

# AULAS TEÓRICO-PRÁTICAS DE COMPILADORES

2º semestre de 2002/2003

---

## AULA Nº 4 (3 horas)

Exercício sobre expressões regulares e autómatos finitos.

### 1 Exercícios com Expressões Regulares

1.1 Escreva expressões regulares para os seguintes exemplos:

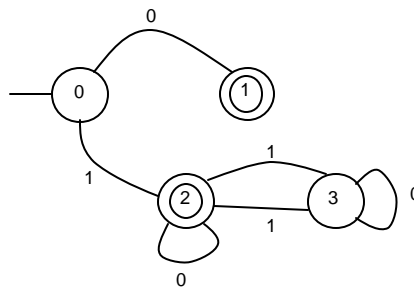
- (a) números binários;
- (b) URLs da forma: `http://www.ualg.pt` (em que o campo intermédio “ualg” é o único campo variável).
- (c) IPs da forma: `140.192.33.37` (considere que todos os IPs têm o mesmo número de dígitos por cada campo).
- (d) números binários que representam inteiros sem zeros supérfluos;
- (e) Strings sobre o alfabeto  $\{a, b, c\}$  com número ímpar de a's;
- (f) Strings sobre o alfabeto  $\{a, b, c\}$  em que o primeiro “a” precede o primeiro “b”;
- (g) Números binários múltiplos de 4;
- (h) Números binários maiores do que 101001;
- (i) [TPC] A linguagem das constantes em vírgula flutuante (notação utilizada em Java).

1.2 Acha que é possível escrever expressões regulares para os seguintes exemplos? No caso de ser possível apresente uma solução:

- (j) números binários que começam e acabam com o mesmo dígito;
- (k) seqüências de algarismos que formam capicuas;

### 2 Exercícios com Autómatos Finitos

2.1 Considere o autómato finito seguinte:



- (a) O autómato é determinista ou não determinista?
- (b) Qual é o seu estado de início? Quais são os estados de aceitação (finais)?

- (c) Este autômato aceita a sequência 110100? Qual a sequência de estados visitados no reconhecimento desta String?
- (d) Qual é a String mais pequena que o autômato aceita?
- (e) Pode indicar a String maior que o autômato aceita?
- (f) Por palavras, qual é a linguagem que o autômato aceita?

2.2 Desenhe os NFAs para as expressões regulares seguintes. Depois converta cada um deles para DFA.

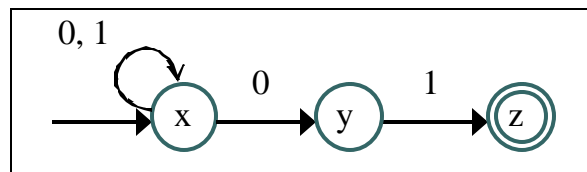
(a)  $[01]$

(b)  $1^+$

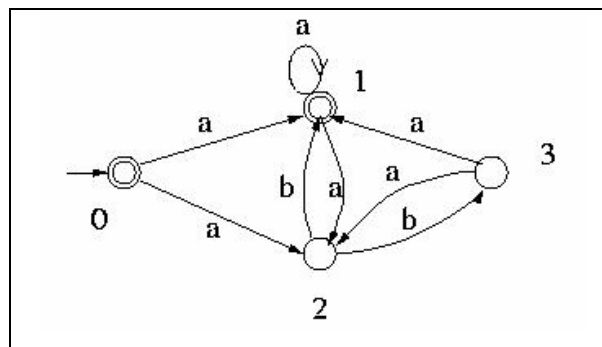
(c)  $0 \mid 1 [0 \mid 1]^+$

2.3 Converta os NFAs seguintes para DFAs:

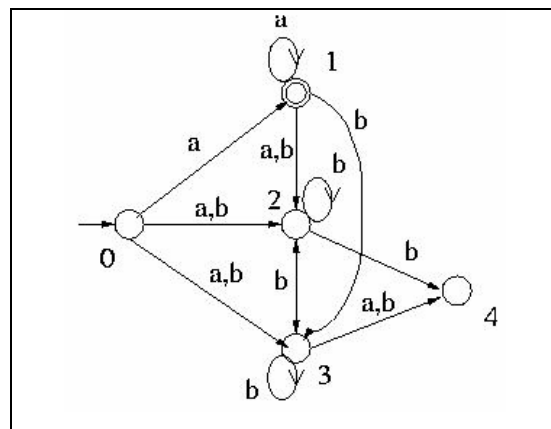
(a)



(b)



(c)



2.4 Um analisador lexical baseado num interpretador de um DFA utiliza duas tabelas:

**edges**: indexada pelo número do estado e símbolo de entrada, retorna o número do estado, e

**final**: indexada pelo número do estado, retorna 0 ou um número representativo da acção a realizar.

Considerando a seguinte especificação:

$(aba)^+$  ? acção número 1

$a(b^*)a$  ? acção número 2

$(a \mid b)$  ? acção número 3

Apresente as tabelas **edge** e **final** para o analisador lexical.