

Documentação TP2 - Compiladores

Samuel Rodrigues Viana de Faria
0057497

1. Introdução

O trabalho prático 2 consiste na implementação do modo pânico e da tabela de símbolos no compilador do trabalho 1. O compilador final é composto por três partes principais: o módulo léxico (`lexico.py`), o módulo sintático (`sintatico.py`) e o módulo da tabela de símbolos (`tabela.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

O módulo sintático agora implementa o modo pânico, que lida de maneira mais robusta com erros sintáticos, permitindo a recuperação do compilador após um erro. Além disso, foi adicionada a construção e utilização de uma tabela de símbolos para armazenar as informações sobre as variáveis declaradas no código.

3.3. Modo Pânico

O modo pânico é acionado quando ocorre um erro sintático. Em vez de encerrar imediatamente, o compilador tenta encontrar um ponto de sincronização na gramática para retomar a análise. Isso ajuda a identificar mais erros no código e a gerar mensagens de erro mais úteis.

3.4. Tabela de Símbolos

A tabela de símbolos armazena informações sobre as variáveis declaradas no código, como seus nomes e tipos. Essa tabela é consultada durante a análise para garantir que as variáveis sejam usadas de acordo com suas declarações.

3.5. Estrutura

O módulo sintático foi adaptado para lidar com as novas funcionalidades, integrando o modo pânico e a tabela de símbolos à análise sintática. As funções foram ajustadas para considerar essas mudanças.

4. Módulo Tabela de Símbolos (tabela.py)

4.1. Descrição

O módulo da tabela de símbolos é responsável por gerenciar a tabela de símbolos do compilador, onde são armazenados os identificadores declarados no programa, juntamente com seus tipos.

4.2. Funcionalidades

- Verificar a existência de um identificador na tabela.
- Declarar um identificador na tabela.
- Obter o tipo associado a um identificador.

4.3. Estrutura

A classe TabelaSimbolos possui métodos para realizar as operações mencionadas acima.

5. Execução do Compilador

5.1. Alterações

O modo de execução do compilador foi modificado para receber como entrada o nome do arquivo a ser compilado. Além disso, foi adicionada a opção de salvar a tabela de símbolos em um arquivo separado.

5.2. Exemplo de Execução

Para compilar um programa, basta fornecer o nome do arquivo como entrada. Caso deseje salvar a tabela de símbolos, é possível incluir a opção **-t** seguida do nome do arquivo desejado.

6. Considerações Finais

O compilador resultante do trabalho prático 2 apresenta melhorias significativas em relação à sua versão anterior. A implementação do modo pânico permite que o compilador continue sua análise mesmo após a detecção de erros sintáticos, facilitando a identificação de múltiplos erros no código fonte. Além disso, a tabela de símbolos foi integrada ao compilador, permitindo o gerenciamento eficiente de identificadores declarados no programa.