

Teoria da Computação

Nome: \_\_\_\_\_

Número: \_\_\_\_\_

Segundo Semestre 2017/2018

Exame Época Especial

27/07/2018

Duração: 180 Minutos

Classificar (Sim/Não) \_\_\_\_\_

---

## Atenção:

Este enunciado tem 9 páginas (incluindo esta) e 9 questões.

Não é permitido o uso de qualquer tipo de material auxiliar ou electrónico enquanto estiver na sala em que decorre a prova. Não é permitida a divulgação deste enunciado antes do final da prova.

Apenas volte a página quando o professor assim o disser. Responda no enunciado. Pode usar o verso das folhas como rascunho, se necessitar.

Todos os alunos devem entregar o enunciado e assinar a folha de presenças para ter o seu exame classificado.

Justifique cuidadosamente todas as respostas (respostas sem justificação não serão consideradas).

Grupo	Valor	Pontuação
I	6	
II	7	
III	7	
Total	20	

---

## Grupo I (6 Valores)

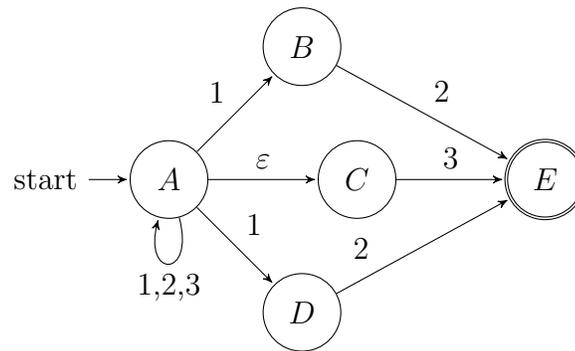
1. Considere a função **seq** que dado um número natural  $n$  devolve o conjunto com todos os naturais até  $n$ .
  - i) Defina indutivamente a função **seq**.
  - ii) Apresente uma derivação que prove que  $\mathbf{seq}(3) = \{0, 1, 2, 3\}$ .
  - iii) Considere a união dos conjuntos  $\mathbf{seq}(n)$  para todo o natural  $n$ . Explique se o conjunto é contável ou infinito não contável.

2. Uma loja online permite que os clientes registados definam listas de produtos. Cada cliente é identificado por um nome único e pode ter um número arbitrário de listas. Uma lista tem um nome (único, nas listas do cliente a que pertence) e um conjunto de nomes de produtos. Inicialmente, o conjunto das listas do novo cliente é vazio.
- i) Modele a aplicação como uma estrutura, em que cada elemento é um tuplo com o nome do cliente e o seu conjunto de listas. Considere que o nome do cliente, de cada lista e dos produtos são strings.
  - ii) Defina a função que verifica se um cliente, identificado pelo seu nome, existe no sistema.
  - iii) Defina a função (total ou parcial) que adiciona um cliente ao sistema.
  - iv) Defina a função (total ou parcial) que dado um nome de cliente devolve o conjunto das suas listas.
  - v) Defina a função (total ou parcial) que dado um nome de cliente e um nome de uma lista devolve o número de produtos na lista (0, se não a lista não existir).
  - vi) Defina a função (total ou parcial) que dado um nome de cliente, um nome de uma lista e um produto, adiciona o produto à lista respectiva do cliente. Se a lista não existe, cria-a.

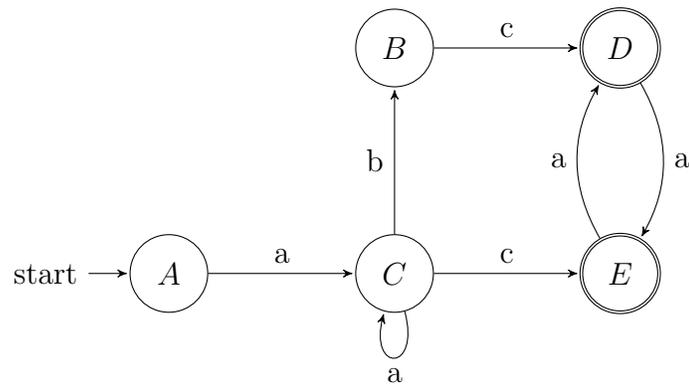
**Grupo II (7 Valores)**

1. Considere a expressão regular  $(ac)^*(b+d)^*ab^*c$ .
  - i) Verifique se a palavra  $dac$  pertence à sua linguagem.
  - ii) Converta a expressão num Autómato Finito Não determinista (AFN), apresentando todos os passos do processo, usando o algoritmo dado nas aulas.

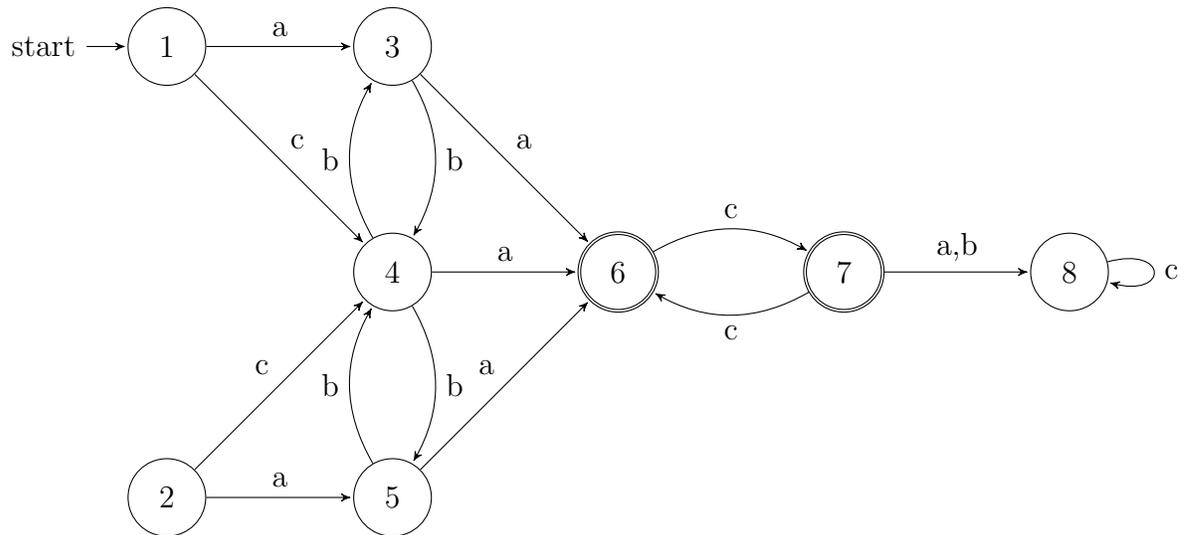
2. Converta o AFN seguinte num Autómato Finito Determinista (AFD), apresentando todos os passos do processo e usando o algoritmo dado nas aulas.



3. Converta o seguinte AFD numa expressão regular, apresentando todos os passos do processo e usando o algoritmo dado nas aulas.



4. Minimize o AFD seguinte, usando um algoritmo dado nas aulas.



5. Verifique se o AFD dado na alínea anterior aceita a palavra *abab*.

## Grupo III (7 Valores)

1. Considere a gramática  $G = \langle V, T, P, S \rangle$  em que

- $V = \{S, R\}$
- $T = \{\text{begin}, \text{open}, \text{close}\}$
- $P = \{S \rightarrow \text{open } R \text{ close}, S \rightarrow \text{begin } S, S \rightarrow \epsilon, R \rightarrow \text{open } R \text{ close}, R \rightarrow \epsilon\}$

- i) Apresente uma derivação *leftmost* para a palavra **begin open open close close**.
- ii) A linguagem gerada por  $S$  é regular? Justifique usando o Lema da Bombagem.
- iii) Verifique se a gramática é LL(1).
- iv) Construa a tabela de transição.
- v) Verifique, usando a tabela de transição, se é reconhecida a palavra **begin begin open open close**.

2. Defina a função de transição, o estado inicial e os finais de uma máquina de Turing que verifica se dada palavra sobre  $\{a, b\}$ , colocada na posição de memória 1 na forma de lista, só tem ocorrências do carácter 'a' ou só tem ocorrências do carácter 'b', escrevendo o resultado (`true` ou `false`) na posição de memória 2.

Note que a palavra *abba*, por exemplo, como lista escreve-se  $(a, (b, (b, (a, \text{null}))))$ .