Instituto de Computação

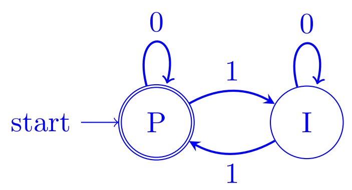
MATA50 – Linguagens Formais e Autômatos – 2023.1
Professor: Roberto Freitas Parente

Gabarito – Prova 01

Questão 1 (3 pontos)**.** *Construa automatos finitos para as seguintes linguagens:*

1. $L_1 = \{w \in \{0,1\}^* \colon$ *a soma dos dígitos de* w *é par*$\}$.

Resposta.



□

2. $L_2 = \{w \in \{0,1,2,3,4\}^* \colon$ *o último digito em* w *já apareceu em* $w\}$.

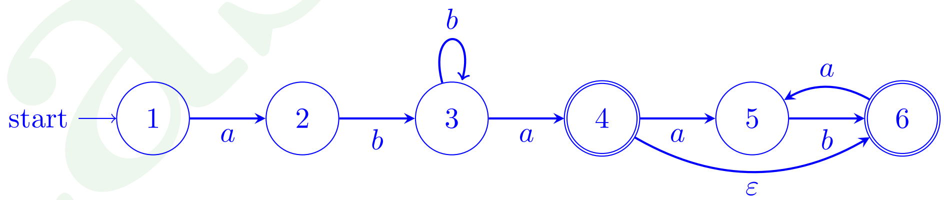
Resposta.

	0	1	2	3	4
→I	{0}	{1}	{2}	{3}	{4}
0	{0,F}	{0,1}	{0,2}	{0,3}	{0,4}
1	{0,1}	{1,F}	{1,2}	{1,3}	{1,4}
2	{0,2}	{1,2}	{2,F}	{2,3}	{2,4}
3	{0,3}	{1,3}	{2,3}	{3,F}	{3,4}
4	{0,4}	{1,4}	{2,4}	{3,4}	{4,F}
*F	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Quando o autômato ler um simbolo ele irá para o estado que o representa e vai ficar nesse estado até o fim da "computação" para que ao ler um simbolo que já foi lido ele ir para o estado final. □

3. $L_3 = \{ab^m ba(ab)^n \colon m, n \geq 0\}$

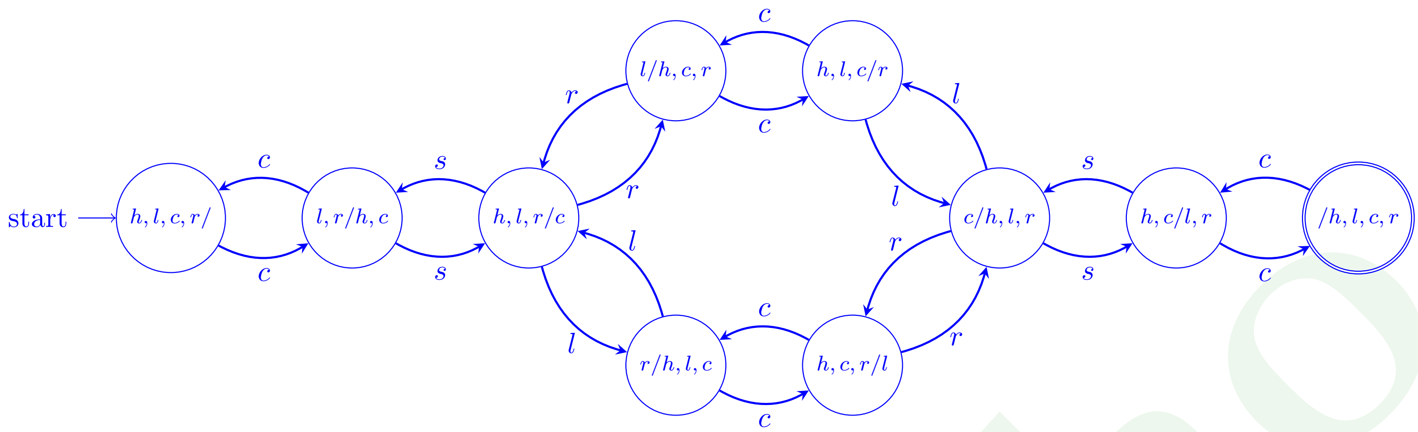
Resposta.



□

Questão 2 (2 pontos)**.** *Por razões desconhecidas, um homem viaja com um lobo, uma cabra e um enorme repolho. Ele enfrenta dificuldades imprevistas quando encontra seu caminho bloqueado por um rio, que só pode ser atravessado usando um pequeno barco ligado à costa. De fato, o barco é tão pequeno que ele só pode ter uma de suas possessões nele por vez. Além disso, seus companheiros animais estão se mostrando menos cooperativos e é evidente que nem o lobo e a cabra nem a cabra e o repolho podem ser deixados em terra juntos sem supervisão, para que não haja um evento de consumo indesejável. Construa um autômato finito que modele o problema utilizando um alfabeto adequado para representar todas as ações que o viajante pode tomar. Projete seu AFND de forma que o viajante atravessar o rio com todas as suas posses significa partir do estado inicial e chegar a um estado final.*
Obs: *O autômato projetado deve permitir todas os possíveis movimentos de travessia do viajante e não apenas o mais curto.*

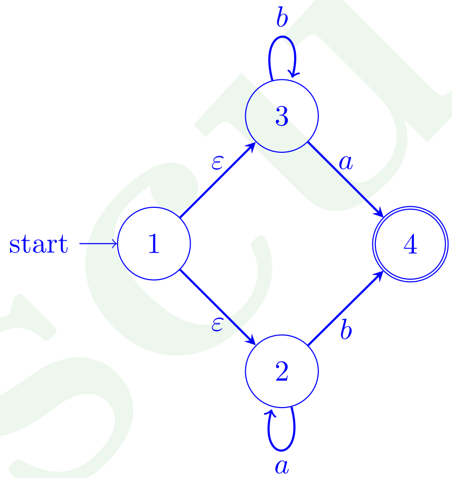
Resposta. Seja $\Sigma = \{l, c, r, s\}$ que representa o lobo, cabra, repolho e o homem viajando sozinho respectivamente.



E h representa o homem. □

Questão 3 (3 pontos)**.** *Apresente um ε-AFND com* **quatro estados** *que reconheça a linguagem $\{a\}^*\{b\} \cup \{b\}^*\{a\}$. Posteriormente aplique o algoritmo visto em sala para transformar no AFD equivalente e, por fim, aplique a minimização ao AFD encontrado.* ***Obs:*** *não esqueça de apresentar os ε-fechamento.*

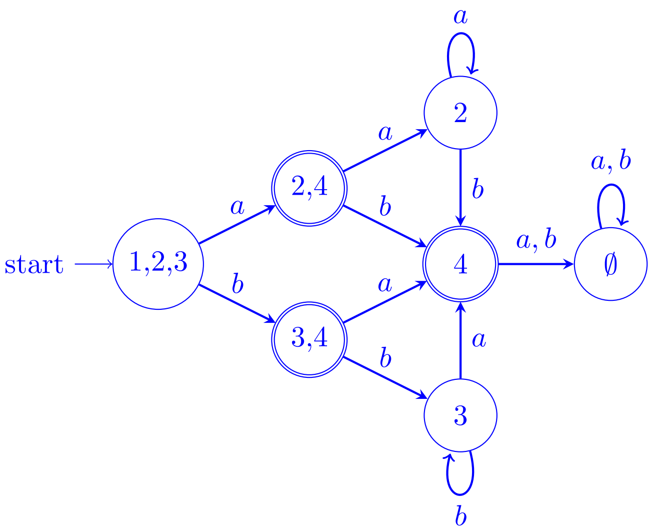
Resposta. Um dos $\varepsilon - AFND$ com quatro estados é:



Os $\varepsilon - fechamento$ são:

- $\varepsilon - fechamento(1) = \{1, 2, 3\}$
- $\varepsilon - fechamento(2) = \{2\}$
- $\varepsilon - fechamento(3) = \{3\}$
- $\varepsilon - fechamento(4) = \{4\}$

Criando o AFD equivalente utilizando somente os estados alcançáveis:



A tabela de minimização

2	X	-	-	-	-	-
3	X	X	-	-	-	-
4	X	X	X	-	-	-
24	X	X	X	X	-	-
34	X	X	X	X	X	-
∅	X	X	X	X	X	X
	123	2	3	4	24	34

Logo o AFD minimizado é ele mesmo $\square$

Questão 4 (2 pontos)**.** *Seja $L \subseteq \Sigma^*$ uma linguagem aceita pelo autômato finito $A = (\mathcal{Q}, \Sigma, \delta, q_0, F)$, i.e., $L = L(A)$. Mostre que $L^R = \{w^R \in \Sigma^* : w \in L\}$ é aceita por um autômato finito.*

Resposta. Informalmente desejamos criar um ε-AFND onde "invertemos" as setas no diagrama de transição, o estado inicial será estado final e criaremos um novo estado inicial que terá ε-transição para cada um do antigos estados finais. Formalizando, seja um autômato finito $B = (Q', \Sigma, \delta', q_0', \{q_0\})$, sendo $Q' = Q \ \cup \ \{q_0'\}$. δ' captura a noção de "inversão" das setas, ou seja, $\forall_{q \in Q} \forall_{a \in \Sigma}$ temos $\delta'(q, a) = \{p \in Q' : \delta(p, a) = q\}$ Por fim, as ε-transições podem ser definidas como $\forall_{q \in F}$ temos $\delta'(q_0', \varepsilon) = \{q\}$. $\square$

Questão 5 (1 ponto)**.** *Mostre que a equivalência de estados de um AFD é uma relação transitiva.*

Resposta. Denotando a relação de equivalencia de estados por E. Suponha que aEb e bEc. Seja w uma palavra se $\hat{\delta}(a, w)$ é final temos que como aEb então $\hat{\delta}(b, w)$ também é final, como bEc temos que $\hat{\delta}(c, w)$ também é final. Logo para toda palavra que é aceita a partir de a também vai ser aceita c. O caso de que w não é aceita a partir de a é análogo. Então aEc. $\square$

Questão 6 (2 pontos)**.** *Vimos que transformar um AFD em um AFND equivalente bastaria utilizar o "equivalente" (no sentido do digrama de estados), mas precisamos provar tal afirmação. Seja $A = (\mathcal{Q}, \Sigma, \delta_D, q_0, F)$ um AFD. Mostre como devemos construir o AFND $N = (\mathcal{Q}, \Sigma, \delta_N, q_0, F)$ e prove que $L(A) = L(N)$.*

Resposta. Para $q \in \mathcal{Q}$ e $a \in \Sigma$, definimos δ_N da seguinte forma: $\delta_D(q, a) = p$ se, e somente se, $\delta_N(q, a) = \{p\}$. Vamos provar por indução no comprimento da palavra de entrada que $\hat{\delta}_D(q_0, w) \in F \Longleftarrow \hat{\delta}_D(q_0, w) \cap F \neq \emptyset$.

Base: Seja $|w| = 0$, o mesmo que $w = \varepsilon$. Por definição da função de transição de um AFD e de um AFND, $\hat{\delta}_D(q_0, \varepsilon) = \{q_0\}$ e $\delta_N(q_0, \varepsilon) = q_0$

Hipótese indutiva: Agora suponha que vale para $|w| = n - 1$.

Passo indutivo: Seja $|w| = n$ e $w = w'a$ com $a \in \Sigma$. Por H.I. temos que $\hat{\delta}_D(q_0, w') = p$ e $\hat{\delta}_N(q_0, w') = \{p\}$. Se perda de generalidade, suponha que $\delta_D(p, a) = q$;

Pela definição de função de transição extendidade temos

$$\hat{\delta}_D(w, a) = \delta_D(\hat{\delta}_D(q_0, w'), a) = \delta_D(p, a) = q.$$

Por outro lado, pela definição de δ_N temos que $\delta_N(p, a) = \{q\}$ e assim

$$\hat{\delta}_N(q_0, w) = \bigcup_{r \in \delta_N(q_0, w')} \delta_N(r, a) = \delta_N(p, a) = \{q\}.$$

Assim, temos que $\hat{\delta}_D(q_0, w) = p \iff \hat{\delta}_N(q_0, w) = \{p\}$. Por fim, pela definição de $L(A)$ e $L(N)$ temos que uma palavra w só será aceita em A se, e somente se, $\hat{\delta}_N(q_0, w) \cap F \neq \emptyset$ que é a definição de aceitação em $L(N)$. $\square$