

# ICI 4242 - Autómatas y compiladores

## Análisis Sintáctico

Rodrigo Olivares  
Mg. en Ingeniería Informática  
`rodrigo.olivares@uv.cl`

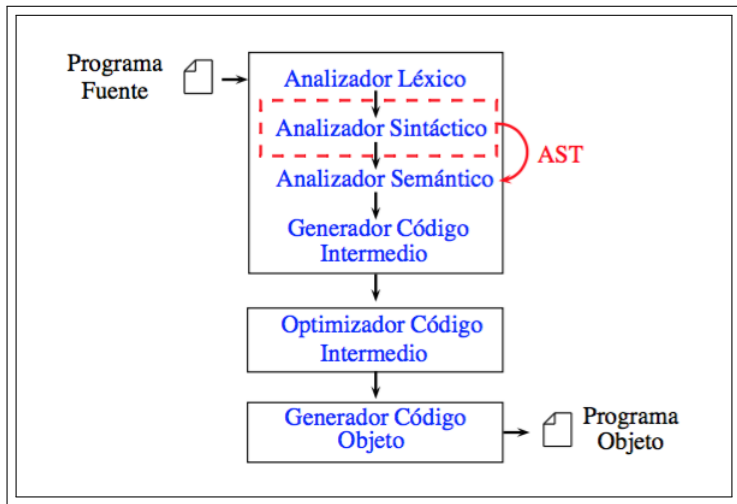
1er Semestre

# Funciones del analizador sintáctico

## Definición

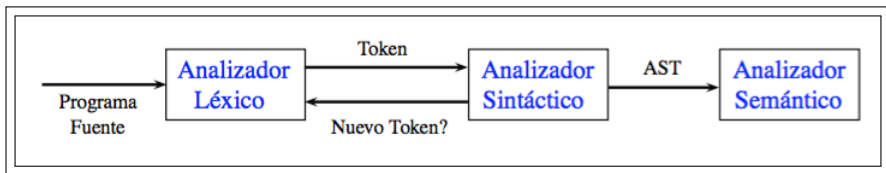
**Analizador Sintáctico:** *Corresponde a la segunda fase de un compilador. En esta fase se construye un **AST** (árbol de sintaxis abstracta) para capturar la jerarquía de la entrada.*

# Funciones del analizador sintáctico



# Funciones del analizador sintáctico

- Construir un **AST** a partir de los tokens recibidos por el analizador léxico.
- Detección de **errores sintácticos**.
- El analizador sintáctico también se conoce como **parser**.



# Herramientas del analizador sintáctico

## Generadores de analizadores sintácticos:

→ **Yacc**

(<http://dinosaur.compilertools.net/>)

→ **Bison**

(<http://www.gnu.org/software/bison/>)

→ **PLY** (Python Lex-Yacc)

(<http://www.dabeaz.com/ply/>)

→ **ANTLR**

(<http://www.antlr.org/>)

...

# Implementación del analizador sintáctico en ANTLR

```

tokens {
    PROGRAM
    VAR_DEC
    ASSIGN
    ...
}

program : VAR_RW! var_dec BEGIN_RW! body END_RW!
        {## = #( #[PROGRAM, "PROGRAM"] ,##);};

var_dec : (type IDENT SEMICOLON!)*
        {## = #( #[VAR_DEC, "VAR_DEC"] ,##);};

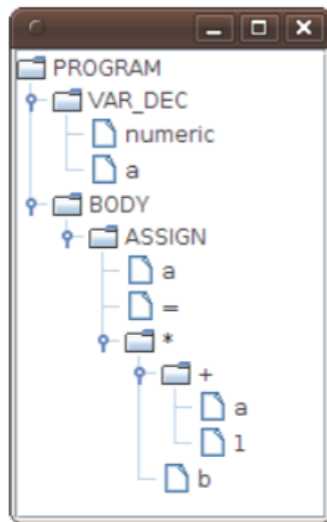
type : NUMERIC_TYPE | STRING_TYPE;

assign : IDENT ASSIG expr SEMICOLON!
        {## = #( #[ASSIGN, "ASSIGN"] ,##);};

```

# Implementación del analizador sintáctico en ANTLR

```
var
    numeric a;
begin
    a = (a + 1) * b
end
```



# Implemente el analizador sintáctico del lenguaje MiLe

## Gramática

`program ::= "var" var_dec "begin" body "end"`

`var_dec ::= type ident ";"`

`type ::= "numeric" | "string"`

`assig ::= ident "=" exp ";"`

`body ::= assig | st | print | read`

`st ::= for | if`



# Implemente el analizador sintáctico del lenguaje MiLe

## Gramática

```
for ::= "for" for_header "" body ""  
for_header ::= "(" assign ";" expr ";" number ")"  
if ::= "if" "(" cond ")" "" body "" else?  
else ::= "else" "" body ""  
read ::= "read" "(" ident ")" ";"  
print ::= "print" "(" string "," ident ")" ";"
```

# Preguntas

Preguntas ?