



INSTITUTO TECNOLÓGICO DE MÉXICO

CAMPUS PACHUCA

LENGUAJES Y AUTOMATAS

4.1 Tabla de tokens para proyecto de Analizador léxico

Baume Lazcano Rodolfo

Espinosa Antelis Gabriela Dyvheke

Garcia Cruz Andres

14 MAYO 2024

Para elaborar un analizador léxico (o lexer), es fundamental definir las reglas básicas del lenguaje de programación que se desea analizar. Estas reglas determinan cómo se identificarán y clasificará cada componente del código fuente.

1. Palabras Clave (Keywords)

Las palabras clave son términos reservados por el lenguaje de programación que tienen un significado específico y no pueden usarse como identificadores (nombres de variables, funciones, etc.).

KEYWORDS	TOKEN	PATRON REGULAR
if	condicion	Palabras reservadas por el lenguaje
then	entonces	
else	sino	
case	caso	
switch	alterna	
while	mientras	
for	para	
(ALL DATA TYPES)		

2. Identificadores

Los identificadores son nombres utilizados para variables, funciones, clases, etc. Las reglas para los identificadores varían entre lenguajes:

Se identifican con '___'

Deben comenzar con una letra (mayúscula o minúscula).

Pueden contener letras y dígitos después del primer carácter. Mas no puede empezar con numeros

No pueden ser una palabra clave.

IDENTIFIERS		PATRON REGULAR	TOKENS
__	Definidos por '___'	$\wedge_([a-zA-Z][a-zA-Z0-9]*)^*\$$	$\wedge_([a-zA-Z][a-zA-Z0-9]*)^*\$$

3. Operadores

Los operadores son símbolos que representan operaciones. Se dividen en varias categorías:

Aritméticos: +, -, *, /, %.

Relacionales: ==, !=, <, >, <=, >=.

Lógicos: &&, ||, !.

4. Símbolos Especiales

Estos son caracteres o combinaciones de caracteres que tienen significados especiales en el lenguaje:

Delimitadores: {}, (), [] (para bloques de código, parámetros de funciones, arreglos).

Punto y coma: ; (para terminar declaraciones en lenguajes como C, C++, Java).

Comas: , (para separar elementos en listas, parámetros en funciones).

OPERATORS/SYMBOLS		PATRON REGULAR	TOKENS
=	asignacion	Operadores aritmeticos, logicos y caracteres especiales	Símbolo en cuestion
+	adicion		
-	substraccion		
/	division		
*	multiplicacion		
<	mayor que		
%	modulo		
==	identico a		
>=	mayor igual que		
<=	menor igual que		
>	menor que		
&&	and		
	or		
:	dos puntos		
;	punto y coma		
.	punto		
,	coma		
(parentesis		
)	parentesis		
{	llave		
}	llave		
[corchete		
]	corchete		

5. Comentarios

Los comentarios son anotaciones en el código fuente que el compilador o intérprete ignora. Pueden ser:

De una sola línea: Comienzan con // en C, C++, Java, y # en Python.

De múltiples líneas: Encerrados entre /* y */ en C, C++, Java.

COMMENTS		PATRON REGULAR	TOKENS
//	Definidos por '/'	\\\[a-zA-Z0-9]*\$	\\\[a-zA-Z0-9]*\$

6. Tipos de dato

Se encuentran dentro de las palabras reservadas, sin embargo, al manipularse mediante números, también podemos asignarle su patrón regular para que nuestro analizador conozca como es que debe manejarlos.

DATA TYPE	TOKEN	PATRON REGULAR
int	entero	[+]?[0-9]+
float	decimal	[+]?([0-9]*\.[0-9]+ [0-9]+\.[0-9]*)
char	carácter	[a-zA-Z_][a-zA-Z0-9_]*
long	entero largo	[+]?[0-9]+L

7. Validaciones específicas

VALIDATIONS	PATRON REGULAR	TOKENS
Correo electronico	^[a-zA-Z0-9_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$	^[a-zA-Z0-9_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$
Telefono	^\+[1-9]\d{1,10}\$	^\+[1-9]\d{1,10}\$
Código Postal	^\d{5}\$	^\d{5}\$
Suma	^\s*\d+(\.\d+)?\s*\+ \s*\d+(\.\d+)?\s*\$	Operador suma
Resta	^\s*\d+(\.\d+)?\s*\- \s*\d+(\.\d+)?\s*\$	Operador resta
Multiplcacion	^\s*\d+(\.\d+)?\s** \s*\d+(\.\d+)?\s*\$	Operador multiplicacion
Division	^\s*\d+(\.\d+)?\s*\/ \s*\d+(\.\d+)?\s*\$	Operador division