



**E.T.S.I. Informáticos**  
**Universidad Politécnica de Madrid**



## **Procesadores de Lenguajes**

**Práctica**

---

# **Evaluación de TDS con Analizadores Sintácticos Descendentes Recursivos**

**José Luis Fuertes**

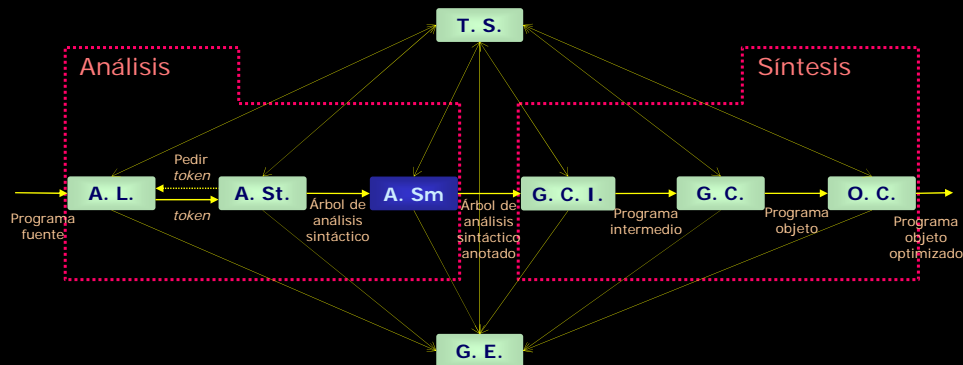
diciembre de 2020

## **Contenido**

---

1. Análisis Sintáctico Descendente Recursivo
2. Traductor Descendente Recursivo
3. Ejemplo

## Compilador



3

Análisis Semántico

## Análisis Sintáctico Descendente Recursivo

- En un A. St. D. las acciones se ejecutarán en el mismo momento en que se expandiría un símbolo que estuviera en esa misma posición en el árbol de análisis sintáctico
- Analizador Sintáctico Descendente Predictivo Recursivo
  - ♦ Consiste en un programa con un procedimiento para cada no terminal
    - Si en la regla hay un no terminal, se llama al procedimiento de dicho no terminal
    - Si en la regla hay un terminal y coincide con el último *token* recibido, se pide al Analizador Léxico el siguiente *token*
  - ♦ La gramática debe ser LL(1)

4

Práctica-ASm

## Evaluación Descendente de TDS

- Diseño de un Traductor Descendente Recursivo
  - ♦ Se parte de:
    - G tipo 2 LL(1)
    - Esquema de Traducción
  - ♦ Consiste en un programa con una función para cada no terminal
  - ♦ Función:
    1. Para cada No Terminal A, se construye una función que tenga un parámetro por cada atributo heredado de A y que devuelva los valores de los atributos sintetizados de A
    2. El código de la función depende de la regla:
      - a. Si hay un Terminal X (*token*), con un atributo x, se guarda X.x en una variable local. Si coincide con el último *token* recibido, se pide al Analizador Léxico el siguiente *token*
      - b. Si hay un No Terminal B, se genera la instrucción  $s := B(b_1, b_2, \dots, b_n)$ 
        - $b_i$ : son los atributos heredados de B
        - $s$ : variable local que recoge los atributos sintetizados de B
      - c. Si hay una Acción Semántica, se implementa dentro de la función sustituyendo las referencias a atributos por las variables correspondientes

5

Práctica-ASm

## Evaluación Descendente de TDS

### Ejemplo

```

D → T {L.tipo:= T.tipo}
    L ;
T → int {T.tipo:= ent}
T → float {T.tipo:= real}
L → id {AñadeTipoTS (id.pos, L.tipo)
        R.tipo:= L.tipo}
    R
R → , {L.tipo:= R.tipo}
    L
R → λ {}
  
```

```

Procedure D ()
{
  var t: tipo
  t:= T ()
  L (t)
  equipara (<pyc>)
}
  
```

```

Function T (): tipo
{
  var t: tipo
  if (sig_tok=<int>) then
  {
    equipara (<int>)
    t:= ent
  }
  else
  {
    equipara (<float>)
    t:= real
  }
  return t
}
  
```

6

Práctica-ASm

## Evaluación Descendente de TDS

### Ejemplo

```

D → T {L.tipo:= T.tipo}
    L ;
T → int {T.tipo:= ent}
T → float {T.tipo:= real}
L → id {AñadeTipoTS (id.pos, L.tipo)
        R.tipo:= L.tipo}
    R
R → , {L.tipo:= R.tipo}
    L
R → λ {}

```

```

Procedure L (t: tipo)
{
    var p: posTS
    if (sig_tok=<id>) then
    {
        p:= sig_tok.atrib
        equipara (<id>)
        AñadeTipoTS (p, t)
        R (t)
    }
    else Error (...)
}

```

```

Procedure R (t: tipo)
{
    if (sig_tok=<coma>) then
    {
        equipara (<coma>)
        L (t)
    }
}

```