

UNIVERSIDAD DE GUADALAJARA CUCEI



Nombre: Ivan nudelstejer gomez

Codigo: 218130122

Materia: Seminario de solución de problemas de traductores de lenguaje II

Profesor: Michel Emanuel lopez franco

Reporte de la Etapa del Proyecto: Analizador Léxico Completo

El proyecto consiste en un analizador léxico implementado en Python utilizando la biblioteca Tkinter para la interfaz gráfica de usuario (GUI). El analizador léxico divide una cadena de entrada en una secuencia de tokens (símbolos y tipos) siguiendo las reglas definidas en el código.

Estructura del Proyecto

El proyecto está organizado en tres archivos principales:

1. `main.py`: Contiene la clase `AnalizadorLexicoApp`, que implementa la interfaz gráfica de usuario utilizando Tkinter. Esta clase permite al usuario ingresar una cadena de código y analizarla utilizando el analizador léxico.
2. `lexico.py`: Contiene las clases `TipoCadena` y `AnalizadorLexico`. La clase `TipoCadena` define los tipos de símbolos que pueden aparecer en la cadena de entrada. La clase `AnalizadorLexico` implementa los métodos para analizar la cadena y determinar los tipos de cada símbolo.
3. `lexico.py`: Contiene las clases `TipoCadena` y `AnalizadorLexico`. La clase `TipoCadena` define los tipos de símbolos que pueden aparecer en la cadena de entrada. La clase `AnalizadorLexico` implementa los métodos para analizar la cadena y determinar los tipos de cada símbolo.

Funcionalidades

El programa permite al usuario ingresar una cadena de código en la interfaz gráfica y luego presionar un botón para iniciar el análisis léxico. Después del análisis, se muestran los resultados en la misma interfaz, con cada símbolo de la cadena seguido de su tipo correspondiente.

Análisis Léxico

El análisis léxico se realiza utilizando un autómata finito determinista (DFA) implementado en la clase `AnalizadorLexico`. Este DFA tiene estados que representan diferentes situaciones en el análisis de la cadena, como identificación de letras, números, operadores, etc. Basado en el estado actual y el carácter actual, el autómata transita a un nuevo estado hasta que la cadena se agote.

- 1. Mejoras en la Interfaz de Usuario:** Se podrían agregar más elementos de diseño y funcionalidades a la interfaz gráfica para hacerla más atractiva y útil para el usuario.
- 2. Manejo de Errores Mejorado:** Actualmente, el programa informa que la cadena no es válida si está vacía. Se podrían agregar más comprobaciones de validez y proporcionar mensajes de error más descriptivos.
- 3. Optimización del Análisis Léxico:** Se podría mejorar la eficiencia del análisis léxico mediante la optimización del código y la reducción del tiempo de ejecución.

Conclusión

El proyecto del analizador léxico es funcional y cumple con su propósito de analizar cadenas de código en busca de tokens específicos. Con algunas mejoras adicionales, podría convertirse en una herramienta aún más poderosa y útil para los desarrolladores.

Captura:

