

Trabalho Prático

A construção de um compilador para uma linguagem imperativa simplificada

Prática No.1 - Implementação da Tabela de Símbolos e Analisador Léxico

Nesta fase deverão ser implementados a tabela de símbolos e o analisador léxico do compilador.

1. Defina o alfabeto para a linguagem e numere os tokens.
2. Implemente o tipo abstrato de dados “tabela de símbolos”, onde serão armazenados apenas as palavras reservadas e os identificadores. A tabela deverá armazenar registros que conterão, a princípio, campos para o número do token (byte) e o lexema (arranjo de caracteres) da palavra ou identificador. Devem ser implementadas duas funções:
 - Uma função que pesquisa a tabela em busca de um lexema e retorna o endereço do registro correspondente, caso exista, ou NULL se este não estiver na tabela.
 - Uma função que insere dinamicamente um registro na tabela, com o token e seu lexema, retornando o endereço de inserção.

Para maior eficiência, utilize uma função de espalhamento para pesquisar e inserir itens. Antes do início da análise léxica, o compilador deve inserir todas as palavras reservadas na tabela de símbolos.

3. Teste a estrutura da tabela de símbolos, listando na tela os itens inseridos. **Após o ter certeza que a tabela está correta, desabilite os testes.**
4. Implemente o analisador léxico na forma de um procedimento que será chamado pelo analisador sintático. Este procedimento deverá ler o programa-fonte e identificar o próximo token. Comentários, espaços em branco e quebras de linha deverão ser considerados delimitadores e desprezados. O token reconhecido deverá ser representado por um registro único de escopo global contendo campo para o número do token, lexema, endereço de inserção na tabela (somente para identificadores e palavras reservadas), tipo e tamanho (somente para o token *constante*). Observe que o registro léxico é diferente do registro da tabela de símbolos.

A diferenciação entre identificador e palavra reservada é feita através de uma pesquisa na tabela de símbolos. Se um identificador não existir na tabela, deverá ser

inserido. Para a construção dos lexemas, utilize um autômato de estados finitos determinístico com rotinas associadas às suas transições. Erros na análise léxica devem ser reportados através de mensagens. O processo de compilação se encerra após um erro léxico.

As mensagens de erro devem ter os seguintes formatos, onde *nn* é o número da linha onde o erro foi detectado e *lex* é o lexema encontrado:

nn
caractere invalido.

nn
lexema nao identificado [*lex*].

nn
fim de arquivo nao esperado.

A contagem das linhas começa com valor “1”. No caso de sucesso na compilação a mensagem será:

nn linhas compiladas.

onde *nn* é o número de linhas do programa. Cada quebra de linha conta uma linha, mesmo dentro de comentários. A linha finalizada pelo fim do arquivo também é contabilizada.

Até que o analisador sintático seja feito, o programa principal do compilador deve chamar repetidamente o analisador léxico, listando na tela os registros léxicos encontrados, até o fim do arquivo-fonte. **Este teste deverá ser desabilitado posteriormente.**

Obs: Leia as especificações gerais contidas no documento “Descrição do trabalho”.