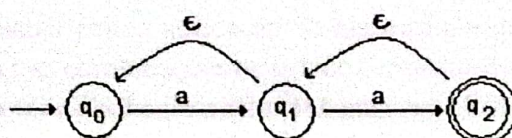
	Primeira Avaliação	Nota: 8,0
Curso:	Ciência da Computação	
Disciplina:	Linguagens Formais e Autômatos	
Aluno(a):		Data: 19/10/22

1) Considere a seguinte expressão regular: $(a + b + c) c^* a^* + d + (b + c)^*$. Marque a opção que apresenta uma palavra que não seja gerada por ela. (2 pts).

- a) ☐ d
- b) ☐ cb
- c) ☐ aaaa
- d) ☐ ccca
- e) ☒ bbca

2) Com o uso do algoritmo $AF\epsilon \rightarrow AFN$ construa o AFN equivalente ao $AF\epsilon$ abaixo e marque a afirmativa correta com relação ao autômato gerado: (2 pts)



- a) ☐ Possui 6 transições
- b) ☐ Possui 7 transições
- c) ☐ Possui 8 transições
- d) ☐ Possui 9 transições
- e) ☒ Nenhuma das respostas anteriores.

3) Qual das opções denota a função programa estendida (δ) de um $AF\epsilon$? (2 pts)

- a) ☒ $2^Q \times \Sigma^* \rightarrow 2^Q$
- b) ☐ $2^Q \times \Sigma^* \rightarrow Q$
- c) ☐ $Q \times \Sigma \rightarrow 2^Q$
- d) ☐ $Q \times \Sigma \rightarrow Q$
- e) ☐ Nenhuma das respostas anteriores.

4) Marque a opção que corresponde a uma expressão regular que especifique a linguagem aceita pelo AFN M dado. (2 pts)

$$M = (\{x, y, z\}, \{q_0, q_1, q_2\}, \delta, q_0, \{q_2\})$$

$\delta:$	x	y	z
q_0	$\{q_1\}$	$\{q_1\}$	$\{q_1\}$
q_1	$\{q_0, q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_2\}$
q_2	-	-	-

- a) ☐ $(xyz)^*xyz$
b) ☐ $(x + y + z)^*$
c) ☐ $(x + y + z)^*(x + y + z)$
d) ☐ $(x + y + z)(x + y + z)^*(x + y + z)$
e) ☒ Nenhuma das respostas anteriores.

5) Marque a opção que apresenta as afirmativas que são verdadeiras com relação ao algoritmo de minimização de autômatos. (1 pt):

I. Dois estados q_i e q_j são equivalentes quando para qualquer palavra w pertencente a Σ^* , $\delta(q_i, w)$ e $\delta(q_j, w)$ resultam ambos em estados não finais.

II. Um dos pré-requisitos para a aplicação do algoritmo é que o autômato seja um AFN.

III. Um dos pré-requisitos para a aplicação do algoritmo é que a função programa seja definida para todas as combinações de estados com símbolos do alfabeto.

IV. O algoritmo de minimização de autômatos gera um AFN com o menor número de estados possível.

- a) ☐ II
b) ☒ III
c) ☐ I, II, III
d) ☐ II, III, IV
e) ☐ I, II, III, IV

6) Dada a GR G_1 , qual a ER que gera a linguagem por ela denotada? (1 pt)

$G_1 = (\{S, A, B, C\}, \{a, b, c\}, P, S)$

P:

$S \rightarrow aS \mid A$

$A \rightarrow aaB \mid C$

$B \rightarrow cB \mid \epsilon$

$C \rightarrow bC \mid B$

- a) ☐ $a^*(aa + b)c^*$
b) ☒ $a^*(b^* + aa)c^*$
c) ☐ $a^*(aa + b)^*c$
d) ☐ $a^*(b^* + aa^*)c$
e) ☐ Nenhuma das respostas anteriores

Algoritmo AF ϵ \rightarrow AFN

Seja $M = (\Sigma, Q, \delta, q_0, F)$ um AF ϵ qualquer. Seja $M' = (\Sigma, Q, \delta', q_0, F')$ um AFN construído a partir de M como segue:

δ'

Tal que $\delta': Q \times \Sigma \rightarrow 2^Q$ onde $\delta'(q, a) = \underline{\delta}(\{q\}, a)$

F'

Conjunto de todos os estados q pertencentes a Q tal que algum elemento do $F\epsilon(q)$ pertence a F .

$$\underline{\delta}(P, \epsilon) = F\epsilon(P)$$

$$\underline{\delta}(P, wa) = F\epsilon(R) \text{ onde } R = \{r \mid r \in \delta(s, a) \text{ e } s \in \underline{\delta}(P, w)\}$$

Algoritmo AFN \rightarrow AFD

Seja $M = (\Sigma, Q, \delta, q_0, F)$ um AFN qualquer. Seja $M' = (\Sigma, Q', \delta', \langle q_0 \rangle, F')$ um AFD construído a partir de M como segue:

Q'

Conjunto de todas as combinações, sem repetições, de estados de Q as quais são denotadas por $\langle q_1 q_2 \dots q_n \rangle$, onde q_i pertence a Q , para i em $\{1, 2, \dots, n\}$. Note-se que a ordem dos elementos não distingue mais combinações. Por exemplo, $\langle q_u q_v \rangle = \langle q_v q_u \rangle$;

δ'

Tal que $\delta'(\langle q_1 \dots q_n \rangle, a) = \langle p_1 \dots p_m \rangle$ se, e somente se, $\delta(\{q_1, \dots, q_n\}, a) = \{p_1, \dots, p_m\}$. Ou seja, um estado de M' representa uma imagem dos estados de todos os caminhos alternativos de M ;

$\langle q_0 \rangle$

Estado inicial;

F'

Conjunto de todos os estados $\langle q_1 q_2 \dots q_n \rangle$ pertencentes a Q' tal que alguma componente q_i pertence a F , para i em $\{1, 2, \dots, n\}$.

Simplificação

- Todo estado que possua somente setas chegando a ele e que não seja um estado final deve ser eliminado, bem como as transições.
- Todo estado que possua apenas setas saindo de si e que não seja o estado inicial deve ser eliminado, bem como as transições.