Instituto tecnológico de nacional de México

Instituto Tecnológico de Ciudad Madero

Ingeniero:

Fernando Manzanares Gonzalez.

Alumno:

Ramos Hernández Marco Antonio.

Materia:

Lenguajes y Autómatas 1

Hora:

19:00-20:00

Introducción

El análisis léxico y sintáctico es una parte crucial en el proceso de compilación de un lenguaje de programación. Consiste en analizar la estructura gramatical del código fuente para verificar si cumple con las reglas léxicas y sintácticas del lenguaje. En este documento, se presenta la implementación de un analizador para el lenguaje de programación Pascal.

El proyecto se trabajó en el IDE: PyCharm el cual es un entorno de desarrollo especializado en Python.

Funcionamiento

```python
from Lexico import Lexicon
from Sintactico import Sintax

file = "Codigos/Suma.txt"
lexico = Lexicon(file, print_result=True)
if not lexico.error_found:
    print("Análisis léxico completado exitosamente")
    sintaxis = Sintax(lexico.head, print_result=True)
    if not sintaxis.error_found:
        print("Análisis Sintactico completado exitosamente")
```

Este código Python importa dos módulos, "Lexico" y "Sintactico", que están diseñados para realizar análisis léxico y sintáctico, respectivamente. Luego, se importan los módulos necesarios de estos paquetes:

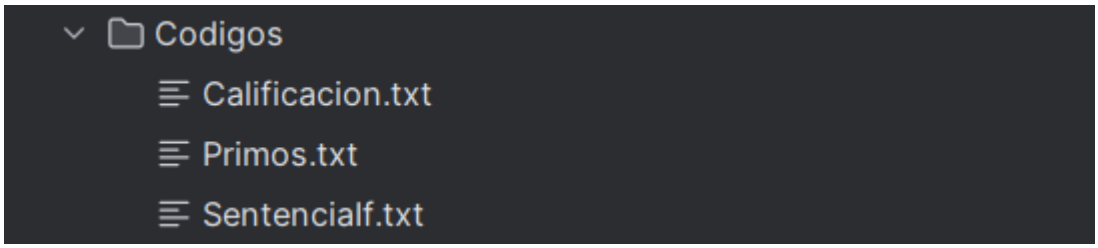
python

Copiar código

from Lexico import Lexicon

from Sintactico import Sintax

A continuación, se define una variable file que almacena la ruta de un archivo de texto llamado "Suma.txt" dentro del directorio "Codigos".



python

Copiar código

```
file = "Codigos/Suma.txt"
```

Luego, se crea una instancia de la clase Lexicon con el archivo especificado y se imprime el resultado del análisis léxico.

python

Copiar código

```
lexico = Lexicon(file, print_result=True)
```

Si no se encuentra ningún error léxico (error_found es False), se imprime un mensaje indicando que el análisis léxico se completó exitosamente.

python

Copiar código

```
if not lexico.error_found:
    print("Análisis léxico completado exitosamente")
```

Después, se realiza un análisis sintáctico utilizando la salida del análisis léxico (lexico.head). Se imprime el resultado del análisis sintáctico.

python

Copiar código

```
sintaxis = Sintax(lexico.head, print_result=True)
```

Si no se encuentra ningún error sintáctico (error_found es False), se imprime un mensaje indicando que el análisis sintáctico se completó exitosamente.

python

Copiar código

```
if not sintaxis.error_found:
    print("Análisis Sintactico completado exitosamente")
```

En resumen, el código realiza análisis léxico y sintáctico de un archivo de texto llamado "Suma.txt" ubicado en el directorio "Codigos", e imprime mensajes indicando si estos análisis se completaron exitosamente o si se encontraron errores, igualmente funciona con el directorio “Errores” poniendo la extensión .txt dentro del directorio del error que se quiere probar.

Clase Sintax

La clase Sintax es responsable de realizar el análisis sintáctico del código fuente en Pascal. Utiliza la lista de tokens generados por el analizador léxico y verifica si la secuencia de tokens sigue las reglas gramaticales del lenguaje.

Métodos

- __init__(self, head, print_result=True): Constructor de la clase, inicializa los atributos y llama al método parse_program para iniciar el análisis.

- parse_program(self): Analiza la estructura básica de un programa Pascal, incluyendo declaraciones de variables y bloques de código.

- parse_block(self): Analiza un bloque de código delimitado por las palabras clave begin y end.

- parse_statement_list(self): Analiza una lista de declaraciones separadas por punto y coma (;).

- parse_statement(self): Analiza una declaración individual, como asignaciones, condicionales y bucles.

- parse_expression(self): Analiza expresiones aritméticas.

- parse_term(self): Analiza términos en una expresión.

- parse_factor(self): Analiza factores en una expresión, como identificadores, enteros o expresiones entre paréntesis.

- match(self, token): Verifica si el token actual coincide con el esperado y avanza al siguiente token.

- error(self, message): Maneja errores de sintaxis y muestra un mensaje de error.

El análisis léxico es la primera etapa en el proceso de compilación de un programa de computadora. Consiste en convertir una secuencia de caracteres (código fuente) en una secuencia de tokens, que son unidades léxicas con significado en el lenguaje de programación. En este documento, se presenta la implementación de un analizador léxico para el lenguaje de programación Pascal.

Clase `Lexicon`

La clase `Lexicon` es responsable de realizar el análisis léxico del código fuente en Pascal. Recibe el archivo de código fuente como entrada y produce una lista enlazada de tokens, cada uno representando una unidad léxica del programa.

Métodos

- `__init__(self, filename, print_result=True)`: Constructor de la clase, inicializa los atributos y comienza el análisis léxico.

- `printNodes(self, p)`: Método auxiliar para imprimir la lista enlazada de tokens.

- `validateReservedWord(self, lexeme, token)`: Verifica si un identificador es una palabra reservada del lenguaje y asigna el token correspondiente.

- `insertNode(self, lexeme, token, row, head, p)`: Inserta un nuevo nodo en la lista enlazada de tokens.

- `match(self, token)`: Verifica si el token actual coincide con el esperado y avanza al siguiente token.

- `error(self, message)`: Maneja errores léxicos y muestra un mensaje de error.

Uso

```python
file = "Pruebas/Temp.txt"
lexico = Lexicon(file, print_result=True)
if not lexico.error_found:
```

```
print("Análisis léxico completo")
```