

PROCESADORES DE LENGUAJES

FASE 1.1:

DESARROLLO DE ANALIZADOR LÉXICO PARA TINY(1).



Grupo 07:

HongXiang Chen y
Andrés Teruel Fernández

1. CLASES LÉXICAS DE TINY(1):

En esta sección incluimos un listado con todas las clases léxicas del lenguaje Tiny(1) 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 '='.

$$\text{Asignacion} \rightarrow =$$

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 → /

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

Mult → *

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

And → and

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

Or → or

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

Not → not

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

Mayor → >

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

Menor → <

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

MayorIgual → >=

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

MenorIgual → <=

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

Igual → ==

25. Diferente: una clase univaluada para el operador de suma '!='.

Diferente → !=

26. Porcentaje: una clase univaluada para el operador '%'.

Porcentaje → %

27. CorcheteAbierto: una clase univaluada para el símbolo '['.

CorcheteAbierto → [

28. CorcheteCerrado: una clase univaluada para el símbolo ']'.

CorcheteCerrado →]

29. LlaveAbierta: una clase univaluada para el símbolo '{'.

LlaveAbierta → {

30. LlaveCerrada: una clase univaluada para el símbolo '}'.

LlaveCerrada → }

31. Punto: una clase univaluada para el símbolo '.'.

Punto → .

32. Flecha: una clase univaluada para el símbolo '->'.

Flecha → ->

33. Coma: una clase univaluada para el símbolo ','.

Coma → ,

34. Ampersand: una clase univaluada para el símbolo '&'.

Ampersand → &

35. String: una clase univaluada para el nombre de tipo 'string'.

String → string

36. Null: una clase univaluada para el nombre de tipo 'null'.

Null → null

37. Proc: una clase univaluada para el nombre de tipo 'proc'.

Proc → proc

38. If: una clase univaluada para el nombre de tipo 'if'.

If → if

39. Then: una clase univaluada para el nombre de tipo 'then'.

Then → then

40. Else: una clase univaluada para el nombre de tipo 'else'.

Else → else

41. EndIf: una clase univaluada para el nombre de tipo 'EndIf'.

EndIf → endif

42. While: una clase univaluada para el nombre de tipo 'while'.

While → while

43. Do: una clase univaluada para el nombre de tipo 'do'.

Do → do

44. Endwhile: una clase univaluada para el nombre de tipo 'endwhile'.

Endwhile → endwhile

45. Call: una clase univaluada para el nombre de tipo 'call'.

Call → call

46. Record: una clase univaluada para el nombre de tipo 'record'.

Record → record

47. Array: una clase univaluada para el nombre de tipo 'array'.

Array → array

48. Of: una clase univaluada para el nombre de tipo 'of'.

Of → of

49. Pointer: una clase univaluada para el nombre de tipo 'pointer'.

Pointer → pointer

50. New: una clase univaluada para el nombre de tipo 'new'.

New → new

51. Delete: una clase univaluada para el nombre de tipo 'delete'.

Delete → delete

52. Read: una clase univaluada para el nombre de tipo 'read'.

Read → read

53. Write: una clase univaluada para el nombre de tipo 'write'.

Write → write

54. Nl: una clase univaluada para el nombre de tipo 'nl'.

Nl → nl

55. Var: una clase univaluada para el nombre de tipo 'var'.

$\text{Var} \rightarrow \text{var}$

56. Type: una clase univaluada para el nombre de tipo 'type'.

$\text{Type} \rightarrow \text{type}$

57. Cadenas ignorables: una clase multivaluada para comentar una línea, que comienza por # y seguido de 0 o más caracteres, a excepción de un salto de línea.

$\text{comentario} \rightarrow \#(\overline{NL|EOF})^*$

$\text{ignorable} \rightarrow \text{ESPACIO} | \backslash n | \backslash b | \backslash r$

58. EOF: una clase univaluada para el indicador de fin del fichero.

$\text{EOF} \rightarrow \text{EOF}$

59. Cadena: una clase multivaluada para todos los literales cadena. Comienzan y acaban por dobles comillas y entre medias pueden tener cualquier carácter distinto de: ("), retroceso (\b), retorno de carro (\r), y salto de línea (\n).

$\text{Cadena} \rightarrow \text{"}(\overline{\backslash b | \backslash n | \backslash r | EOF | "})^*\text{"}$