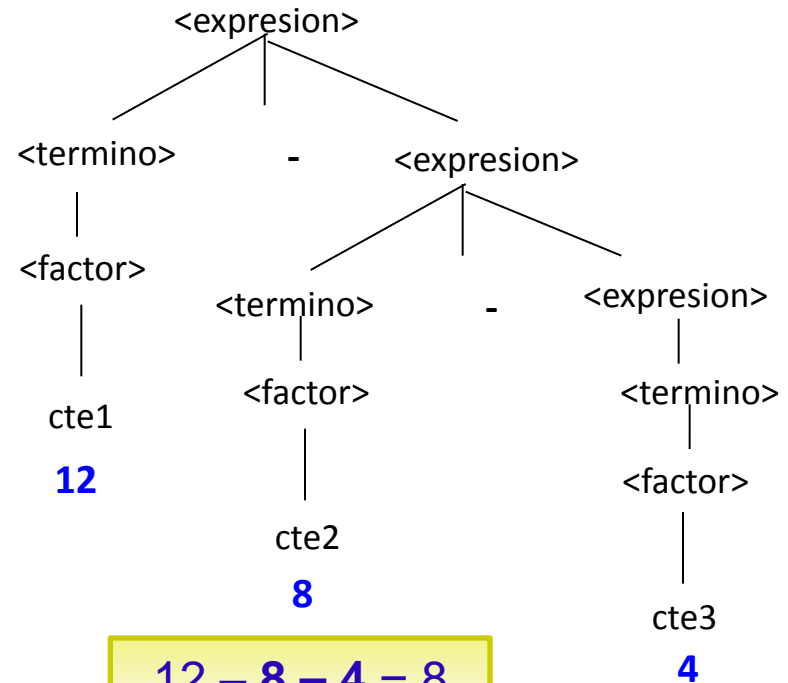
$\langle \text{expresion} \rangle ::= \langle \text{expresion} \rangle + \langle \text{termino} \rangle$
 $\langle \text{expresion} \rangle ::= \langle \text{expresion} \rangle - \langle \text{termino} \rangle$
 $\langle \text{expresion} \rangle ::= \langle \text{expresion} \rangle * \langle \text{termino} \rangle$
 $\langle \text{expresion} \rangle ::= \langle \text{expresion} \rangle / \langle \text{termino} \rangle$
 $\langle \text{expresion} \rangle ::= \langle \text{termino} \rangle$
 $\langle \text{termino} \rangle ::= \text{cte}$



$$12 - 8 - 4 = 0$$

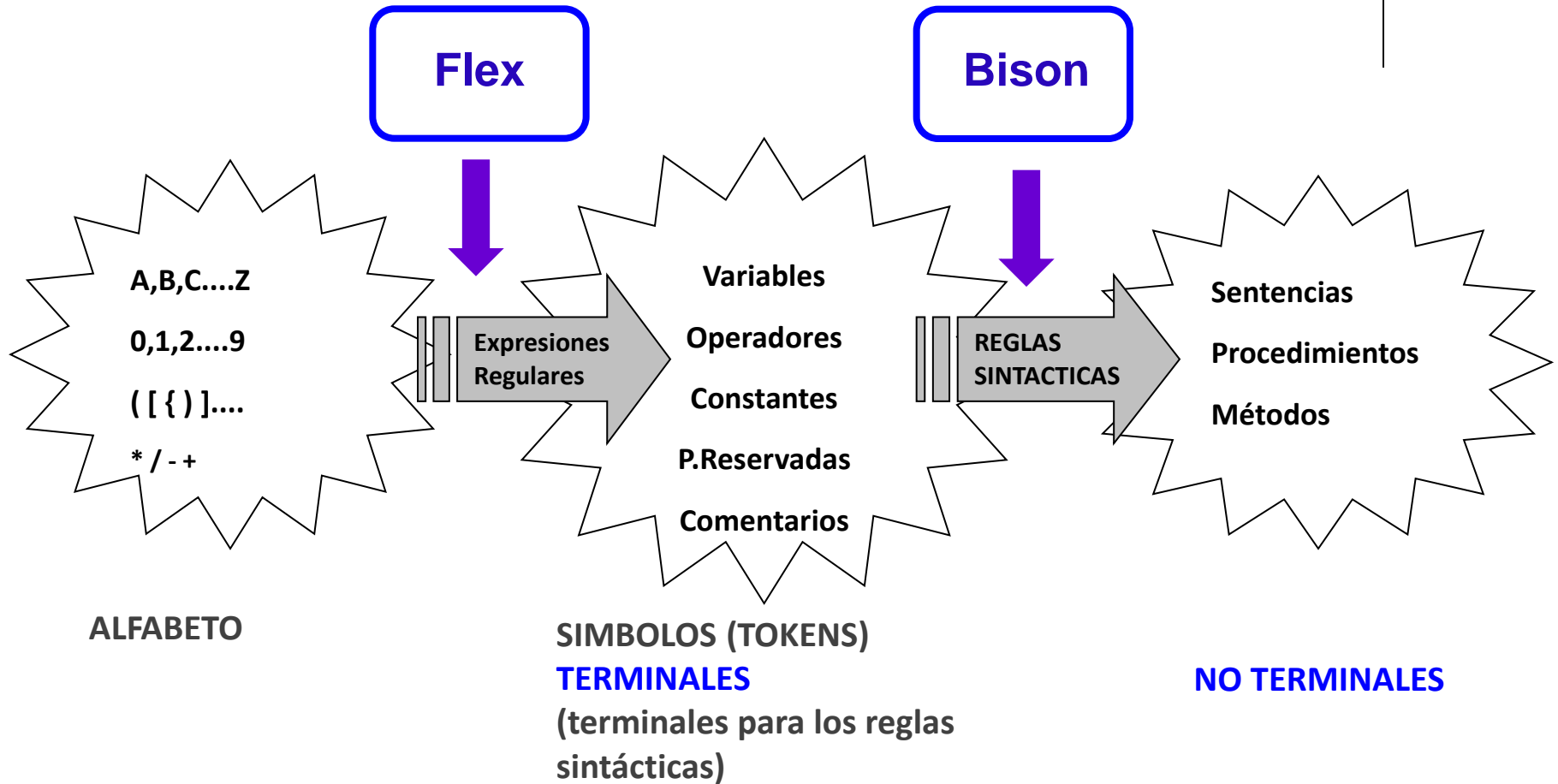
Gramática 2

$\langle \text{expresion} \rangle ::= \langle \text{termino} \rangle + \langle \text{expresion} \rangle$
 $\langle \text{expresion} \rangle ::= \langle \text{termino} \rangle - \langle \text{expresion} \rangle$
 $\langle \text{expresion} \rangle ::= \langle \text{termino} \rangle$
 $\langle \text{termino} \rangle ::= \langle \text{factor} \rangle * \langle \text{termino} \rangle$
 $\langle \text{termino} \rangle ::= \langle \text{factor} \rangle / \langle \text{termino} \rangle$
 $\langle \text{termino} \rangle ::= \langle \text{factor} \rangle$
 $\langle \text{factor} \rangle ::= \text{cte}$



$$12 - 8 - 4 = 8$$

Sintaxis



Sintaxis

¿Preguntas?