

UNIVERSIDAD DEL VALLE DE GUATEMALA
Diseño de lenguajes de programación
Sección 30
Ing. Pablo Koch



[M2] Actividad Práctica: Introducción al análisis léxico

Ihan Marroquin – 23108

GUATEMALA, 29 de enero del 2025

Problema 1: 25%

Considere los siguientes fragmentos de código:

Fragmento 1 - Código en C:

```
1 float limitedSquare(x)
2 float x;
3 {
4     /* returns x-squared, but never more than 100 */
5     return (x<=-10.0||x>=10.0)?100:x*x;
6 }
```

Fragmento 2 - Código HTML:

```
1 Here is a photo of <b>my house</b>:
2 <p><img src = "house.gif"><br>
3 See <a href = "morePix.html">More Pictures</a> if you liked that one.<p>
```

- Realice el análisis léxico completo de ambos fragmentos de código. Para cada fragmento, identifique todos los lexemas y clasifíquelos por tipo.
- Determine cuáles lexemas de cada fragmento deben obtener valores léxicos asociados (atributos).
- Para cada lexema que requiera un valor léxico, especifique el tipo de valor que debe asociarse, el valor específico, y justifique por qué es necesario.
- Diseñe una estructura de datos unificada que pueda representar tokens de diferentes lenguajes de programación. Su estructura debe ser lo suficientemente flexible para manejar tanto lenguajes como C como lenguajes de marcado como HTML. Demuestre su uso representando al menos 5 tokens de cada fragmento.

Fragmento 1:

No	Lexema	Tipo	Valores léxicos asociados (¿sí/no?)	Tipo de valor léxico asociado	Valor léxico asociado	Justificación
1	float	Keyword	No	-	-	Palabra reservada; el token en sí basta.
2	limitedSquare	Identificador (ID)	Sí	string ()	"limitedSquare"	Necesario para la tabla de símbolos.s

3	(Puntuación	No	-	-	Marca inicio de lista de parámetros.
4	x	Identificador (ID)	Sí	string	"x"	Identificador del parámetro.
5)	Puntuación	No	-	-	Cierre de lista de parámetros.
6	float	Keyword	No	-	-	Declaración del tipo del parámetro.
7	x	Identificador (ID)	Sí	string	"x"	Nombre del parámetro en la declaración.
8	;	Puntuación	No	-	-	Separador de declaración.
9	{	Puntuación	No	-	-	Inicio de bloque.
10	/* returns x-square d, but never more than 100 */	Comentario	No	-	-	Se ignora el comentario
11	return	Keyword	No	-	-	Palabra reservada de control de flujo.
12	(Puntuación	No	-	-	Abre la expresión.

13	x	Identificador (ID)	Sí	string	"x"	Operando; referencia a variable.
14	<=	Operador.	No	-	-	Operador.
15	-	Operador.	No	-	-	Operador.
16	10.0	Constante	Sí	float	10.0	Valor numérico.
17		Keyword	No	-	-	-
18	x	Identificador (ID)	Sí	string	"x"	Referencia a variable.
19	>=	Operador	No	-	-	Operador; no requiere atributo.
20	10.0	Constante	Sí	float	10.0	Igual que arriba.
21)	Puntuación	No	-	-	Cierra expresión.
22	?	Operador.	No	-	-	Operador; token suficiente.

23	100	Constante	Sí	int	100	Literal entero; used para valor máximo.
24	:	Puntuación.	No	-	-	Separador del operador ternario.
25	x	Identificador (ID)	Sí	string	"x"	Operando.
26	*	Operador.	No	-	-	Operador.
27	x	Identificador (ID)	Sí	string	"x"	Operando.
28	;	Puntuación.	No	-	-	Final de sentencia.
29	}	Puntuación.	No	-	-	Fin de bloque/función .

Fragmento 2:

No .	Lexema	Tipo	¿Valores léxicos asociados?	Tipo de valor léxico asociado	Valor léxico asociado	Justificación
1	Here is a photo of	Constantes	Sí	string	"Here is a photo of "	Texto literal fuera de tags
2		Opening tag	No	-	-	Marca inicio de tag.

3	my house	Constantes	Sí	string	"my house"	Texto interior de la etiqueta.
4		Closing tag	No	-	-	Marca inicio de tag de cierre.
5	:	Puntuación	No	-	-	Carácter de puntuación.
6	<p>	Opening tag	No	-	-	Inicio de <p>.
7	<img	Opening tag	No	-	-	Inicio de .
8	src	Identificador (ID)	Sí	string	"src"	Nombre del atributo que indica el origen de la imagen.
9	=	Operador	No	-	-	Asignación entre nombre de atributo y su valor.
10	"house.gif "	Constantes	Sí	string	"house.gif "	Valor literal del atributo src.
11	>	Closing tag	No	-	-	Cierra el tag .
12	 	Opening tag	No	-	-	Inicio de .
13	See	Constantes	Sí	string	"See "	Texto literal antes del enlace.
14	<a	Opening tag	No	-	-	Inicio de <a>.

15	<code>href</code>	Identificador (ID)	Sí	string	" <code>href</code> "	Nombre del atributo que indica destino del enlace.
16	=	Operadores	No	-	-	Asigna el valor al atributo <code>href</code> .
17	"morePix.h tml"	Constantes	Sí	string	"morePix.h tml"	Valor del atributo <code>href</code>
18	>	Closing tag	No	-	-	Cierre del start tag <code><a . . . ></code> .
19	More Pictures	Constantes	Sí	string	"More Pictures"	Texto que se muestra como enlace.
20	<code></code>	Closing tag	No	-	-	Inicio del cierre <code></code> .
21	if you liked that one.	Constantes	Sí	string	" if you liked that one . "	Texto posterior al enlace.
22	<code><p></code>	Opening tag	No	-	-	Inicio de <code><p></code> .

Estructura de datos unificada:

- Lenguaje: C o HTML
- Lexema
- Tipo: Keyword, Operador, Identificador (ID), Constante o Puntuación
- Tipo_Value_Asociado: int, String, Float, null, etc.
- Valor_Asociado: 10.0, "My House", "See", null, etc.

Ejemplos C:

1. float
 - a. Lenguaje: C
 - b. Lexema: float
 - c. Tipo: Keyword
 - d. Tipo_Value_Asociado: null
 - e. Valor_Asociado: null
2. limitedSquare
 - a. Lenguaje: C
 - b. Lexema: limitedSquare
 - c. Tipo: Identificador (ID)
 - d. Tipo_Value_Asociado: String
 - e. Valor_Asociado: limitedSquare
3. return
 - a. Lenguaje: C
 - b. Lexema: return
 - c. Tipo: Keyword
 - d. Tipo_Value_Asociado: null
 - e. Valor_Asociado: null
4. (
 - a. Lenguaje: C
 - b. Lexema: (
 - c. Tipo: Puntuación
 - d. Tipo_Value_Asociado: null
 - e. Valor_Asociado: null
5. 100
 - a. Lenguaje: C
 - b. Lexema: 100
 - c. Constante
 - d. Tipo_Value_Asociado: int
 - e. Valor_Asociado: 100

Ejemplos HTML:

1. Here is a photo of
 - a. Lenguaje: HTML
 - b. Lexema: Here is a photo of
 - c. Tipo: Constante
 - d. Tipo_Valor_Asociado: String
 - e. Valor_Asociado: Here is a photo of
2.
 - a. Lenguaje: HTML
 - b. Lexema:
 - c. Tipo: Opening tag
 - d. Tipo_Valor_Asociado: null
 - e. Valor_Asociado: null
3.
 - a. Lenguaje: HTML
 - b. Lexema:
 - c. Tipo: Closing tag
 - d. Tipo_Valor_Asociado: null
 - e. Valor_Asociado: null
4. href
 - a. Lenguaje: HTML
 - b. Lexema: href
 - c. Tipo: Identificador (ID)
 - d. Tipo_Valor_Asociado: String
 - e. Valor_Asociado: "href"
5. "house.gif"
 - a. Lenguaje: HTML
 - b. Lexema: "house.gif"
 - c. Tipo: Constante
 - d. Tipo_Valor_Asociado: String
 - e. Valor_Asociado: "house.gif"

Problema 2: 25%

Considere el siguiente fragmento de código en Java que contiene errores léxicos:

```
1 public int calculate Total(int x, double y) {  
2     int result = x * y;  
3     String message = "Result is: + result;  
4     double pi = 3.14.59;  
5     return result;  
6 }
```

- Identifique todos los errores léxicos presentes en el código.
- Para cada error léxico identificado, proponga una estrategia de recuperación específica que permita al analizador léxico continuar el análisis. Justifique por qué su estrategia es apropiada.
- Explique la diferencia entre un error léxico y un error sintáctico, dando un ejemplo de cada uno basado en el código anterior.

Errores léxicos detectados:

- Error léxico 1 — Línea 3: cadena no terminada
 - Texto problemático: String message = "Result is: + result;
 - Por qué es léxico: el analizador léxico leyó un " (comilla doble) y no encontró la comilla de cierre hasta el final de la línea/archivo, por lo que no puede formar un token válido.
- Error léxico 2 — Línea 4: literal numérico mal formado (puntos múltiples)
 - Texto invalido: double pi = 3.14.59;
 - Por qué es léxico: la secuencia 3.14.59 no encaja en la sintaxis de literales numéricos de Java.