Pontifícia Universidade Católica de Minas Gerais
ICEI - Curso de Ciência da Computação
Disciplina: Compiladores
Prof. Alexei Machado

Trabalho Prático

A construção de um compilador para uma linguagem gráfica que executa scripts de animação

Parte I - Implementação do Analisador Léxico

Data de entrega: 16/04/2013

Nesta fase deverão ser implementados o analisador léxico e parte do tratamento da tabela de símbolos do compilador. O programa deverá ler um arquivo contendo o programa-fonte, o qual deve ser **passado como parâmetro** (argumento) para o compilador.

1. Defina o alfabeto para a linguagem GL 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 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 e espaços em branco deverão ser considerados delimitadores e desprezados. Números reais podem começar ou(exclusivo) terminar com ponto. O sinal deve ser considerado um token a parte. Declare uma variável de escopo global (registro léxico) contendo campos para o número do token, lexema, endereço de inserção na tabela (somente para identificadores), tipo de dado (no caso de

constantes), valor numérico da parte inteira e parte fracionária (no caso de constante). A cada token identificado no programa-fonte, as informações correspondentes são copiadas no registro léxico para que o analisador sintático tenha acesso a elas, sobrescrevendo o token anterior. No caso da constante ser do tipo real, sua parte fracionária será representada por um valor inteiro correspondente em décimos de milésimo. Observe que o registro léxico é diferente do registro da tabela de símbolos. 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.

5. 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. 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.

6. 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.**

O que entregar:

- Códigos-fontes
- Definição do alfabeto e padrão de formação dos lexemas
- Desenho do AFD

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