

Documentação TP1 - Compiladores

Samuel Rodrigues Viana de Faria
0057497

1. Introdução

O trabalho prático 1 consiste na implementação de um compilador que realiza a análise léxica e sintática de um código fonte seguindo a gramática específica criada pelo professor. O compilador atual é composto por duas partes principais: o módulo léxico (`lexico.py`) e o módulo sintático (`sintatico.py`).

2. Módulo Léxico (`lexico.py`)

2.1. Descrição

O módulo léxico é responsável por realizar a análise léxica do código fonte, identificando e classificando os tokens presentes no arquivo de entrada, tal módulo é chamado pelo sintático para que o mesmo receba os tokens formados pelo léxico.

2.2. Funcionalidades

- Reconhecimento de palavras reservadas, identificadores, números inteiros/reais, operadores, pontuações, cadeias de caracteres e comentários.
- Tratamento de erros léxicos, como identificadores muito longos.

2.3. Estrutura

2.3.1. Classes

- **TipoToken**: Enumeração que define os tipos de tokens reconhecidos pelo compilador.
- **Token**: Classe que representa um token, armazenando informações como tipo, lexema e linha.

2.3.2. Métodos Principais

- **abreArquivo**: Abre o arquivo de entrada para leitura.
- **fechaArquivo**: Fecha o arquivo de entrada.
- **getChar**: Lê um caractere do arquivo.
- **ungetChar**: Desfaz a leitura de um caractere.
- **is_alpha**: Verifica se o caractere faz parte do alfabeto sem caracteres especiais.

- **getToken:** Realiza a análise léxica, formando e classificando os tokens presentes no código fonte.

3. Módulo Sintático (sintatico.py)

3.1. Descrição

O módulo sintático é responsável por realizar a análise sintática do código fonte em conjunto com o módulo léxico, aplicando as regras definidas na gramática.

3.2. Funcionalidades

- Implementação de funções correspondentes às regras gramaticais.
- Tratamento de erros sintáticos, exibindo mensagens informativas.

3.3. Estrutura

3.3.1. Classes

- **Sintatico:** Classe que contém os métodos para realizar a análise sintática.

3.3.2. Métodos Principais

- **interprete:** Inicia o processo de interpretação do código fonte a partir da chamada do módulo léxico que retorna os tokens lidos do arquivo.
- Métodos correspondentes às regras gramaticais definidas na gramática.

4. Execução do Compilador

Para executar o compilador, basta criar uma instância da classe Sintatico e chamar o método interprete, passando o nome do arquivo contendo o código fonte como parâmetro.

```
if __name__ == "__main__":  
    nome = input("Entre com o nome do arquivo: ")  
    parser = Sintatico()  
    parser.interprete(nome)  
    print('O código fonte presente no arquivo escolhido foi compilado com SUCESSO!!!!')
```

5. Considerações Finais

Este compilador é uma implementação básica que utiliza somente dos módulos léxicos e sintáticos de um compilador, o mesmo ainda não conta com o módulo semântico que deve ser implementado na próxima parte do trabalho. A estrutura modular facilita a manutenção e extensão do projeto.