

Fase 1: Desarrollo de analizador léxico para Tiny(0) y Tiny

G01

Esther Babon Arcauz

Pablo Campo Gómez

Claudia López-Mingo Moreno

José Antonio Ruiz Heredia

Índice

1. Enumeración de las clases léxicas de Tiny(0)	2
2. Especificación formal del léxico del lenguaje mediante definiciones regulares	3
3. Diseño de un analizador léxico para el lenguaje mediante un diagrama de transiciones	4
4. Enumeración de las clases léxicas de Tiny	5
5. Especificación formal del léxico del lenguaje mediante definiciones regulares	6

1. Enumeración de las clases léxicas de Tiny(0):

Aunque no exista una receta universal para identificar las clases léxicas, vamos a seguir algunas guías:

- Cada símbolo de puntuación y cada operador da lugar a una clase léxica univaluada separada.
- Cada palabra reservada da lugar a una clase léxica univaluada separada.
- Cada tipo literal a una clase léxica multivaluada.
- Lo mismo pasa con los identificadores: clase léxica multivaluada identificador.

En Tiny(0) encontramos las siguientes clases léxicas:

- **Variables:** Palabras reservadas que tienen asociadas un valor específico. Comienzan con una letra o un guión bajo, seguido de letras, dígitos o más guiones bajos.
- **Literales enteros:** Número incluido en el conjunto de los números enteros. Pueden incluir un signo opcional (+ o -) seguido de una secuencia de uno o más dígitos, sin ceros no significativos al principio.
- **Literales reales:** Número incluido en el conjunto de los números reales. Pueden tener las siguientes partes:
 - Parte entera: Sigue el formato de los literales enteros.
 - Parte decimal: Comienza con un punto seguido de uno o más dígitos.
 - Parte exponencial: Comienza con 'e' o 'E' seguido de uno o más dígitos.
- **Literales booleanos:** Representación de un valor de tipo booleano, es decir que se incluye dentro del conjunto de los valores 'true' y 'false'.
- **Operadores:** Símbolos y signos que se utilizan para realizar las siguientes operaciones:
 - Operadores aritméticos: suma, resta, multiplicación, división.
 - Operadores lógicos: and, not, or.
 - Operadores relacionales: mayor, menor, mayor igual, menor igual, igual, desigual.
 - Operador de asignación.
- **Símbolos de puntuación:** Elementos gramaticales utilizados para estructurar y organizar el código.
 - Paréntesis de apertura y paréntesis de cierre.
- **Palabras reservadas:** Palabras especiales que tienen un significado específico en el lenguaje y no pueden ser usadas como identificadores.
 - Tipos de datos: int, real, bool.
 - Valores: true, false.
 - Operadores: and, not, or.

2. Especificación formal del léxico del lenguaje mediante definiciones regulares:

- **Alfabetos**

- letra $\equiv [a-z, A-Z, _]$
- digitoPositivo $\equiv [1-9]$
- digito $\equiv \{\text{digitoPositivo}\} \mid 0$
- parteEntera $\equiv (\{\text{digitoPositivo}\} \{\text{digito}\}^* \mid 0)$
- parteDecimal $\equiv (\{\text{digito}\}^* \{\text{digitoPositivo}\} \mid 0)$
- parteExponencial $\equiv [\backslash e, \backslash E] [\backslash +, \backslash -]? (\{\text{digitoPositivo}\} \{\text{digito}\}^* \mid 0)$

- **Palabras reservadas**

- int $\equiv (i|I)(n|N)(t|T)$
- real $\equiv (r|R)(e|E)(a|A)(l|L)$
- bool $\equiv (b|B)(o|O)(o|O)(l|L)$
- true $\equiv (t|T)(r|R)(u|U)(e|E)$
- false $\equiv (f|F)(a|A)(l|L)(s|S)(e|E)$
- and $\equiv (a|A)(n|N)(d|D)$
- not $\equiv (n|N)(o|O)(t|T)$
- or $\equiv (o|O)(r|R)$

- **Literales**

- literalEntero $\equiv [\backslash +, \backslash -]? \{\text{parteEntera}\}$
- literalReal $\equiv \{\text{literalEntero}\} (\backslash . \{\text{parteDecimal}\})? \{\text{parteExponencial}\}?$
- variable $\equiv \{\text{letra}\} (\{\text{letra}\} \mid \{\text{digito}\})^*$

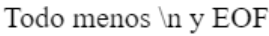
- **Operadores**

- suma $\equiv \backslash +$
- resta $\equiv \backslash -$
- mul $\equiv \backslash *$
- div $\equiv /$
- mayor $\equiv \backslash >$
- menor $\equiv \backslash <$
- mayorIgual $\equiv \backslash >=$
- menorIgual $\equiv \backslash <=$
- igual $\equiv ==$
- desigual $\equiv !=$
- asig $\equiv =$

- **Símbolos de puntuación**

- parA $\equiv \backslash ($
- parC $\equiv \backslash)$
- arroba $\equiv \backslash @$

diagrama de transiciones:



4. Enumeración de las clases léxicas de Tiny:

Aunque no exista una receta universal para identificar las clases léxicas, vamos a seguir algunas guías:

- Cada símbolo de puntuación y cada operador da lugar a una clase léxica univaluada separada.
- Cada palabra reservada da lugar a una clase léxica univaluada separada.
- Cada tipo literal a una clase léxica multivaluada.
- Lo mismo pasa con los identificadores: clase léxica multivaluada identificador.

En Tiny encontramos las siguientes clases léxicas:

- **Variables:** Palabras reservadas que tienen asociadas un valor específico. Comienzan con una letra o un guión bajo, seguido de letras, dígitos o más guiones bajos.
- **Literales enteros:** Número incluido en el conjunto de los números enteros. Pueden incluir un signo opcional (+ o -) seguido de una secuencia de uno o más dígitos, sin ceros no significativos al principio.
- **Literales reales:** Número incluido en el conjunto de los números reales. Pueden tener las siguientes partes:
 - Parte entera: Sigue el formato de los literales enteros.
 - Parte decimal: Comienza con un punto seguido de uno o más dígitos.
 - Parte exponencial: Comienza con 'e' o 'E' seguido de uno o más dígitos.
- **Literales cadena:** Representa un valor de tipo string, comienza con comillas dobles (") seguidas por la cadena vacía o una cadena que contiene cualquier caracter que no sean comillas y termina con comillas dobles.
- **Operadores:** Símbolos y signos que se utilizan para realizar las siguientes operaciones:
 - Operadores aritméticos: suma, resta, multiplicación, división, módulo.
 - Operadores lógicos: and, not, or.
 - Operadores relacionales: mayor, menor, mayor igual, menor igual, igual, desigual.
 - Operador de asignación.
- **Símbolos de puntuación:** Elementos gramaticales utilizados para estructurar y organizar el código.
 - Paréntesis de apertura y cierre, punto y coma, corchete de apertura y cierre, llave de apertura y cierre, punto, & , &&, arroba
- **Palabras reservadas:** Palabras especiales que tienen un significado específico en el lenguaje y no pueden ser usadas como identificadores.
 - Tipos de datos: int, real, bool, string, struct.
 - Valores: true, false, null.
 - Operadores: and, not, or.

- Condicionales: if, else, while
- Instrucciones: proc, new, delete, read, write, nl, type, call

5. Especificación formal del léxico del lenguaje mediante definiciones regulares:

● Alfabetos

- letra $\equiv [a-z, A-Z, _]$
- digitoPositivo $\equiv [1-9]$
- digito $\equiv \{\text{digitoPositivo}\} \mid 0$
- parteEntera $\equiv (\{\text{digitoPositivo}\} \{\text{digito}\}^* \mid 0)$
- parteDecimal $\equiv (\{\text{digito}\}^* \{\text{digitoPositivo}\} \mid 0)$
- parteExponencial $\equiv [\backslash e, \backslash E] [\backslash +, \backslash -]? (\{\text{digitoPositivo}\} \{\text{digito}\}^* \mid 0)$

● Palabras reservadas

- int $\equiv (i|I)(n|N)(t|T)$
- real $\equiv (r|R)(e|E)(a|A)(l|L)$
- bool $\equiv (b|B)(o|O)(o|O)(l|L)$
- string $\equiv (s|S)(t|T)(r|R)(i|I)(n|N)(g|G)$
- null $\equiv (n|N)(u|U)(l|L)(l|L)$
- proc $\equiv (p|P)(r|R)(o|O)(c|C)$
- if $\equiv (i|I)(f|F)$
- else $\equiv (e|E)(l|L)(s|S)(e|E)$
- while $\equiv (w|W)(h|H)(i|I)(l|L)(e|E)$
- struct $\equiv (s|S)(t|T)(r|R)(u|U)(c|C)(t|T)$
- new $\equiv (n|N)(e|E)(w|W)$
- delete $\equiv (d|D)(e|E)(l|L)(e|E)(t|T)(e|E)$
- read $\equiv (r|R)(e|E)(a|A)(d|D)$
- write $\equiv (w|W)(r|R)(i|I)(t|T)(e|E)$
- nl $\equiv (n|N)(l|L)$
- type $\equiv (t|T)(y|Y)(p|P)(e|E)$
- call $\equiv (c|C)(a|A)(l|L)(l|L)$
- true $\equiv (t|T)(r|R)(u|U)(e|E)$
- false $\equiv (f|F)(a|A)(l|L)(s|S)(e|E)$
- and $\equiv (a|A)(n|N)(d|D)$
- not $\equiv (n|N)(o|O)(t|T)$
- or $\equiv (o|O)(r|R)$

● Literales

- identificador $\equiv \{\text{letra}\} \{\{\text{letra}\} \{\text{digito}\}\}^*$
- literalEntero $\equiv [\backslash +, \backslash -]? \{\text{parteEntera}\}$
- literalReal $\equiv \{\text{literalEntero}\} (\backslash. \{\text{parteDecimal}\})? \{\text{parteExponencial}\}?$

- literalCadena \equiv " [^"]? "

- **Operadores**

- suma \equiv \+
- resta \equiv \-
- mul \equiv *
- div \equiv \/
- mod \equiv \%
- mayor \equiv \>
- menor \equiv \<
- mayorIgual \equiv \>=
- menorIgual \equiv \<=
- igual \equiv \==
- desigual \equiv \!=
- asig \equiv \=
- parA \equiv \((
- parC \equiv \)
- puntoYComa \equiv \;
- punto \equiv \.
- corcheteA \equiv \[
- corcheteC \equiv \]
- llaveA \equiv \{
- llaveC \equiv \}
- arroba \equiv \@
- ampersand \equiv \&
- ampersand2 \equiv \&&

- **Cadenas ignorables**

separador \equiv [, \t, \r, \b, \n]

comentario \equiv ## [^\n]?