ESCOLA SUPERIOR DE TECNOLOGIA IPCA

Trabalho Prático

Processamento de Linguagens — 2017/2018

Licenciatura em Engenharia de Sistemas Informáticos

Versão 0.1 — 14 Jan 2018

Instruções

- Este enunciado corresponde ao trabalho prático de Processamento de Linguagens, regimes diurno e pós-laboral, no ano letivo 2017/2018, para a avaliação em época de exames (Janeiro);
- O trabalho prático poderá ser realizado em Grupo, com um **máximo de 3 alunos**;
- A data limite para a entrega do Trabalho Prático é o **dia do exame da UC**;
- As apresentações dos trabalhos práticos é feita por **todos os elementos do grupo**;
- Para além da implementação em C + flex + bison do projeto, cada grupo deverá incluir um conjunto de testes que demonstrem as funcionalidades implementadas;
- Deverá ser preparado um pequeno relatório que explique de que forma o enunciado foi interpretado, e quais as decisões tomadas na sua implementação. Deverá, obrigatoriamente, incluir a gramática implementada, em notação BNF.
- Qualquer detalhe que não esteja claro no enunciado deve ser extrapolado pelos alunos, optando pela interpretação que lhes parecer mais lógica/funcional.

Aritmética Fraccionária

Pretende-se a implementação de uma calculadora capaz de realizar operações com números inteiros e fraccionários (e nunca números reais). Assim, enquanto que uma operação como 4/2 deverá indicar como resultado o valor 2, uma operação como 3/2 deverá indicar como resultado o valor fraccionário $\frac{3}{2}$.

A linguagem deve suportar as operações aritméticas básicas (soma, subtracção, multiplicação e divisão) garantindo sempre que o resultado é um valor inteiro ou fraccionário. Neste último caso, o valor deverá ser simplificado (ou seja, 6/4 deverá ser apresentado como 3/2). Para além das quatro operações aritméticas, o utilizador deverá poder aplicar parentesis para alterar a prioridade dos operadores. Além das operações básicas, pretende-se que se implemente, também, a exponenciação (usando o caracter circunflexo), com expoentes inteiros (e bases inteiras ou fraccionárias).

Cada expressão termina com uma mudança de linha (*new line*). Assim, duas expressões em linhas diferentes devem ser avaliadas separadamente, e apresentados dois resultados distintos. Quando numa expressão se usar o caracter '`.`', este deve ser substituído pelo resultado da expressão anterior. Assim, a sequência:

```
4 * 3 * 2
. + 3
```

Deverá apresentar o resultado 24 para a primeira linha, e 27 para a segunda.

Pretende-se, ainda, a possibilidade de definir funções simples, apenas com um argumento, usando a seguinte sintaxe:

```
def quadrado _ ^ 2
```

Ou seja, nas funções, o caracter sublinhado (`_`) deve ser substituído pelo argumento da função.