

Processamento de Linguagens

LEI (3ºano)

Trabalho Prático nº 2 (Yacc)

Ano lectivo 14/15

1 Objectivos e Organização

Este trabalho prático tem como principais **objectivos**:

- (genericamente) aumentar a experiência em *engenharia de linguagens* e em *programação generativa (gramatical)*;
- (especificamente) desenvolver processadores de linguagens segundo o método da *tradução dirigida pela sintaxe*, suportado numa gramática tradutora;
- (especificamente) desenvolver um **compilador** gerando código para uma **máquina de stack virtual**;
- (especificamente) utilizar *geradores de compiladores* baseados em *gramáticas tradutoras*, como o Yacc ou o AnTLR.

e como **objectivos** secundários:

- aumentar a experiência de uso do ambiente Linux, da linguagem imperativa C ou em alternativa Java (para codificação das estruturas de dados e respectivos algoritmos de manipulação), e de algumas ferramentas de apoio à programação;
- rever e aumentar a capacidade de escrever *gramáticas independentes de contexto* que satisfaçam a condição LR() usando BNF-puro, ou em alternativa que satisfaçam a condição LL(k) usando extended-BNF.

Para o efeito, esta folha contém apenas 1 enunciado.

O programa desenvolvido será apresentado aos membros da equipa docente, totalmente pronto e a funcionar (acompanhado do respectivo relatório de desenvolvimento) e será defendido por todos os elementos do grupo, na última semana de Janeiro, em dia a combinar.

O **relatório** a elaborar, deve ser claro e, além do respectivo enunciado, da descrição do problema, das decisões que lideraram o desenho da linguagem/gramática e as regras de tradução para **Assembly** da VM (incluir as especificações Yacc), deverá conter exemplos de utilização (textos fontes diversos e respectivo resultado produzido). Como é de tradição, o relatório será escrito em L^AT_EX.

2 Enunciado

Pretende-se que comece por definir uma linguagem de programação imperativa simples (LPIS), a seu gosto.

Apenas deve ter em consideração que a LPIS terá de permitir manusear variáveis do tipo inteiro (escalar ou array) e realizar as operações básicas como atribuições de expressões a variáveis, ler do *standard input* e escrever no *standard output*. As instruções vulgares para controlo do fluxo de execução—*condicional* e *cíclica*—devem estar também previstas.

Sobre inteiros, estão disponíveis as habituais operações aritméticas, relacionais e lógicas, bem como a operação de indexação sobre arrays.

Como é da praxe neste tipo de linguagens, as variáveis deverão ser declaradas no início do programa e não pode haver

re-declarações, nem utilizações sem declaração prévia; se nada for explicitado, o valor da variável após a declaração é indefinido.

Note que o desenho da LPIS deve ser validado com o docente antes de avançar para a fase seguinte.

Desenvolva, então, um compilador para LPIS, com base na GIC criada acima e recurso ao Gerador Yacc/ Flex, ou ao AnTLR.

O compilador de LPIS deve gerar **pseudo-código**, **Assembly** da Máquina Virtual VM cuja documentação completa será disponibilizada no Bb.

Note que neste projeto, muitas tarefas são deixadas à criatividade do grupos de trabalho.