

Practica 2:

Generación del analizador léxico con JLex

Diseño de la sintaxis

Análisis sintáctico descendente

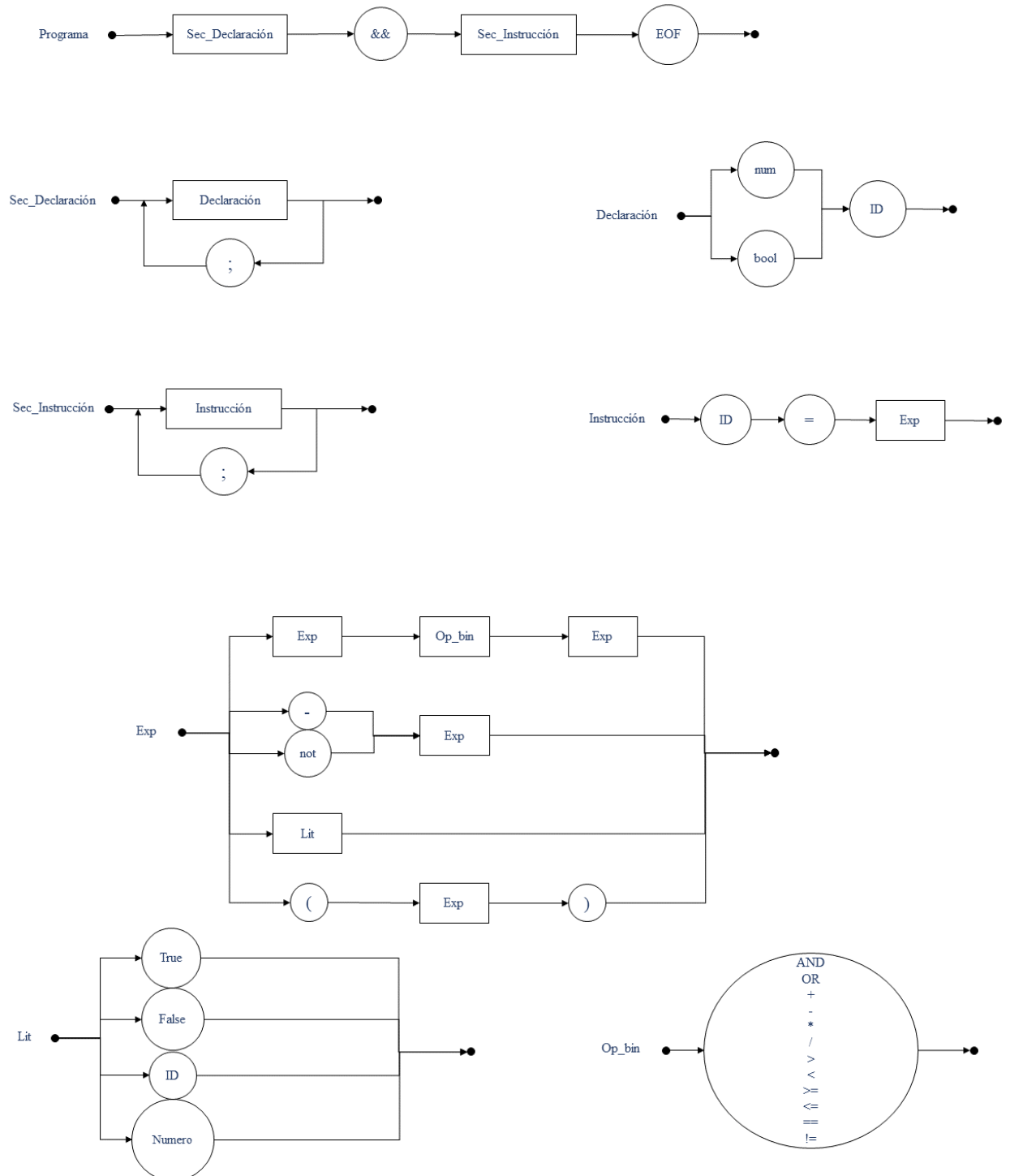
Grupo 11:

Youssef El Faqir El Rhazoui

Enrique Ávila Rodríguez

1. Conceptualización mediante diagramas

Pasamos a describir el lenguaje mediante los diagramas.



2. Gramática Incontextual

Ahora pasamos a especificar la gramática, teniendo en cuenta la asociatividad y prioridad de los operadores dados por la siguiente tabla.

Operadores	Prioridad	Asociatividad
(binario) +, -	0	• Asoc. Izq
And, or	1	• And asoc derch. • Or no asoc
==, !=, <, <=, >, >=	2	• No asoc
*, /	3	• Asoc izq
- (unario), not	4	• - asoc • Not no asoc

Programa \rightarrow Sec_Declaración && Sec_Instrucción EOF

Sec_Declaración \rightarrow Declaración

Sec_Declaración \rightarrow Declaración ; Sec_Declaración

Declaración \rightarrow Tipo ID

Tipo \rightarrow num

Tipo \rightarrow bool

Sec_Instrucción \rightarrow Instrucción

Sec_Instrucción \rightarrow Instrucción ; Sec_Instrucción

Instrucción \rightarrow ID = Exp

Exp \rightarrow Exp0

Exp0 \rightarrow Exp0 Op0 Exp1

Exp0 \rightarrow Exp1

Op0 \rightarrow +

Op0 \rightarrow -

Exp1 \rightarrow Exp2 and Exp1

Exp1 \rightarrow Exp2 or Exp2

Exp1 \rightarrow Exp2

Exp2 \rightarrow Exp3 Op2 Exp3

Exp2 \rightarrow Exp3

Op2 \rightarrow $=$ | $==$ | $<$ | $<=$ | $>$ | $>=$

Exp3 \rightarrow Exp3 Op3 Exp4

Exp3 \rightarrow Exp4

Op3 \rightarrow $*$

Op3 \rightarrow $/$

Exp4 \rightarrow $-$ Exp4

Exp4 \rightarrow not Exp5

Exp4 \rightarrow Exp5

Exp5 \rightarrow Lit

Exp5 \rightarrow (Exp0)

Lit \rightarrow ID

Lit \rightarrow True

Lit \rightarrow False

Lit \rightarrow Numero

3. Transformaciones para gramática LL1

Para poder hacer un análisis descendente práctico, la gramática anterior no nos vale debido a que presenta problemas de recursión a la izquierda y de factor común.

Resolución de Factor Común

(1)

Exp1 \rightarrow Exp2 and Exp1

Exp1 \rightarrow Exp2 or Exp2

Exp1 \rightarrow Exp2

Exp1 \rightarrow Exp2 Exp1p

Exp1p \rightarrow and Exp1

Exp1p \rightarrow or Exp2

Exp1p \rightarrow ϵ

(2)

$\text{Exp2} \rightarrow \text{Exp3 Op2 Exp3}$

$\text{Exp2} \rightarrow \text{Exp3}$

$\text{Exp2} \rightarrow \text{Exp3 Exp2p}$

$\text{Exp2p} \rightarrow \text{Op2 Exp3}$

$\text{Exp2p} \rightarrow \varepsilon$

(3)

$\text{Sec_Declaración} \rightarrow \text{Declaración}$

$\text{Sec_Declaración} \rightarrow \text{Declaración ; Sec_Declaración}$

$\text{Sec_Declaración} \rightarrow \text{Declaración Sec_Declaración_p}$

$\text{Sec_Declaración_p} \rightarrow ; \text{Sec_Declaración}$

$\text{Sec_Declaración_p} \rightarrow \varepsilon$

(4)

$\text{Sec_Instrucción} \rightarrow \text{Instrucción}$

$\text{Sec_Instrucción} \rightarrow \text{Instrucción ; Sec_Instrucción}$

$\text{Sec_Instrucción} \rightarrow \text{Instrucción Sec_Instrucción_p}$

$\text{Sec_Instrucción_p} \rightarrow ; \text{Sec_Instrucción}$

$\text{Sec_Instrucción_p} \rightarrow \varepsilon$

Resolución de recursión a la izq

(5)

$\text{Exp0} \rightarrow \text{Exp0 Op0 Exp1}$

$\text{Exp0} \rightarrow \text{Exp1}$

$\text{Exp0} \rightarrow \text{Exp1 Exp0p}$

$\text{Exp0p} \rightarrow \text{Op0 Exp1 Exp0p}$

$\text{Exp0p} \rightarrow \varepsilon$

(6)

$\text{Exp3} \rightarrow \text{Exp3 Op3 Exp4}$

$\text{Exp3} \rightarrow \text{Exp4}$

$\text{Exp3} \rightarrow \text{Exp4 Exp3p}$

$\text{Exp3p} \rightarrow \text{Op3 Exp4 Exp3p}$

$\text{Exp3p} \rightarrow \varepsilon$

Con lo cual, la gramática LL1 resultante es:

$\text{Programa} \rightarrow \text{Sec_Declaración} \&\& \text{Sec_Instrucción} \text{EOF}$

$\text{Sec_Declaración} \rightarrow \text{Declaración Sec_Declaración_p}$

$\text{Sec_Declaración_p} \rightarrow ; \text{Sec_Declaración}$

$\text{Sec_Declaración_p} \rightarrow \varepsilon$

$\text{Declaración} \rightarrow \text{Tipo ID}$

$\text{Tipo} \rightarrow \text{num}$

$\text{Tipo} \rightarrow \text{bool}$

$\text{Sec_Instrucción} \rightarrow \text{Instrucción Sec_Instrucción_p}$

$\text{Sec_Instrucción_p} \rightarrow ; \text{Sec_Instrucción}$

$\text{Sec_Instrucción_p} \rightarrow \varepsilon$

$\text{Instrucción} \rightarrow \text{ID} = \text{Exp}$

$\text{Exp} \rightarrow \text{Exp0}$

$\text{Exp0} \rightarrow \text{Exp1 Exp0p}$

$\text{Exp0p} \rightarrow \text{Op0 Exp1 Exp0p}$

$\text{Exp0p} \rightarrow \varepsilon$

$\text{Op0} \rightarrow +$

$\text{Op0} \rightarrow -$

$\text{Exp1} \rightarrow \text{Exp2 Exp1p}$

$\text{Exp1p} \rightarrow \text{and Exp1}$

$\text{Exp1p} \rightarrow \text{or Exp2}$

$\text{Exp1p} \rightarrow \varepsilon$

$\text{Exp2} \rightarrow \text{Exp3 Exp2p}$

$\text{Exp2p} \rightarrow \text{Op2 Exp3}$

$\text{Exp2p} \rightarrow \varepsilon$

$\text{Op2} \rightarrow |=|==|<|<=|>|>=$

$\text{Exp3} \rightarrow \text{Exp4 Exp3p}$

$\text{Exp3p} \rightarrow \text{Op3 Exp4 Exp3p}$

$\text{Exp3p} \rightarrow \varepsilon$

$\text{Op3} \rightarrow *$

$\text{Op3} \rightarrow /$

$\text{Exp4} \rightarrow - \text{Exp4}$

$\text{Exp4} \rightarrow \underline{\text{not}} \text{Exp5}$

$\text{Exp4} \rightarrow \text{Exp5}$

$\text{Exp5} \rightarrow \text{Lit}$

$\text{Exp5} \rightarrow (\text{Exp0})$

$\text{Lit} \rightarrow \underline{\text{ID}}$

$\text{Lit} \rightarrow \underline{\text{True}}$

$\text{Lit} \rightarrow \underline{\text{False}}$

$\text{Lit} \rightarrow \underline{\text{Numero}}$

4. Primeros, siguientes y directores

Nos hemos apoyado en [proletool](#) para generarlos, adjuntamos el archivo “proletool_file.txt” usado en esta herramienta.

Anulables					
Sec_Instruccion_p	Exp1p	Exp0p	Sec_Declaracion_p	Exp3p	Exp2p

No terminal	Iniciales
Instruccion	ID
Sec_Instruccion	ID
Sec_Declaracion	bool num
Op0	+ -
Op2	menor_igual mayor_igual mayor menor dist igual
Programa	bool num
Declaracion	bool num
Exp0	Numero not (True ID False -
Tipo	bool num
Op3	* /
Exp2	Numero not (True ID False -
Exp1	Numero not (True ID False -
Sec_Instruccion_p	;
Exp4	Numero not (True ID False -
Lit	Numero True ID False
Exp3	Numero not (True ID False -
Exp1p	or and
Exp0p	+ -
Exp5	Numero (True ID False
Sec_Declaracion_p	;
Exp	Numero not (True ID False -
Exp3p	* /
Exp2p	menor_igual mayor_igual mayor menor dist igual

No terminal	Siguientes
Instruccion	; EOF
Sec_Instruccion	EOF
Sec_Declaracion	&
Op0	Numero not (True ID False -
Op2	Numero not (True ID False -
Programa	\$
Declaracion	& ;
Exp0) ; EOF
Tipo	ID
Op3	Numero not (True ID False -
Exp2	or and) + ; - EOF
Exp1) + ; - EOF
Sec_Instruccion_p	EOF
Exp4	menor_igual mayor_igual or mayor dist) * + - / menor and igual ; EOF
Lit	menor_igual mayor_igual or mayor dist) * + - / menor and igual ; EOF
Exp3	menor_igual mayor_igual or mayor dist) + - menor and igual ; EOF
Exp1p) + ; - EOF
Exp0p) ; EOF
Exp5	menor_igual mayor_igual or mayor dist) * + - / menor and igual ; EOF
Sec_Declaracion_p	&
Exp	; EOF
Exp3p	menor_igual mayor_igual or mayor dist) + - menor and igual ; EOF
Exp2p	or and) + ; - EOF