

Processamento de Linguagens – MiEI

Exame de Recurso

19 de Junho de 2019 (13h00)

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

Questão 1: Expressões Regulares e Autómatos (3v)

Responda às seguintes alíneas:

a) Considere a seguinte gramática:

```
S -> b A a
    | c d B d
A -> b A
    | &
B -> a B d
    | e
```

e depois de a analisar com muito cuidado desenhe o autómato determinista que define a mesma linguagem. Por fim construa a ER equivalente. No caso de não ser possível uma equivalência total, indique-o e desenhe o autómato e a ER mais aproximada possível.

b) Considere a ER abaixo, escrita em notação do Flex:

```
[+\-]?[012][0-9]{2} ". "
```

Mostre, usando uma cadeia de derivação, uma frase válida que essa ER *apanha* e transforme a expressão num autómato não-determinista equivalente aplicando as regras de transformação formais.

c) O comando `sed`, disponível em Linux, permite (entre outras coisas) fazer substituições de uma expressão regular, `expReg`, por uma `string` em todas as ocorrências desse padrão num dado ficheiro de texto `file`, sendo usado do seguinte modo

```
sed -re 's/expReg/string/g' file
```

Construa, 3 comandos `sed` que, usando uma única substituição:

- substitua todas as imagens HTML (``) existentes em `file.html` pelo poster `Bauhaus.jpg`¹.
- remova os comentários Awk (`#` até ao fim de linha), incluídos em `file`.
- normalize os espaços junto às vírgulas no texto contido em `file` (p.ex. `'a ,b'` deve ficar `'a, b'`).

Questão 2: Filtros de Texto em Flex e Gawk (6v = 3+3)

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

¹Bauhaus 1919 - cujo centenário se celebra em 2019

- a) Em língua natural uma lista é normalmente enunciada por elementos separados por vírgulas e pela conjunção 'e' junto ao último elemento

$$x_1, x_2, x_3, \dots, x_m \text{ e } x_n$$

Faça um filtro Flex para extrair as lista homogêneas monopalavra de um texto, bem como o número da linha em que começam. Aplicada ao seguinte texto:

Em Janeiro e Fevereiro estará aberto uma exposição contendo
alguns dos mais brilhantes desenhos, pinturas,
textos e esculturas de Leonardo d'Avinci.

a saída pretendida é

```
linha1 : Janeiro :: Fevereiro
linha2 : desenhos :: pinturas :: textos :: esculturas
```

- b) Considere o seguinte ficheiro de texto contendo uma gramática:

```
axioma : frase

frase : nominal predicado .

nominal : pronome
         | artigo nome

frase_adjectiva : adjectivo
                 |

# verbo com ou sem complemento directo
predicado : verbo
           | verbo nominal

artigo : o
        | um
```

Considere ainda que:

- o axioma chama-se sempre "axioma"
- todas as linhas não vazias têm um ':' ou um '|', ou são comentários (linhas começadas por '#')
- existe sempre um ou mais espaços a separar os elementos.

Escreva uma ou mais scripts em GAwk para:

- b1) Calcular os símbolos terminais da gramática (aparecem no lado direito mas não no lado esquerdo).
b2) Calcular os símbolos não terminais não alcançáveis (não referidos em nenhum lado direito).

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

Para ensinar às crianças o pensamento computacional que está na base da programação pretende-se criar um interpretador simples que permita descrever inicialmente uma base de conhecimento (BC) na forma de um conjunto de triplos <sujeito, predicado, objeto> e depois permita fazer uma ou mais perguntas sobre essa BC. As perguntas podem ser de vários tipos de modo a permitir saber todas as relações com um dado **sujeito** ou um dado **predicado**, ou então saber todos os pares <sujeito, objeto> relacionados através de um dado **predicado**. Os resultados das perguntas devem poder ser imprimidos de imediato ou ficar guardados em variáveis (que à moda do AWK podem ser usadas sem ser declaradas)

Neste contexto, responda às alíneas seguintes:

- a) Imagine uma linguagem simples e intuitiva que possa ser usada por uma criança para fazer a programação da BC e respetivas perguntas e, então, escreva uma GIC (Gramática Independente do Contexto) que defina a linguagem que conceber.
- b) Usando a notação do Flex especifique um analisador léxico para a linguagem em causa.

Questão 4: Gramáticas, Parsing e Tradução (7v=1+1+1+1+3)

Considere a Gramática Independente De Contexto, G , abaixo apresentada, que define uma linguagem para comandar um robot.

Note ainda que as palavras reservadas estão escritas em letras maiúsculas e os restantes símbolos terminais (sinais e classes terminais) estão definidos abaixo (conjunto T). Os símbolos não-terminais estão no lado esquerdo das produções do conjunto P , sendo PrgBot o seu axioma (ou símbolo inicial).

```
T = { '{', '}', '=', ';', '(', ')', num, ... }
P = {
p1: PrgBot  -> Dcl '{' Cmds '}'
p2: Dcl     -> PASSO '=' num
p3:         | &
p4: Cmds    -> Cmd RCmds
p5: RCmds   -> ';' Cmds
p6:         | &
p7: Cmds    -> AV Dest
p8:         | REC num
p9:         | ESQ num
p10:        | DIR num
p11:        | SE Cond '(' Cmds ')'
p12: Cond   -> TOCA-OBJ
p13:        | VE-LUZ
p14: Dest   -> FOREVER
p15:        | num
}
```

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

- Escreva uma *frase válida* da linguagem gerada do G , apresentando a respetiva *árvore de derivação*.
- Calcule o *Lookahead(1)* de cada produção e diga justificando se há **Conflitos LL(1)**.
- Após estender a G dada, construa completamente o *estado inicial* do respetivo *Autómato LR(0)* e os *estados que dele imediatamente derivam*.
Diga, justificando, se em algum desses estados ocorrem de **Conflitos shift-reduce**.
- Escreva as funções de um parser RD-puro (recursivo-descendente) para reconhecer qualquer símbolo terminal e os 2 símbolos não-terminais Dcl e Cmds .
- Transforme a GIC dada numa GT (Gramática Tradutora) desenvolvendo as ações semânticas necessárias para:
 - contar o número de instruções condicionais e o comprimento do bloco de comandos de cada uma.
 - otimizar o programa de modo a somar os passos, ou os graus, se aparecerem instruções de movimento, ou de rotação, do mesmo tipo seguidas.Por exemplo o fragmento de programa

AV 10; AV 5; AV 3; ESQ 20; DIR 10; DIR 10;

deve ser otimizado para

AV 18; ESQ 20; DIR 20;