



# Linguagens Formais e Autómatos

(Ano lectivo de 2013/14)

## Guiões das aulas práticas

Guião #02

Expressões regulares

### Sumário

Resolução de exercícios sobre expressões regulares. Sua utilização em libreoffice, egrep, sed e C.

Sobre o alfabeto  $A = \{a, b, c\}$ , considere as seguintes linguagens

$$L_1 = \{u \in A^* : \exists_i x_i = x_{i-1} = x_{i-2}\}$$

$$L_2 = \{u \in A^* : \#(ab, u) > 1\}$$

$$L_3 = \{u \in A^* : \#(aba, u) > 1\}$$

$$L_4 = \{u \in A^* : \#(b, u) = 0 \vee \#(a, u) \% 2 = 0\}$$

$$L_5 = \{u \in A^* : (i < j \wedge u_i = u_j = b \wedge \forall_{i < k < j} u_k \neq b) \implies (\forall_{i < n < j} \forall_{i < m < j} u_n = u_m)\}$$

$$L_6 = \{u \in A^* : \#(ab, u) = 1\}$$

$$L_7 = \{u \in A^* : \#(aba, u) < 3\}$$

$$L_8 = \{u \in A^* : \#(abc, u) \geq (\#(ca, u) + \#(cb, u))\}$$

Considere ainda o ficheiro `abc.txt` que possui palavras definidas sobre o alfabeto  $A$ .

**Exercício 1** Para cada uma das linguagens  $L_1$  a  $L_8$ , determine uma expressão regular que a representa.

**Exercício 2** O Find/Replace do LibreOffice/Writer possui suporte para expressões regulares. Use esse suporte para identificar as palavras do ficheiro `abc.txt` que pertencem a cada uma das linguagens  $L_1$  a  $L_8$ .

**Exercício 3** O `egrep` é um comando que permite seleccionar as linhas de um ficheiro dado que possuem uma sub-palavra que satisfaz uma determinada expressão regular. É executado na forma

```
egrep 'exp-reg' fich
```

Por exemplo, o comando

```
egrep 'a[bc]' fich
```

selecciona as linhas do ficheiro `fich` que possuem pelo menos um `a` seguido de um `b` ou `c`.

Use o `egrep` para seleccionar as linhas do ficheiro `abc.txt` que pertencem a cada uma das linguagens  $L_1$  a  $L_8$ .

**Exercício 4** O `sed` é um comando unix que permite executar ações de edição sobre as linhas de um ficheiro dado. Suporta expressões regulares. Executado na forma

```
sed -r 's/exp1/exp2/' fich
```

permite procurar nas linhas do ficheiro `fich` a primeira sub-palavra que satisfaz a expressão regular `exp1` e substituí-la pela expressão `exp2`. Se executado na forma

```
sed -r 's/exp1/exp2/g' fich
```

substitui todas as ocorrências de sub-palavras que satisfazem a expressão regular `exp1`. `exp2` pode representar uma simples sequência de caracteres ou representar uma palavra que inclui na sua definição parte ou a totalidade da sequência captada por `exp1`. Por exemplo,

```
sed -r 's/a[bc]/_/' fich
```

substitui a primeira ocorrência de `ab` ou `ac` por `_`. Experimente. Num outro exemplo,

```
sed -r 's/a[bc]/"/g' fich
```

coloca todas as ocorrências de um `a` seguido de um `b` ou de um `c` entre aspas. Finalmente, o exemplo

```
sed -r 's/a\([bc]\)/_\1/g' fich
```

transforma todos os `a` que têm um `b` ou `c` à sua direita em `_`.

Use o `sed` para realçar as linhas do ficheiro `abc.txt` que pertencem a cada uma das linguagens  $L_1$  a  $L_8$ .

**Exercício 5** A biblioteca `regex` do C permite a manipulação de expressões regulares. Leia o seu manual (`man regex`). Construa um programa em C que, usando a biblioteca `regex`, realce as linhas do ficheiro `abc.txt` que pertencem a cada uma das linguagens  $L_1$  a  $L_8$ .