

PROCESADORES DE LENGUAJES

FASE 1.1:

DESARROLLO DE ANALIZADOR LÉXICO PARA TINY(0).



Grupo 07:

HongXiang Chen y
Andrés Teruel Fernández

1. CLASES LÉXICAS DE TINY(0):

En esta sección incluimos un listado con todas las clases léxicas del lenguaje Tiny(0) junto con sus respectivas especificaciones formales.

Antes de enumerar las clases, consideramos conveniente declarar las siguientes **definiciones regulares** a tener en cuenta para las especificaciones formales de cada clase:

- digito $\rightarrow [0 - 9]$
- digitoPos $\rightarrow [1 - 9]$
- letra $\rightarrow [a - z] \mid [A - Z]$
- ParteDecimal $\rightarrow . (digito^* digitoPos \mid 0)$
- ParteExp (e | E) NumEnt

Clases léxicas:

1. **Identificador:** una clase multivaluada que engloba los identificadores de las variables. Un identificador siempre empieza por una letra, tras esto puede contener cualquier sucesión de letras, dígitos y ‘_’.

Identificador $\rightarrow letra (letra \mid digito \mid _)^*$

2. **Separador:** una clase univaluada dedicada a la cadena ‘&&’.

Separador $\rightarrow \&\&$

3. **Int:** clase univaluada para el nombre de tipo ‘int’.

Int $\rightarrow int$

4. **Real:** clase univaluada para el nombre de tipo ‘real’.

Real $\rightarrow real$

5. **Bool:** clase univaluada para el nombre de tipo ‘bool’.

Bool $\rightarrow bool$

6. **NumEnt:** clase multivaluada para expresiones básicas de tipo int. Un número entero puede opcionalmente empezar por '+' o '-' a lo que le sigue una serie de uno o más dígitos (no puede haber ceros no significativos a la izquierda).

$$\text{NumEnt} \rightarrow (+ \mid -)? (\text{digitoPos } (\text{digito})^* \mid 0)$$

7. **NumReal:** clase multivaluada para expresiones básicas de tipo real. Un número real puede opcionalmente empezar por '+' o '-' a lo que le sigue un número entero seguido de una parte decimal (un '.' seguido de una serie de números que no puede tener ceros no significativos a la izquierda), una parte exponencial (una 'e' o una 'E' seguida de un número entero) o por una parte decimal seguida de una exponencial.

$$\text{NumReal} \rightarrow \text{NumEnt} (\text{ParteDecimal} \mid \text{ParteExp} \mid \text{ParteDecimal ParteExp})$$

8. **Boolean:** clase multivaluada para expresiones básicas de tipo bool.

$$\text{Boolean} \rightarrow \text{true} \mid \text{false}$$

9. **Asignacion:** clase univaluada para el símbolo de asignación '='.

$$\text{Asignación} \rightarrow =$$

10. **PuntoComa:** una clase univaluada para el símbolo de ';'.

$$\text{PuntoComa} \rightarrow ;$$

11. **ParAbierto:** una clase univaluada para el símbolo de '('.

$$\text{ParAbierto} \rightarrow ($$

12. **ParCerrado:** una clase univaluada para el símbolo de ')'.

$$\text{ParCerrado} \rightarrow)$$

13. **Suma:** una clase univaluada para el operador de suma '+'.

$$\text{Suma} \rightarrow +$$

14. **Resta:** una clase univaluada para el operador de resta '-'.

$$\text{Resta} \rightarrow -$$

15. Div: una clase univaluada para el operador de suma '/'.

Div \rightarrow /

16. Mult: una clase univaluada para el operador de suma '*'.

Mult \rightarrow *

17. And: una clase univaluada para el operador lógico 'and'.

And \rightarrow and

18. Or: una clase univaluada para el operador lógico 'or'.

Or \rightarrow or

19. Not: una clase univaluada para el operador lógico 'not'.

Not \rightarrow not

20. Mayor: una clase univaluada para el operador de suma '>'.

Mayor \rightarrow >

21. Menor: una clase univaluada para el operador de suma '<'.

Menor \rightarrow <

22. MayorIgual: una clase univaluada para el operador de suma '>='.

MayorIgual \rightarrow >=

23. MenorIgual: una clase univaluada para el operador de suma '<='.

MenorIgual \rightarrow <=

24. Igual: una clase univaluada para el operador de suma '=='.

Igual \rightarrow ==

