

## PCS2056 – Linguagens e Compiladores

**Assunto:** Semântica informal - declarações

**Objetivo:** Estudo da semântica das declarações em uma linguagem imperativa.

**Palavras-chave:**

Escopos: nomes locais e globais, estrutura de blocos aninháveis

Declarações de variáveis simples:

Declarações de vetores e matrizes: linearização do espaço de endereçamento

Notação para selecionar elementos de matrizes e vetores

Declarações de registros

Notação para selecionar campos de registros

Declarações de novos tipos

Declarações de variáveis ou agregados usando os novos tipos

Declarações de procedimentos e funções com parâmetros

Chamadas de procedimentos e funções paramétricas

**Questões:**

- 1) Explique como deve ser interpretada uma linguagem que apresente estrutura de blocos aninháveis:
  - a) o que significam os diversos níveis de aninhamento dos blocos?
  - b) qual é o âmbito em que têm significado os nomes de variáveis, matrizes, etc., que tenham sido declarados em um determinado bloco?
  - c) o que acontece aos elementos declarados em um dado bloco quando o controle da execução do programa não se encontra dentro do escopo ao qual eles pertencem?
  - d) o que deve acontecer quando se declara um nome já declarado no mesmo escopo?
  - e) e se o mesmo nome pertencer ao conjunto de nomes de algum escopo envolvente?
  - f) o que ocorre se forem declarados nomes iguais em blocos paralelos (não aninhados)?
- 2) Deduza uma fórmula de linearização para calcular, em relação ao endereço do seu primeiro elemento, os endereços virtuais de um elemento genérico de uma matriz cujos elementos possam ser bytes, palavras de 32 bits ou agrupamentos de 64 bits conforme o tipo do elemento (char, int ou double, respectivamente).
- 3) Em um cadastro, cada um dos 200 registros do arquivo correspondente guarda as seguintes informações: (a) o nome do cliente (máximo 128 caracteres); (b) data de nascimento (três inteiros - para dia, mês e ano); endereço [campos: logradouro (máximo 128 caracteres), número da casa (inteiro), CEP (inteiro), bairro (máximo 32 caracteres), cidade (máximo 32 caracteres)]. Reservando-se para os registros a área suficiente para armazenar cada campo em seu formato máximo, estabeleça uma fórmula geral para calcular o endereço de cada um dos campos específicos para cada um dos registros que compõem o cadastro.
- 4) Para cada um dos tipos de dados apresentados acima, levante as informações junto com as quais os dados devem ser armazenados para que seja possível executar os respectivos algoritmos de endereçamento.
- 5) Em relação a novos tipos de dados, definidos pelo usuário, faça um estudo de como deve ser efetuado o seu tratamento, de tal modo que possam ser declaradas variáveis ou agregados contendo elementos dos novos tipos. Como podem ser eles endereçados no momento de seu uso? É possível estipular uma fórmula geral para esse endereçamento?

## PCS2056 – Linguagens e Compiladores

**Assunto:** Semântica informal – comandos

**Objetivo:** Estudo informal da semântica dos comandos em uma linguagem imperativa.

**Palavras-chave:**

Expressões aritméticas e booleanas

Comandos imperativos: atribuição de valor, chamada de sub-rotina, read, print

Rótulos e comandos de desvio incondicional, condicional e múltiplo

Comandos de controle de fluxo: if, while, do-until, for, case

**Questões:**

- 1) Mostre, através de um conjunto de exemplos significativos, de que forma pode ser escrito, em uma linguagem de baixo nível (do tipo assembly para uma arquitetura baseada em registradores), um programa que calcula o valor de uma expressão aritmética dada. Considerar que tal expressão aritmética envolve apenas constantes e variáveis aritméticas simples, podendo utilizar as operações aritméticas usuais, e um número arbitrário de parênteses aninhados.
- 2) Repita o exercício anterior para uma arquitetura baseada em pilha.
- 3) Repita mais uma vez, considerando que a arquitetura disponível é uma simples máquina de calcular à sua escolha, e que a linguagem a ser usada tem como conjunto de instruções o conjunto de teclas disponíveis nessa máquina.
- 4) Amplie as expressões em questão inserindo, um de cada vez, os seguintes elementos: (a) variáveis indexadas – vetores ou matrizes aritméticas; (b) chamadas de funções aritméticas (atenção à forma como os parâmetros devem ser passados); (c) elementos aritméticos, selecionados de algum dos campos de uma estrutura.
- 5) Mostre como escrever programas, em linguagem de baixo nível, que sejam equivalentes a um comando responsável pela atribuição de um valor aritmético a: (a) alguma variável aritmética; (b) alguma posição de vetor ou de matriz aritmética; (c) algum elemento aritmético selecionado de algum campo pertencente a uma estrutura.
- 6) Mostre como representar, em linguagem de baixo nível, o cálculo de uma expressão booleana: (a) formada pela comparação de duas expressões aritméticas; (b) formada a partir de outras expressões booleanas mais simples, combinadas com operadores booleanos usuais e aninhamento arbitrário de parênteses; (c) combinando os casos (a) e (b); (d) permitindo ainda elementos booleanos extraídos de agregados quaisquer; (e) permitindo adicionalmente chamadas de funções booleanas.
- 7) Repita, para o caso booleano, o exercício 5, usando as expressões criadas no exercício 6.
- 8) Mostre como escrever, em linguagem de baixo nível, programas equivalentes aos comandos de desvios incondicionais, condicionais e múltiplos.
- 9) Como programar, na linguagem de baixo nível, um programa que realize os comandos de entrada e saída da linguagem de alto nível? Lembre-se que a entrada e saída em linguagem de alto nível não se limita a efetuar transferências de dados através das portas do computador.
- 10) Mostre como construir programas em linguagem de baixo nível, equivalentes aos comandos de controle de fluxo – if, while, do-until, for, case.

**Uma referência sobre semântica** (aulas desta semana)

Tremblay e Sorenson *The theory and practice of compiler writing*. McGraw-Hill, 1985