

Memoria: Procesadores de Lenguaje - Lenguaje Tiny

Fase 2: Análisis sintáctico

Grupo G03:

Burgos Sosa Rodrigo, Cassin Gina Andrea,
Estebán Velasco Luis, Rabbia Santiago Elias

Curso 2024

1 Introducción

En el siguiente documento se expondrá una memoria sobre el desarrollo de analizadores sintácticos aplicado sobre dos lenguajes de programación, Tiny y Tiny(0) - un subconjunto de Tiny. Se presentará una especificación sintáctica de ambos lenguajes, con acondicionamiento de la gramática para permitir la implementación de un analizador sintáctico descendente predictivo recursivo, y se proporcionarán los directores de cada regla de la gramática acondicionada para solamente Tiny(0).

2 Análisis sintáctico

Es un componente central de un procesador de lenguaje, ya que dicha estructura gramatical será la base para articular los subsecuentes procesamiento del lenguaje: procesamiento dirigido por la sintaxis. Esto es lo que se especificará a continuación:

2.1 Tiny(0)

2.1.1 Especificación sintáctica (gramática)

- programa \rightarrow bloque
- bloque $\rightarrow \{ \text{declaraciones instrucciones} \}$
- declaraciones $\rightarrow \text{declaracionesAux} \&\&$
- declaraciones $\rightarrow \varepsilon$
- declaracionesAux $\rightarrow \text{declaracionesAux} ; \text{declaracionVar}$
- declaracionesAux $\rightarrow \text{declaracionVar}$
- declaracionVar $\rightarrow \text{tipo identificador}$
- tipo $\rightarrow \text{int}$
- tipo $\rightarrow \text{real}$
- tipo $\rightarrow \text{bool}$
- instrucciones $\rightarrow \text{instruccionesAux}$

- instrucciones $\rightarrow \varepsilon$
- instruccionesAux \rightarrow instruccionesAux ; instruccion
- instruccionesAux \rightarrow instruccion
- instruccion $\rightarrow @$ expr

Las siguientes expresiones se construyen de acuerdo a las siguientes prioridades (de menor a mayor prioridad):

1. Operador de asignación, es binario e infijo.
2. Operadores relacionales, son binarios e infijos.
3. +, - (binario)
4. and, or
5. *, / son binarios e infijos.
6. - (unario), not son binarios y prefijos.

- expr \rightarrow e0
- e0 \rightarrow e1 = e0 (*asocia por derecha*)
- e0 \rightarrow e1
- e1 \rightarrow e1 op1 e2 (*asocia por izquierda*)
- e1 \rightarrow e2
- e2 \rightarrow e2 + e3 (*asocia por izquierda*)
- e2 \rightarrow e3 - e3 (*no asocia*)
- e2 \rightarrow e3
- e3 \rightarrow e4 and e3 (*asocia por derecha*)
- e3 \rightarrow e4 or e4 (*no asocia*)
- e3 \rightarrow e4
- e4 \rightarrow e4 op4 e5 (*asocia por izquierda*)

- $e4 \rightarrow e5$
- $e5 \rightarrow op5\ e5$ (*prefijo*)
- $e5 \rightarrow e6$
- $e6 \rightarrow (e0)$
- $e6 \rightarrow exprBase$
- $exprBase \rightarrow literalEntero$
- $exprBase \rightarrow literalReal$
- $exprBase \rightarrow true$
- $exprBase \rightarrow false$
- $exprBase \rightarrow identificador$
- $exprBase \rightarrow null$
- $op1 \rightarrow <$
- $op1 \rightarrow >$
- $op1 \rightarrow <=$
- $op1 \rightarrow >=$
- $op1 \rightarrow ==$
- $op1 \rightarrow !=$
- $op4 \rightarrow *$
- $op4 \rightarrow /$
- $op5 \rightarrow -$
- $op5 \rightarrow not$

2.1.2 Acondicionamiento de la gramática (descendente predictivo recursivo)

Por notación, la eliminación de recursión por izquierda será denotada por Ri y la factorización por Fi .

- $\text{programa} \rightarrow \text{bloque}$
- $\text{bloque} \rightarrow \{ \text{declaraciones instrucciones} \}$
- $\text{declaraciones} \rightarrow \text{declaracionesAux} \&\&$
- $\text{declaraciones} \rightarrow \varepsilon$
- $\text{declaracionesAux} \rightarrow \text{declaracionVar R1}$
- $\text{R1} \rightarrow ; \text{declaracionVar R1}$
- $\text{R1} \rightarrow \varepsilon$
- $\text{declaracionVar} \rightarrow \text{tipo identificador}$
- $\text{tipo} \rightarrow \text{int}$
- $\text{tipo} \rightarrow \text{real}$
- $\text{tipo} \rightarrow \text{bool}$
- $\text{instrucciones} \rightarrow \text{instruccionesAux}$
- $\text{instrucciones} \rightarrow \varepsilon$
- $\text{instruccionesAux} \rightarrow \text{instruccion R2}$
- $\text{R2} \rightarrow ; \text{instruccion R2}$
- $\text{R2} \rightarrow \varepsilon$
- $\text{instruccion} \rightarrow @ \text{expr}$
- $\text{expr} \rightarrow \text{e0}$
- $\text{e0} \rightarrow \text{e1 F1}$
- $\text{F1} \rightarrow = \text{e0}$

- $F1 \rightarrow \varepsilon$
- $e1 \rightarrow e2 \text{ R3}$
- $R3 \rightarrow \text{op1 } e2 \text{ R3}$
- $R3 \rightarrow \varepsilon$
- $e2 \rightarrow e3 \text{ F2 R4}$
- $R4 \rightarrow + e3 \text{ R4}$
- $R4 \rightarrow \varepsilon$
- $F2 \rightarrow - e3$
- $F2 \rightarrow \varepsilon$
- $e3 \rightarrow e4 \text{ F3}$
- $F3 \rightarrow \text{and } e3$
- $F3 \rightarrow \text{or } e4$
- $F3 \rightarrow \varepsilon$
- $e4 \rightarrow e5 \text{ R5}$
- $R5 \rightarrow \text{op4 } e5 \text{ R5}$
- $R5 \rightarrow \varepsilon$
- $e5 \rightarrow \text{op5 } e5$
- $e5 \rightarrow e6$
- $e6 \rightarrow (e0)$
- $e6 \rightarrow \text{exprBase}$
- $\text{exprBase} \rightarrow \text{literalEntero}$
- $\text{exprBase} \rightarrow \text{literalReal}$
- $\text{exprBase} \rightarrow \text{true}$
- $\text{exprBase} \rightarrow \text{false}$

- $\text{exprBase} \rightarrow \text{identificador}$
- $\text{exprBase} \rightarrow \text{null}$
- $\text{op1} \rightarrow <$
- $\text{op1} \rightarrow >$
- $\text{op1} \rightarrow <=$
- $\text{op1} \rightarrow >=$
- $\text{op1} \rightarrow ==$
- $\text{op1} \rightarrow !=$
- $\text{op4} \rightarrow *$
- $\text{op4} \rightarrow /$
- $\text{op5} \rightarrow -$
- $\text{op5} \rightarrow \text{not}$

2.1.3 Directores de cada regla de la gramática condicionada

REGLA	DIRECTORES	ANULABLE
$\text{programa} \rightarrow \text{bloque}$	{	no
$\text{bloque} \rightarrow \{ \text{declaraciones instrucciones} \}$	{	no
$\text{declaraciones} \rightarrow \text{declaracionesAux} \&\&$	tipo	no
$\text{declaraciones} \rightarrow \varepsilon$		si
$\text{declaracionesAux} \rightarrow \text{declaracionVar R1}$	tipo	no
$\text{R1} \rightarrow ; \text{declaracionVar}$	tipo	no
$\text{R1} \rightarrow \varepsilon$		si
$\text{declaracionVar} \rightarrow \text{tipo identificador}$	tipo	no
$\text{tipo} \rightarrow \text{int}$	int	no
$\text{tipo} \rightarrow \text{real}$	real	no
$\text{tipo} \rightarrow \text{bool}$	bool	no
$\text{instrucciones} \rightarrow \text{instruccionesAux}$	@	no

instrucciones $\rightarrow \varepsilon$		si
instruccionesAux \rightarrow instruccion R2	@	no
R2 \rightarrow ; instruccion R2	@	no
R2 $\rightarrow \varepsilon$		si
instruccion \rightarrow @ expr	@	no
expr \rightarrow e0	(- not literalEntero literalReal true false identificador null	no
e0 \rightarrow e1 F1	(- not literalEntero literalReal true false identificador null	no
F1 \rightarrow = e0	=	no
F1 $\rightarrow \varepsilon$		si
e1 \rightarrow e2 R3	(- not literalEntero literalReal true false identificador null	no
R3 \rightarrow op1 e2 R3	< > <= >= == !=	no
R3 $\rightarrow \varepsilon$		si
e2 \rightarrow e3 F2 R4	(- not literalEntero literalReal true false identificador null	no
R4 \rightarrow + e3 R4	+	no
R4 $\rightarrow \varepsilon$		si
F2 \rightarrow e3	-	no
F2 $\rightarrow \varepsilon$		si
e3 \rightarrow e4 F3	(- not literalEntero literalReal true false identificador null	no
F3 \rightarrow and e3	and	no
F3 \rightarrow or e4	or	no
F3 $\rightarrow \varepsilon$		si
e4 \rightarrow e5 R5	(- not literalEntero literalReal true false identificador null	no
R5 \rightarrow op4 e5 R5	* /	no
R5 $\rightarrow \varepsilon$		si
e5 \rightarrow op5 e5	- not	no

$e5 \rightarrow e6$	(- not literalEntero literalReal true false identificador null	no
$e6 \rightarrow (e0)$	(no
$e6 \rightarrow \text{exprBase}$	(- not literalEntero literalReal true false identificador null	no
$\text{exprBase} \rightarrow \text{literalEntero}$	literalEntero	no
$\text{exprBase} \rightarrow \text{literalReal}$	literalEntero	no
$\text{exprBase} \rightarrow \text{true}$	true	no
$\text{exprBase} \rightarrow \text{false}$	false	no
$\text{exprBase} \rightarrow \text{identificador}$	identificador	no
$\text{exprBase} \rightarrow \text{null}$	null	no
$\text{op1} \rightarrow <$	<	no
$\text{op1} \rightarrow >$	>	no
$\text{op1} \rightarrow <=$	<=	no
$\text{op1} \rightarrow >=$	>=	no
$\text{op1} \rightarrow ==$	==	no
$\text{op1} \rightarrow !=$!=	no
$\text{op1} \rightarrow *$	*	no
$\text{op1} \rightarrow /$	/	no
$\text{op1} \rightarrow -$	-	no
$\text{op1} \rightarrow \text{not}$	not	no

2.2 Tiny

2.2.1 Especificación sintáctica (gramática)

- $\text{programa} \rightarrow \text{bloque}$
- $\text{bloque} \rightarrow \{ \text{declaraciones instrucciones} \}$
- $\text{declaraciones} \rightarrow \text{declaracionesAux} \&\&$
- $\text{declaraciones} \rightarrow \varepsilon$
- $\text{declaracionesAux} \rightarrow \text{declaracionesAux} ; \text{declaracion}$
- $\text{declaracionesAux} \rightarrow \text{declaracion}$
- $\text{declaracion} \rightarrow \text{declaracionVar}$

- $\text{declaracion} \rightarrow \text{declaracionTipo}$
- $\text{declaracion} \rightarrow \text{declaracionProc}$
- $\text{declaracionVar} \rightarrow \text{tipo0 identificador}$
- $\text{declaracionTipo} \rightarrow \text{type tipo0 identificador}$
- $\text{declaracionProc} \rightarrow \text{proc identificador paramsFormales bloque}$
- $\text{paramsFormales} \rightarrow (\text{paramsFormalesAux})$
- $\text{paramsFormalesAux} \rightarrow \text{paramsFormalesAux} , \text{param}$
- $\text{paramsFormalesAux} \rightarrow \text{param}$
- $\text{param} \rightarrow \text{tipo referencia identificador}$
- $\text{referencia} \rightarrow \&$
- $\text{referencia} \rightarrow \varepsilon$
- $\text{tipo0} \rightarrow \text{tipo0} [\text{literalEntero}]$
- $\text{tipo0} \rightarrow \text{tipo1}$
- $\text{tipo1} \rightarrow ^ \text{tipo1}$
- $\text{tipo1} \rightarrow \text{tipo2}$
- $\text{tipo2} \rightarrow \text{struct } \{ \text{listaCampos} \}$
- $\text{listaCampos} \rightarrow \text{listaCampos} , \text{campo}$
- $\text{listaCampos} \rightarrow \text{campo}$
- $\text{campo} \rightarrow \text{tipo0 identificador}$
- $\text{tipo2} \rightarrow \text{int}$
- $\text{tipo2} \rightarrow \text{real}$
- $\text{tipo2} \rightarrow \text{bool}$
- $\text{tipo2} \rightarrow \text{string}$
- $\text{tipo2} \rightarrow \text{identificador}$

- $\text{instrucciones} \rightarrow \text{instruccionesAux}$
- $\text{instrucciones} \rightarrow \varepsilon$
- $\text{instruccionesAux} \rightarrow \text{instruccionesAux} ; \text{instruccion}$
- $\text{instruccionesAux} \rightarrow \text{instruccion}$
- $\text{instruccion} \rightarrow @ \text{ expr}$
- $\text{instruccion} \rightarrow \text{if expr bloque}$
- $\text{instruccion} \rightarrow \text{if expr bloque else bloque}$
- $\text{instruccion} \rightarrow \text{while expr bloque}$
- $\text{instruccion} \rightarrow \text{read expr}$
- $\text{instruccion} \rightarrow \text{write expr}$
- $\text{instruccion} \rightarrow \text{nl}$
- $\text{instruccion} \rightarrow \text{new expr}$
- $\text{instruccion} \rightarrow \text{delete expr}$
- $\text{instruccion} \rightarrow \text{call identificador paramsReales}$
- $\text{paramsReales} \rightarrow (\text{ paramsRealesAux })$
- $\text{paramsReales} \rightarrow ()$
- $\text{paramsRealesAux} \rightarrow \text{paramsRealesAux} , \text{ expr}$
- $\text{paramsRealesAux} \rightarrow \text{expr}$

Las siguientes expresiones se construyen de acuerdo a las siguientes prioridades (de menor a mayor prioridad):

1. Operador de asignación, es binario e infijo.
2. Operadores relacionales, son binarios e infijos.
3. $+$, $-$ (binario)
4. and , or
5. $*$, $/$, $\%$ son binarios e infijos.

6. - (unario), not son binarios y prefijos.

7. Operadores de indexación, de acceso a registro y de indirección. Unarios posfijos, asociativos.

- $\text{expr} \rightarrow \text{e0}$
- $\text{e0} \rightarrow \text{e1} = \text{e0}$
- $\text{e0} \rightarrow \text{e1}$
- $\text{e1} \rightarrow \text{e1 op1 e2}$
- $\text{e1} \rightarrow \text{e2}$
- $\text{e2} \rightarrow \text{e2} + \text{e3}$
- $\text{e2} \rightarrow \text{e3} - \text{e3}$
- $\text{e2} \rightarrow \text{e3}$
- $\text{e3} \rightarrow \text{e4 and e3}$
- $\text{e3} \rightarrow \text{e4 or e4}$
- $\text{e3} \rightarrow \text{e4}$
- $\text{e4} \rightarrow \text{e4 op4 e5}$
- $\text{e4} \rightarrow \text{e5}$
- $\text{e5} \rightarrow \text{op5 e5}$
- $\text{e5} \rightarrow \text{e6}$
- $\text{e6} \rightarrow \text{e6 op6}$
- $\text{e6} \rightarrow \text{e7}$
- $\text{e7} \rightarrow (\text{e0})$
- $\text{e7} \rightarrow \text{exprBase}$
- $\text{exprBase} \rightarrow \text{literalEntero}$
- $\text{exprBase} \rightarrow \text{literalReal}$
- $\text{exprBase} \rightarrow \text{true}$

- $\text{exprBase} \rightarrow \text{false}$
- $\text{exprBase} \rightarrow \text{literalCadena}$
- $\text{exprBase} \rightarrow \text{identificador}$
- $\text{exprBase} \rightarrow \text{null}$
- $\text{op1} \rightarrow <$
- $\text{op1} \rightarrow >$
- $\text{op1} \rightarrow <=$
- $\text{op1} \rightarrow >=$
- $\text{op1} \rightarrow ==$
- $\text{op1} \rightarrow !=$
- $\text{op4} \rightarrow *$
- $\text{op4} \rightarrow /$
- $\text{op4} \rightarrow \%$
- $\text{op5} \rightarrow -$
- $\text{op5} \rightarrow \text{not}$
- $\text{op6} \rightarrow [\text{expr}]$
- $\text{op6} \rightarrow . \text{identificador}$
- $\text{op6} \rightarrow ^$

2.2.2 Acondicionamiento de la gramática (descendente predictivo recursivo)

Por notación, la eliminación de recursión por izquierda será denotada por Ri y la factorización por Fi .

- $\text{programa} \rightarrow \text{bloque}$
- $\text{bloque} \rightarrow \{ \text{declaraciones instrucciones} \}$
- $\text{declaraciones} \rightarrow \text{declaracionesAux} \&\&$

- declaraciones $\rightarrow \varepsilon$
- declaracionesAux \rightarrow declaracion R1
- R1 \rightarrow ; declaracion R1
- R1 $\rightarrow \varepsilon$
- declaracion \rightarrow declaracionVar
- declaracion \rightarrow declaracionTipo
- declaracion \rightarrow declaracionProc
- declaracionVar \rightarrow tipo0 identificador
- declaracionTipo \rightarrow type tipo0 identificador
- declaracionProc \rightarrow proc identificador paramsFormales bloque
- paramsFormales \rightarrow (paramsFormalesAux)
- paramsFormalesAux \rightarrow param R2
- R2 \rightarrow , param R2
- R2 $\rightarrow \varepsilon$
- param \rightarrow tipo0 referencia identificador
- referencia \rightarrow &
- referencia $\rightarrow \varepsilon$
- tipo0 \rightarrow tipo1 R3
- R3 \rightarrow [literalEntero]
- R3 $\rightarrow \varepsilon$
- tipo1 \rightarrow ^ tipo1
- tipo1 \rightarrow tipo2
- tipo2 \rightarrow struct { listaCampos }
- listaCampos \rightarrow campo R4

- $R4 \rightarrow , \text{ campo } R4$
- $R4 \rightarrow \varepsilon$
- $\text{campo} \rightarrow \text{tipo0 identificador}$
- $\text{tipo2} \rightarrow \text{int}$
- $\text{tipo2} \rightarrow \text{real}$
- $\text{tipo2} \rightarrow \text{bool}$
- $\text{tipo2} \rightarrow \text{string}$
- $\text{tipo2} \rightarrow \text{identificador}$
- $\text{instrucciones} \rightarrow \text{instruccionesAux}$
- $\text{instrucciones} \rightarrow \varepsilon$
- $\text{instruccionesAux} \rightarrow \text{instruccion } R5$
- $R5 \rightarrow ; \text{ instruccion } R5$
- $R5 \rightarrow \varepsilon$
- $\text{instruccion} \rightarrow @ \text{ expr}$
- $\text{instruccion} \rightarrow \text{if expr bloque } F1$
- $F1 \rightarrow \text{else bloque}$
- $F1 \rightarrow \varepsilon$
- $\text{instruccion} \rightarrow \text{while expr bloque}$
- $\text{instruccion} \rightarrow \text{read expr}$
- $\text{instruccion} \rightarrow \text{write expr}$
- $\text{instruccion} \rightarrow \text{nl}$
- $\text{instruccion} \rightarrow \text{new expr}$
- $\text{instruccion} \rightarrow \text{delete expr}$
- $\text{instruccion} \rightarrow \text{call identificador paramsReales}$

- $\text{paramsReales} \rightarrow (\text{F2}$
- $\text{F2} \rightarrow \text{paramsRealesAux})$
- $\text{F2} \rightarrow)$
- $\text{paramsRealesAux} \rightarrow \text{expr R6}$
- $\text{R6} \rightarrow , \text{expr R6}$
- $\text{R6} \rightarrow \varepsilon$

- $\text{expr} \rightarrow \text{e0}$
- $\text{e0} \rightarrow \text{e1 F3}$
- $\text{F3} \rightarrow = \text{e0}$
- $\text{F3} \rightarrow \varepsilon$
- $\text{e1} \rightarrow \text{e2 R7}$
- $\text{R7} \rightarrow \text{op1 e2 R7}$
- $\text{R7} \rightarrow \varepsilon$
- $\text{e2} \rightarrow \text{e3 F4 R8}$
- $\text{R8} \rightarrow + \text{e3 R8}$
- $\text{R8} \rightarrow \varepsilon$
- $\text{F4} \rightarrow - \text{e3}$
- $\text{F4} \rightarrow \varepsilon$
- $\text{e3} \rightarrow \text{e4 F5}$
- $\text{F5} \rightarrow \text{and e3}$
- $\text{F5} \rightarrow \text{or e4}$
- $\text{F5} \rightarrow \varepsilon$
- $\text{e4} \rightarrow \text{e5 R9}$
- $\text{R9} \rightarrow \text{op4 e5 R9}$

- $R9 \rightarrow \varepsilon$
- $e5 \rightarrow op5\ e5$
- $e5 \rightarrow e6$
- $e6 \rightarrow e7\ R10$
- $R10 \rightarrow op6\ R10$
- $R10 \rightarrow \varepsilon$
- $e7 \rightarrow (e0)$
- $e7 \rightarrow exprBase$
- $exprBase \rightarrow literalEntero$
- $exprBase \rightarrow literalReal$
- $exprBase \rightarrow true$
- $exprBase \rightarrow false$
- $exprBase \rightarrow literalCadena$
- $exprBase \rightarrow identificador$
- $exprBase \rightarrow null$
- $op1 \rightarrow <$
- $op1 \rightarrow >$
- $op1 \rightarrow <=$
- $op1 \rightarrow >=$
- $op1 \rightarrow ==$
- $op1 \rightarrow !=$
- $op4 \rightarrow *$
- $op4 \rightarrow /$
- $op4 \rightarrow \%$

- $\text{op5} \rightarrow -$
- $\text{op5} \rightarrow \text{not}$
- $\text{op6} \rightarrow [\text{expr}]$
- $\text{op6} \rightarrow . \text{identificador}$
- $\text{op6} \rightarrow ^$