ANALIZADOR LÉXICO TINY(1)

Grupo 29
Adrian Estevez Gallego
Alvaro Delgado Gutierrez

1. Identificación y definición informal de las clases léxicas

Suponemos que el lenguaje diferencia entre mayúsculas y minúsculas.

Clases Léxicas

- **EntNum**: Identifica los números enteros. Los números enteros empiezan opcionalmente por un signo + o -. Seguidamente debe aparecer una secuencia de uno o más dígitos, no se admiten 0s no significativos a la izquierda.
- **RealNum**: Identifica los números reales. Los números reales son aquellos con una parte entera (misma especificación que EntNum) y adicionalmente cuentan con una parte decimal y/o una parte exponencial.
- **Variable**: Identifica los nombres de variables. Estos deben comenzar por una letra seguida de una secuencia de 0 o más letras, dígitos o subrayados (“_”).
- **LitCad**: Identifica los strings (cadenas de caracteres), estos empiezan por comilla doble (“) seguida de una secuencia de 0 o más caracteres distintos de “, retroceso (\b), retorno de carro (\r), y salto de línea (\n), seguida de “.
- **True** : Identifica la palabra reservada para un booleano cierto.
- **False** : Identifica la palabra reservada para un booleano falso.
- **Null** : Identifica la palabra reservada *“null”* para valor nulo.
- **Proc** : Identifica la palabra reservada *“proc”* para la creación de un procedimiento.
- **If** : Identifica la palabra reservada *“if”* para la creación de condiciones.
- **Then** : Identifica la palabra reservada *“then”* para la creación de condiciones.
- **Else** : Identifica la palabra reservada *“else”* para la creación de condiciones.
- **Endif**: Identifica la palabra reservada *“endif”* para la creación de condiciones.
- **While** : Identifica la palabra reservada *“while”* para la creación de bucles while.

- **Do**: Identifica la palabra reservada *“do”* para la creación de bucles while.
- **Endwhile** : Identifica la palabra reservada *“endwhile”* para la creación de bucles while.
- **Call** : Identifica la palabra reservada *“call”* para la invocación de procedimientos.
- **Type** : Identifica la palabra reservada *“type”* para la declaración de tipos.
- **Record** : Identifica la palabra reservada *“record”* para la declaración de tipos.
- **New** : Identifica la palabra reservada *“new”* para la reserva de memoria.
- **Delete** : Identifica la palabra reservada *“delete”* para la liberación de memoria.
- **Read** : Identifica la palabra reservada *“read”* para la instrucción de lectura.
- **Write** : Identifica la palabra reservada *“write”* para la instrucción de escritura.
- **NL** : Identifica la palabra reservada *“nl”* para la instrucción de nueva línea.
- **Var** : Identifica la palabra reservada *“var”* para la declaración de variables.
- **Bool**: Identifica la palabra reservada *“bool”* para el tipo booleano.
- **Int**: Identifica la palabra reservada *“int”* para el tipo entero.
- **Real** : Identifica la palabra reservada “*real*” para el tipo de números reales.
- **String** : Identifica la palabra reservada *“string”* para el tipo de cadena de caracteres.
- **Pointer** : Identifica la palabra reservada *“pointer”* para la declaración de punteros.
- **Array** : Identifica la palabra reservada *“array”* para la declaración de arrays.
- **Of** : Identifica la palabra reservada *“of”* para la declaración de arrays.
- **Sum** : Identifica la suma, designada por *“+”*.
- **Res** : Identifica la resta, designada por *“-”*.
- **Mul** : Identifica la multiplicación, designada por *“*”*.

- **Div** : Identifica la división, designada por *"/"*.
- **Mod** : Identifica el módulo, designado por *"%"*.
- **Asig** : Identifica la instrucción de asignación a una variable designada por el símbolo *"="*.
- **Menor** : Identifica el operador relacional *"<"*.
- **Mayor** : Identifica el operador relacional *">"*.
- **Ig** : Identifica el operador relacional *"=="*.
- **MayIg** : Identifica el operador relacional *">="*.
- **MenIg** : Identifica el operador relacional *"<="*.
- **Dist** : Identifica el operador relacional *"!="*.
- **And** : Identifica el operador lógico *"and"*.
- **Or** : Identifica el operador lógico "*or*".
- **Not** : Identifica el operador lógico "*not*".
- **PAp** : Identifica la apertura de un paréntesis *"("*.
- **PCie** : Identifica el cierre de un paréntesis *")"*.
- **CAp** : Identifica la apertura de un corchete *"["*.
- **CCie** : Identifica el cierre de un corchete *"]"*.
- **LAp** : Identifica la apertura de una llave *"{"*.
- **LCie** : Identifica el cierre de una llave *"}"*.
- **Punto** : Identifica el punto *"."*.
- **Coma** : Identifica la coma *","*.
- **Andvers** : Identifica el símbolo *"&"*.
- **Flecha** : Identifica la cadena *"->"*.
- **PunCo** : Identifica el final de una sentencia designado por *";"*.
- **Sep** : Identifica el separador entre la sección de declaraciones y la sección de instrucciones. Este separador es *"&&"*.

Cadenas ignorables

- Espacio
- Tabulación
- Salto de línea
- Retroceso
- Retorno de carro
- Comentario

2. Especificación formal de las clases léxicas

Definiciones auxiliares

- DigitoPos = [1-9]
- Digito = (0 | [1-9])
- Ent = ((DigitoPos Digito*) | 0)
- PDec = \. (Digito* DigitoPos| 0)
- PExp = (e | E) (\- | \+)? ((DigitoPos Digito*) | 0)
- letra = [a-z, A-Z]

Clases léxicas

- EntNum = (\- | \+)? Ent
- RealNum = (\- | \+)? Ent (PDec | PExp | PDec PExp)
- Variable = letra (letra | Digito | _)*
- LitCad = \" [^\" ,^\b , ^\r , ^\n]* \"
- True = true
- False = false
- Null = null
- Proc = proc
- If = if
- Then = then
- Else = else
- Endif = endif
- While = while
- Do = do
- Endwhile = endwhile
- Call = call
- Type = type
- Record = record
- New = new
- Delete = delete
- Read = read
- Write = write
- NL = nl

- Var = var
- Bool = bool
- Int = int
- Real = real
- String = string
- Pointer = pointer
- Array = array
- Of = of
- Sum = \+
- Res = \-
- Mul = *
- Div = \/
- Mod = \%
- Asig = \=
- Menor = <
- Mayor = >
- Ig = \= \=
- MayIg = > \=
- MenIg = < \=
- Dist = ! \=
- And = and
- Or = or
- Not = not
- PAp = \(
- PCie = \)
- CAp = \[
- CCie = \]
- LAp = \{
- LCie = \}
- Punto = \.
- Coma = \,
- Andvers = \&
- Flecha = \- \>
- PunCo = \;
- Sep = \&\&

Cadenas ignorables

- Esp = <espacio>
- Tab = \t
- Salto = \n
- Back = \b
- Retorno = \r
- Comentario = \# ^\n