Processamento de Linguagens
LEI (3º ano)

1ª Exame

Data: 18 de Julho de 2008
Hora: 09:30

Dispõe de 1:30 + 1:30 horas para realizar este exame

I Parte (20v)

1 Filtros de Texto e Analisadores Léxicos (8v)

Supondo que vai usar o gerador Flex para construir automaticamente os programas abaixo descritos, escreva a respectiva especificação:

a) Um filtro que expanda abreviaturas à moda do próprio LaTeX. Para isso as abreviaturas (*palavra simples* apenas formada por letras ou dígitos e necessariamente terminada por um espaço, fim-de-linha, ou caracter de pontuação) aparecem no meio do texto, em qualquer lado, sempre iniciadas por um '*backslash*' (i.é, uma barra para a esquerda ou \). As abreviaturas são definidas no próprio texto, em qualquer lado embora só tenham efeito a partir do ponto de definição, através do comando

```
\def\abrev{texto-expandido}
```

Note que no texto podem aparecer muitas palavras iniciadas da mesma forma por um '*backslash*' e que se não tiverem sido definidas como abreviaturas serão consideradas comandos e devem ser mantidas inalteradas.

b) Um Analisador Léxico, a integrar no tradutor gerado pelo Yacc, para a linguagem definida pela GIC do exercício 4. O Analisador Léxico a gerar deve devolver os Símbolos Terminais com o respectivo valor semântico (quando necessário).
Para isso note que `id` é uma palavra formada letras e dígitos, desde que comece por uma letra; que `str` é uma sequência de caracteres entre aspas; que `ano` é formado por 4 dígitos começados por '2'; e que `num` é uma qualquer sequência de dígitos possivelmente começada por um sinal e que também poderá conter uma parte decimal separada da parte inteira por um '.'.

2 Expressões Regulares e Autómatos (12v)

Considere as seguintes ERs:

$$e1 \;=\; a \;+\; a^*\; b$$
$$e2 \;=\; a\; a^*\; b$$
$$e3 \;=\; a \;+\; (c \;+\; b)^*$$
$$e4 \;=\; a \;+\; (c^* \;+\; b^*)$$
$$e5 \;=\; a \;+\; (c\; b)^*$$

e os seguintes Autómatos:

```
AD1 :: < {'a', 'b', 'c'}, {1,2,3,4}, 1, {1,2,4}, delta >
  sendo delta 1 'a' = 2
        delta 1 'c' = 3
        delta 3 'b' = 4
        delta 4 'c' = 3

AD2 :: < {'a', 'b', 'c'}, {1,2,3}, 1, {1,2,3}, delta >
  sendo delta 1 'a' = 2
        delta 1 'c' = 3
        delta 1 'b' = 3
        delta 3 'b' = 3
        delta 3 'c' = 3

AND3 :: < {'a', 'b', 'c'}, {1,2,3,4,5,6}, 1, {3,5,6}, delta >
  sendo delta 1  &  = 2       delta 4  &  = 6
        delta 1  &  = 4       delta 5 'b' = 5
        delta 2 'a' = 3       delta 5 'c' = 6
        delta 4 'b' = 5       delta 6 'c' = 5
```

Responda, então, às seguintes questões:

a) explique por palavras suas porque é que $e1$ e $e2$ não são equivalentes.

b) use agora os respectivos Autómatos Deterministas (que terá de construir informalmente) para dar a mesma justificação da alínea anterior sobre a não-equivalência de $e1$ e $e2$.

c) usando a respectiva *cadeia de derivação*, mostre que a frase "`aaab`"pertence à linguagem gerada por $e2$.

d) diga justificando se a frase "`aaab`"também podia ser gerada por $e1$.

a) explique por palavras suas a diferença entre $e3$, $e4$ e $e5$.

f) sabendo que AD1 e AD2 são equivalentes a duas das expressões regulares $e3$, $e4$ e $e5$, estabeleça essa correspondência.

g) construa, aplicando as regras sistemáticas, um autómato não-determinista equivalente a $e5$.

h) escreva um reconhecedor para a linguagem gerada por $e4$ usando um conjunto de funções recursivas.

i) o que o leva a dizer que `AD1` e `AD2` são Autómatos Deterministas e `AND3` é Não-Determinista?

j) considerando o autómato `AND3`, calcule o $\varepsilon fecho$ do estado 1.

k) diga justificando se a frase "`ccbc`"pertence à linguagem gerada por `AND3`.

l) escreva uma expressão regular $e6$ para descrever a linguagem formada por sequências de pares de 'a', ou por sequências de 'b' começadas e terminadas por um 'c'(p.ex., "`aaaaaa`"ou "`cbbbc`").

II Parte (20v)

3 Desenho/especificação de uma Linguagem (6v)

Pretende-se definir uma nova Linguagem que permita descrever as praias de Portugal, sabendo-se que a nossa costa está dividida em zonas (Costa Verde, Costa Brava, Costa de Prata, etc.). Cada zona costeira tem, além do nome, um código e uma área, podendo abarcar mais do que um concelho (caracterizado também por um código e nome) e oferece um conjunto de parques naturais. A nova Linguagem deve permitir descrever as zonas costeiras (através das características referidas) e associar a cada uma as praias (nome, comprimento e descrição) que nela se situam.
Escreva então uma Gramática Independente de Contexto, *GIC*, que especifique a Linguagem pretendida (note que o estilo da linguagem (mais ou menos verbosa) e o seu desenho são da sua responsabilidade).

4 Gramáticas, Linguagens, Parsing e Tradução (14v)

A gramática independente de contexto, *GIC*, abaixo apresentada, define uma linguagem específica para apoio na gestão de projectos.
O Símbolo Inicial é `Projs`, 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 `&`; os restantes (sempre começados por maiúsculas) serão os Símbolos Não-Terminais.

```
p0:  Projectos   -->  Projs '.'
p1:  Projs       -->  Proj
p2:                |  Projs ';' Proj
p3:  Proj        -->  PROJECTO Caracteriza  TAREFAS '{' Tarefas '}'
p4:  Caracteriza -->  Nome Tema Coordenador PERIODO Inic Fim CUSTO Valor
P5:  Nome        -->  id
p6:  Tema        -->  str
p7:  Coordenador -->  id
p8:  Inic        -->  ano
p9:  Fim         -->  ano
p10: Valor       -->  num
p11: Tarefas     -->  Taref
p12:               |  Tarefas '-' Taref
p13: Taref       -->  CodTar Desc Inic Fim Equipe
p14: Desc        -->  str
p15: Equipe      -->  id
p16:               |  Equipe ',' id
p17: CodTar      -->  id
```

Neste contexto e após analisar a *GIC* dada, responda às alíneas seguintes.

a) escreva uma Frase válida da linguagem gerada pela *GIC* dada, mostrando a respectiva Árvore de Derivação.

b) altere a gramática de modo a permitir que o projecto tenha mais de um Coordenador.

c) reduza a gramática *GIC* eliminando as produções inúteis, isto é, aquelas que derivam apenas em um símbolo.

d) o par de produções `p1/p2` define uma lista com recursividade à esquerda. Altere esse par para usar recursividade à direita e mostre, através das árvores de derivação, a diferença entre ambos os esquemas iterativos.

e) considere os seguintes iterativos (S1 e S2):

```
S1:                          S2:
L    --> ele ',' Cont        L    --> ele Cont
Cont --> ele                 Cont --> &
      |  L                          |  ',' ele Cont
```

e explique porque não são equivalentes.

f) explique as diferenças entre um parser Top-Down e um parser Bottom-Up.

g) embora siga também a estratégia Top-Down o parser Recursivo-Descende Puro é muito distinto do LL(1); esclareça essa diferença.

h) transforme a *GIC* dada numa **gramática tradutora**, *GT*, reconhecível pelo yacc, para:

- calcular e imprimir: o número de projectos; o número de tarefas de cada projecto; o numero total de colaboradores afectos às várias tarefas de um projecto.
- calcular e imprimir: a duração de cada projecto.
- identificar e listar por ordem alfabética os Coordenadores dos vários projectos.
- verificar se o período de realização de cada tarefa está dentro do período global do projecto.