

# Processamento de Linguagens I

## LESI + LMCC (3<sup>o</sup> ano)

Exame de 2<sup>a</sup> Época

Data: 23 de Julho de 2004  
Hora: 09:30

Dispõe de 2:30 horas para realizar este exame

### Questão 1: linguagem e gramática de atributos

Uma equipa de historiadores está a levantar informação, nos vários arquivos do País, sobre individualidades do Clero Catedralício português em determinado período da Idade Média e referente às 9 dioceses existentes nessa época.

De modo a facilitar o trabalho desses investigadores, quer-se definir uma linguagem que permita processar automaticamente (no sentido de validar, normalizar e criar uma base de dados) as notas extraídas de cada documento consultado. Para isso a linguagem terá de possibilitar a descrição dos documentos (identificação do arquivo e cota do documento, sua data e data da consulta, bem como o redactor e o tipo) e, para cada documento, a identificação de todos clérigos que sejam mencionados (seu nome e apelido, diocese a que pertence, categoria eclesiástica e papel na evento relatado no documento em causa).

O Arquivo e Tipo de Documento, bem como a Diocese e a Categoria Eclesiástica de cada clérigo, só podem tomar valores dentro de conjuntos finitos e conhecidos à partida; além disso, a Cota dentro de cada arquivo não pode ser repetida. Assim sendo o seu processador terá de validar o texto processado, garantido que estas restrições semânticas são observadas.

O que se pede nesta questão é que desenhe a Gramática de Atributos que especifica a sintaxe e semântica da linguagem pretendida.

Assuma que a todos os símbolos terminais estão associados sempre e somente três *atributos intrínsecos*: `lexval`, do tipo `string`, que representa a sequência de caracteres lida pelo analisador léxico ao reconhecer o símbolo em causa; `lin` e `col`, ambos do tipo `inteiro`, representando a posição do símbolo no texto fonte.

Responda, então, às seguintes alíneas:

- Construa uma **gramática independente de contexto abstracta** (GICA) que defina a linguagem pretendida (identifique cada produção).
- Usando uma tabela para mais fácil leitura, associe a cada símbolo não-terminal os respectivos **atributos herdados e sintetizados**, indicando o tipo de cada um.
- Escreva as **regras para o cálculo dos atributos**, a associar a cada produção da GICA.
- Acrescente às produções convenientes as **condições contextuais** para garantir a correcção semântica das frases da linguagem, indicando a mensagem de erro a emitir em cada caso de violação da condição.
- Acrescente ainda às produções convenientes as **regras de tradução** necessárias para mostrar, no fim, a informação que se vai acumulando relativamente a cada clérigo, contendo referência a cada documento onde essa individualidade foi citada.
- Se resolvesse usar o LRC para gerar automaticamente o processador pretendido para as frases da linguagem descrita pela GA que construiu, diga de que maneira resolvia a dificuldade inerente ao facto de não existirem em LRC os

conceitos de condição de contexto e de regra de tradução. Como organizava (em quantos módulos e ficheiros) a gramática de atributos para submeter ao Gerador em causa?

## Questão 2: parsing e gramática tradutora

Para auxiliar um Joalheiro, concebeu-se uma linguagem que lhe permite descrever as peças constantes de uma determinada colecção, mostrando-se abaixo a respectiva especificação sintáctica.

Analise atentamente a seguinte gramática independente de contexto, cujo Símbolo Inicial é *Joias* em que os Símbolos Terminais são escritos em minúsculas (pseudo-terminais), ou em maiúscula (palavras-reservadas), ou entre apostrofes (sinais de pontuação) e a string nula é denotada por  $\epsilon$ .

```
p1: Joias    --> Joia OutrasJ
p2: OutrasJ --> Joia
p3:         | OutrasJ Joia
p4: Joia    --> Ref Desc Mat Classe Preco
p5: Mat     --> OURO
p6:         | PRATA
p7: Classe  --> BRINCO
p8:         | COLAR
p9:         | ANEL
p10:        | PULSEIRA
p11:        | OUTRO
p12: Desc   --> str
p13: Preco  --> num
p14: Ref    --> pal
```

Recordando que o carácter \$ representa o fim-de-ficheiro (do texto de entrada) responda, então, às alíneas seguintes:

a) Mostre que o texto

```
B004001 "pecaXXXX" OURO BRINCO 250.00 BP040012 "pecaYYY" PRATA BRINCO 389.00
```

é uma frase válida dessa linguagem, apresentando a respectiva árvore de derivação.

- b) Altere a GIC apresentada de modo a que a linguagem obrigue a agrupar as jóias por classe (brinco, colar, ..., outro).
- c) Construa, justificando o processo, a Tabela de Decisão LL(1) completa, isto é com entradas para todos os símbolos não-terminais e terminais.
- d) Mostre, observando a Tabela de Decisão construída na alínea anterior, que a gramática apresentada não é LL(1).
- e) Modifique a GIC acima para resolver os conflitos LL(1) e, considerando essa nova gramática, diga como inicializava a *stack de parsing* caso pretendesse usar um *parser top-down* do tipo LL(1). Qual será o conteúdo dessa *stack de parsing* após ler todo o texto fonte caso este seja uma frase válida da linguagem.
- f) Mostre, construindo o autómato determinista LR, que a gramática apresentada não é LR(0).
- g) Prove que a adopção da estratégia SLR(1) resolvía os conflitos acima detectados, e mostre a ordem pela qual os nodos da árvore de derivação da alínea a) seriam reconhecidos por um *parser* desse tipo.
- h) Para reduzir o número de estados do autómato LR, e diminuir o tamanho das respectivas Tabelas de Decisão, é vulgar acrescentar ao *parser LR* mais um tipo de acção dito *shift-reduce(p)* (*transita-reduz(p)*) que se usa num estado  $q$  sempre que há uma transição de  $q$  para  $q_1$  por  $t$  (símbolo terminal) e em  $q_1$  há apenas uma redução por  $p$  (produção da GIC); desta forma elimina-se o estado  $q_1$  e poupa-se uma transição. Se aplicasse essa estratégia ao autómato LR(0) que construiu na alínea e), quantos e quais estados pouparia?
- i) Transforme a GIC dada numa GT (gramática tradutora) reconhecível pelo yacc para produzir uma listagem final (após processar toda a descrição da colecção) em que as peças estariam ordenadas por referência e agrupadas pelo tipo de material (ouro ou prata).