

# Desarrollo de Lenguajes y Compiladores

## [MII774]

### Capítulo 3: Análisis Sintáctico

**Dr. Ricardo Soto**

[[ricardo.soto@ucv.cl](mailto:ricardo.soto@ucv.cl)]

[<http://www.inf.ucv.cl/~rsoto>]

Escuela de Ingeniería Informática  
Pontificia Universidad Católica de Valparaíso

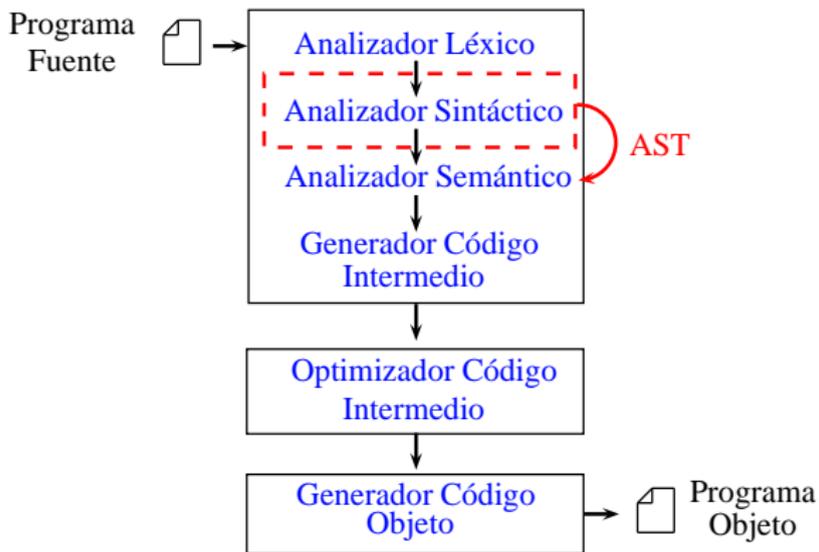


PONTIFICIA UNIVERSIDAD  
CATOLICA  
DE VALPARAISO



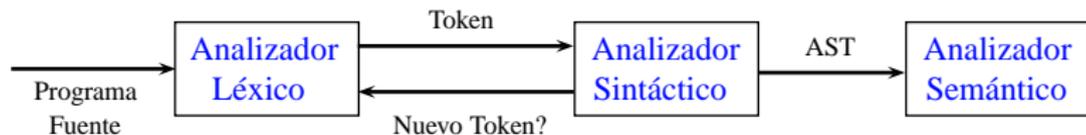
# 1. Introducción

El **analizador sintáctico** es la fase que sigue al análisis léxico. En esta fase se construye un AST (árbol de sintaxis abstracta) para capturar la jerarquía de la entrada.



## 2. Funciones del Analizador Sintáctico

- **Construir** un AST a partir de los **tokens** recibidos por el analizador léxico.



- Detección de **errores sintácticos**

### Nota

El **analizador sintáctico** también se conoce como parser.

# 3. Herramientas para implementar analizadores sintácticos

- 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/>)
  - ...

# 4. Implementación de analizadores sintácticos en ANTLR

## Definición de tokens y reglas

```

tokens {
  PROGRAM
  VAR_DEC
  ASSIGN
  ...
}
...

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

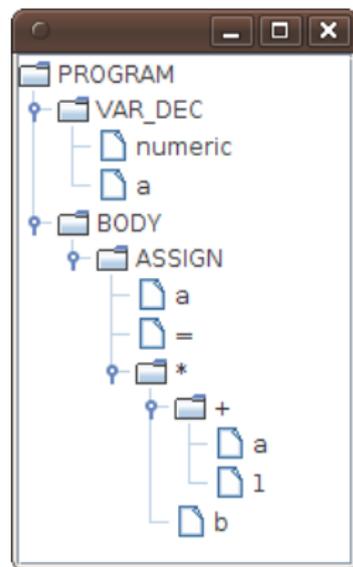
var_dec : (type IDENT SEMICOLON!)*
        {## = #( #[VAR_DEC, "VAR_DEC"] ,##);};

type:    NUMERIC_TYPE|STRING_TYPE;

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

```

## AST



# 4. Implementación de analizadores sintácticos en ANTLR

## Definición de tokens y reglas

```
tokens {
  PROGRAM
  VAR_DEC
  ASSIGN
  ...
}
```

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

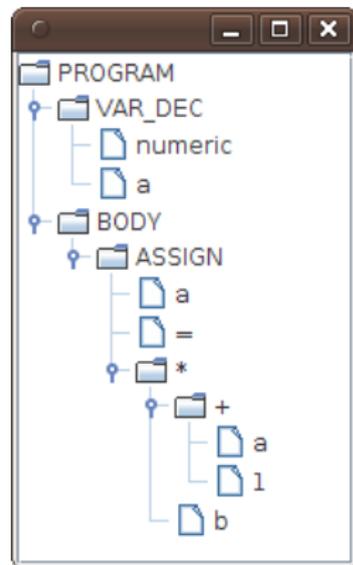
```
program : VAR_RW! var_dec BEGIN_RW! body END_RW!
        {## = #( #[PROGRAM, "PROGRAM"] ,##);};

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

type:    NUMERIC_TYPE|STRING_TYPE;

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

## AST



# 5. Ejercicios

## Implemente el analizador sintáctico del lenguaje MiLe (Micro Lenguaje)

### Gramática

```

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

var_dec ::= type ident ";"
type ::= "numeric" | "string"

assign ::= ident "=" exp ";"
body ::= assign | st | print | read
st ::= for_st | if_st

for_st ::= "for" for_header "{" body "}"
for_header ::= "(" assign expr ";" number ")"

if_st ::= "if" "(" cond ")" "{" body "}" else_st?
else_st ::= "else" "{" body "}"

read ::= "read" "(" ident ")" ";"
print ::= "print" "(" string "," ident ")" ";"

```