# Conceptos de Lenguajes de Programación

SINTAXIS

# SINTAXIS

Un lenguaje de programación es una notación formal para describir algoritmos a ser ejecutados en una computadora

# SINTAXIS

**Sintáxis:** Conjunto de reglas que definen como componer letras, dígitos y otros caracteres para formar los programas

# SINTAXIS

La sintáxis establece reglas que definen cómo deben combinarse las componentes básicas, llamadas “word”, para formar sentencias y programas.

**Elementos de la sintáxis:**

Alfabeto o conjunto de caracteres

- identificadores
- Operadores
- Palabra clave y palabra reservada
- Comentarios y uso de blancos

# SINTAXIS

¿Cómo definir la sintáxis?

Se necesita una descripción finita para definir un conjunto infinito (conjunto de todos los programas bien escritos)

Formas para definir la sintaxis:

- Lenguaje natural. Ej.: Fortran
- Utilizando la gramática libre de contexto, definida por Backus y Naun: BNF. Ej: Algol
- Diagramas sintácticos son equivalentes a BNF pero mucho mas intuitivos

# SINTAXIS

- **BNF (Backus Naun Form)**
  - Es una notación formal para describir la sintaxis
  - Es un metalenguaje
  - Utiliza metasímbolos
    - **< > ::= |**
  - Define las reglas por medio de “producciones”

Ejemplo:

**<** digito **>** **::=** 0|1|2|3|4|5|6|7|8|9

**No terminal** Se define como **Terminales**

**Metasímblo**

# SINTAXIS - BNF

- **Gramática**
  - Conjunto de reglas finita que define un conjunto infinito de posibles sentencias válidas en el lenguaje.
  - Una gramática esta formada por una 4-tupla

**G = ( N, T, S, P)**

Conjunto de símbolos no terminales

Conjunto de símbolos terminales

Símbolo distinguido de la gramática que pertenece a N

Conjunto de producciones

# SINTAXIS - BNF

## Ejemplo Simple: Números Enteros

**G=(N,T,S,P)**

**N={<numEntero>,** <digito>}

**T={**“0”,“1”,“2”,“3”,“4”,“5”,“6”,“7”,“8”,“9” }

**S={<numEntero>}**

**P={**

**<numEntero>::=<digito>|<digito> <numEntero>**

} <digito>::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”

}

Producción
RECURSIVA!!!

# SINTAXIS - BNF

**Producciones recursivas:**

- Regla recursiva por la izquierda: La asociatividad es por la izquierda. El símbolo no terminal de la parte izquierda de una regla de producción aparece al comienzo de la parte derecha
- Regla recursiva por la derecha: La asociatividad es por la derecha. El símbolo no terminal de la parte izquierda de una regla de producción aparece al final de la parte derecha

# SINTAXIS - BNF

## Gramáticas ambiguas:

Una gramática es ambigua si una sentencia puede derivarse de mas de una forma

G= ( N, T, S, P)

N = { <id>, <exp>, <asig>}

T = { A,B,C, +,*,-,/,:=}

S = <asig>

P = {

<asig> ::= <id> := <exp>

<exp> ::= <exp>+<exp>|<exp>*<exp>|<exp>- <exp>|<exp>/ <exp>|<id>

<id> ::= A | B | C

}

**recursión**

# SINTAXIS

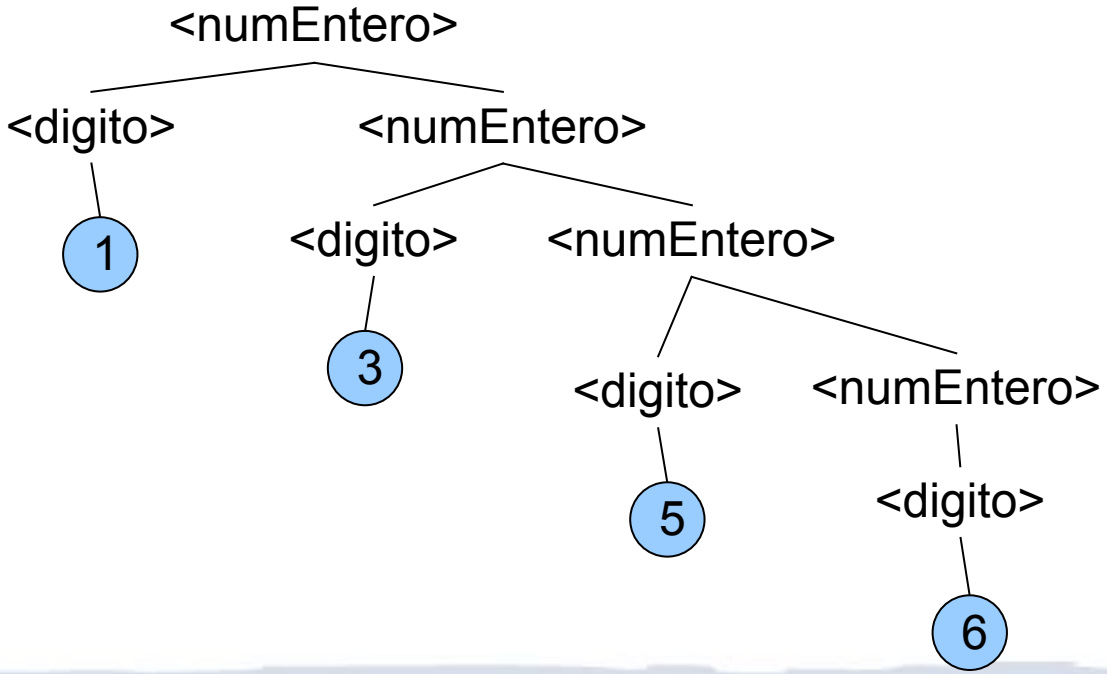
Se necesita de un Método de análisis (reconocimiento) que permita determinar si un string dado es valido o no en el lenguaje: Parsing.

El parse, para cada sentencia construye un “árbol sintáctico o árbol de derivación”

# SINTAXIS

G=(N,T,S,P)
N={<numEntero>, <digito>}
T={“0”,“1”,“2”,“3”,“4”,“5”,“6”,“7”,“8”,“9” }
S={<numEntero>}
P={
<numEntero>::=<digito>|<digito> <numEntero>
<digito>::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”
}

Ej.: 1356



# SINTAXIS

## Ejemplo con gramática ambigua:

G= ( N, T, S, P)

N = { <id>, <exp>, <asig>}

T = { A,B,C, +,*,-,/,:=}

S = <asig>

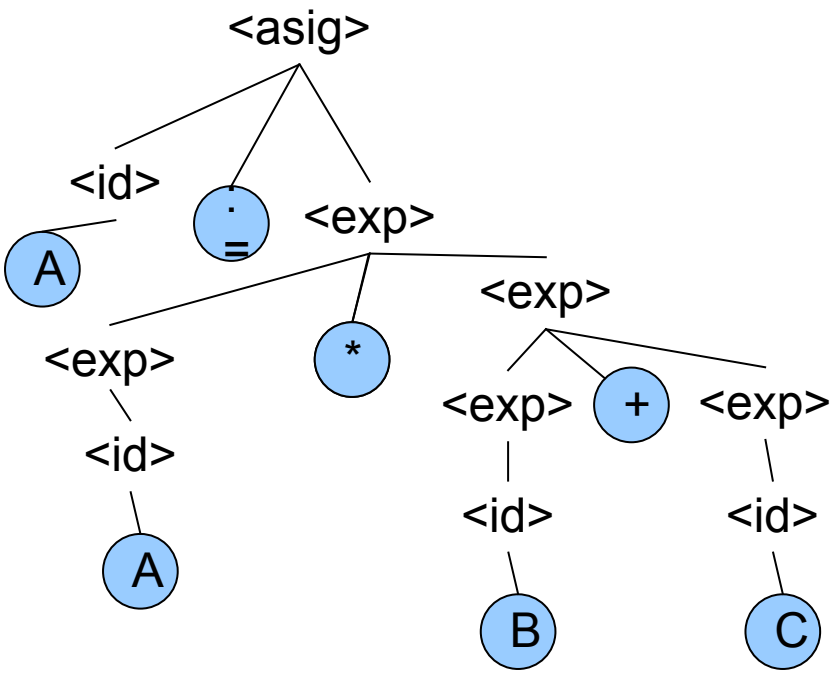
P = {

<asig> ::= <id> := <exp>

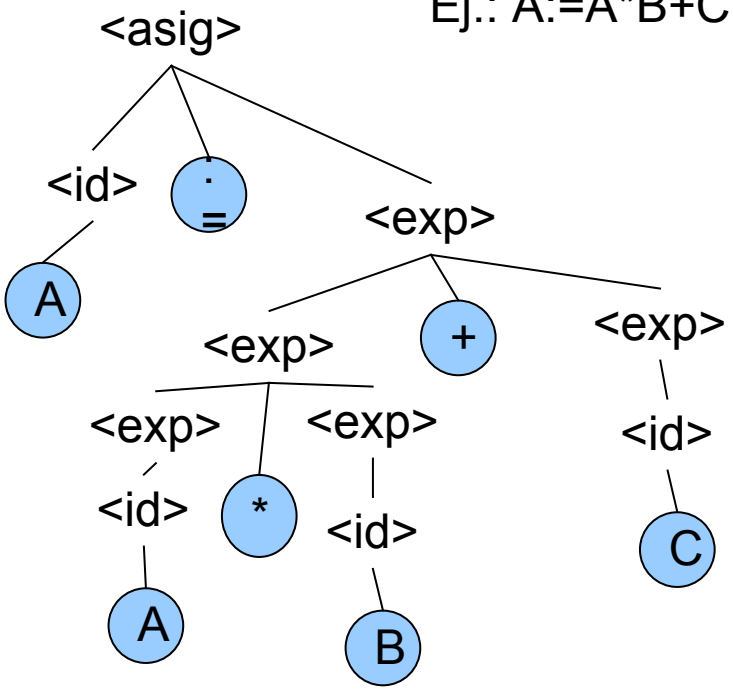
<exp> ::= <exp>+<exp>|<exp>*<exp>|<exp>- <exp>|<exp>/ <exp>|<id>

<id> ::= A | B | C

}




Ej.: A:=A*B+C



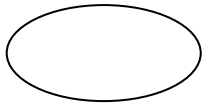
# SINTAXIS

## EBNF:

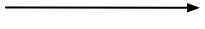
| Meta símbolos utilizados por | | Significado |
|---|---|---|
| BNF | EBNF | |
| < > | < > | Definición de un elemento no terminal |
| ::= | ::= | Definición de una producción |
| \| | (\|) | Selección de una alternativa |
| < p > < p1 > | { } | Repetición |
| | * | Repetición de 0 o más veces |
| | + | Repetición de 1 o más veces |
| | [ ] | Opcional está presente o no lo está |
| Nota: p y p1 son producciones si | | |

# SINTAXIS

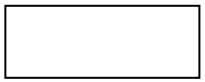
## Diagramas sintácticos (CONWAY):



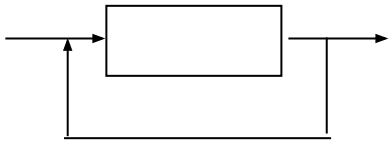
**Terminales**



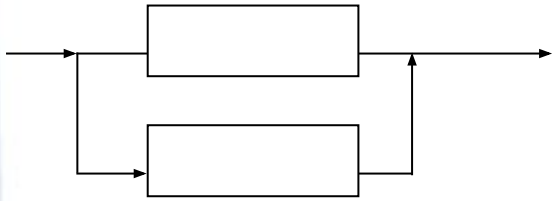
**Flujo**



**No terminales**



**Repetición**



**Selección**

**Ej:**