

Analizador léxico

Equipo de 4 personas

Objetivo

- Escriba un programa en el lenguaje de su preferencia que realice el análisis léxico sobre un programa fuente escrito en C ANSI dado como entrada

Especificaciones de entrada y salida

- El programa deberá recibir desde la consola el nombre del archivo que contiene el programa, el nombre del archivo en el cual se escribirá la tabla de símbolos y el nombre del archivo en el cual se escribirán los errores encontrados
- La sintaxis es la siguiente:

```
programa <archivo_entrada> <tabla_simbolos> <archivo_errores>
```

Especificaciones del programa

- El programa deberá realizar las siguientes tareas:
 - Abrir y leer el archivo de entrada
 - Identificar los componentes léxicos del archivo de entrada
 - Almacenar los componentes léxicos y sus atributos en la tabla de símbolos
 - Identificar los errores léxicos del archivo de entrada

Componentes léxicos

- Palabras reservadas
- Identificadores
- Números enteros
- Números flotantes
- Operadores
- Símbolos especiales

Palabras reservadas

- Las palabras reservadas que se requieren identificar son:
 - Estructuras selectivas (if, else, switch, case)
 - Estructuras repetitivas (for, while, do)
 - Tipos de datos (void, int, float, double, char)
 - Calificadores de tipo de dato (short, long, signed, unsigned)
 - Instrucciones del preprocesador (include, define)
 - return, main

Números enteros y flotantes

- Los números pueden tener variantes en su formato (revisarlos directamente en un compilador de C ANSI):
 - 4, 786
 - 4.3, .3, 3.
- No se considerarán para el analizador la notación hexadecimal, ni la exponencial
- También se descartan números con signo

Operadores

- Aritméticos: +, -, *, /, %
- Lógicos: &&, ||, !
- Relacionales: >, <, >=, <=, ==, !=
- Asignación: =, +=, -=, *=, /=
- Incremento: ++, --

Símbolos especiales

- (
-)
- {
- }
- [
-]
- ;
- ,
- .
- :
- #
- '
- “

Tabla de símbolos

- La tabla de símbolos generada tiene la siguiente estructura (El archivo “Listado de valores” contiene el resto de los valores que deben establecer a cada componente léxico)

Componente léxico	Lexema	Valor
IDENTIFICADOR	suma	suma
NÚMERO ENTERO	4	4
ESTRUCTURA SELECTIVA	if	IF
OPERADOR RELACIONAL	>	MAYOR-QUE
SÍMBOLO ESPECIAL	(PARÉNTESIS-ABIERTO
TIPO DE DATO	int	INT

Errores léxicos

- El analizador debe detectar errores léxicos e indicar el símbolo que produjo el error, así como la línea en la cual se encontró
- Debe utilizar la estrategia de manejo de errores de modo pánico para recuperarse de estos y continuar con el análisis
- Al final debe escribir en un archivo con la siguiente estructura

Línea de error	Símbolo
1	@
12	^

Listado de valores para los componentes léxicos

Componente léxico	Valor
IDENTIFICADOR	El lexema
NÚMERO ENTERO	El lexema
NÚMERO FLOTANTE	El lexema
ESTRUCTURA SELECTIVA	IF, ELSE, SWITCH, CASE
ESTRUCTURA REPETITIVA	FOR, WHILE, DO
OPERADOR LÓGICO	Y, O, NEGACIÓN, DIFERENTE
OPERADOR RELACIONAL	MAYOR, MENOR, IGUAL, MAYOR-IGUAL, MENOR-IGUAL
OPERADOR ARITMÉTICO	SUMA, RESTA, MULTIPLICACIÓN, DIVISIÓN, MÓDULO
ASIGNACIÓN	ASIGNACIÓN, SUMA-ASIGNACIÓN, RESTA-ASIGNACIÓN, MULTIPLICACIÓN-ASIGNACIÓN, DIVISIÓN-ASIGNACIÓN
INCREMENTO	INCREMENTO-POSITIVO, INCREMENTO-NEGATIVO
TIPO DE DATO	INT, FLOAT, DOUBLE, CHAR, VOID
CALIFICADOR DE TIPO DE DATO	SHORT, LONG, SIGNED, UNSIGNED
INSTRUCCIÓN DEL PREPROCESADOR	INCLUDE, DEFINE, INLINE
RETURN	RETURN
MAIN	MAIN
SÍMBOLO ESPECIAL	PARENTESIS-ABIERTO, PARENTESIS-CERRADO, LLAVE-ABIERTA, LLAVE-CERRADA, CORCHETE-ABIERTO, CORCHETE-CERRADO, PUNTO-Y-COMA, COMA, PUNTO, DOS-PUNTOS, NUMERAL, COMILLA-SIMPLE, COMILLA-DOBLE

Reporte

- Se debe entregar un reporte que incluya lo siguiente:
 - Portada con los nombres de los integrantes del equipo y el grupo
 - El diseño de cada uno de los autómatas creados para su analizador y la especificación de el o los componentes que reconoce.
 - Los estados de los autómatas deben estar numerados de acorde a la implementación que realizaron
 - Cuando sea posible los autómatas deben estar agrupados por componente, por ejemplo: autómata que reconoce operadores lógicos, autómata que reconoce operadores relacionales, etc.