Processamento de Linguagens e Compiladores
LCC (3º ano)

Exame de Recurso

Data: 04 de Fevereiro de 2016
Hora: 09:00

Dispõe de 2:00 horas para realizar este teste

1 Filtros de Texto (8v)

a) Especifique em Flex um filtro que dado um texto de entrada produza um texto de saída em que seja feita a numeração da profundidade das chavetas.
Exemplo: se fosse fornecido como entrada o seguinte texto

```
{
  { }
  { { } } } { } }
```

Na saída iríamos ter

```
{1
  {2 2}
  {2 {3 3} 2} 1} 0} {0 0} -1}
```

(números de chaveta menores ou iguais a '0' correspondem a erros)

b) Analise as ER (expressões regulares) abaixo, escritas na notação do Flex, e explique por palavras suas, com clareza, cada uma dando 2 elucidativos exemplos de frases que concordam com elas:

```
%%
^[EsTtEeAa]+":"[ \n]+
"</"[^>]+">"
"{".*
%%
```

c) Suponha que tem um ficheiro com as datas para as pessoas tomarem a vacina do tétano, mas que por engano algumas pessoas se increveram mais que uma vez. Construa em AWK um filtro para eliminar as repetições, mantendo apenas a primeira ocorrência.

No final o filtro deve ainda escrever o número de vacinas necessárias e de entradas removidas.
Exemplo: se o texto de entrada for

```
João da Silva:dia 12
Rui Meneses:dia 17
Joaquina Maria Santos:dia 19
```

```
Rui Meneses:dia 19
Rui Meneses:dia 21
Fagundes:dia 30
```

A saída correspondente pretendida é:

```
João da Silva:dia 12
Rui Meneses:dia 17
Joaquina Maria Santos:dia 19
Fagundes:dia 30

## vacinas necessárias: 4
## marcações removidas: 2
```

d) ?????????????? Considere a seguinte script GAWK:

```
#!/usr/bin/gawk -f
BEGIN  { RS="href=[\"']"; FS="[\"']"; }
NR > 1 { print $1}
```

Indique o que ela faz quando aplicada a um ficheiro HTML. Para ilustrar a sua resposta, escreva um pequeno exemplo HTML e a respetiva saída. ??????????????????

2 Expressões Regulares e Autómatos (1v)

Considere as seguintes ER:

$$e1 \;=\; (a \;+\; b)\; c^{*}\; d\; (b^{+} \;+\; a\; c)$$

Responda, então, às seguintes questões:

a) usando a respectiva *cadeia de derivação*, diga se a frase "`bccdbba`"pertence à linguagem gerada por $e1$.

b) construa informalmente o Autómato Determinista equivalente a $e1$.

3 Desenho/especificação de uma Linguagem (3+1v)

Um arquiteto, para desenhar a planta de cada piso de um edifício tem por hábito definir os blocos que representam cada divisão (retângulos, quadrados e triângulos) e depois compô-los usando os operadores de posição relativa "à-esquerda-de" (`esq`) e "por-cima-de" (`sobre`), como se exemplifica a seguir

```
G esq ( (Cz esq Q1) sobre S1)
```

para desenhar uma casa com uma Garagem ao lado esquerdo da Sala que tem por cima uma Cozinha à esquerda de um Quarto.
Naturalmente que o desenho só se poderá fazer se cada Bloco (`G, Cz, Q1, S1`)for previamente declarado indicando que figura geométrica é (conforme os 3 tipos acima enunciados) e as suas dimensões.
Neste contexto, responda às alíneas seguintes.

a) Escreva, em notação do Yacc, uma Gramática Independente de Contexto, *GIC*, que especifique uma Linguagem concreta para o arquiteto poder declarar os seus blocos e depois desenhar um piso ou mais (os quais devem ser devidamente identificados).

b) Escreva em Flex a especificação do Analisador Léxico para a linguagem definida pela *GIC* acima.

4 Gramáticas, e Parsing Top-Down (4v)

Considere a gramática independente de contexto, GIC, abaixo apresentada, atendendo a que os símbolos terminais T e não-terminais NT são definidos antes do conjunto de produções P, sendo Z o seu axioma ou símbolo inicial.

```
T  = { t, d, a, b, v, s1, s2 }
NT = { Z, D, Ld, Dd, RL, S }

p0:  Z   -->  D S t
p1:  D   -->  d Ld
p2:  Ld  -->  Dd RL
p3:  Dd  -->  a b
p4:  RL  -->  &
p6:        |   v Ld
p4:  S   -->  s1 S
p7:        |   s2
```

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

a) Mostre que a frase '`d a b s1 s2 t`' pertence à linguagem, construindo a respectiva Árvore de Derivação.

b) Calcule o `lookahead()` das 8 produções.

c) Construa a Tabela de Parsing LL(1) que indica para cada símbolo NT e para cada símbolo T qual a produção a usar para continuar o reconhecimento (ou terminar com erro, se o terminal não for aceite nesse momento).

d) Escreva as funções de um parser RD-puro (recursivo-descendente) para reconhecer os Símbolos Ld, Dd e RL.

5 Gramáticas, Tradução e Parsing Bottom-Up (3v)

A gramática independente de contexto, GIC, abaixo escrita em BNF, define uma linguagem de domínio específico para descrição descrever os grupos de trabalho formados pelos alunos de uma turma.
O Símbolo Inicial é `turma`, os Símbolos Terminais são escritos só em maiúsculas (terminais-variáveis) ou entre apostrofes (palavras-reservadas ou sinais-de-pontuação), e a string nula é denotada por `&`; os restantes (sempre em minúsculas) serão os Símbolos Não-Terminais.

```
p0: turma : alunos grupos ;
p1: alunos: aluno
p2:       | alunos ';' aluno ;
p3: aluno : CODAL NOME ;
p4: grupos: &
p5:       | grupos grupo '.' ;
p6: grupo : 'Gr' NUM CODAL CODAL CODAL;
```

Neste contexto e após analisar esta GIC, responda às alíneas seguintes.

a) Após estender a GIC dada, construa o estado inicial do autómato LR(0) e os estados que dele saem.

b) Modifique a GIC anterior para permitir que um grupo tenha de 1 a 3 alunos.

c) Transforme a GIC dada numa **gramática tradutora**, GT, reconhecível pelo Yacc, para:

- calcular e imprimir o número de alunos declarados e o número de grupos definidos.
- verificar que todos os alunos que são incluídos em cada grupo existem e que um aluno não é alocado a mais de um grupo.