Processamento de Linguagens – LEI

Teste final

19 de Junho de 2015 (9h00)

Dispõe de **2:00 horas** para realizar este teste.

Questão 1: Expressões Regulares e Autómatos (4v = 1+1+1+1)

Responda às seguintes alíneas:

a) Escreva uma expressão regular para definir *endereços IP*: 192.136.20.2 , 202.23.3.254 , etc;

b) Escreva uma expressão regular para definir *endereços de email*: jcr@di.uminho.pt , leo.lopes87@gmail.com , etc;

c) Diga, **justificando** apropriadamente, se as expressões regulares abaixo, escritas em notação do Flex, são equivalentes:

```
((K|k)(E|e)(Y|y))+
[kKEeYy]+
```

d) Construa um autómato finito não-determinista para a primeira ER da alínea anterior (usando as regras formais que foram ensinadas) e construa informalmente um autómato determinista para a segunda ER.

Questão 2: Filtros de Texto em Flex e GAWK (4v = 1+1+1+1)

Especifique filtros de texto com base em expressões regulares e regras de produção (padrão - ação) para resolver as seguintes alíneas:

a) Especifique em Flex um filtro que dado um texto de entrada produz um texto de saída em que as datas no formato anglo-saxónico são convertidas para o formato ANSI ISO.
Exemplo: Se as seguintes datas fossem encontradas no meio do texto:

```
Apr 25, 1974
Jan 1, 2001
Jun 20, 2014
```

As mesmas teriam de ser convertidas para:

```
1974-04-25
2001-01-01
2014-06-20
```

O restante texto deverá ser copiado para a saída sem alterações.

b) Especifique em Flex um filtro que acumula os números inteiros encontrados num texto, podendo o acumulador ser a multiplicação ou a adição, conforme o utilizador decidir. Para isso, o filtro deverá ser sensível a três comandos que podem ser encontrados no texto de entrada por qualquer ordem e quantidade:

acc off Este comando desliga o "comportamento acumulador"do filtro; a partir daqui o filtro não deverá multiplicar/adicioanr mais números e deve imprimir o valor acumulado até ao momento, limpando então o acumulador; este é o comportamento por *omissão* (ou seja, o que está ativo no início do processamento.

mul on Este comando liga o "comportamento multiplicador"do filtro.

add on Este comando liga o "comportamento somador"do filtro.

c) Suponha que se pretende construir um índice remissivo (para inserir no fim de um livro) a partir do ficheiro que se mostra abaixo. Escreva um programa GAWK para gerar o dito índice.

```
pagina 1: grego, troiano, historia
pagina 2: camoes, historia, lusiadas
pagina 3: historia, poetas, camoes, pessoa
```

d) Analise o programa GAWK abaixo e explique com muita clareza e rigor o que dará como resultado quando aplicado a um texto.

```
BEGIN                { FS=";"; conta=0 }
NR==1                { print "A processar: " FILENAME }
NR>=10 && NR<=100    { cont[$1]++ }
                     { conta+=NF }
/ana.*sa|ana.*costa/ { print $1 " -> " $6 }
$1 ~ "rui" && $2!=40 { print $2 " =» " $NF }
END                  { print "....." conta/NR }
```

Questão 3: Desenho/especificação de uma Linguagem (4v)

Considere uma situação em que se pretende simular o comportamento de uma máquina de vender tabaco.
Para desenvolver tal simulador, deve ser criada uma linguagem que permita carregar os dados a processar conforme se descreve a seguir.
Primeiro é preciso definir o stock de tabaco (recorde-se que há várias marcas de tabaco, a preços diferentes e com valores de procura diferentes) carregado para a máquina no início do dia. Também é necessário registar a quantia de dinheiro (para trocos) carregada inicialmente. Depois de definir o estado de arranque, pretende-se indicar os registos das vendas ao longo do dia indicando a marca de cada maço escolhido e o dinheiro introduzido. Esta informação diária deve poder ser repetida para vários dias, indicando-se no início da descrição a respetiva data.

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).

Especifique em Flex um Analisador Léxico para reconhecer todos os símbolos terminais da sua linguagem e devolver os respetivos códigos.

1 Gramáticas, Parsing e Tradução (8v)

A gramática independente de contexto, *GIC*, abaixo escrita em *BNF*, define uma linguagem de domínio específico para descrição de transações (movimentos bancários) a serem processados em batch por um serviço automático de determinado banco.
O Símbolo Inicial é `Transacoes`, os Símbolos Terminais são escritos só em minúsculas (terminais-variáveis) ou só em maiúsculas (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:  Transacoes  -->  Cabec Movimentos ETASK
p1:  Cabec       -->  BTASK data
p2:  Movimentos  -->  Move '.'
p3:                |  Movimentos Move '.'
p4:  Move        -->  ContaDest ';' Sinal ';' Quant ';' Ordenante ';' Descr
p5:  ContaDest   -->  id
P6:  Sinal       -->  CREDITO
p7:                |  DEBITO
p8:  Quant       -->  num
p9:  Ordenante   -->  id
p10: Descr       -->  str
```

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

a) Calcule o `lookahead()` das 8 produções;

b) Construa a Tabela de Parsing LL(1) que indica para cada símbolo NT (9 linhas) e para cada símbolo T (10 colunas) qual a produção a usar para continuar o reconhecimento (ou terminar com erro, se o terminal não for aceite nesse momento);

c) Identifique as posições em que há conflitos LL(1) na tabela que construiu e indique como podia modificar a gramática para ultrapassar os conflitos;

d) Escreva a função de um parser RD-puro (recursivo-descendente) para reconhecer os Símbolos `Movimentos` e `Sinal`;

e) Após estender a GIC dada, construa o estado inicial do autómato LR(0) e os estados que dele saem (identifique esses estado e mostre ainda as ações a partir de cada um deles);

f) Escreva uma frase que pertence à linguagem e prove-o construindo a respetiva Árvore de Derivação;

g) Há 4 produções unitárias nesta GIC que poderiam ser eliminadas. Indique quais são e mostre como ficaria a gramática dada após a sua eliminação;

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

- calcular e imprimir: o número total de movimentos a débito e a crédito;
- garantir que o movimento se pode processar (conta destino existe e o saldo é suficiente para se efetuar o débito) — imagine para isso que existe uma BD com a informação das contas.