

# Processamento de Linguagens I

## LESI + LMCC (3º ano)

6ª Ficha Prática

Ano Lectivo de 01/02

### 1 Objectivos

Esta ficha prática contém exercícios para serem resolvidos nas aulas teórico-práticas com vista a sedimentar os conhecimentos relativos a:

- Implementação de parsers com a ferramenta yacc;
- Com cálculo directo;
- Com construção da árvore de sintaxe abstracta.

### 2 Enunciados

#### Processamento de Registos

Especifique através de uma gramática uma linguagem para o registo da avaliação dos alunos. Considere o seguinte texto como um exemplo duma frase da linguagem.

NOTAS

4140:José Carlos Ramalho:14:10:16

4156:Paulo Jorge Domingues:11:15:17:12:10

4238:José Alberto Rodrigues:12:12:10:13:14:11:12

...

FIM

Realize os seguintes pontos:

1. depois de especificar a gramática especifique o analisador léxico em lex;
2. especifique o analisador sintáctico em yacc;
3. especifique uma makefile para gerar o executável;
4. teste o executável obtido;
5. acrescente as acções semânticas necessárias de modo a que o reconhecedor produza uma lista de alunos com a seguinte estrutura:

NOTAS-FINAIS

nome:média

...

FIM

## Percursos de Orientação

A Orientação, na sua variante pedestre, é um desporto com grande desenvolvimento em Portugal. Basicamente, um atleta recebe um mapa que tem um percurso desenhado; nesse percurso, há uma série de *pontos quentes* assinalados que o atleta terá que *visitar* pela ordem em que estes aparecem. A gramática seguinte surgiu na tentativa de especificar a lista de percursos que compõem a prova (um por escalão etário).

O seu vocabulário é  $T = \{ \text{PROVA, FIM, PONTOS, FPONTOS, ";", "(", ",", ")"}, \text{PERCURSOS, FPERCURSOS, ":", ".", id, int, escalao} \}$  (*id* é uma ou mais letras maiúsculas ou minúsculas, *int* é uma sequência de dígitos, *escalao* é um *id*) e  $N = \{ \text{OProva, ListaPontos, Pontos, Ponto, IdPonto, CoordX, CoordY, ListaPercursos, Percursos, Percurso, IdPercurso, ListaRefs} \}$ , o axioma é *OProva* e contém um conjunto de produções que se mostra abaixo:

$$OProva \rightarrow PROVA\ ListaPontos\ ListaPercursos\ FIM \quad (1)$$

$$ListaPontos \rightarrow PONTOS\ Pontos\ FPONTOS \quad (2)$$

$$Pontos \rightarrow Pontos\ ';\ ' Ponto \quad (3)$$

$$Pontos \rightarrow Ponto \quad (4)$$

$$Ponto \rightarrow '( IdPonto\ ',\ CoordX\ ',\ CoordY\ )' \quad (5)$$

$$IdPonto \rightarrow id \quad (6)$$

$$CoordX \rightarrow int \quad (7)$$

$$CoordY \rightarrow int \quad (8)$$

$$ListaPercursos \rightarrow PERCURSOS\ Percursos\ FPERCURSOS \quad (9)$$

$$Percursos \rightarrow Percursos\ ';\ ' Percurso \quad (10)$$

$$Percursos \rightarrow Percurso \quad (11)$$

$$Percurso \rightarrow IdPercurso\ '!\ ' escalao\ '!\ ' ListaRefs\ '!\ ' \quad (12)$$

$$IdPercurso \rightarrow id \quad (13)$$

$$ListaRefs \rightarrow ListaRefs\ '!\ ' IdPonto \quad (14)$$

$$ListaRefs \rightarrow IdPonto \quad (15)$$

A distância entre dois pontos pode ser calculada pela fórmula:

$$dist = \sqrt{|y_2 - y_1|^2 + |x_2 - x_1|^2}$$

1. Derive a gramática abstracta para a gramática apresentada.
2. Usando as regras apresentadas nas aulas teóricas, derive as estruturas de dados em C e as respectivas funções de construção: `oriABS.h` e `oriABS.c`.
3. Construa o analisador léxico em `lex` e o analisador sintático em `yacc`.
4. Acrescente ao parser desenvolvido as acções semânticas para a construção da representação abstracta.
5. Desenvolva uma função semântica em C (ficheiros: `oriSEM.h` e `oriSEM.c`) que realiza uma travessia à representação abstracta e que produz uma lista com a seguinte estrutura:

`IdPercurso:Distancia`