

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

2.1.2 Acondicionamiento de la gramática (descendente predictivo recursivo)

Por notación, la eliminación de recursión por izquierda será denotada comenzando el nombre por *rec* y la factorización por *fac*.

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

- $\text{declaraciones} \rightarrow \text{declaracionesAux} \ \&\&$
- $\text{declaraciones} \rightarrow \varepsilon$
- $\text{declaracionesAux} \rightarrow \text{declaracionVar} \ \text{recDeclaracionesAux}$
- $\text{recDeclaracionesAux} \rightarrow ; \ \text{declaracionVar} \ \text{recDeclaracionesAux}$
- $\text{recDeclaracionesAux} \rightarrow \varepsilon$
- $\text{declaracionVar} \rightarrow \text{tipo} \ \text{identificador}$

- $\text{tipo} \rightarrow \text{int}$
- $\text{tipo} \rightarrow \text{real}$
- $\text{tipo} \rightarrow \text{bool}$

- $\text{instrucciones} \rightarrow \text{instruccionesAux}$
- $\text{instrucciones} \rightarrow \varepsilon$
- $\text{instruccionesAux} \rightarrow \text{instruccion} \ \text{recInstruccionesAux}$
- $\text{recInstruccionesAux} \rightarrow ; \ \text{instruccion} \ \text{recInstruccionesAux}$
- $\text{recInstruccionesAux} \rightarrow \varepsilon$
- $\text{instruccion} \rightarrow @ \ \text{expr}$

- $\text{expr} \rightarrow \text{e0}$
- $\text{e0} \rightarrow \text{e1} \ \text{facE0}$
- $\text{facE0} \rightarrow = \ \text{e0}$
- $\text{facE0} \rightarrow \varepsilon$
- $\text{e1} \rightarrow \text{e2} \ \text{recE1}$
- $\text{recE1} \rightarrow \text{op1} \ \text{e2} \ \text{recE1}$
- $\text{recE1} \rightarrow \varepsilon$
- $\text{e2} \rightarrow \text{e3} \ \text{facE2} \ \text{recE2}$

- $\text{recE2} \rightarrow + \text{e3 recE2}$
- $\text{recE2} \rightarrow \varepsilon$
- $\text{facE2} \rightarrow - \text{e3}$
- $\text{facE2} \rightarrow \varepsilon$
- $\text{e3} \rightarrow \text{e4 facE3}$
- $\text{facE3} \rightarrow \text{and e3}$
- $\text{facE3} \rightarrow \text{or e4}$
- $\text{facE3} \rightarrow \varepsilon$
- $\text{e4} \rightarrow \text{e5 recE4}$
- $\text{recE4} \rightarrow \text{op4 e5 recE4}$
- $\text{recE4} \rightarrow \varepsilon$
- $\text{e5} \rightarrow \text{op5 e5}$
- $\text{e5} \rightarrow \text{e6}$
- $\text{e6} \rightarrow (\text{e0})$
- $\text{e6} \rightarrow \text{literalEntero}$
- $\text{e6} \rightarrow \text{literalReal}$
- $\text{e6} \rightarrow \text{true}$
- $\text{e6} \rightarrow \text{false}$
- $\text{e6} \rightarrow \text{identificador}$
- $\text{op1} \rightarrow <$
- $\text{op1} \rightarrow >$
- $\text{op1} \rightarrow <=$
- $\text{op1} \rightarrow >=$
- $\text{op1} \rightarrow ==$

- $op1 \rightarrow !=$
- $op4 \rightarrow *$
- $op4 \rightarrow /$
- $op5 \rightarrow -$
- $op5 \rightarrow \text{not}$

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
$\text{declaracionesAux} \rightarrow \text{declaracionVar} \text{recDeclaracionesAux}$	tipo	no
$\text{recDeclaracionesAux} \rightarrow ; \text{declaracionVar}$	tipo	no
$\text{recDeclaracionesAux} \rightarrow \varepsilon$		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
$\text{instrucciones} \rightarrow \varepsilon$		si
$\text{instruccionesAux} \rightarrow \text{instruccion} \text{recInstruccionesAux}$	@	no
$\text{recInstruccionesAux} \rightarrow ; \text{instruccion} \text{recInstruccionesAux}$;	no
$\text{recInstruccionesAux} \rightarrow \varepsilon$		si
$\text{instruccion} \rightarrow @ \text{expr}$	@	no
$\text{expr} \rightarrow e0$	(- not literalEntero literalReal true false identificador	no

$e0 \rightarrow e1 \text{ facE0}$	(- not literalEntero literalReal true false identificador	no
$\text{facE0} \rightarrow = e0$	=	no
$\text{facE0} \rightarrow \varepsilon$		si
$e1 \rightarrow e2 \text{ recE1}$	(- not literalEntero literalReal true false identificador	no
$\text{recE1} \rightarrow \text{op1 } e2 \text{ recE1}$	< > <= >= == !=	no
$\text{recE1} \rightarrow \varepsilon$		si
$e2 \rightarrow e3 \text{ facE2 recE2}$	(- not literalEntero literalReal true false identificador	no
$\text{recE2} \rightarrow + e3 \text{ recE2}$	+	no
$\text{recE2} \rightarrow \varepsilon$		si
$\text{facE2} \rightarrow e3$	-	no
$\text{facE2} \rightarrow \varepsilon$		si
$e3 \rightarrow e4 \text{ facE3}$	(- not literalEntero literalReal true false identificador	no
$\text{facE3} \rightarrow \text{and } e3$	and	no
$\text{facE3} \rightarrow \text{or } e4$	or	no
$\text{facE3} \rightarrow \varepsilon$		si
$e4 \rightarrow e5 \text{ recE4}$	(- not literalEntero literalReal true false identificador	no
$\text{recE4} \rightarrow \text{op4 } e5 \text{ recE4}$	* /	no
$\text{recE4} \rightarrow \varepsilon$		si
$e5 \rightarrow \text{op5 } e5$	- not	no
$e5 \rightarrow e6$	(- not literalEntero literalReal true false identificador	no
$e6 \rightarrow (e0)$	(no
$e6 \rightarrow \text{literalEntero}$	literalEntero	no
$e6 \rightarrow \text{literalReal}$	literalEntero	no
$e6 \rightarrow \text{true}$	true	no
$e6 \rightarrow \text{false}$	false	no
$e6 \rightarrow \text{identificador}$	identificador	no

op1 \rightarrow <	<	no
op1 \rightarrow >	>	no
op1 \rightarrow <=	<=	no
op1 \rightarrow >=	>=	no
op1 \rightarrow ==	==	no
op1 \rightarrow !=	!=	no
op4 \rightarrow *	*	no
op4 \rightarrow /	/	no
op5 \rightarrow -	-	no
op5 \rightarrow not	not	no

2.2 Tiny

2.2.1 Especificación sintáctica (gramática)

- programa \rightarrow bloque
- bloque \rightarrow { declaraciones instrucciones }
- declaraciones \rightarrow declaracionesAux &&
- declaraciones $\rightarrow \varepsilon$
- declaracionesAux \rightarrow declaracionesAux ; declaracion
- declaracionesAux \rightarrow declaracion
- 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 paramsFormalesLista
- paramsFormalesAux $\rightarrow \varepsilon$

- $\text{paramsFormalesLista} \rightarrow \text{paramsFormalesLista} , \text{param}$
- $\text{paramsFormalesLista} \rightarrow \text{param}$
- $\text{param} \rightarrow \text{tipo referencia identificador}$
- $\text{referencia} \rightarrow \&$
- $\text{referencia} \rightarrow \varepsilon$
- $\text{tipo0} \rightarrow \text{tipo0} [\text{literalEntero}]$
- $\text{tipo0} \rightarrow \text{tipo1}$
- $\text{tipo1} \rightarrow ^ \text{tipo1}$
- $\text{tipo1} \rightarrow \text{tipo2}$
- $\text{tipo2} \rightarrow \text{struct} \{ \text{listaCampos} \}$
- $\text{listaCampos} \rightarrow \text{listaCampos} , \text{campo}$
- $\text{listaCampos} \rightarrow \text{campo}$
- $\text{campo} \rightarrow \text{tipo0 identificador}$
- $\text{tipo2} \rightarrow \text{int}$
- $\text{tipo2} \rightarrow \text{real}$
- $\text{tipo2} \rightarrow \text{bool}$
- $\text{tipo2} \rightarrow \text{string}$
- $\text{tipo2} \rightarrow \text{identificador}$
- $\text{instrucciones} \rightarrow \text{instruccionesAux}$
- $\text{instrucciones} \rightarrow \varepsilon$
- $\text{instruccionesAux} \rightarrow \text{instruccionesAux} ; \text{instruccion}$
- $\text{instruccionesAux} \rightarrow \text{instruccion}$
- $\text{instruccion} \rightarrow @ \text{expr}$
- $\text{instruccion} \rightarrow \text{if expr bloque}$

- instruccion \rightarrow if expr bloque else bloque
- instruccion \rightarrow while expr bloque
- instruccion \rightarrow read expr
- instruccion \rightarrow write expr
- instruccion \rightarrow nl
- instruccion \rightarrow new expr
- instruccion \rightarrow delete expr
- instruccion \rightarrow call identificador paramsReales
- paramsReales \rightarrow (paramsRealesAux)
- paramsRealesAux \rightarrow paramsRealesLista
- paramsRealesAux $\rightarrow \varepsilon$
- paramsRealesLista \rightarrow paramsRealesLista , expr
- paramsRealesLista \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.
 7. Operadores de indexación, de acceso a registro y de indirección. Unarios posfijos, asociativos.
- expr \rightarrow e0
 - e0 \rightarrow e1 = e0

- $e_0 \rightarrow e_1$
- $e_1 \rightarrow e_1 \text{ op1 } e_2$
- $e_1 \rightarrow e_2$
- $e_2 \rightarrow e_2 + e_3$
- $e_2 \rightarrow e_3 - e_3$
- $e_2 \rightarrow e_3$
- $e_3 \rightarrow e_4 \text{ and } e_3$
- $e_3 \rightarrow e_4 \text{ or } e_4$
- $e_3 \rightarrow e_4$
- $e_4 \rightarrow e_4 \text{ op4 } e_5$
- $e_4 \rightarrow e_5$
- $e_5 \rightarrow \text{op5 } e_5$
- $e_5 \rightarrow e_6$
- $e_6 \rightarrow e_6 \text{ op6 }$
- $e_6 \rightarrow (e_0)$
- $e_6 \rightarrow \text{literalEntero}$
- $e_6 \rightarrow \text{literalReal}$
- $e_6 \rightarrow \text{true}$
- $e_6 \rightarrow \text{false}$
- $e_6 \rightarrow \text{literalCadena}$
- $e_6 \rightarrow \text{identificador}$
- $e_6 \rightarrow \text{null}$
- $\text{op1} \rightarrow <$
- $\text{op1} \rightarrow >$

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

2.2.2 Acondicionamiento de la gramática (descendente predictivo recursivo)

Por notación, la eliminación de recursión por izquierda será denotada comenzando el nombre por *rec* y la factorización por *fac*.

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

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

- $\text{tipo2} \rightarrow \text{real}$
- $\text{tipo2} \rightarrow \text{bool}$
- $\text{tipo2} \rightarrow \text{string}$
- $\text{tipo2} \rightarrow \text{identificador}$

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

- $\text{recParamReal} \rightarrow , \text{expr recParamReal}$
- $\text{recParamReal} \rightarrow \varepsilon$
- $\text{expr} \rightarrow \text{e0}$
- $\text{e0} \rightarrow \text{e1 facE1}$
- $\text{facE1} \rightarrow = \text{e0}$
- $\text{facE1} \rightarrow \varepsilon$
- $\text{e1} \rightarrow \text{e2 recOp1}$
- $\text{recOp1} \rightarrow \text{op1 e2 recOp1}$
- $\text{recOp1} \rightarrow \varepsilon$
- $\text{e2} \rightarrow \text{e3 facE3 recSuma}$
- $\text{recSuma} \rightarrow + \text{e3 recSuma}$
- $\text{recSuma} \rightarrow \varepsilon$
- $\text{facE3} \rightarrow - \text{e3}$
- $\text{facE3} \rightarrow \varepsilon$
- $\text{e3} \rightarrow \text{e4 facE4}$
- $\text{facE4} \rightarrow \text{and e3}$
- $\text{facE4} \rightarrow \text{or e4}$
- $\text{facE4} \rightarrow \varepsilon$
- $\text{e4} \rightarrow \text{e5 recOp4}$
- $\text{recOp4} \rightarrow \text{op4 e5 recOp4}$
- $\text{recOp4} \rightarrow \varepsilon$
- $\text{e5} \rightarrow \text{op5 e5}$
- $\text{e5} \rightarrow \text{e6}$
- $\text{e6} \rightarrow \text{e6Aux recOp6}$

- $\text{recOp6} \rightarrow \text{op6 recOp6}$
- $\text{recOp6} \rightarrow \varepsilon$
- $\text{e6Aux} \rightarrow (\text{e0})$
- $\text{e6Aux} \rightarrow \text{literalEntero}$
- $\text{e6Aux} \rightarrow \text{literalReal}$
- $\text{e6Aux} \rightarrow \text{true}$
- $\text{e6Aux} \rightarrow \text{false}$
- $\text{e6Aux} \rightarrow \text{literalCadena}$
- $\text{e6Aux} \rightarrow \text{identificador}$
- $\text{e6Aux} \rightarrow \text{null}$
- $\text{op1} \rightarrow <$
- $\text{op1} \rightarrow >$
- $\text{op1} \rightarrow <=$
- $\text{op1} \rightarrow >=$
- $\text{op1} \rightarrow ==$
- $\text{op1} \rightarrow !=$
- $\text{op4} \rightarrow *$
- $\text{op4} \rightarrow /$
- $\text{op4} \rightarrow \%$
- $\text{op5} \rightarrow -$
- $\text{op5} \rightarrow \text{not}$
- $\text{op6} \rightarrow [\text{expr}]$
- $\text{op6} \rightarrow . \text{identificador}$
- $\text{op6} \rightarrow ^$