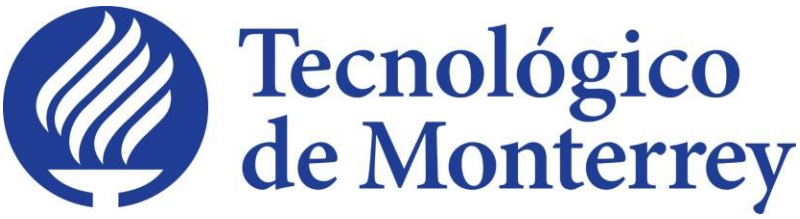

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Guadalajara

Implementación de métodos computacionales

Analizador Sintáctico

Jose Manuel Martinez Morales-A01734279

Profesor

Vladimir Mendoza

Fecha de entrega

27 de mayo del 2024

El Analizador Sintáctico es una herramienta diseñada para analizar la estructura gramatical de un programa. Utiliza un conjunto de reglas gramaticales predefinidas para analizar el código fuente y generar un árbol de derivación sintáctica que representa la estructura del programa. Este manual proporcionará instrucciones detalladas sobre cómo compilar, ejecutar y utilizar el programa del Analizador Sintáctico.

Como utilizar el analizador sintáctico

Se utiliza el cmd de Windows, primero se utiliza el comando cd para ir al directorio en donde se tenga el código de Python. En mi caso:

cd C:\Users\josem\Downloads\AnalizadorSintactico

Después simplemente se escribe el comando:

Python analizadorSintactico.py

Que hace el analizador sintáctico

Los pasos utilizar el analizador sintáctico son los siguientes:

Entrada de Expresiones Aritméticas: Ingrese el el texto que se quiere analizar en “prueba.txt” en el formato del lenguaje específico. Asegúrese de seguir las convenciones de sintaxis y los tokens definidos en el lenguaje.

Interpretación de la Salida: Después de ingresar el texto, el Analizador Sintáctico generará un árbol de derivación sintáctica que representa la estructura gramatical del programa. La salida se presentará en la consola o terminal en forma de árbol, donde cada nodo representa un elemento gramatical del programa.

Interpretación de los Mensajes de Error: Si el texto fuente ingresado contiene errores sintácticos, el Analizador Sintáctico generará mensajes de error detallados que indicarán la naturaleza y la ubicación de los errores. Estos mensajes de error ayudarán a identificar y corregir errores en el texto.

Como funciona el código

El código fuente del Analizador Sintáctico consta de varias partes principales:

Tokenizador: Utiliza expresiones regulares para dividir el código fuente en tokens significativos, como palabras clave, identificadores, operadores y símbolos de puntuación.

Analizador Sintáctico: Implementa un conjunto de reglas gramaticales para analizar la estructura del programa basado en los tokens generados por el tokenizador. Utiliza un enfoque de análisis ascendente para construir un árbol de derivación sintáctica que representa la estructura del programa.

Clase Nodo: Define una clase para representar los nodos del árbol de derivación sintáctica. Cada nodo tiene un tipo, un valor opcional y puede tener varios hijos que representan elementos gramaticales anidados.

Casos de prueba

Prueba 1

```
Programa {
  principal()
    {
        Entero a = 4;
        Real b = 5.8E-8;
        Entero serie = a * b * c + 9;
    }
}
```

En este caso la prueba es valida y se despliega el árbol con su análisis léxico y sintactico

Prueba 2

```
Programa {
  principal()
    {
        a = 4;
        Real b = 5.8E-8;
        Entero serie = a *  * c;
    }
}
```

En este caso la prueba es invalida porque falta una variable en:

Entero serie = a * * c;

Y no sigue el formato <type> <variable_name> = <value>

Referencias

¿Qué es el análisis sintáctico en Python – Explicado! – Kanaries. (n.d.). https://docs.kanaries.net/es/topics/Python/what-is-parse-python

OpenAI. (2024). ChatGPT: *A large language model trained by OpenA*I. OpenAI. Retrieved from https://www.openai.com/

parser — Acceder a árboles de análisis sintáctico de Python — documentación de Python - 3.9.19. (n.d.). https://docs.python.org/es/3.9/library/parser.html