

## PCS2056 – Linguagens e Compiladores

**Assunto:** Tradução das declarações

**Objetivo:** Com base na especificação do ambiente de execução, desenvolvido anteriormente, projetar o mecanismo de tradução para as declarações de uma linguagem imperativa.

**Palavras-chave:**

tabelas de símbolos e de atributos

**Atividades de Projeto:**

- 1) Projetar uma rotina responsável pela montagem e pela busca de informação em uma **tabela de símbolos** (= **tabela de nomes**), a ser usada para guardar os nomes das variáveis, e de uma **tabela de atributos**, na qual são informados o seu tipo, uma indicação sobre o fato de já ter sido declarada ou não, referenciada ou não, e um indicador da posição de memória onde as mesmas foram alocadas.
- 2) Projetar um algoritmo para a **tradução das declarações de variáveis simples: inteira, booleana, real**. Este algoritmo deve promover a alocação de espaço para as variáveis, e estabelecer a correspondência entre as variáveis indicadas na declaração e sua respectiva área alocada na memória, contabilizando o uso da memória e registrando as variáveis e seus atributos nas tabelas de símbolos e de atributos.
- 3) Estender o algoritmo anterior de forma que traduza também **declarações de agregados** homogêneos (vetores e matrizes). Esse procedimento deverá alocar espaço para conter na memória os elementos do agregado, e também construir e depositar na memória, para cada matriz ou vetor declarado, o correspondente descritor, registrando tudo isso nas tabelas de símbolos e de atributos.
- 4) Projetar um algoritmo que, dados os índices e o descritor de um agregado homogêneo, **localize** na memória o elemento referenciado, **extraíndo** daí o valor corrente ou então nele **depositando** um novo valor fornecido.
- 5) Repetir as duas últimas atividades para **agregados heterogêneos**.

## PCS2056 – Linguagens e Compiladores

**Assunto:** Tradução das expressões.

**Objetivo:** Com base na especificação do ambiente de execução, desenvolvido anteriormente, projetar o mecanismo de tradução para as expressões aritméticas e booleanas de uma linguagem imperativa.

**Palavras-chave:**

tabelas de símbolos e de atributos  
expressões aritméticas  
expressões booleanas

**Atividades de projeto:**

- 1) Escolher uma linguagem de saída para a qual as expressões deverão ser traduzidas. Escolher as convenções de representação das variáveis e constantes aritméticas e booleanas: formatos internos para números inteiros e reais, e a convenção para os valores true e false.
- 2) Projetar um esquema de tradução de expressões aritméticas simples, sem parênteses, envolvendo variáveis simples e constantes aritméticas e as cinco operações aritméticas binárias usuais: soma, subtração, multiplicação, divisão, potenciação.
- 3) Completar o projeto de expressões aritméticas ampliando as expressões da questão anterior para incorporarem operadores + e – unários, parênteses e chamadas de funções.
- 4) Projetar um esquema de tradução para a comparação entre expressões aritméticas usando operadores de comparação >, <, =, ≠, ≥, ≤.
- 5) Projetar a compilação de expressões booleanas simples, envolvendo variáveis e constantes booleanas, e os operadores booleanos usuais and, or, not.
- 6) Ampliar as expressões booleanas incorporando as comparações entre expressões aritméticas, chamadas de funções booleanas e parênteses.
- 7) Completar as expressões booleanas permitindo a comparação entre expressões booleanas através dos operadores de comparação = e ≠.