

Processamento de Linguagens e Compiladores — LCC (3ºano)

1º Teste

Data: 03 de Janeiro de 2019

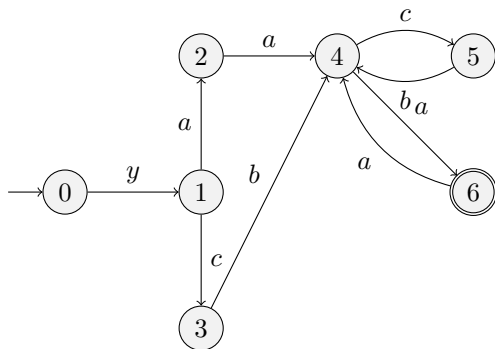
Hora: 09:00

Dispõe de 2 horas para realizar este exame

Questão 1: Expressões Regulares e Autômatos (4v)

Responda, então, às seguintes questões:

- a) Desenhe um autômato determinístico correspondente a: $(aab)^+ c (d \mid abf)^* j$
- b) Qual a expressão regular correspondente ao seguinte autômato:



- c) Sabendo que em \LaTeX um comando se escreve seguindo sempre o formato que se mostra nos dois exemplos a seguir

```
\textbf{bla bla bla}
\chapter{Inicio do Documento}
```

Escreva uma expressão regular '**expreg**' que usando o comando do Linux

```
grep -oP expreg
```

permita extrair de um texto '**.tex**' todas as linhas que tenham comandos \LaTeX .

- d) Considere com atenção a especificação FLex baseada em expressões regulares

```
%%
"<Bb>"      { acc1(yytext); }
"<" [Bb] ">" { acc2(yytext); }
\<\/?(B|b)\> { acc3(yytext); }
[<Bb>]      { acc4(yytext); }
%%
```

e diga se há um ou mais pares de ERs equivalentes, ou se são todas diferentes; justifique corretamente a sua resposta.

Considerando o texto abaixo, digam que ações semânticas seriam executadas e porque ordem

<HTML>

aquí vai um exemplo aparentemente simples mas bem complexo.

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

a) Especifique em FLex um normalizador de espaço branco: tabs, mudanças de linha e espaços. O texto de entrada deverá ser passado para a saída com as seguintes transformações:

- Caracteres brancos no início da linha são retirados;
- Caracteres brancos no fim da linha são reduzidos apenas à mudança de linha;
- Qualquer sequência de um ou mais caracteres brancos no meio de uma linha (tabs e espaços) é reduzida a um único espaço.
- Qualquer palavra após um sinal de pontuação deve começar com uma maiúscula.

b) Explique o que faz a script *GAWK* abaixo apresentada.

```
BEGIN {FS=";|:"}
$1 ~ /bd/ {alunos[$2]=$3; n++}
$1 ~ /aa/ {for (i=4; i<=NF; i++) { split($i,nuc,"="); s+=nuc[2] }
          notas[$2][$3]=s/(i-3.0)}
END {print "Alunos registados: " n;
     for(a in alunos)
       {print "Registo académico de " a " -- " alunos[a];
        for (ano in notas[a])
          { print ano " : " notas[a][ano] }
       }
     }
```

recorrendo ao exemplo de texto fonte que se segue.

```
bd:a11234:Ana Silva;cc1111;morada1;contactos1
aa:a11234:16-17:uc1=10;uc2=11;uc3=12
aa:a11234:17-18:uc4=14;uc5=16;
bd:a22234:Rui Sousa;cc2222;morada2;contactos2
aa:a22234:16-17:uc1=12;uc2=12;uc3=12;uc4=12
```

Questão 3: Desenho/especificação de uma Linguagem (4v=3+1)

Pretende-se descrever um leilão online em que está à venda um conjunto de peças (cada peça identificada por uma referência única, acrescida de uma descrição e fotografia). Esse conjunto está dividido por categorias: *mobiliário*; *louça*; e *pintura*. A cada peça descrita inicialmente é associado um valor base e uma lista de lances, sendo cada lance constituído por um valor de licitação e um código que representa o licitador.

- a) Escreva então, usando a notação do Yacc, uma Gramática Independente de Contexto, *GIC*, que especifique a Linguagem pretendida
- b) Especifique em Flex um Analisador Léxico para reconhecer todos os símbolos terminais da sua linguagem e devolver os respetivos códigos.

Questão 4: Gramáticas, Parsing e Tradução (8v)

Considere a seguinte gramática que define uma linguagem para descrever as prendas trocadas num jantar de Natal entre vários amigos (os símbolos terminais estão escritos em minúsculas ou entre apostrofes).

```
p01: Amigos → Cabec Ofertas

p02: Cabec   → &
p03:         | data local

p04: Ofertas → Amigo
p05:         | Ofertas ',' Amigo

p06: Amigo → id ':' Oferece ',' Recebe

p07: Oferece → id Descric

p08: Recebe → id Descric

p09: Descric → string
```

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

- a) Apresente uma frase desta linguagem e a respetiva árvore de derivação.
- b) Desenhe o autómato LR0.
- c) Diga porque é que esta gramática não é LL1 e escreva uma que, definindo exatamente a mesma linguagem, seja LL1.
- d) Usando a estratégia de parsing recursivo descendente, escreva as funções para reconhecer os símbolos **Amigos** e **Cabec**.
- e) Usando a notação do Yacc, estenda a GIC com ações semânticas para calcular o número de amigos distintos que estava no jantar.
- f) Usando a notação do Yacc, estenda a GIC com ações semânticas para verificar que nenhum amigo oferece nem recebe de si próprio
- g) Usando a notação do Yacc, estenda a GIC com ações semânticas para garantir que se A oferece a B, então B recebe de A.
- h) Se quiser implementar um parser Top-Down iterativo e table-driven, diga qual o conteúdo da stack de parsing LL(1) antes de iniciar o ciclo de reconhecimento.