

Práctica
Procesadores de Lenguajes

David Antuña Rodríguez
Javier Carrión García

Contenidos

1	Fase 1	1
1.1	Clases léxicas	1
1.2	Especificación formal	3
1.3	Diagrama de transiciones	4

1 Fase 1

1.1 Clases léxicas

La descripción de las clases léxicas identificadas se hará de manera informal, en lenguaje natural.

- **SPROG**

Es el separador && que indica el fin de la sección de declaraciones y el comienzo de la de instrucciones.

- **LREAL**

Empiezan con un signo (+ o -) opcional, a continuación aparecen uno o más dígitos cualesquiera. Seguida de esta parte puede aparecer una decimal que consta de un punto seguido de uno o más dígitos cualesquiera. Por último, tiene una E o e seguida de un signo (+ o -), opcional, y de uno o más dígitos cualesquiera.

- **ID**

Comienza por una letra cualquiera y la sigue una secuencia de cero o más letras, dígitos o subrayado(_).

- **BOOL**

Es una palabra reservada que se conforma por las letras minúsculas: b, o, o, l. En ese orden.

- **NUM**

Palabra reservada formada por las letras minúsculas: n, u, m. En ese orden.

- **TRUE**

Es una palabra reservada compuesta por las letras minúsculas: t, r, u, e. En ese orden.

- **FALSE**

Palabra reservada que contiene las siguientes letras minúsculas: f, a, l, s, e. En ese orden.

- **PLUS**

Representa una suma, \+.

- **MINUS**

Representa una resta, \-.

- **MUL**

Representa la multiplicación, *.

- **DIV**
Representa la división, $/$.
- **IS**
Representación de la asignación, $=$.
- **EQ**
Representa una comparación, $==$.
- **GT**
Representa el mayor que, $>$.
- **GEQ**
Representa el mayor o igual que, $>=$.
- **LT**
Representa el menor que, $<$.
- **LEQ**
Representa el menor o igual que, $<=$.
- **NEQ**
Representa una desigualdad, $!=$.
- **AND**
Representa el operador lógico and.
- **OR**
Representa el operador lógico or.
- **NOT**
Representa el operador lógico not.
- **POP**
Representa un paréntesis de apertura, $($.
- **PCL**
Representa un paréntesis de cierre, $)$.
- **EOL**
Representa el punto y coma como separador especial, $;$.

1.2 Especificación formal

Vamos a utilizar DR_s para dar una descripción formal del lenguaje que conforman las clases léxicas del apartado 1.1.

- (*) $SPROG \equiv \&\&$
- (*) $LREAL \equiv \underline{L}ENT \ \underline{P}DEC? \ \underline{P}EXP?$
 $PDEC \equiv \backslash. \ \underline{Dig}^* \ \underline{Dig}$
 $PEXP \equiv (E \mid e) \ \underline{L}ENT$
 $LENT \equiv \underline{Sign}? \ \underline{Dig}^* \ \underline{Dig}$
 $Sign \equiv [\backslash+, \backslash-]$
 $Dig \equiv [0-9]$
- (*) $ID \equiv \underline{Letter} \ (\underline{Letter} \mid \underline{Dig} \mid -)^*$
 $Letter \equiv [a-z, A-Z]$
 $Dig \equiv [0-9]$
- (*) $BOOL \equiv b \ o \ o \ l$
- (*) $NUM \equiv n \ u \ m$
- (*) $TRUE \equiv t \ r \ u \ e$
- (*) $FALSE \equiv f \ a \ l \ s \ e$
- (*) $PLUS \equiv \backslash+$
- (*) $MINUS \equiv \backslash-$
- (*) $MUL \equiv \backslash*$
- (*) $DIV \equiv /$
- (*) $IS \equiv =$
- (*) $EQ \equiv ==$
- (*) $GT \equiv >$
- (*) $GEQ \equiv >=$
- (*) $LT \equiv <$
- (*) $LEQ \equiv <=$
- (*) $NEQ \equiv !=$
- (*) $AND \equiv a \ n \ d$
- (*) $OR \equiv o \ r$
- (*) $NOT \equiv n \ o \ t$
- (*) $POP \equiv \backslash($
- (*) $PCL \equiv \backslash)$

$$(*) \text{ EOL} \equiv ;$$

[I] SEP \equiv [' ', \t, \n, \r, \b]

1.3 Diagrama de transiciones

