

Teste 2 A da UC de Teoria da Computação (MIEI 2015/2016, FCT UNL)

4 de Junho de 2016

Informações e instruções

1. Justifique cuidadosamente a resposta a cada pergunta.
2. O teste tem a duração de 2h. Os alunos podem sair ao fim de 1h.
3. Todos os alunos devem assinar a folha de presenças e entregar este caderno antes de sair da sala. Quem não quiser o teste avaliado (quiser “desistir”) deve escrever nesta página DESISTO e assinar por baixo.
4. Responda a cada grupo na folha respectiva. Solicite a um docente na sala folhas de rascunho, se precisar.
5. O teste é sem consulta e deve ser respondido a caneta. A prova deve ser realizada em respeito pelo código de ética do DI.

Identificação

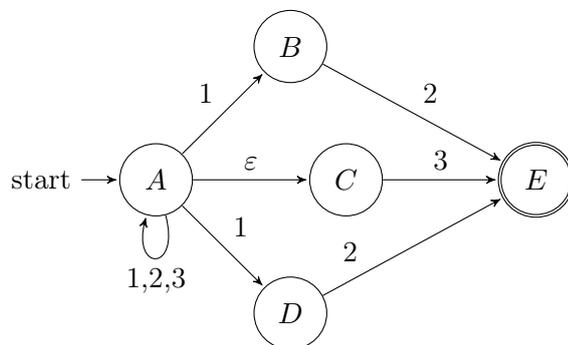
1. NOME:
2. NÚMERO:
3. TOTAL DE FOLHAS ENTREGUES (INCLUINDO ESTA):

Grupo I

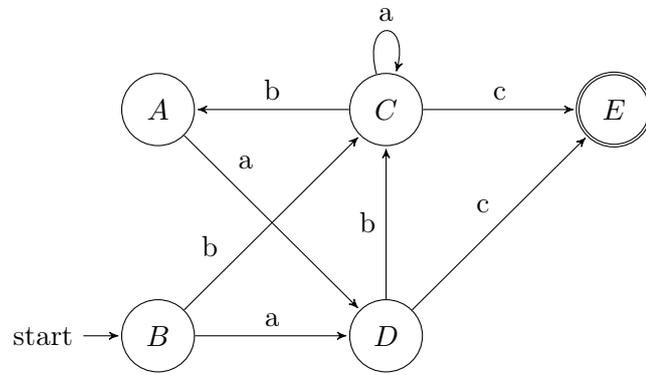
(8 valores)

1. Converta a expressão regular $(ac)^*(b+d)^*c$ num NFA, apresentando todos os passos do processo.

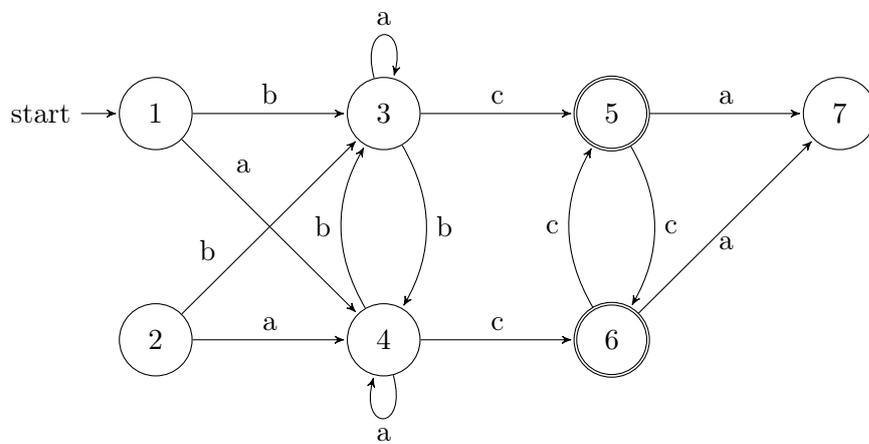
2. Converta o NFA seguinte num DFA, apresentando todos os passos do processo.



3. Converta o seguinte DFA numa expressão regular, apresentando todos os passos do processo.



4. Minimize o DFA seguinte, usando um algoritmo dado nas aulas, apresentando todos os passos do processo.



Grupo II

(8 valores)

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

- $V = \{S, R\}$
- $T = \{\text{empty}, \text{push}, \text{pop}\}$
- $P = \{S \rightarrow \text{push } R \text{ pop}, S \rightarrow \text{empty } S, S \rightarrow \epsilon, R \rightarrow \text{push } R \text{ pop}, R \rightarrow \epsilon\}$

1. Apresente uma derivação *leftmost* para a palavra **empty empty push push pop pop**.
2. A linguagem gerada por S é regular? Justifique, usando o Lema da Bombagem.
3. Construa a tabela de transição da função δ , justificando cada caso da definição da função.
4. Verifique, usando a tabela de transição, se é reconhecida a palavra **empty empty push push pop**.

Grupo III

(4 valores)

Defina a função de transição duma SBTM que, tendo uma lista de valores na posição 1 da memória e um valor na posição 2 da memória, verifica se o valor ocorre na lista, escrevendo na posição 3 da memória o resultado como um valor booleano.