Nome: ____nº _______________

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

2015-2016 **TEORIA DA COMPUTAÇÃO** 11/Nov./2015 17h

Duração: 120m· 1ª Frequência

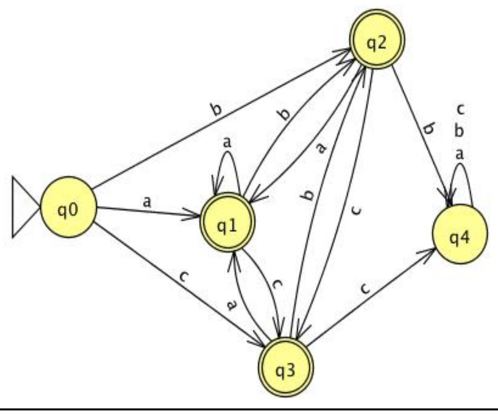
Leia atentamente:

1º- A prova é **sem** consulta.
2º- Responda na folha do enunciado.
3º- Não responda à sorte: respostas (de escolha) erradas têm pontuação negativa; respostas em branco têm pontuação nula.
4º- Para responder só pode utilizar os espaços do enunciado. Seja conciso e diga só o essencial. Quando a resposta for de escolha, assinale com X a que julgar certa.
5º- Coloque o nome e o nº de estudante em **todas** as folhas da prova.

1. Escreva uma gramática para a linguagem no alfabeto $\Sigma = \{0, 1\}$ que inclui todas as cadeias em que se verifica uma das duas situações seguintes: (a) tem pelo menos um *1* e não tem nenhum '0'; (b) cada subcadeia de *0s* consecutivos é seguida de uma subcadeia de *1s* com comprimento igual ou maior (incluindo comprimento zero).

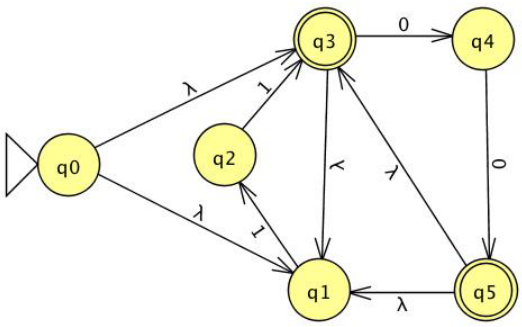
$S \rightarrow A \mid B$
$A \rightarrow 1A \mid 1$
$B \rightarrow \lambda \mid 1B \mid CB$
$C \rightarrow \lambda \mid 0C1$

2. Desenhe um autómato finito determinístico, com o menor número de estados, que aceite a linguagem no alfabeto $\Sigma = \{a, b, c\}$ composta por todas as cadeias **não vazias**, em que nunca ocorrem nem dois '*b*' consecutivos nem dois '*c*' consecutivos. Por exemplo, as cadeias '*cbba*' , '*accab*' , '*abccabbca*', não devem ser aceites.



Nome: ____nº _______________

3. Seja o seguinte autómato finito não determinístico no alfabeto Σ ={0, 1}.

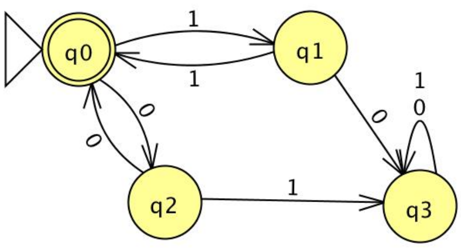


a) Converta-o num autómato finito determinístico com o menor número de estados possível. Desenhe previamente a respetiva tabela de transições.

Tabela de transições

	0	**1**
q013	q4	q2
q4	q135	-
q2	-	q31
q135	q4	q2
q13	q4	q2
-	-	-

Grafo:

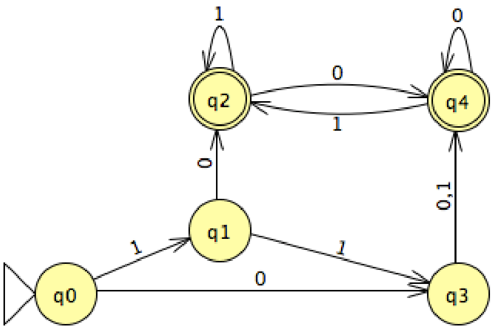


b) Descreva a linguagem por ele aceite

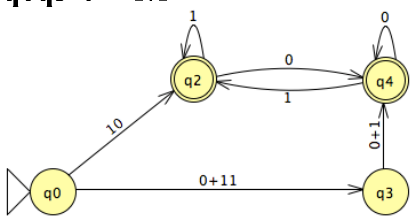
Todas as cadeias em que as corridas de 0s e 1s têm comprimento par, incluindo a cadeia vazia.

4. Com base no seguinte autómato e, com o alfabeto Σ = {0, 1}, construa a sua expressão regular apresentando todos os passos do algoritmo de eliminação de estados abordado nas aulas.

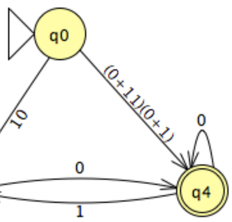
Nome: ____ nº _______________



Eliminando q1:
q0q2 –∅ + 1.0
q0q3-0 + 1.1



Eliminado q3:
q0q4- ∅ + (0+11).(0+1)



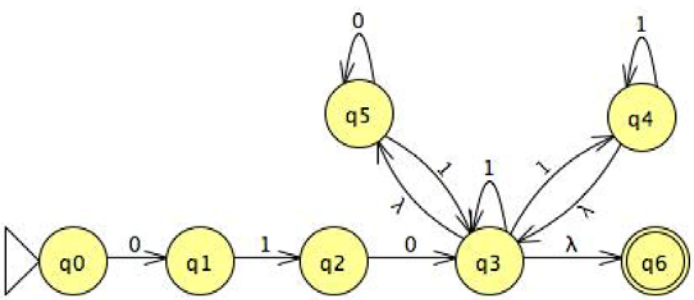
Eliminando q2:
q0q4- (0+11)(0+1) + 101*0
q4q4- 0+11*0
ER: ((0+11)(0+1)+101*0)(0+11*0)*
Eliminando q4:
q0q2- 10 + ((0+11)(0+1)0*1)
q2q2- 1*+00*1

ER: (10 + ((0+11)(0+1)0*1))(1*+00*1)*

ER final: (10 + ((0+11)(0+1)0*1))(1*+00*1)* + ((0+11)(0+1)+101*0)(0+11*0)*

5. Considere a seguinte Expressão Regular 010 (11*+0*1+1) * no alfabeto Σ = {0, 1}.

a) Converta-a no seu autómato finito não determinístico.



Nome: ____nº ______________

b) Converta o autómato que obteve na alínea anterior na sua Gramática Regular utilizando o menor número de produções possível.

S->010 A
A->1A | 0BA | λ
B->0B | 1

6. Classifique as seguintes afirmações como verdadeiras ou falsas:

(i) Qualquer gramática linear à esquerda é regular. V [X] F [ ]

Dê um exemplo de uma tal gramática (completa) com três produções:

$S \rightarrow Aab$
$A \rightarrow Aab$
$A \rightarrow \lambda$

(ii) Uma linguagem gerada por qualquer gramática linear é regular. V [ ] F [X]

(iii) A gramática dada a seguir é regular. V [ ] F [F]

Gramática:

$S \rightarrow Aab$
$A \rightarrow BAb \mid \lambda$
$B \rightarrow aA \mid \lambda$

Justifique:

Por duas razões:
-Tem uma produção linear à esquerda e outra linear à direita
- Tem uma produção não linear

7. Sendo L, M, N expressões regulares, diga se as seguintes igualdades são verdadeiras ou falsas:

a) (L + M) N*= LN* + MN* Verdadeira: x☐ / Falsa: ☐
b) (NLN + NMN) = N(L+ M)N Verdadeira: x☐ / Falsa: ☐
c) (L*M*)* = (M + L)* + L* +M* Verdadeira: X☐ / Falsa: ☐
Justifique c) :

(M + L)* =(L*M*)* (propriedade demonstrável do fecho estrela). L* pode ser gerada por (L*λ) e M* por (λM*). Logo L* e M* estão contidas em (L*M*)*, ou seja, estão contidas em (M+L)*, donde (M+L)*+M*+L*=(M+L)*=(L*M*)*, q.e.d.

Nome: ____nº ______________

8. Sendo A, B e C linguagens regulares, e L complemento de L, as linguagens seguintes são regulares:

(i) A-B Verdadeiro: x☐ / Falso: ☐

Justifique (i)
A família das linguagens regulares é fechada em relação à diferença (e por isso A-B é regular) e em relação à complementação (e por isso o complemento da diferença é regular)

Justifique (ii)

A família das linguagens regulares é fechada em relação à complementação, à interseção, e á união.

9. Prove pelo lema da bombagem que a linguagem $L = \{ ba^pba^pb^{2p}ab, p>0 \}$ não é regular.

Para qualquer *m* que me apresentes considere-se a cadeia

$$ba^mba^mb^{2m}ab$$

Ela é de comprimento $4m+4$, e portanto maior do que m, e pertence à linguagem. Nela poderemos contradizer o lema; se o conseguirmos provamos que a linguagem é não-regular. **A identificação da cadeia na qual vamos trabalhar é importante na demonstração**. Note-se que ao escolhermos esta cadeia **assegura-se que os primeiros *m* caracteres serão necessáriamente *b* seguido de *m-1 a*'s**. A decomposição $xy<=m$, como tem que estar no princípio da cadeia , é composta apenas por $b+a$´s e portanto y será sempre composta ou por a's ou por a's precedidos por b ; y pode ser a, ou a^2, ou a^3, ... ou a^{m-1}, ou ba^{m-1}. A bombagem só produz a's ou então a's precedidos de b.. Por exemplo se $y = a$, x= ba^{m-2}, y^5 =*aaaaa* vai produzir uma cadeia com $m+4$ a's, mantendo-se os m a's no meio da cadeia e os $2m$ b's na parte final. Portanto quebrando a regra das cadeias. Bombeou-se assim para fora da linguagem. Se y contém a's precedido de b, então a bombagem de y para além de desequilibrar o número de a's introduz b's em posições erradas, produzindo cadeias cuja estrutura não obedece à linguagem. Qualquer bombagem de y, excepto y^1, resulta numa cadeia que não pertence a L (mas bastava até que só acontecesse para um caso).. Não existe a tal decomposição *xyz*.
Como tudo isto acontece para quaquer m, não existe nenhuma m que satisfaça o lema da bombagem, e por isso a linguagem não é regular.

10. Dada uma linguagem regular num certo alfabeto, existe algum algoritmo para decidir se a uma qualquer cadeia nesse alfabeto pertence à linguagem ? Se sim qual ?

Existe: dada a linguagem regular L, constrói-se o DFA M(L). Dada qualquer cadeia dá-se a M(L). Se o autómato ler a cadeia e terminar num estado aceitador, a cadeia pertence a L; se terminar num estado não aceitador, a cadeia não pertence a L.