

Trabalho Prático

A construção de um compilador para uma linguagem imperativa simplificada

Prática No.2 - Implementação do Analisador Léxico

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) e tipo (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 itens léxicos, 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 o primeiro erro ser encontrado.

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.

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

O que entregar:

- Códigos-fontes
- Desenho do AFD

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