

Bloco 3

Construção de interpretadores e compiladores

Resumo:

- Concepção e construção de gramáticas simples.
- Concepção e implementação de interpretadores e compiladores

Exercício 3.1

Neste problema pretende-se a concepção e desenvolvimento de um interpretador para processar tabelas resultantes de ficheiros *CSV*¹. As escolhas a tomar no desenvolvimento da gramática (ou das gramáticas, caso se justifique mais do que uma), é livre, mas deve ser possível a realização de pelo menos as seguintes operações (para as quais devem existir instruções nesse interpretador):

- Ler ficheiros *CSV*, registando o seu conteúdo numa estrutura de dados adequada, tipo tabela com cabeçalho (nome dos campos) e conteúdo;
- Definição de variáveis tipo tabela;
- Escrever tabelas em ficheiros ou na consola;
- Criar novas tabelas por selecção de colunas de uma tabela existente.
- Juntar duas tabelas numa nova tabela, tomando uma coluna de cada uma como termo de comparação (por exemplo, se tivermos duas tabelas com classificações de duas avaliações numa disciplina, deve ser possível juntar as duas tabelas numa única usando como critério, por exemplo, o número mecanográfico).

¹ *Comma separated values.*

- Definir expressões numéricas elementares (operações aritméticas), e permitir que sejam aplicadas a tabelas. O nome do campo servirá como variável nessas expressões. Caso a célula da tabela não seja numérica, a operação não é aplicada a essa célula, devendo o utilizador ser adequadamente informado dessa impossibilidade.
- Definir expressões de texto elementares: concatenar (soma), apagar (subtracção), em condições de utilização similares às das expressões aritméticas.
- Guardar o estado do interpretador de forma persistente (i.e. em ficheiro), e permitir que se carregue estados previamente guardados.

No desenvolvimento deste problema deve seguir os seguintes passos:

- a. Criar exemplos simples e testar as operações “à mão”;
- b. Criar a gramática passo a passo, testando as operações que vão sendo criadas (obviamente que o primeiro passo deve ser uma gramática para ler ficheiros *CSV*).

Exercício 3.2

Construa uma linguagem e respectivo compilador, para uma calculadora de números inteiros de precisão arbitrária. O compilador deve gerar código em Java, utilizando a classe `java.math.BigInteger` para suportar contas com precisão arbitrária.

Tome em consideração os seguintes desafios (que devem ser implementados um a um):

- a. Suporte para a realização das operações aritméticas elementares – soma, subtracção, multiplicação, quociente e resto da divisão inteira – com as precedências habituais;
- b. Suporte para uso de variáveis;
- c. Suporte para tipos de dados inteiros (precisão arbitrária) e booleanos.
- d. Suporte para predicados (expressões booleanas geradas com operadores relacionais sobre inteiros e operadores booleanos envolvendo conjunção, disjunção e negação).
- e. Suporte para instruções condicionais e repetitivas;
- f. Suporte para funções.