

# Memoria práctica 2:

Procesadores de Lenguajes:

Integrantes: Suhuai Chen, Daniel Pizarro, David Starry y Jiahui You

# Índice

1. Tiny (o) .....	2
1.1 Gramática .....	2
Terminales: .....	2
No Terminales: .....	3
2. Tiny .....	5
2.1 Gramática .....	5
Terminales: .....	5
No terminal: .....	6

## 1. Tiny (o)

### 1.1 Gramática

#### Terminales:

Los terminales se corresponden con clases léxicas.

- Una clase léxica por cada **símbolo de puntuación**: igual { } &&;
- Clase léxica para booleanos : true, false
- Operadores:
  - Nivel 0: Operador de asignación = (binario infijo asociativo a derechas)
  - Nivel 1: Operadores relacionales <, >, <=, >=, ==, != (binario infijo asociativo a izquierdas)
  - Nivel 2: + (binario infijo asociativo a izquierdas), - (binario infijo no asociativo)
  - Nivel 3: **and** asociativo a derechas, y **or** no asociativo.
  - Nivel 4: \*, / Operadores binarios, infijos, asociativos a izquierdas.
  - Nivel 5: - (unario) y **not**. Operadores unarios, prefijos, asociativos.

Estas prioridades y asociatividades pueden modificarse mediante el uso de paréntesis, tal y como es habitual.

- identificador
- literalEntero
- literalReal
- paréntesis de apertura
- paréntesis de cierre
- @

## No Terminales:

programa, bloque, declaraciones\_opt, declaraciones, declaración, instrucciones, instrucción, E0, E1, E2, E3, E4, E5, E6, OP1, OP4, OP5, TipoBasico

## Patrones:

// Comentarios

Directores

Gramática equivalente

Reutilización, nivel de abstracción equilibrado, opcionalidad, variantes, listas y expresiones.

programa  $\rightarrow$  { bloque } # Símbolo inicial (Primera regla) ({} )

bloque  $\rightarrow$  declaraciones\_opt instrucciones\_opt (int, real, bool)

declaraciones\_opt  $\rightarrow$  declaraciones && (int, real, bool)

declaraciones\_opt  $\rightarrow \epsilon$

instrucciones\_opt  $\rightarrow$  instrucciones (@)

instrucciones\_opt  $\rightarrow \epsilon$

declaraciones  $\rightarrow$  declaraciones ; declaracion

Declaraciones  $\rightarrow$  declaracion

declaraciones  $\rightarrow$  declaracion DR (int, real, bool)

DR  $\rightarrow$  ; declaracion DR(;

DR  $\rightarrow \epsilon$

//declaraciones  $\rightarrow$  declaracion ; declaraciones

//Declaraciones  $\rightarrow$  declaracion

//declaraciones  $\rightarrow$  declaración DF (int, real, bool)

//DF  $\rightarrow$  ; declaraciones (;

//DF  $\rightarrow \epsilon$

declaracion  $\rightarrow$  TipoBasico identificador (int, real, bool)

TipoBasico  $\rightarrow$  int (int)

TipoBasico  $\rightarrow$  real (real)

TipoBasico  $\rightarrow$  bool (bool)

instrucciones  $\rightarrow$  instrucciones ; instrucción

instrucciones  $\rightarrow$  instruccion

instrucciones  $\rightarrow$  instrucción IR (@)

IR  $\rightarrow$  ; instruccion IR (;)

IR  $\rightarrow \epsilon$

//instrucciones  $\rightarrow$  instruccion ; instrucciones

//instrucciones  $\rightarrow$  instruccion

//instrucciones  $\rightarrow$  instrucción IF (@)

//IF  $\rightarrow$  ; instrucciones IF (;)

//IF  $\rightarrow \epsilon$

instruccion  $\rightarrow$  @ Eo (@)

Eo  $\rightarrow$  E1 = Eo

Eo  $\rightarrow$  E1

Eo  $\rightarrow$  E1 FE1 (not, -, identificador, literalEntero, literalReal, true, false, ())

FE1  $\rightarrow$  = Eo (not, -, identificador, literalEntero, literalReal, true, false, ())

FE1  $\rightarrow \epsilon$

E1  $\rightarrow$  E1 OP1 E2 //izquierdas

E1  $\rightarrow$  E2

E1  $\rightarrow$  E2 RE1 (not, -, identificador, literalEntero, literalReal, true, false, ())

RE1  $\rightarrow$  OP1 E2 RE1 (<, >, <=, >=, ==, !=)

RE1  $\rightarrow \epsilon$

OP1  $\rightarrow$  < (<)

OP1  $\rightarrow$  > (>)

OP1  $\rightarrow$  <= (<=)

OP1  $\rightarrow$  >= (>=)

OP1  $\rightarrow$  == (==)

OP1  $\rightarrow$  != (!=)

E2  $\rightarrow$  E2 + E3 //izquierdas

E2  $\rightarrow$  E3 - E3

E2  $\rightarrow$  E3

E2  $\rightarrow$  E3 FE2 RE2 (not, -, identificador, literalEntero, literalReal, true, false, ())

RE2  $\rightarrow$  + E3 RE2 (not, -, identificador, literalEntero, literalReal, true, false, ())

RE2  $\rightarrow \epsilon$

FE2  $\rightarrow$  - E3 (-)

FE2  $\rightarrow \epsilon$

$E_3 \rightarrow E_4 \text{ and } E_3$

$E_3 \rightarrow E_4 \text{ or } E_4$

$E_3 \rightarrow E_4$

$E_3 \rightarrow E_4 \text{ FE}_3 (\text{not}, -, \text{identificador}, \text{literalEntero}, \text{literalReal}, \text{true}, \text{false}, ())$

$\text{FE}_3 \rightarrow \text{and } E_3$

$\text{FE}_3 \rightarrow \text{or } E_4$

$\text{FE}_3 \rightarrow \varepsilon$

$E_4 \rightarrow E_4 \text{ OP}_4 E_5 // \text{izquierdas}$

$E_4 \rightarrow E_5$

$E_4 \rightarrow E_5 \text{ RE}_4 (\text{not}, -, \text{identificador}, \text{literalEntero}, \text{literalReal}, \text{true}, \text{false}, ())$

$\text{RE}_4 \rightarrow \text{OP}_4 E_5 \text{ RE}_4$

$\text{RE}_4 \rightarrow \varepsilon$

$\text{OP}_4 \rightarrow * (*)$

$\text{OP}_4 \rightarrow / (/)$

$E_5 \rightarrow \text{OP}_5 E_5 (\text{not}, -)$

$E_5 \rightarrow E_6 (\text{identificador}, \text{literalEntero}, \text{literalReal}, \text{true}, \text{false}, ())$

$\text{OP}_5 \rightarrow \text{not} (\text{not})$

$\text{OP}_5 \rightarrow - (-)$

$E_6 \rightarrow \text{identificador} (\text{identificador})$

$E_6 \rightarrow \text{literalEntero} (\text{literalEntero})$

$E_6 \rightarrow \text{literalReal} (\text{literalReal})$

$E_6 \rightarrow \text{true} (\text{true})$

$E_6 \rightarrow \text{false} (\text{false})$

$E_6 \rightarrow ( E_6 ) (())$

## 2. Tiny

### 2.1 Gramática

#### Terminales:

Los terminales se corresponden con clases léxicas.

- Una clase léxica por cada **símbolo de puntuación**: igual { } &&;
- Clase léxica para booleanos : true, false
- Operadores:
  - Nivel 0: Operador de asignación = (binario infijo asociativo a derechas)
  - Nivel 1: Operadores relacionales <, >, <=, >=, ==, != (binario infijo asociativo a izquierdas)
  - Nivel 2: + (binario infijo asociativo a izquierdas), - (binario infijo no asociativo)
  - Nivel 3: **and** asociativo a derechas, y **or** no asociativo.
  - Nivel 4: \*, / Operadores binarios, infijos, asociativos a izquierdas.
  - Nivel 5: - (unario) y **not**. Operadores unarios, prefijos, asociativos.
  - Nivel 6: Operadores de indexación, de acceso a registro y de indirección. Todos estos operadores son operadores unarios posfijos, asociativos. Los operadores de indexación son operadores de la forma [ E ], con E una expresión. Los operadores de acceso a registro son operadores de la forma . c, con c un nombre de campo (un identificador). El operador de indirección es ^.

Estas prioridades y asociatividades pueden modificarse mediante el uso de paréntesis, tal y como es habitual.

- identificador
- literalEntero
- literalReal
- paréntesis de apertura
- paréntesis de cierre
- @

(Terminales para CUP, copiado desde ClaseLexica)

IDEN, PAP, PCIERRE, ASIG, COMA, MAS, MENOS, POR, DIV, EVAL, EOF, GT, GE, LT, LE, EQ, NE, PYC, LLAP, LLCIERRE, SEP, E, LITERALENTERO, LITERALREAL, TRUE, FALSE, AND, OR, NOT, BOOL, ENT, REAL, MOD, CAP, CCIERRE, PUNTO, CIRCUNFLEJO, AMP, LITERALCADENA, STRING, NULL, PROC, IF, ELSE, WHILE, STRUCT, NEW, DELETE, READ, WRITE, NL, TYPE, CALL

### No terminales:

programa, bloque, declaraciones\_opt, instrucciones\_opt, tipo1, tipo2, tipo3, tipoBasico, typeDeclaracion, declaraciones, declaracion, header, variable, parametrosFormales\_opt, parametrosFormales, parametroFormal, referencia\_opt, instrucciones, instruccion, parametrosReales\_opt, parametrosReales, E0, E1, E2, E3, E4, E5, E6, E7, OP1, OP4, OP5, OP6

### Patrones:

// Comentarios

Directores

Gramática equivalente

programa → bloque // Símbolo inicial (Primera regla) ({} )

bloque → { declaraciones\_opt instrucciones\_opt } ({} )

declaraciones\_opt → declaraciones && (^, int, real, bool, string, type, proc)

declaraciones\_opt → ε

instrucciones\_opt → instrucciones (@, if, else, while, read, write, nl, new, delete, call, {} )

instrucciones\_opt → ε

- Tipos // cambiar prioridades [] es tiene más prioridad que el puntero

tipo1 → tipo1[literalEntero]

tipo1 → tipo2 (int, real, bool, string)

tipo1 → tipo2 rtipo1 (int, real, bool, string)

rtipo1 → [literalEntero] rtipo1 ([])

rtipo1 → ε

tipo2 → ^tipo2 (^)

tipo2 → tipo3

tipo3 → tipoBasico (int, real, bool, string)

tipo3 → struct { typeDeclaracion } ({} )

tipo3 → identificador (identificador)

tipoBasico  $\rightarrow$  int (int)  
tipoBasico  $\rightarrow$  real (real)  
tipoBasico  $\rightarrow$  bool (bool)  
tipoBasico  $\rightarrow$  string (string)

typeDeclaracion  $\rightarrow$  variable , typeDeclaracion  
typeDeclaracion  $\rightarrow$  variable  
typeDeclaracion  $\rightarrow$  variable FtypeDeclaracion (^, int, real, bool, string)  
FtypeDeclaracion  $\rightarrow$  , typeDeclaracion (,)  
FtypeDeclaracion  $\rightarrow$   $\epsilon$

#### - Declaraciones

declaraciones  $\rightarrow$  declaraciones ; declaración  
declaraciones  $\rightarrow$  declaración  
declaraciones  $\rightarrow$  declaración DR (^, int, real, bool, string, type, proc)  
DR  $\rightarrow$  ; declaración DR(;)  
DR  $\rightarrow$   $\epsilon$

declaracion  $\rightarrow$  variable (^, int, real, bool, string)  
declaracion  $\rightarrow$  type variable (type)  
declaracion  $\rightarrow$  header bloque (proc)  
header  $\rightarrow$  proc identificador ( parametrosFormales\_opt )

variable  $\rightarrow$  tipo1 identificador (^, int, real, bool, string)

parametrosFormales\_opt  $\rightarrow$  parametrosFormales  
parametrosFormales\_opt  $\rightarrow$   $\epsilon$

parametrosFormales  $\rightarrow$  parametrosFormales, parametroFormal  
parametrosFormales  $\rightarrow$  parametroFormal  
parametrosFormales  $\rightarrow$  parametroFormal RparametrosFormales (^, int, real, bool, string)  
RparametrosFormales  $\rightarrow$  , parametroFormal RparametrosFormales (,)  
RparametrosFormales  $\rightarrow$   $\epsilon$

parametroFormal  $\rightarrow$  tipo1 referencia\_opt identificador (^, int, real, bool, string)  
referencia\_opt  $\rightarrow$  & (&)  
referencia\_opt  $\rightarrow$   $\epsilon$



### - Instrucciones

instrucciones  $\rightarrow$  instrucciones ; instrucción

instrucciones  $\rightarrow$  instruccion

instrucciones  $\rightarrow$  instrucción IR (@, if, else, while, read, write, nl, new, delete, call, {})

IR  $\rightarrow$  ; instruccion IR (;)

IR  $\rightarrow \epsilon$

instruccion  $\rightarrow$  @ Eo (@)

instruccion  $\rightarrow$  if Eo bloque

instruccion  $\rightarrow$  if Eo bloque else bloque

instruccion  $\rightarrow$  if Eo bloque IFR (if)

IFR  $\rightarrow$  else bloque (else)

IFR  $\rightarrow \epsilon$

instruccion  $\rightarrow$  while Eo bloque (while)

instruccion  $\rightarrow$  read Eo (read)

instruccion  $\rightarrow$  write Eo (write)

instruccion  $\rightarrow$  nl // NI IDEA (nl)

instruccion  $\rightarrow$  new Eo (new)

instruccion  $\rightarrow$  delete Eo (delete)

instruccion  $\rightarrow$  call identificador ( parametrosReales\_opt ) (call)

instruccion  $\rightarrow$  bloque ({} )

parametrosReales\_opt  $\rightarrow$  parametrosReales

parametrosReales\_opt  $\rightarrow \epsilon$

parametrosReales  $\rightarrow$  parametrosReales, Eo

parametroReales  $\rightarrow$  Eo

parametrosReales  $\rightarrow$  Eo RparametrosReales (not, -, identificador, leteralEntero, literalReal, true, false, literalCadena, (, null)

RparametrosReales  $\rightarrow$  , Eo RparametrosReales (not, -, identificador, leteralEntero, literalReal, true, false, literalCadena, (, null)

RparametrosReales  $\rightarrow \epsilon$

$E_0 \rightarrow E_1 = E_0$

$E_0 \rightarrow E_1$

$E_0 \rightarrow E_1 \text{ FE}_1(\text{not}, -, \text{identificador}, \text{leteralEntero}, \text{leteralReal}, \text{true}, \text{false}, \text{leteralCadena}, (, \text{null}))$

$\text{FE}_1 \rightarrow = E_0 (=)$

$\text{FE}_1 \rightarrow \varepsilon$

$E_1 \rightarrow E_1 \text{ OP}_1 E_2 // \text{izquierdas}$

$E_1 \rightarrow E_2$

$E_1 \rightarrow E_2 \text{ RE}_1(\text{not}, -, \text{identificador}, \text{leteralEntero}, \text{leteralReal}, \text{true}, \text{false}, \text{leteralCadena}, (, \text{null}))$

$\text{RE}_1 \rightarrow \text{OP}_1 E_2 \text{ RE}_1(<, >, <=, >=, ==, !=)$

$\text{RE}_1 \rightarrow \varepsilon$

$\text{OP}_1 \rightarrow < (<)$

$\text{OP}_1 \rightarrow > (>)$

$\text{OP}_1 \rightarrow <= (<=)$

$\text{OP}_1 \rightarrow >= (>=)$

$\text{OP}_1 \rightarrow == (==)$

$\text{OP}_1 \rightarrow != (!=)$

$E_2 \rightarrow E_2 + E_3 // \text{izquierdas}$

$E_2 \rightarrow E_3 - E_3$

$E_2 \rightarrow E_3$

$E_2 \rightarrow E_3 \text{ FE}_3 \text{ RE}_2 (\text{not}, -, \text{identificador}, \text{leteralEntero}, \text{leteralReal}, \text{true}, \text{false}, \text{leteralCadena}, (, \text{null}))$

$\text{RE}_2 \rightarrow + E_3 \text{ RE}_2 (+)$

$\text{RE}_2 \rightarrow \varepsilon$

$\text{FE}_3 \rightarrow - E_3 (-)$

$\text{FE}_3 \rightarrow \varepsilon$

$E_3 \rightarrow E_4 \text{ and } E_3$

$E_3 \rightarrow E_4 \text{ or } E_4$

$E_3 \rightarrow E_4$

$E_3 \rightarrow E_4 \text{ FE}_4 (\text{not}, -, \text{identificador}, \text{leteralEntero}, \text{leteralReal}, \text{true}, \text{false}, \text{leteralCadena}, (, \text{null}))$

$\text{FE}_4 \rightarrow \text{and } E_3 (\text{and})$

$\text{FE}_4 \rightarrow \text{or } E_4 (\text{or})$

$\text{FE}_4 \rightarrow \varepsilon$

$E_4 \rightarrow E_4 \text{ OP}_4 E_5$

$E_4 \rightarrow E_5$

$E_4 \rightarrow E_5 \text{ RE}_4$  (not, -, identificador, literalEntero, literalReal, true, false, literalCadena, (, null)

$\text{RE}_4 \rightarrow \text{OP}_4 E_5 \text{ RE}_4$  (\*, /, %)

$\text{RE}_4 \rightarrow \varepsilon$

$\text{OP}_4 \rightarrow *$  (\*)

$\text{OP}_4 \rightarrow /$  (/)

$\text{OP}_4 \rightarrow \%$  (%)

$E_5 \rightarrow \text{OP}_5 E_5$  (not, -)

$E_5 \rightarrow E_6$  (identificador, literalEntero, literalReal, true, false, literalCadena, (, null)

$\text{OP}_5 \rightarrow \text{not}$  (not)

$\text{OP}_5 \rightarrow -$  (-)

$E_6 \rightarrow E_6 \text{ OP}_6$

$E_6 \rightarrow E_7$

$E_6 \rightarrow E_7 \text{ RE}_6$  (identificador, literalEntero, literalReal, true, false, literalCadena, (, null)

$\text{RE}_6 \rightarrow \text{OP}_6 \text{ RE}_6$  ([, ., ^)

$\text{RE}_6 \rightarrow \varepsilon$

$\text{OP}_6 \rightarrow [ \text{ Eo } ]$  // E es una expresión, a.c[x + 1] ([)

$\text{OP}_6 \rightarrow . \text{ identificador }$  // . un nombre de campo (un identificador) (.)

$\text{OP}_6 \rightarrow ^$  // 1^.sig (^)

$E_7 \rightarrow \text{identificador}$  (identificador)

$E_7 \rightarrow \text{literalEntero}$  (literalEntero)

$E_7 \rightarrow \text{literalReal}$  (literalReal)

$E_7 \rightarrow \text{true}$  (true)

$E_7 \rightarrow \text{false}$  (false)

$E_7 \rightarrow \text{literalCadena}$  (literalCadena)

$E_7 \rightarrow ( \text{ Eo } )$  ((

$E_7 \rightarrow \text{null}$  (null)