

# Processamento de Linguagens I

## LESI + LMCC (3º ano)

3º Ficha Prática

Ano Lectivo de 04/05

### 1 Objectivos

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

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.

Com esse fim em vista e assumindo que o 1º nível já está implementado (graças ao uso do Flex para gerar o **Analísador Léxico (AL)**), propõe-se para esta aula 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 cada um dos exercícios, proceda em três etapas:

1. escreva a GIC e gere o *Parser* (Analísador Sintáctico) 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 Máquina de Venda de Chocolates

Considere uma situação em que se pretende simular o funcionamento de uma máquina de vender chocolates. Dado o stock no início do dia (nome, preço e quantidade de cada produto disponível), a quantia inicial de trocos e os registos das vendas diárias (nome do chocolate escolhido e a quantia introduzida), o objectivo é calcular *o stock final e o dinheiro acumulado*, considerando todos os pedidos válidos ao longo do dia. O sistema deve ainda assinalar todos os pedidos que não puderam ser satisfeitos, por o chocolate pedido não existir em stock, ou a quantia não ser suficiente.

No contexto desta aula o que se pretende é: que desenvolva, com auxílio do Gerador Yacc, o processador (reconhecedor+calculador e verificador) pretendido, tomando por base a gramática—da linguagem para descrever o estado inicial da máquina e os registos de vendas efectuadas durante o dia—criada na aula anterior (Ficha Prática 2); utilize o AL desenvolvido também nessa aula.

## 2.2 Anuário dos Medicamentos brancos

Considere agora uma outra situação em que, para auxiliar o Instituto Farmacêutico do Ministério da Saúde na gestão do novo lote de medicamentos brancos, se pretende criar um sistema de consulta a esses medicamentos acessível a qualquer farmácia via um browser HTML. Esse sistema deve mostrar a informação agrupada por: classe de medicamentos no Symposium Terapêutico (uma página por classe, com os medicamentos ordenados alfabeticamente); ou por fabricante (uma página única, com os medicamentos agrupados por fabricante).

Sobre cada medicamento é fornecida a seguinte documentação: nome, código, classe, composição química, preço recomendado, fabricantes disponíveis.

No contexto desta aula o que se pretende é: que desenvolva, com auxílio do Gerador Yacc, o processador (reconhecedor+calculador e verificador) pretendido, tomando por base a gramática—da linguagem para definir o ano a que o Symposium Terapêutico diz respeito, a lista das classes e de fabricantes válidos, e descrever a informação envolvida no lote de medicamentos a considerar—criada na aula anterior (Ficha Prática 2); utilize o AL desenvolvido também nessa aula.

## 2.3 Documento anotado em XML

Como sabe um Documento XML é um texto vulgar semeado de anotações, ou marcas, que são identificadores especiais intercalados entre os caracteres "<" e ">". Num documento XML bem formado, a cada marca de abertura corresponderá uma marca de fecho, que tem o mesmo identificador, mas que começa por "</". Dentro de cada marca, isto é entre "<" e ">", além do identificador do elemento, ainda podem aparecer triplos formados por um outro identificador (de atributo), pelo sinal "=" e pelo respectivo valor que é qualquer texto entre aspas. Cada fragmento do documento entre marcas deve ser considerado em bloco como sendo o símbolo PCDATA.

Tomando por base esta descrição de Documento XML, pretende-se desenvolver um programa que valide se toda a marca que abre fecha, pela ordem correcta (uma marca aberta dentro de outra, terá de fechar antes da primeira), e que liste todas as marcas encontradas indicando a frequência de cada uma (número de ocorrências sobre o total de marcas). O processador também deve verificar que a mesma marca abre sempre associada ao mesmo conjunto de atributos.

No contexto desta aula o que se pretende é: que desenvolva, com auxílio do Gerador Yacc, o processador (reconhecedor+calculador e verificador) pretendido, tomando por base a gramática de um Documento XML criada na aula anterior (Ficha Prática 2); utilize o AL desenvolvido também nessa aula.