

# Desarrollo de analizadores sintácticos ascendentes y descendentes

FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
COMPLUTENSE  
MADRID

Asignatura de Procesadores de Lenguajes

Curso 2021-2022

Grupo 20

Adrián Martín Tiscar  
Gema Blanco Núñez

# 1. Especificación sintáctica

Programa  $\rightarrow$  Decs '&&' Instrucciones

Programa  $\rightarrow$  Instrucciones

Decs  $\rightarrow$  Decs ';' Dec

Decs  $\rightarrow$  Dec

Dec  $\rightarrow$  var tipo id

Dec  $\rightarrow$  type tipo id

Dec  $\rightarrow$  proc id ParamForm bloque

ParamForm  $\rightarrow$  '(' ListParamForm ')'

ParamForm  $\rightarrow$  '(' ')'

ListParamForm  $\rightarrow$  ListParamForm ',' Parametro

ListParamForm  $\rightarrow$  Parametro

Parametro  $\rightarrow$  tipo '&' id

Parametro  $\rightarrow$  tipo id

bloque  $\rightarrow$  '{' Programa '}'

tipo  $\rightarrow$  int

tipo  $\rightarrow$  bool

tipo  $\rightarrow$  real

tipo  $\rightarrow$  string

tipo  $\rightarrow$  id

tipo  $\rightarrow$  array '[' numeroEntero ']' of tipo

tipo  $\rightarrow$  record '{' campos '}'

tipo  $\rightarrow$  pointer tipo

campos  $\rightarrow$  campos ';' campo

campos  $\rightarrow$  campo

campo  $\rightarrow$  tipo id

Instrucciones  $\rightarrow$  Instrucciones ';' Inst

Instrucciones  $\rightarrow$  Inst

- Instrucción de asignación:

Inst  $\rightarrow$  Expresion '=' Expresion

- Instrucción if-then:

Inst  $\rightarrow$  if Expresion then InstOp endif

- Instrucción if-then-else:

Inst  $\rightarrow$  if Expresion then InstOp else InstOp endif

- Instrucción while:

Inst  $\rightarrow$  while Expresion do InstOp endwhile

InstOp  $\rightarrow$  Instrucciones

InstOp  $\rightarrow \epsilon$

- Instrucción de lectura

Inst  $\rightarrow$  read Expresion

- Instrucción de escritura

Inst  $\rightarrow$  write Expresion

- Instrucción de nueva línea

Inst  $\rightarrow$  nl

- Instrucción de reserva de memoria

Inst  $\rightarrow$  new Expresion

- Instrucción de liberación de memoria

Inst  $\rightarrow$  delete Expresion

- Instrucción de invocación a procedimiento

Inst  $\rightarrow$  call id paramReales

paramReales  $\rightarrow$  '(' ListaExpresiones ')'

paramReales  $\rightarrow$  '(' ')'

ListaExpresiones  $\rightarrow$  ListaExpresiones ',' Expresion

ListaExpresiones  $\rightarrow$  Expresion

- Instrucción compuesta

Inst  $\rightarrow$  bloque

Expresion  $\rightarrow$  E0

E0  $\rightarrow$  E1 '+' E0

E0  $\rightarrow$  E1 '-' E1

E0  $\rightarrow$  E1

E1  $\rightarrow$  E1 OP1 E2

E1  $\rightarrow$  E2

E2  $\rightarrow$  E2 OP2 E3

E2  $\rightarrow$  E3

E3  $\rightarrow$  E4 OP3 E4

E3  $\rightarrow$  E4

E4  $\rightarrow$  '-' E5

E4 → not E4

E4 → E5

E5 → E5 OP5

E5 → E6

E6 → '\*' E6

E6 → E7

E7 → numeroEntero

E7 → numeroReal

E7 → id

E7 → true

E7 → false

E7 → literalCad

E7 → null

E7 → (E0)

OP1 → and

OP1 → or

OP2 → '>'

OP2 → '>='

OP2 → '<'

OP2 → '<='

OP2 → '=='

OP2 → '!='

OP3 → '\*'

OP3 → '/'

OP3 → '%'

- Operadores de indexación:

OP5 → '[' Expresion ']'

- Operadores de acceso a registro:

OP5 → '.' id

OP5 → '->' id

## 2. Acondicionamiento de la gramática

Con el fin de construir un analizador sintáctico descendente predictivo recursivo se aplican dos transformaciones:

- Eliminación de factores comunes
- Eliminación de recursión a izquierdas

Eliminación de recursión a izquierdas	
Sin acondicionar	Acondicionada

Decs $\rightarrow$ Decs ';' Dec Decs $\rightarrow$ Dec	Decs $\rightarrow$ Dec RDecs RDecs $\rightarrow$ ';' Dec RDecs RDecs $\rightarrow \epsilon$
ListParamForm $\rightarrow$ ListParamForm ';' Parametro ListParamForm $\rightarrow$ Parametro	ListParamForm $\rightarrow$ Parametro RListParamForm RListParamForm $\rightarrow$ ';' Parametro RListParamForm RListParamForm $\rightarrow \epsilon$
campos $\rightarrow$ campos ';' campo campos $\rightarrow$ campo	campos $\rightarrow$ campo RCampos RCampos $\rightarrow$ ';' campo RCampos RCampos $\rightarrow \epsilon$
Instrucciones $\rightarrow$ Instrucciones ';' Inst Instrucciones $\rightarrow$ Inst	Instrucciones $\rightarrow$ Inst RInstrucciones RInstrucciones $\rightarrow$ ';' Inst RInstrucciones RInstrucciones $\rightarrow \epsilon$
ListaExpresiones $\rightarrow$ ListaExpresiones ';' Expresion ListaExpresiones $\rightarrow$ Expresion	ListaExpresiones $\rightarrow$ Expresion RListaExpresiones RListaExpresiones $\rightarrow$ ';' Expresion RListaExpresiones RListaExpresiones $\rightarrow \epsilon$
E1 $\rightarrow$ E1 OP1 E2 E1 $\rightarrow$ E2	E1 $\rightarrow$ E2 RE1 RE1 $\rightarrow$ OP1 E2 RE1 RE1 $\rightarrow \epsilon$
E2 $\rightarrow$ E2 OP2 E3 E2 $\rightarrow$ E3	E2 $\rightarrow$ E3 RE2 RE2 $\rightarrow$ OP2 E3 RE2 RE2 $\rightarrow \epsilon$
E5 $\rightarrow$ E5 OP5 E5 $\rightarrow$ E6	E5 $\rightarrow$ E6 RE5 RE5 $\rightarrow$ OP5 RE5 RE5 $\rightarrow \epsilon$
<b>Eliminación de factores comunes</b>	
<b>Sin acondicionar</b>	<b>Acondicionada</b>
ParamForm $\rightarrow$ '(' ListParamForm ')' ParamForm $\rightarrow$ '(' '('	ParamForm $\rightarrow$ '(' restoParamForm ')' restoParamForm $\rightarrow$ ListParamForm restoParamForm $\rightarrow \epsilon$
Parametro $\rightarrow$ tipo '&' id Parametro $\rightarrow$ tipo id	Parametro $\rightarrow$ tipo restoParametro restoParametro $\rightarrow$ Amp id restoParametro $\rightarrow$ id
Inst $\rightarrow$ if Expresion then InstOp endif	Inst $\rightarrow$ if Expresion then InstOp RIf

Inst $\rightarrow$ if Expresion then InstOp else InstOp endif	RIf $\rightarrow$ endif Rif $\rightarrow$ else InstOp endif
paramReales $\rightarrow$ '(' ListaExpresiones ')' paramReales $\rightarrow$ '(' ')'	paramReales $\rightarrow$ '(' RparamReales ')' RparamReales $\rightarrow$ ListaExpresiones RparamReales $\rightarrow \epsilon$
E0 $\rightarrow$ E1 '+' E0 E0 $\rightarrow$ E1 '-' E1 E0 $\rightarrow$ E1	E0 $\rightarrow$ E1 RE0 RE0 $\rightarrow$ '+' E0 RE0 $\rightarrow$ '-' E1 RE0 $\rightarrow \epsilon$
E3 $\rightarrow$ E4 OP3 E4 E3 $\rightarrow$ E4	E3 $\rightarrow$ E4 RE3 RE3 $\rightarrow$ OP3 E4 RE3 $\rightarrow \epsilon$
OP2 $\rightarrow$ '>' OP2 $\rightarrow$ '>=' OP2 $\rightarrow$ '<' OP2 $\rightarrow$ '<='	OP2 $\rightarrow$ '<' ROP2 OP2 $\rightarrow$ '>' ROP2 ROP2 $\rightarrow$ '=' ROP2 $\rightarrow \epsilon$

Aplicando las anteriores transformaciones. obtenemos la siguiente gramática resultante acondicionada:

Programa  $\rightarrow$  Decs '&&' Instrucciones  
Programa  $\rightarrow$  Instrucciones

Decs  $\rightarrow$  Dec RDecs  
RDecs  $\rightarrow$  ';' Dec RDecs  
RDecs  $\rightarrow \epsilon$

Dec  $\rightarrow$  var tipo id  
Dec  $\rightarrow$  type tipo id  
Dec  $\rightarrow$  proc id ParamForm bloque

ParamForm  $\rightarrow$  '(' restoParamForm ')'  
restoParamForm  $\rightarrow$  ListParamForm  
restoParamForm  $\rightarrow \epsilon$

ListParamForm  $\rightarrow$  Parametro RListParamForm  
RListParamForm  $\rightarrow$  ';' Parametro RListParamForm  
RListParamForm  $\rightarrow \epsilon$

Parametro  $\rightarrow$  tipo restoParametro  
restoParametro  $\rightarrow$  '&' id  
restoParametro  $\rightarrow$  id

bloque  $\rightarrow$  '{ Programa '}

tipo  $\rightarrow$  int

tipo  $\rightarrow$  bool

tipo  $\rightarrow$  real

tipo  $\rightarrow$  string

tipo  $\rightarrow$  id

tipo  $\rightarrow$  array '[' numeroEntero ']' of tipo

tipo  $\rightarrow$  record '{ campos '}

tipo  $\rightarrow$  pointer tipo

campos  $\rightarrow$  campo RCampos

RCampos  $\rightarrow$  ';' campo RCampos

RCampos  $\rightarrow$   $\epsilon$

campo  $\rightarrow$  tipo id

Instrucciones  $\rightarrow$  Inst RInstrucciones

RInstrucciones  $\rightarrow$  ';' Inst RInstrucciones

RInstrucciones  $\rightarrow$   $\epsilon$

- Instrucción de asignación:

Inst  $\rightarrow$  Expresion '=' Expresion

- Instrucción if-then:

Inst  $\rightarrow$  if Expresion then InstOp RIf

RIf  $\rightarrow$  endif

Rif  $\rightarrow$  else InstOp endif

- Instrucción while:

Inst  $\rightarrow$  while Expresion do InstOp endwhile

InstOp  $\rightarrow$  Instrucciones

InstOp  $\rightarrow$   $\epsilon$

- Instrucción de lectura

Inst  $\rightarrow$  read Expresion

- Instrucción de escritura

Inst  $\rightarrow$  write Expresion

- Instrucción de nueva línea

Inst  $\rightarrow$  nl

- Instrucción de reserva de memoria

Inst  $\rightarrow$  new Expresion

- Instrucción de liberación de memoria

Inst  $\rightarrow$  delete Expresion

- Instrucción de invocación a procedimiento

Inst  $\rightarrow$  call id paramReales

paramReales  $\rightarrow$  '(' RparamReales ')'

RparamReales  $\rightarrow$  ListaExpresiones

RparamReales  $\rightarrow$   $\epsilon$

ListaExpresiones  $\rightarrow$  Expresion RListaExpresiones

RListaExpresiones  $\rightarrow$  ',' Expresion RListaExpresiones

RListaExpresiones  $\rightarrow$   $\epsilon$

- Instrucción compuesta

Inst  $\rightarrow$  bloque

Expresion  $\rightarrow$  E0

E0  $\rightarrow$  E1 RE0

RE0  $\rightarrow$  '+' E0

RE0  $\rightarrow$  '-' E1

RE0  $\rightarrow$   $\epsilon$

E1  $\rightarrow$  E2 RE1

RE1  $\rightarrow$  OP1 E2 RE1

RE1  $\rightarrow$   $\epsilon$

E2  $\rightarrow$  E3 RE2

RE2  $\rightarrow$  OP2 E3 RE2

RE2  $\rightarrow$   $\epsilon$

E3  $\rightarrow$  E4 RE3

RE3  $\rightarrow$  OP3 E4

RE3  $\rightarrow$   $\epsilon$

E4  $\rightarrow$  '-' E5

E4  $\rightarrow$  not E4

E4  $\rightarrow$  E5

E5  $\rightarrow$  E6 RE5

RE5  $\rightarrow$  OP5 RE5

RE5  $\rightarrow$   $\epsilon$

E6  $\rightarrow$  '\*' E6

E6  $\rightarrow$  E7

E7  $\rightarrow$  numeroEntero



E7  $\rightarrow$  numeroReal  
E7  $\rightarrow$  id  
E7  $\rightarrow$  true  
E7  $\rightarrow$  false  
E7  $\rightarrow$  literalCad  
E7  $\rightarrow$  null  
E7  $\rightarrow$  (E0)

OP1  $\rightarrow$  and  
OP1  $\rightarrow$  or

OP2  $\rightarrow$  '<' ROP2  
OP2  $\rightarrow$  '>' ROP2  
ROP2  $\rightarrow$  '='  
ROP2  $\rightarrow$   $\epsilon$   
OP2  $\rightarrow$  '=='  
OP2  $\rightarrow$  '!='

OP3  $\rightarrow$  '\*'  
OP3  $\rightarrow$  '/'  
OP3  $\rightarrow$  '%'

- Operadores de indexación:  
OP5  $\rightarrow$  '[' Expresion ']

- Operadores de acceso a registro:  
OP5  $\rightarrow$  '.' id  
OP5  $\rightarrow$  '->' id