

## 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* ou *caractere*. Estes valores serão definidos para identificadores de variáveis e constantes, no momento de sua declaração.
  - **Tamanho**, que indicará se o identificador é um vetor. No caso de ser escalar, o tamanho é 0. Isso evitará a necessidade de se usarem expressões de tipo.
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 o tamanho de declaração dos vetores que não podem ultrapassar 4 kbytes.
4. Verifique a compatibilidade de classes de identificadores nas regras da gramática. Por exemplo, um comando de atribuição só pode atribuir valores a variáveis.
5. Verifique a compatibilidade de tipos das expressões, em cada comando:
  - Inteiros não são compatíveis com caracteres;
  - Vetores de inteiros só podem ser manipulados elemento a elemento, ou seja não podem ser lidos, escritos, comparados e operados como vetores completos;
  - Comparações resultam em tipo lógico (0 ou 1);
  - Operadores *or*, *and* e *not* só operam tipos lógicos;
  - Constantes strings podem ser atribuídas a vetores de caracteres, desde que seu conteúdo útil mais o fim de string (\$) caibam no vetor. Um vetor de caracteres só pode ser atribuído a outro de tamanho maior ou igual. Elementos de vetores de caracteres podem ter seus elementos acessados normalmente;
  - Strings e vetores só podem ser comparados quanto à sua igualdade (=);
  - Comandos de testes exigem expressões lógicas;

- Comando de repetição exige tipos inteiros.

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

6. 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 incompatíveis.

*nn*  
tamanho do vetor excede o maximo permitido.

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 SGA:

- Esquema de tradução

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