

Manual de Usuario del Analizador Léxico, Sintáctico y Semántico

Introducción

Este programa permite analizar código fuente escrito por el usuario a través de tres etapas: análisis léxico, análisis sintáctico y análisis semántico. El análisis léxico convierte el código en una secuencia de tokens, el análisis sintáctico construye un árbol sintáctico (AST), y el análisis semántico comprueba la coherencia de las variables y las operaciones.

Requisitos del sistema

- Sistema Operativo: Windows, MacOS, o Linux.
- Python: Versión 3.6 o superior.
- Bibliotecas: Debes tener instalada la biblioteca tkinter (viene instalada por defecto con Python en la mayoría de las distribuciones).

Instrucciones de instalación

1. Descarga el código: Descarga el archivo Python que contiene el programa de análisis.
2. Instala Python: Asegúrate de tener instalado Python 3.6 o superior. Puedes descargarlo desde python.org.
3. Ejecuta el programa: Abre una terminal o línea de comandos y navega a la carpeta donde se encuentra el archivo. Ejecuta el programa con el comando:
`python nombre_del_archivo.py`

Descripción de la Interfaz

La aplicación tiene una interfaz gráfica sencilla con las siguientes secciones principales:

1. Área de entrada de código: Un área donde el usuario puede escribir o pegar el código a analizar.
2. Botón "Analizar": Presiona este botón para iniciar el proceso de análisis léxico, sintáctico y semántico del código ingresado.
3. Resultados del análisis: Tres cuadros de texto para mostrar los resultados del análisis léxico (tokens), del AST, y del análisis semántico.

Pasos para usar la aplicación

1. Introducir código:

Escribe o pega tu código fuente en el área de texto "Introduce el código". Un ejemplo de

código válido es:

```
variable1 = 42 + 8;
```

2. Iniciar el análisis:

Haz clic en el botón "Analizar" para comenzar el análisis. El programa realizará tres pasos:

- Análisis léxico: Convertirá el código en tokens y los mostrará en la sección "Resultado Léxico (Tokens)".
- Análisis sintáctico: Construirá el árbol sintáctico (AST) y lo mostrará en la sección "Árbol Sintáctico (AST)".
- Análisis semántico: Comprobará el uso de variables y generará la tabla de símbolos en la sección "Resultado Semántico".

3. Revisar los resultados:

- Tokens: Los tokens generados se mostrarán en el primer cuadro de texto.
- Árbol Sintáctico (AST): El árbol que representa la estructura del código se mostrará en un formato indentado.
- Resultado Semántico: Si no hay errores, verás un mensaje sobre las variables declaradas y sus tipos.

Posibles errores

Si se encuentra algún error en tu código, el programa te lo indicará a través de un cuadro de diálogo:

- Errores léxicos: Por ejemplo, si hay un símbolo desconocido en tu código.
- Errores sintácticos: Si la estructura del código no sigue las reglas del lenguaje.
- Errores semánticos: Si usas variables no declaradas o cometes otros errores lógicos.

Ejemplo de uso completo

1. Ingresa el siguiente código en el área de texto:

```
x = 10 + 20;
```

2. Presiona "Analizar".

3. En "Resultado Léxico (Tokens)" deberías ver algo similar a:

```
IDENTIFIER: x  
ASSIGN: =  
NUMBER: 10  
OPERATOR: +  
NUMBER: 20  
SEMICOLON: ;
```

4. En "Árbol Sintáctico (AST)" deberías ver:

```
ASSIGN: =  
  IDENTIFIER: x
```

OPERATION: +
NUMBER: 10
NUMBER: 20

5. En "Resultado Semántico", deberías ver:
Variable 'x' asignada con tipo NUMBER

Solución de problemas

- No se muestra ningún token: Asegúrate de que el código esté correctamente formateado y termine con un punto y coma ;.
- Errores léxicos: Revisa si hay símbolos no permitidos o errores tipográficos en el código.
- Errores sintácticos o semánticos: Asegúrate de que el código siga las reglas del lenguaje admitido.

Conclusión

Este programa facilita el análisis de código fuente básico a través de tres fases: léxica, sintáctica y semántica. Con su interfaz gráfica, puedes realizar análisis de código de manera intuitiva y comprender cómo se interpreta paso a paso.