Métodos de Programação III
LESI + LMCC (3º ano)

Exame de 1ª Época – 2ª Chamada

Data: 01 de Fevereiro de 2003
Hora: 09:30

Dispõe de 2:30 horas para realizar este exame

1 Questão: Definição de ERs

Considere as expressões regulares, p e q, abaixo e as linguagens por elas definidas, $\mathcal{L}_p$ e $\mathcal{L}_q$:

$$p = (a^* \ + \ b^*) \ (a \ + \ b)$$

$$q = (a^+ \ + \ b^+)$$

e responda às seguintes alíneas:

a) É um facto que as frases de $\mathcal{L}_p$ e $\mathcal{L}_q$ tem no mínimo um símbolo e começam sempre por um `a` ou um `b`. Explique, então, qual a diferença entre essas linguagens.

b) Mostre que $'aaa' \in \mathcal{L}_p$, apresentando o processo de derivação sucessiva.

2 Questão: Modelação com ERs

Em Linux o comando para listar o conteúdo de uma directoria, ou os atributos de um ficheiro, é formado pelo nome de operação `ls`, seguido de zero ou mais letras precedidas pelo sinal '-' (as quais indicam as opções do comando que se quer activar), seguido ainda, opcionalmente, por um *pathname* (que identifica a directoria ou o ficheiro).
Escolhendo os símbolos apropriados, defina a sintaxe desse comando, através de uma Expressão Regular.

3 Questão: ERs, Sistemas de Produção e Lex

Pretende-se criar um sistema que receba pedidos dos alunos por email, numa sintaxe flexível, como a apresentada na coluna da esquerda.

Os alunos podem pedir um ou mais exames, em que cada exame é identificado pelo tipo (chamada ou exame) e pela data.

Os alunos podem pedir a resposta para o email ou para o telemóvel. Para isso, é preciso gerar uma mensagem num formato adequado, de acordo com os exemplos apresentados à direita (gere só a mensagem; ignore que seria preciso também anexar o enunciado propriamente dito).

```
Bom dia,
Eu sou o aluno Luís Vaz de Camões, e gostaria
de receber a 1ª Chamada de 12-01-2002, para o
meu email luis_camoes@mail.prazeres.pt
Cumprimentos,
Luís
```

```
MAIL FROM: mpiii@di.uminho.pt
RCPT TO: luis_camoes@mail.prazeres.pt
Subject: Exames de MPIII
Caro(a) Luís,
Junto envio:
Chamada de 12-01-2002.
Adeus.
```

```
Olá, queria o Exame de 26-09-2002 e também a
2ª Chamada de 24-02-2002, por favor, para o
meu telemóvel 919000001.
Cumprimentos,
Inês de Castro
```

```
FROM:351253604470
TO:351919000001
Caro(a) Inês de Castro,
Junto envio:
Exame de 26-09-2002.
Chamada de 24-02-2002.
Adeus.
```

```
Poderiam-me enviar, por favor, o Exame de 14-11-2002,
para o meu email pimenta@espanha.pt
Cumprimentos,
Pimenta
```

```
MAIL FROM: mpiii@di.uminho.pt
RCPT TO: pimenta@espanha.pt
Subject: Exames de MPIII
Caro(a) Pimenta,
Junto envio:
Exame de 14-11-2002.
Adeus.
```

a) Escreva um programa Lex, com as respectivas acções, que implemente a transformação pretendida.

b) Imagine que precisava de gerar, para a segunda resposta, uma mensagem com a seguinte forma:

```
FROM:351253604470
TO:351919000001
NUMTOTALCHARS:148
Caro(a) Inês de Castro,
Junto envio:
Exame de 26-09-2002.
Chamada de 24-02-2002.
Adeus.
```

Indique que alterações precisava de fazer ao programa Lex para gerar uma resposta, em que aparecesse o tamanho da própria resposta.

4 Questão: Modelação com Autómatos

Determinada aplicação designada por $\mathcal{X}$, desenvolvida para apoio à análise de registos individuais e familiares em demografia, oferece no primeiro nível 4 opções: Indivíduos; Famílias; Análises; e Terminar.
Escolhendo Terminar volta-se ao nível 0 (ponto de arranque).
Escolhendo Análises pode-se optar por mais 3 operações: Análise Multidimensional; Mineração de dados; Análise Geográfica—findas as quais se volta automaticamente a este submenu.
Escolhendo Indivíduos pode-se optar por mais 5 alternativas: Baptizados; Casamentos; Óbitos; Procura Individuo; Sair. A opção Sair faz voltar ao menu do nível 1; a opção Baptizados permite Inserir um registo; as opções Casamentos e Óbitos permitem Procurar um Individuo e caso exista permite a seguir Inserir um registo. Estas três últimas opções voltam sempre automaticamente ao mesmo submenu.

A opção Procurar lê um termo de pesquisa. A aplicação responde à pesquisa com a indicação se existe ou não o registo. A Inserção lê o nome do indivíduo a inserir.
De momento , ignoram-se as opções relativas à escolha Famílias.

Desenhe um Autómato Determinista que modele o comportamento do programa $\mathcal{X}$.

5 Questão: ERs, ANDs e ADs

Considere a expressão regular: $(a+b+c)(abc)^{+}(a+b+c)$

a) Desenhe o correspondente autómato não determinista, de acordo com estas duas regras:

1. terá que ter três estados iniciais.
2. usando as regras de conversão sugeridas nas fichas teórico-práticas.

b) Relembre a seguinte definição de AFND em Haskell:

```
data Ndfa st sy = Ndfa  [sy] -- Finite set of alfabet symbols
                        [st] -- Finite set of states
                        [st] -- The set of start states
                        [st] -- The set of final states
                        ( st -> Maybe sy -> [ st ]) -- Transition Function
```

Declare, em Haskell, o autómato desenhado na alínea anterior.

c) Calcule o e-fecho de conjunto de estados iniciais, do autómato anterior.

d) Calcule o autómato finito determinista, que define a mesma linguagem que o autómato não determista da primeira alínea. Apresente a tabela de conversão utilizada.

6 Questão: escrita de uma gramática

Pretende-se desenvolver uma linguagem para descrever livros que entram numa biblioteca. A linguagem permitirá indicar o título, editora e ano de edição do livro, a lista dos autores (um ou mais) e, opcionalmente, os temas (1 palavra por tema) em que o livro pode ser classificado.
Responda, então, às seguintes alíneas:

a) Escreva uma gramática independente de contexto (GIC) que defina a linguagem pretendida.

b) Dê um exemplo de uma frase válida da sua linguagem, apresentando a respectiva árvore de derivação.

7 Questão: Gramáticas, Linguagens e Parsing

Analise atentamente a seguinte GIC, cujo Símbolo Inicial é `Opers`, em que todos os símbolos escritos em minúsculas são pseudo-terminais, as palavras-reservadas estã escritas em maiúsculas, os sinais estão entre apóstrofos e `&` representa a string nula.

```
Opers    -->  Op Outras  '.'
Outras   -->  Op Outras
Outras   -->  &
Op       -->  SOMA  Ns
Op       -->  MULTIPLICA  Ns
Ns       -->  num Resto
Resto    -->  Ns
Resto    -->   &
```

Responda, então, às alíneas seguintes:

a) Comente a seguinte afirmação, avaliando a sua veracidade: *num parser recursivo descendente, para a GIC acima, em que se optimize o reconhecimento dos terminais, podem poupar-se 3 funções, sem que isso implique uma significativa degradação de eficiência.*

b) Escreva as funções de um parser recursivo descendente para reconhecer os símbolos `Op` e `Outras`.

c) Admitindo que se pretende usar a linguagem Haskell para implementar as funções de reconhecimento e processamento das frases da linguagem gerada pela gramática acima, suponha que já existem funções para fazer o parsing dos símbolos `num` e `Resto` que retornam como resultado, respectivamente, um número inteiro e uma lista de inteiros. Escreva, então em Haskell e usando *combinadores de parsing*, uma função para reconhecer o símbolo `Ns` e retornar uma nova lista de inteiros.