

# Processamento de Linguagens I

## LESI + LMCC (3º ano)

3º Ficha Prática

Ano Lectivo de 05/06

### 1 Objectivos

Esta ficha prática contém 1 exercício para ser resolvido nas aulas teórico-práticas com vista a consolidar definitivamente os conhecimentos práticos relativos a:

- uso de Gramáticas Independentes de Contexto (GIC) para definir a sintaxe de Linguagens;
- uso de Gramáticas Tradutoras (GT) para definir a semântica estática e dinâmica de Linguagens;
- uso de GT para desenvolver programas eficientes, baseados em algoritmos standard guiados por Tabelas de Decisão (construídas a partir de Autómatos Finitos Deterministas com stack), para reconhecer e processar Linguagens, desencadeando Acções Semânticas específicas ao reconhecer as produções da gramática;
- geração automática de programas a partir de especificações formais;
- uso da ferramenta Yacc, disponível em ambiente Linux, para geração automática de processadores de linguagens, nomeadamente para criação de *Tradutores Dirigidos pela Sintaxe* (*Analísadores Sintácticos* e *Analísadores Semânticos*).

### 2 Tradutores Dirigidos pela Sintaxe

Na medida em que esta ficha tem por objectivo reforçar os conhecimentos e aptidões que os alunos devem ter adquirido ao resolver a Ficha 2, começa-se por relembrar os conceitos básicos aí introduzidos.

No contexto do desenvolvimento de Compiladores, ou mais genericamente de Processadores de Linguagens, o primeiro nível, ou tarefa a implementar, é a **análise léxica** que tem por missão ler o texto fonte e converter todas as palavras correctas em símbolos terminais dessa linguagem. O segundo nível é a **análise sintáctica** que pega nos símbolos recebidos do AL e verifica se a sequência em causa respeita as regras de derivação, ou produções, da gramática. O terceiro nível é a **análise semântica** que calcula o valor, ou significado, exacto da frase e, então, valida se a sequência de símbolos sintacticamente correcta cumpre todas as condições de contexto que a tornam semanticamente válida. O quarto nível é a **tradução** que pega no significado exacto da frase válida e constrói, ou calcula, o resultado final.

Tendo em vista o a consolidação da capacidade de desenvolver PLs com recurso a Geradores de Compiladores, propõe-se para esta aula o recurso à ferramenta Flex para gerar o **Analísador Léxico (AL)**, e o recurso à ferramenta Yacc para gerar os **Analísadores Sintáctico e Semântico** e o **Tradutor** a partir da Gramática Tradutora da Linguagem a processar.

Para resolver o exercício proposto, proceda em etapas:

1. escreva a GIC; identifique os símbolos terminais, descreva-os à custa de Expressões Regulares e gere o AL; gere o *Parser* (Analisador Sintático) que lhe permitirá, apenas, verificar a correcção sintáctica das frases;
2. escreva, depois, uma primeira versão da GT, acrescentando à GIC anterior as Acções Semânticas necessárias para, apenas, validar a correcção semântica das frases;
3. escreva, a seguir, uma versão final da GT, acrescentando à GT anterior as Acções Semânticas necessárias para calcular e escrever o resultado desejado.

## 2.1 Gerador de Indices Remissivos

Considere uma situação em que se pretende gerar automaticamente, a partir de uma descrição como a que se ilustra abaixo, o Índice Remissivo a inserir no fim de um determinado documento. Recorda-se que um Índice Remissivo é uma lista, ordenada alfabeticamente, de todos os termos a destacar num documento, com a indicação das páginas do documento onde cada termo foi definido. A descrição inicial, escrita de acordo com uma linguagem específica que deverá ser definida para o efeito, permite indicar para cada página (referida pelo seu número ou nome do apêndice) a lista de termos a destacar. Abaixo ilustra-se uma frase válida da dita Linguagem:

INDICE

```
1 = processador, linguagem, compilador
2 = compilador, interpretador, gramática
3 = gramática, GR, GIC
4 = gramática, linguagem, reconhecedor
A = gramática, YACC
B = LRC
```

FIM DO INDICE

O Índice que deve ser gerado para este exemplo é apresentado a seguir:

ÍNDICE REMISSIVO

```
compilador: 1, 2.
GIC: 3.
GR: 3.
gramática: 2, 3, 4, Apx A.
interpretador: 2.
linguagem: 1, 4.
LRC: Apx B.
processador: 1.
reconhecedor: 4.
YACC: Apx A.
```

No contexto desta aula o que se pretende é: que escreva a gramática que define a linguagem pretendida e que desenvolva, com auxílio dos Geradores *Lex* e *Yacc* e seguindo os passos acima, o processador (reconhecedor+calculador e verificador) necessário, tomando por base essa gramática.