

Processamento de Linguagens e Compiladores

LCC (2º ano)

1º Exame

Data: 26 de Junho de 2009
Hora: 09:30

Dispõe de 3 horas para realizar este exame

1 Filtros de Texto e Analisadores Léxicos (5v)

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

- a) Um filtro que receba um texto qualquer começado por uma lista de palavras — iniciada por '<', terminada por '>', uma palavra por linha, cada palavra formada por letras, podendo conter pelo meio caracteres especiais como '.', '@', '-', '_' — e retire do corpo do texto (que aparece após essa lista) todas as ocorrências dessas palavras. Por exemplo, se entra

```
<
Pedro
prh@di.uminho.pt
>
O email do Pedro é prh@di.uminho.pt mas ele nunca o usa.
```

irá sair

```
O email do é mas ele nunca o usa.
```

- b) Um validador que leia um texto formado por mensagens sms, uma por linha, e copie para a saída cada mensagem que não cumpra as regras de escrita descritas a seguir. De acordo com o código criptográfico definido por dois amigos, cada mensagem será formada segundo uma das regras:

- Os caracteres de pontuação { '.', ';', ':', '!', '?' } podem aparecer mas sempre isolados em 1 mensagem.
- Sequências de caracteres, excepto letras, são possíveis mas só se a mensagem começar com um *backslash* '\'.
- Sequências de uma ou mais letras podem aparecer mas sempre entre chavetas '{' e '}'.
- Expressões aritméticas em Post-fix também são permitidas. Serão formadas por números inteiros (com um ou mais dígitos) e pelos tradicionais operadores aritméticos { '+', '-', '*', '/' }; operandos (números inteiros) separam-se entre si por vírgulas ','; operadores aparecem sempre depois dos seus operandos. Note que assim uma mensagem terá de começar sempre por, no mínimo, dois números e terminar sempre num operador (exemplo: 32,1+45,2-*).

2 Expressões Regulares e Autómatos (5v)

Considere as seguintes ERs:

$$\begin{aligned}e1 &= (a + b)^* c \\e2 &= (a c + b c)^* \\e3 &= b c + c (b c)^*\end{aligned}$$

e os seguintes Autómatos:

```
AD1 :: < {'a', 'b', 'c'}, {1,2,3,4,5}, 1, {3,5}, delta >
  sendo delta 1 'b' = 2
         delta 1 'c' = 4
         delta 2 'c' = 3
         delta 3 'a' = 3
         delta 4 'b' = 5
         delta 5 'a' = 5

AND2 :: < {'a', 'b', 'c'}, {1,2,3,4,5,6,7}, 1, {4,7}, delta >
  sendo delta 1 & = 2      delta 3 'c' = 4      delta 7 & = 5
         delta 1 & = 5      delta 5 'b' = 6
         delta 2 'a' = 3      delta 5 & = 7
         delta 2 & = 3      delta 6 'c' = 7
```

Responda, então, às seguintes questões:

- explique por palavras suas porque é que $e1$ e $e2$ não são equivalentes.
- 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 $e2$.
- usando a respectiva *cadeia de derivação*, mostre que a frase "acbc" pertence à linguagem gerada por $e2$.
- construa, aplicando as regras sistemáticas, um autómato não-determinista equivalente a $e3$.
- escreva a expressão regular que define a linguagem aceite pelo autómato AD1.
- diga justificando se a frase "cbba" pertence à linguagem gerada por AD1.
- considerando o autómato AND2, calcule o ε fecho do estado 1.

3 Desenho/especificação de uma Linguagem (3v)

Black é uma jovem empresa portuguesa que desenha, produz e comercializa tudo pra Góticos, desde vestimentas, meias e sapatos até adereços e tatuagens.

A Black quer ter um *sítio WWW*, como agora é moda, mas que seja alimentado facilmente por eles (cada época que a colecção é alterada) sem contudo saberem nada de HTML, nem terem eles de montar o *site*. Para isso pretende-se definir uma nova Linguagem que permita à Black descrever a sua colecção. Além da data da última actualização, a linguagem deverá permitir descrever acessórios e vestimentas.

No caso de se tratar de um acessório, além da referência e descrição, deverá ser possível incluir o url para uma ou mais imagens (opcional), o tipo do acessório (bracelete, colar, bolsa, anel, piercing); o seu preço e o material de que é feito (pele, cabedal, tecido, metal).

No caso de se tratar de vestimenta, deverá ser possível diferenciar entre *Para Ele* e *Para Ela*. Além da referência e descrição da vestimenta, deverá ser incluso o url para uma ou mais imagens, o tipo da vestimenta (vestido, calça, tshirt), o preço e os tamanhos disponíveis (XS, S, M, L, XL).

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 (7v)

A gramática independente de contexto, *GIC*, abaixo apresentada, define uma linguagem específica para descrever os livros e CDs disponíveis numa biblioteca e os estados que lhe são associados (livres ou emprestados), à semelhança do que acontece nas bibliotecas da UM.

O Símbolo Inicial é *Biblioteca*, os Símbolos Terminais são escritos em maiúsculas (pseudo-terminais) ou em maiúscula entre apostrofes (palavras-reservadas e sinais de pontuação), e a string nula é denotada por *&*; os restantes (sempre em minúsculas) serão os Símbolos Não-Terminais.

```
p0: Biblioteca --> Registos
p1: Registos   --> Registo
p2:           | Registos ',' Registo
p3: Registo    --> '[' REGISTO Descricao EXISTENCIAS Existencias ']'
p4: Descricao  --> Referencia Tipo Titulo '(' Autor ')' Editora Ano Catalogo
p5: Referencia --> id
p6: Tipo       --> LIVRO
p7:           | CDROM
p8:           | OUTRO
p9: Titulo     --> string
p10: Autor     --> string
p11: Editora   --> string
p12: Ano       --> num
p13: Catalogo --> BGUM
p14:           | ALFA
p15:           | OUTRO
p16: Existencias --> LOCAL Local '(' Estados ')'
p18: Local      --> string
p19: Estados    --> Estado
p20:           | Estados ',' Estado
p21: Estado     --> CodigoBarras Disponib
p22: CodigoBarras --> id
p23: Disponib   --> Estante
p24:           | PERMANENTE
p25:           | EMPRESTADO DataDev
p26: DataDev    --> Ano '-' Mes '-' Dia
p27: Mes        --> num
p28: Dia        --> num
```

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

- escreva uma Frase válida da linguagem gerada pela *GIC* dada, mostrando a respectiva Árvore de Derivação.
- altere a gramática de modo a permitir que cada livro tenha mais de um Autor.
- 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.
- escreva as funções de um parser RD-puro (recursivo-descendente) para reconhecer o Símbolo *Estado* e seus derivados.
- construa o estado inicial do autómato LR(0) pra gramática dada e os estados que dele saiem.
- transforme a *GIC* dada numa **gramática tradutora**, *GT*, reconhecível pelo *yacc*, para:
 - calcular e imprimir: o número de registos; e o número de livros existentes para cada registo.
 - identificar e listar por ordem alfabética os títulos dos livros.
 - verificar se não existem registos com a mesma referência.