PUC Minas

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

Trabalho Prático

A construção de um compilador para uma linguagem imperativa simplificada

Prática No.3 – Verificação de unicidade, classes e tipos

Nesta fase, o analisador sintático construído na segunda parte deverá ser transformado em um esquema de tradução que fará a verificação semântica do programa-fonte. Esta fase é um passo intermediário para a obtenção do tradutor (compilador completo), que será capaz de gerar código Assembly para a linguagem L.

1. Modifique a tabela de símbolos, acrescentando os seguintes campos:

- **Classe**, que poderá assumir os valores *classe-var* e *classe-const*. Estes valores serão definidos para cada identificador, no momento de sua declaração;
- **Tipo**, que poderá assumir os valores *inteiro, lógico, string, real* ou *caractere*. Estes valores serão definidos para identificadores de variáveis e constantes, no momento de sua declaração.

2. Verifique a unicidade dos identificadores: cada identificador não pode ser declarado mais de uma vez, nem pode ser usado sem ter sido antes declarado.

3. Verifique a compatibilidade de classes de identificadores nas regras da gramática. Por exemplo, um comando de atribuição só pode atribuir valores a variáveis.

4. Verifique a compatibilidade de tipos das expressões, em cada comando:

   - Inteiros não são compatíveis com caracteres;
   - Comparações resultam em tipo lógico (true ou false);
   - Operadores *||, &&* e ! só operam tipos lógicos;
   - Constantes strings podem ser atribuídas a identificadores de strings. Elementos de strings podem ser acessados normalmente;
   - Strings só podem ser comparados quanto à sua igualdade (=);
   - Comandos de testes exigem expressões lógicas;

   Verifique o arquivo contendo a descrição da linguagem para obter outras informações sobre a compatibilidade de tipos de cada operador.

   O tipo e tamanho das expressões podem ser verificados recursivamente, com o auxílio de atributos sintetizados passados como parâmetros por referência.

5. Teste o analisador semântico com exemplos de programas-fontes corretos e exemplos de erros semânticos.

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

   nn
   identificador nao declarado [*lex*].

   nn
   identificador ja declarado [*lex*].

   nn
   classe de identificador incompatível [*lex*].

   nn
   tipos incompativeis.

   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.

O que entregar no Canvas:

- Esquema de tradução

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