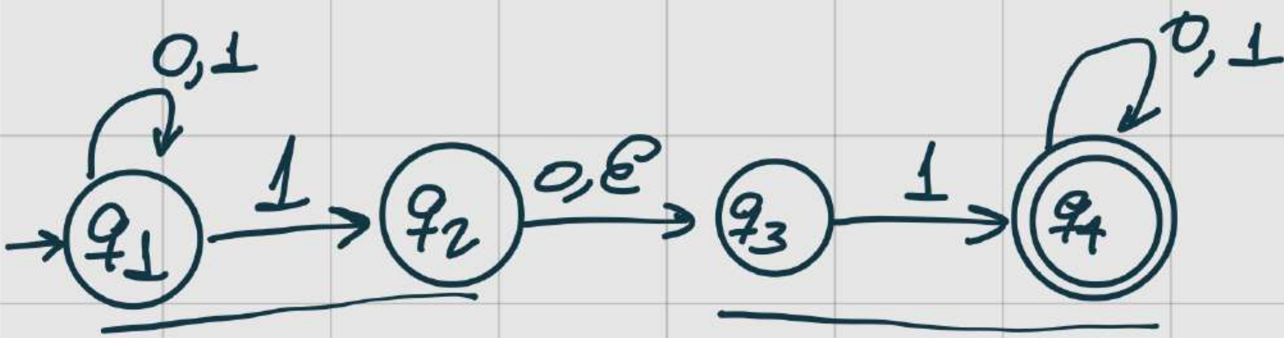
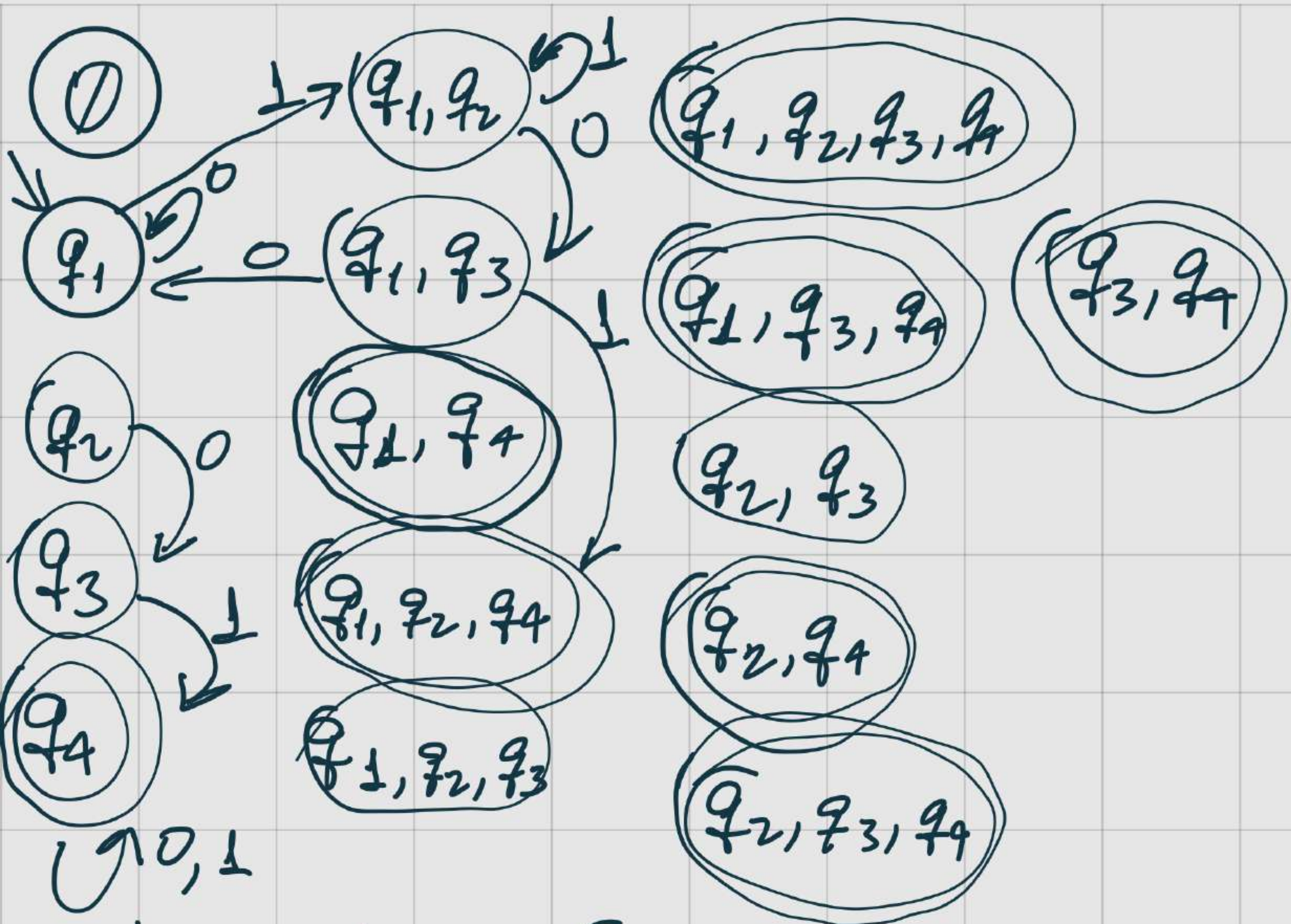
AFN:



Descrição formal $N_1 = (Q, \Sigma_\varepsilon, \delta, q_1, F)$

	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	



	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	

$\{q_1, q_2\} \overset{0}{=} \{q_1, q_3\}$

$\overset{1}{=} \{q_1, q_2\}$

$\{q_1, q_3\} \overset{0}{=} \{q_1\}$

$\overset{1}{=} \{q_1, q_2, q_4\}$

$\{q_1, q_4\} \overset{0}{=} \{q_1, q_4\}$

$\overset{1}{=} \{q_1, q_2, q_4\}$

Equivalência REGEX com AF

↳ Uma linguagem é regular se e somente se alguma expressão regular a descreve

↳ Se uma linguagem é descrita por uma expressão regular, então ela é regular

PROVA| Converter uma expressão regular R em um AFN N.

1. $R=a$ para algum a em Σ. Então $L(R)=\{a\}$, e o seguinte AFN reconhece $L(R)$.

→A máquina é um AFN mas não um AFD

$N=(\{q_1, q_2\}, \Sigma, \delta, q_1, \{q_2\})$, onde δ

$\delta(q_1, a)=\{q_2\}$ e $\delta(r,b)=\emptyset$ para $r \neq q_1$ ou $b \neq a$

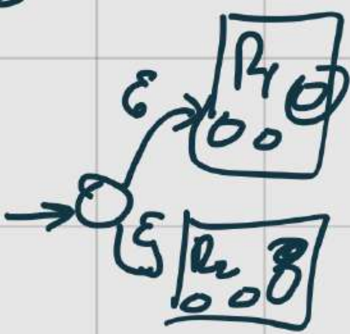
2. $R = \varepsilon$. Então $L(R) = \{\varepsilon\}$, com o AFN

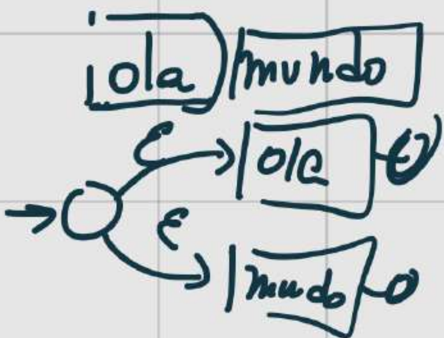
$N = (\{q_1\}, \Sigma, \delta, q_1, \{q_1\})$, onde $\underline{\delta(r,b) = \emptyset}$ para quaisquer r e b

3. $R = \emptyset$. Então $L(R) = \emptyset$

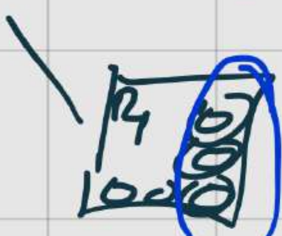
$N = (\{q_1\}, \Sigma, \delta, q, \emptyset)$, onde $\delta(r,b) = \emptyset$ para quaisquer r e b

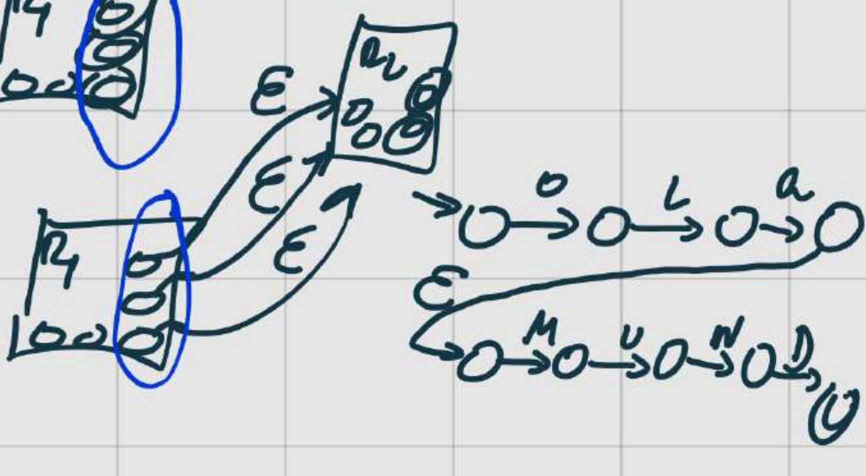
4. $\underline{R} = R_1 | R_2$



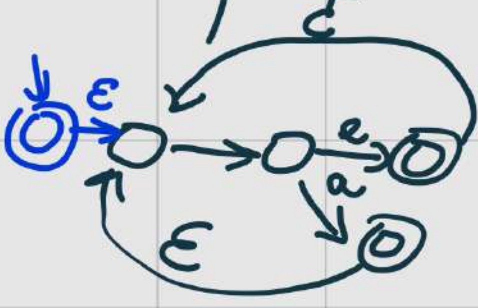


5. $\underline{R} = \underline{R_1} \circ \underline{R_2}$





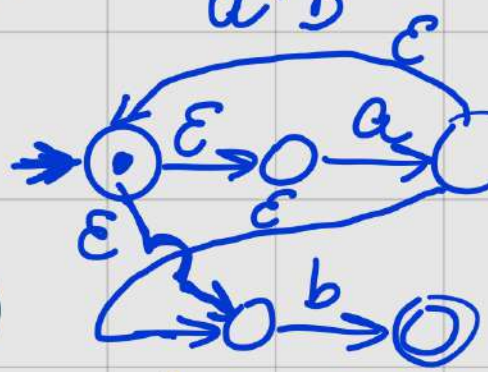
6 = $R = R_1^*$ - R_1^*



exemplo $(ab|a)^*$

$\Sigma = \{[a-z]\}$

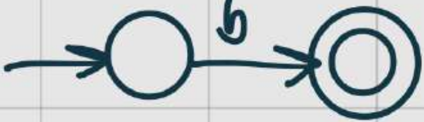
a^*b



aab

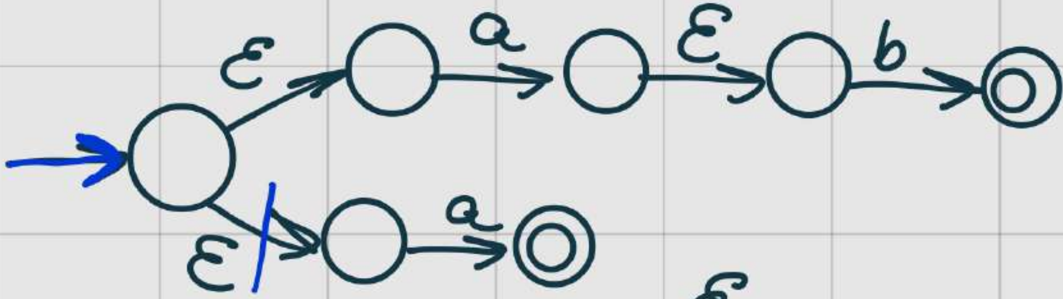
a

b

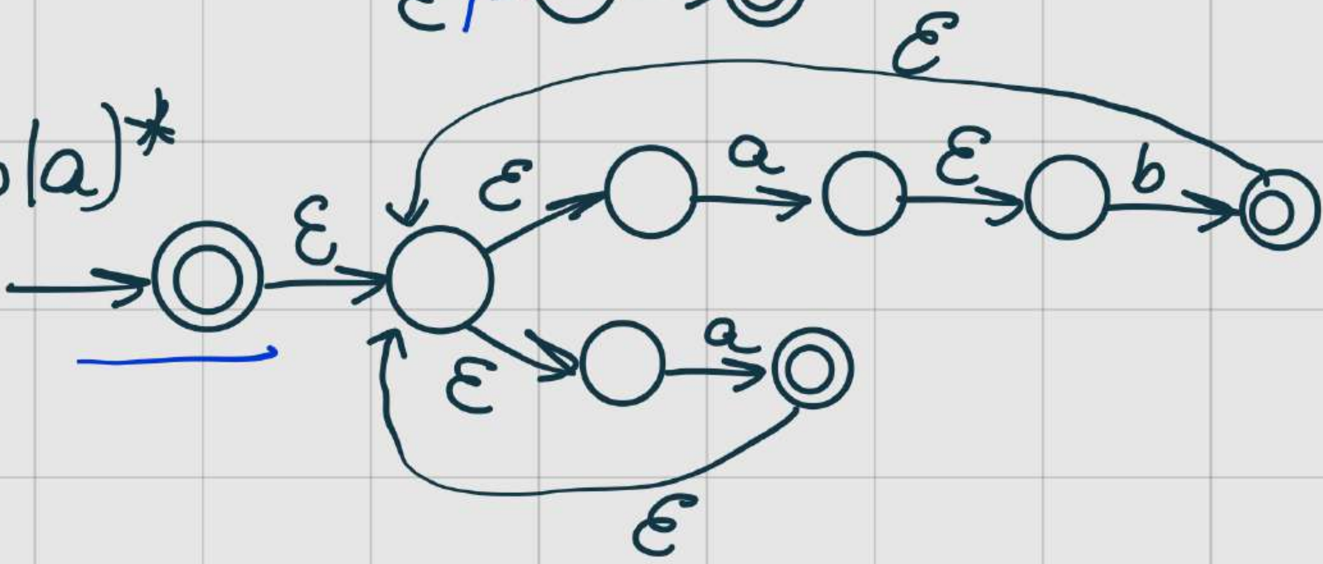


ab

ab|a

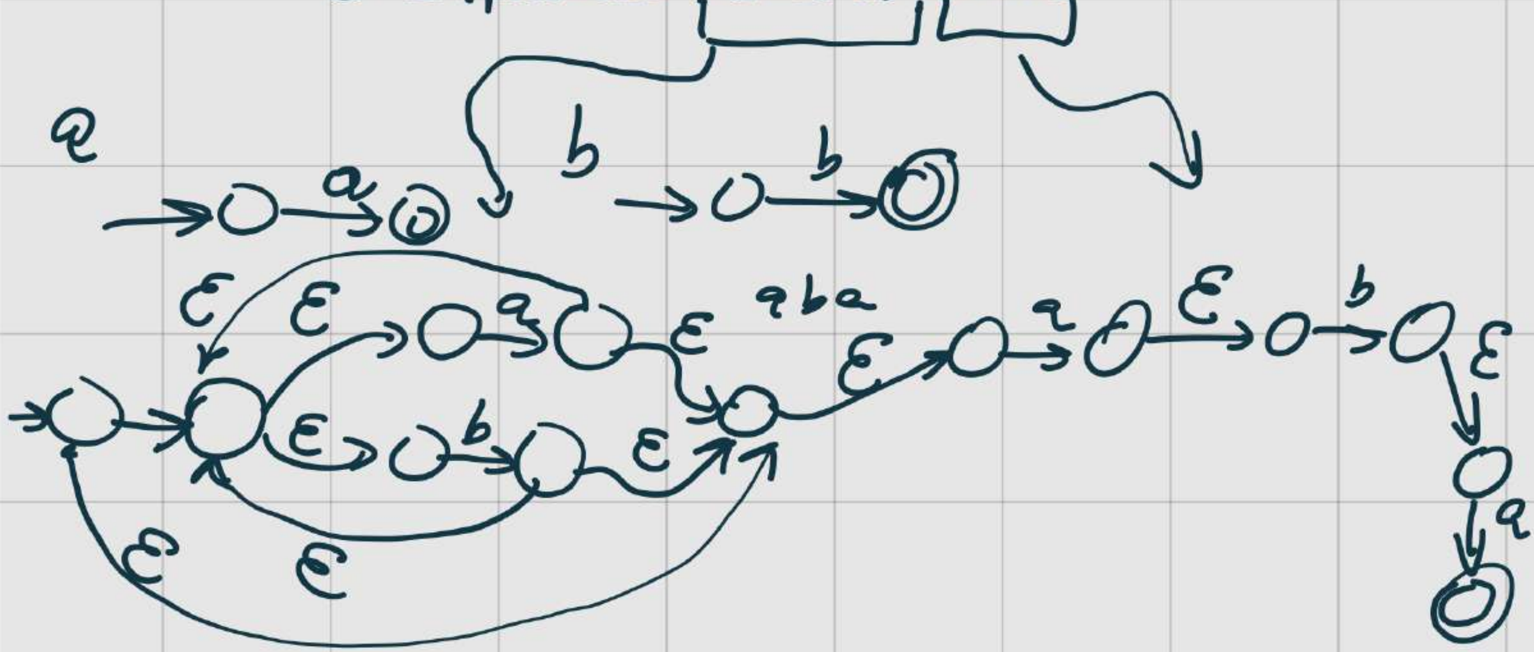


$(ab|a)^*$



faça em casa | Construa o AFN para

a expressão $(a|b)^*aba$



Se uma linguagem é regular, então ela é descrita por uma expressão regular

AFNG | Autômato finito não-determinístico generalizado

→ O estado inicial tem setas de transição saindo para todos os outros estados, mas nenhuma seta chegando de qualquer outro estado.

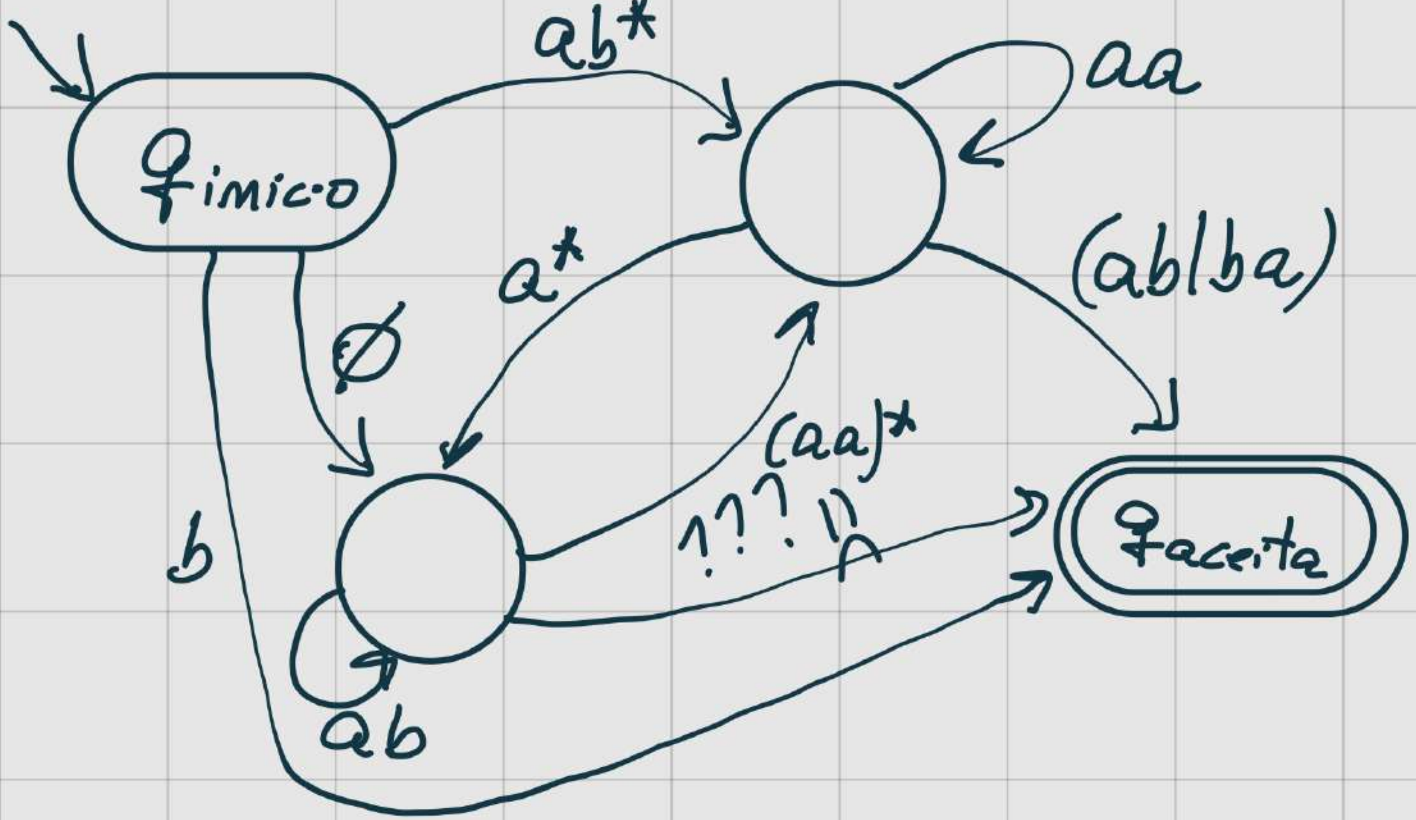
→ Existe apenas 1 estado de aceitação, e ele tem setas chegando de todos outros estados, mas nenhuma saindo. O estado de aceitação NÃO é o mesmo que o inicial.

→ Com exceção dos estados inicial e de aceitação, uma seta sai de cada estado para todos os outros e também de cada estado para ele mesmo.

formalmente um AFNG é uma 5-upla

$(Q, \Sigma, \delta, q_{inicio}, q_{aceita})$, onde

1. Q é o cjto de estados
2. Σ é o alfabeto de entrada
3. $\delta : (Q \setminus \{q_{aceita}\}) \times (Q \setminus \{q_{inicio}\}) \to R$
4. q_{inicio} é o estado inicial
5. q_{aceita} é o estado de aceitação



Um AFNG aceita uma cadeia w em Σ^* se $w = w_1 w_2 \dots w_k$, onde cada w_i está em Σ^*, e existe uma sequência de estados $q_0 q_1 \dots q_k$ tal que

1. $q_0 = q_{inicio}$ é o estado inicial
2. $q_k = q_{aceita}$ é o estado de aceitação, e
3. para cada i, temos $w_i \in L(R_i)$, onde $R_i = \delta(q_{i-1}, q_i)$; ou R_i é a expressão sobre a seta de q_{i-1} a q_i

→ Para converter um AFD para um AFNG adicionando um novo estado inicial e um novo estado de aceitação e setas adicionais conforme necessário. Usamos o procedimento CONVERT(G) que toma um AFNG como entrada e retorna uma expressão regular equivalente

CONVERT(G):

1. Seja k o número de estados de G
2. Se k = 2, então G deve consistir de um estado inicial, um estado de aceitação, e uma única seta conectando os dois rotulada com uma expressão regular R
   ↳ Retorne a expressão R

3. Se $k>2$, selecionemos qualquer $q_{rem} \in Q$ diferente de q_{inicio} e q_{aceita} e seja G' o AFNG$(Q', \Sigma, \delta', q_{inicio}, q_{aceita})$, onde:

→ $\boxed{Q' = Q \setminus \{q_{rem}\}}$,

e para qualquer $q_i \in Q' \setminus \{q_{aceita}\}$ e qualquer $q_j \in Q' \setminus \{q_{inicio}\}$ seja

$$\delta'(q_i,q_j) = ((R_1)(R_2)^*(R_3)) \mid (R_4)$$

para $R_1 = \delta(q_i, q_{rem})$, $R_2 = \delta(q_{rem}, q_{rem})$,

$R_3 = \delta(q_{rem}, q_j)$ e $R_4 = \delta(q_i, q_i)$

ab

$R_1 = a$ $R_4 = \emptyset$

$R_2 = \emptyset$

$R_3 = b$

4. Compute CONVERT(G') e retorne este valor

$$\boxed{\delta'(q_i,q_j)} = ((R_1)(R_2)^*(R_3)) \mid (R_4)$$

exemplo

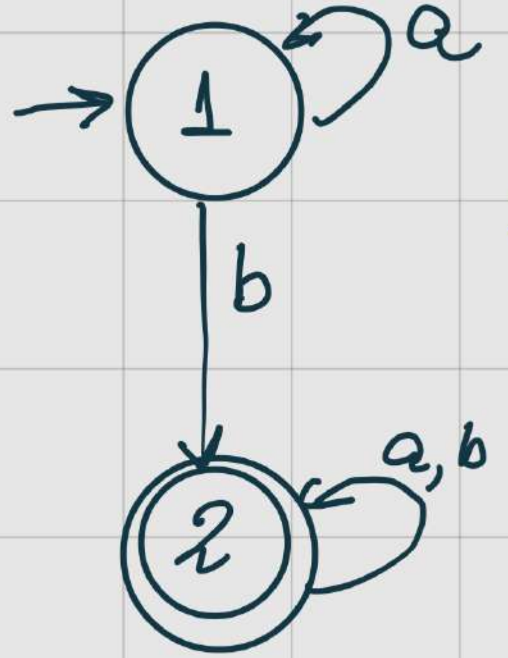
$R_1 = a^*b$

$R_2 = (a|b)$

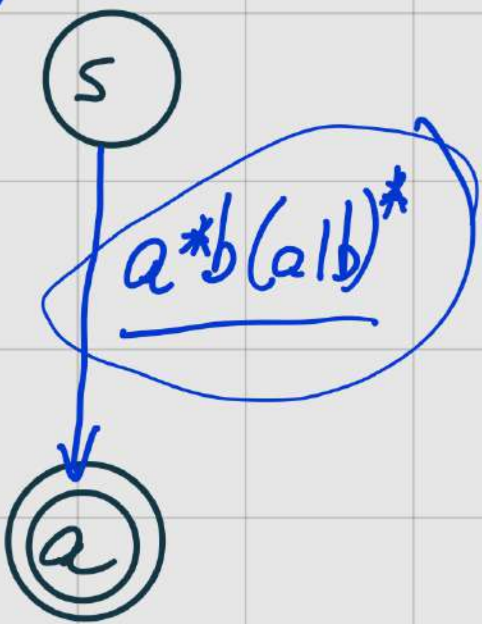
$R_3 = \varepsilon$

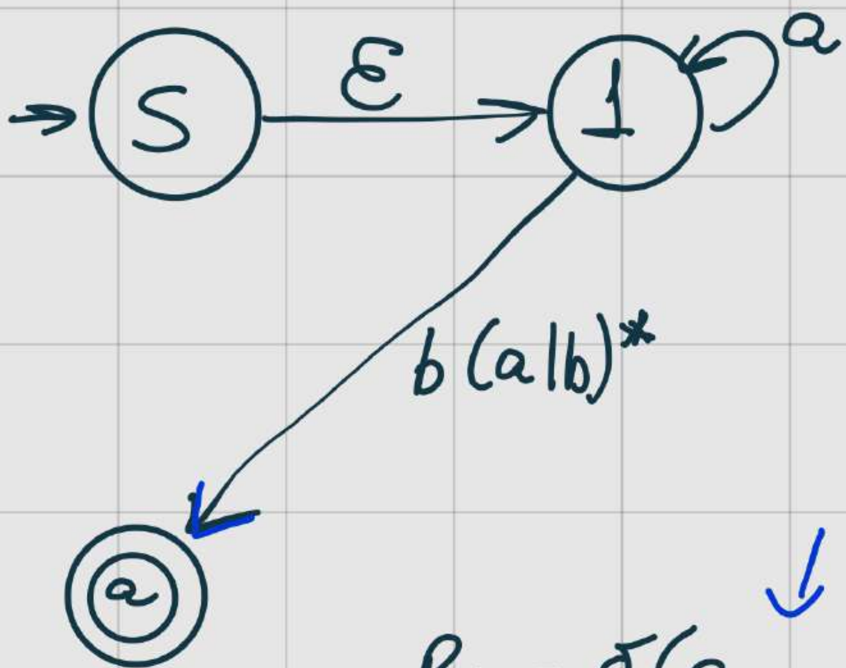
$R_4 = \emptyset$

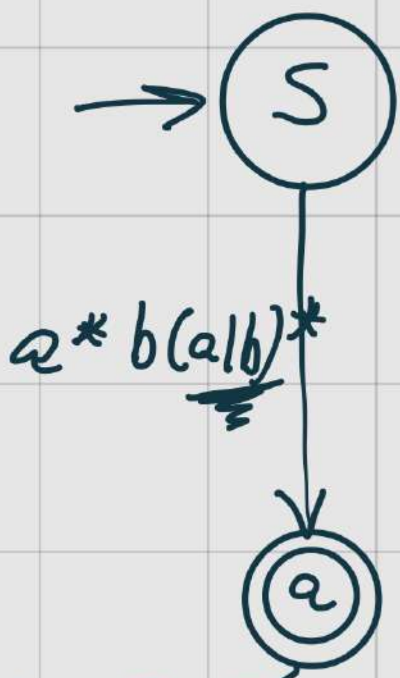
$(a^*b)(a|b)^* \cdot \varepsilon | \emptyset$



$\underline{s}$ $\underline{a}$



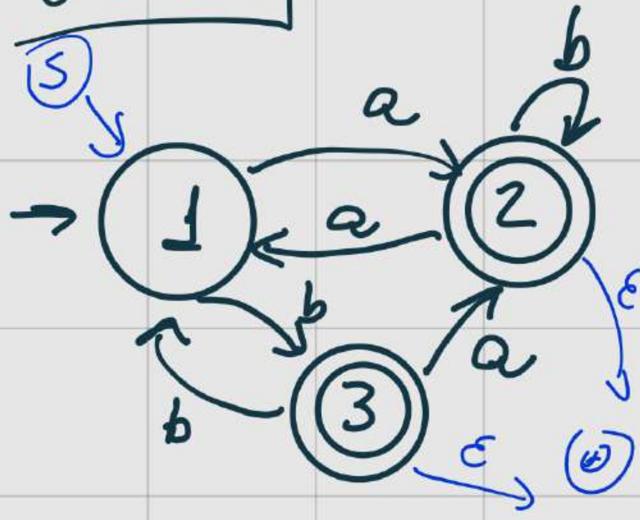




para $\underline{R_1} = \delta(q_i, q_{rem})$, $R_2 = \delta(q_{rem}, q_{rem})$,

$R_3 = \delta(q_{rem}, q_j)$ e $R_4 = \delta(q_i, q_j)$

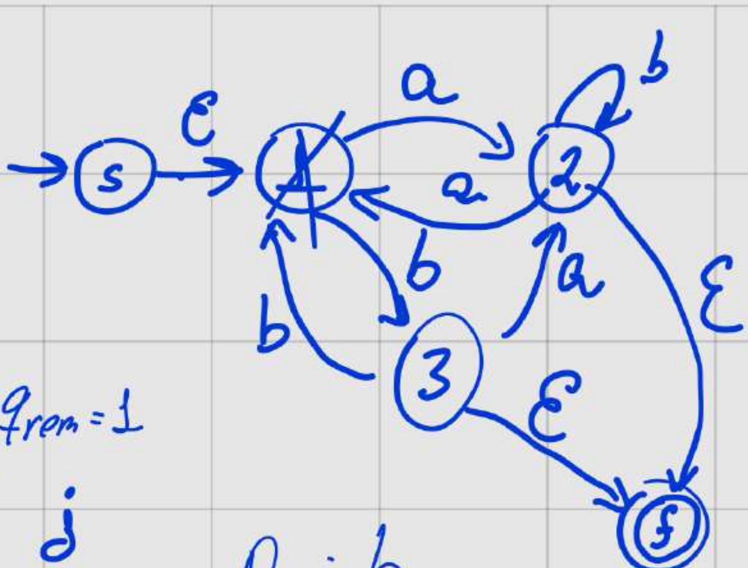
faça



$q_{rem} = 1$

i j

3 3



$R_1 : b$

$R_2 : \emptyset$

$R_3 : b$

$R_4 : \emptyset$

$b\emptyset^* b | \emptyset$

bb

$a\emptyset^* a | b$

$\underline{aa|b}$

S 2

S 3

S f

2 3

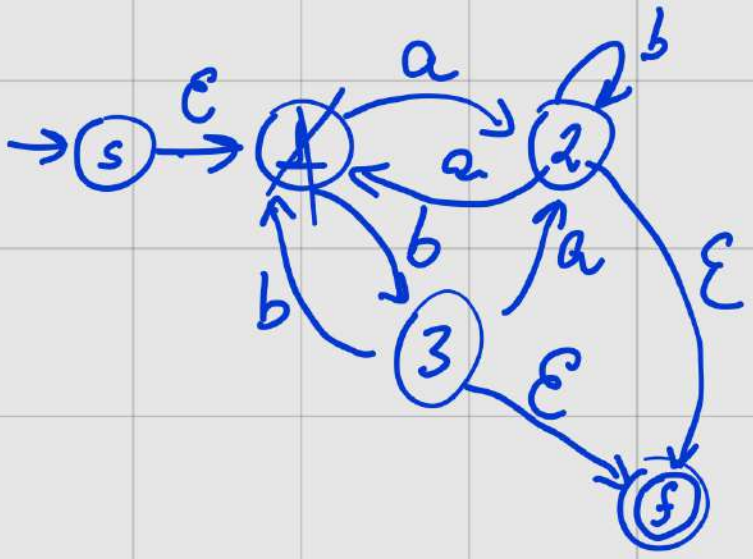
2 f

3 2

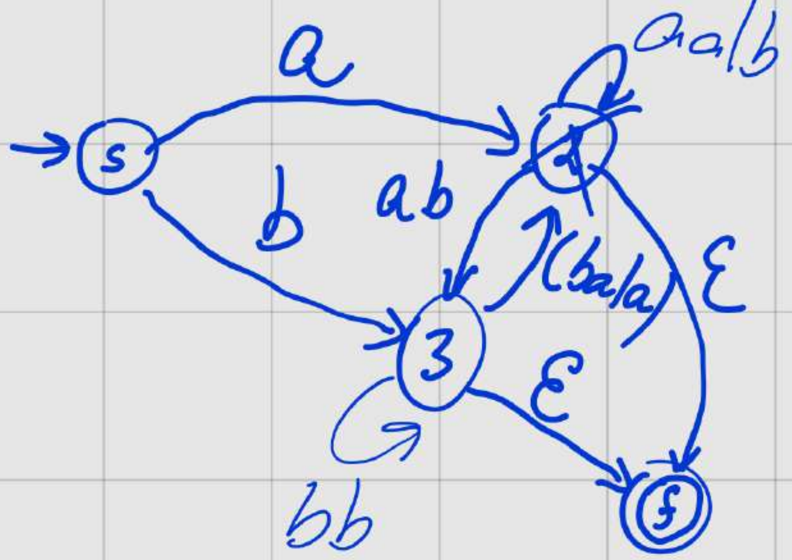
3 f

3 3

2 2



s f
s 2
s 3
2 f
2 3
3 2
3 f
2 2
3 3



S f ✓
S 3 ✓
3 f ✓
3 3

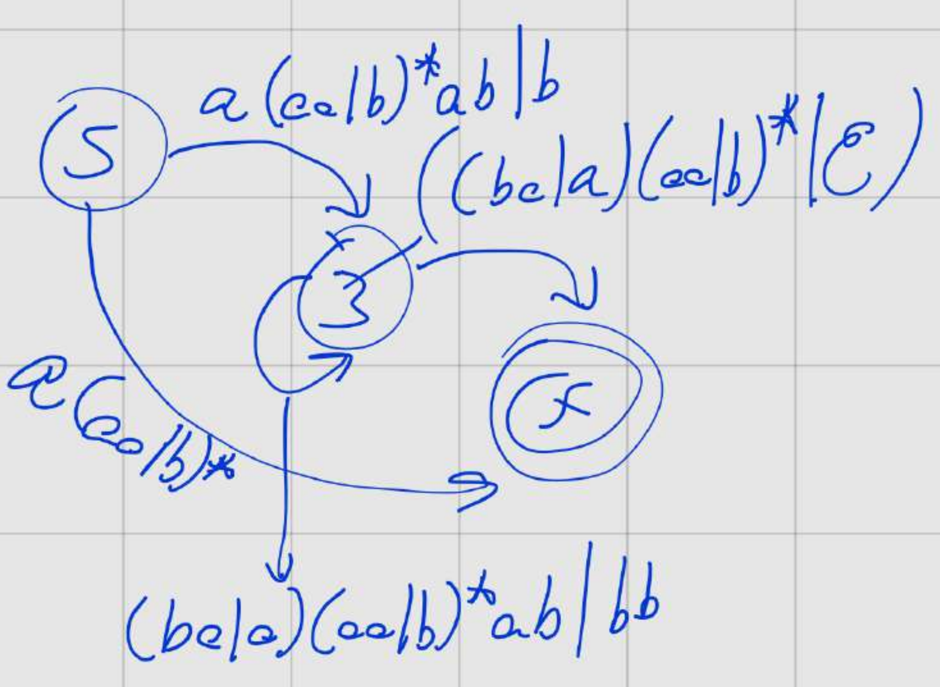
S f

$R_1 = a(aa|b)^* ab | b$

$R_2 (= (ba|a)(aa|b)^* ab | bb)^*$

$R_3 = ((ba|a)(aa|b)^* | ε$

$R_4 = a(aa|b)^*$



$(a(aa|b)^*ab|b)((ba|a)(aa|b)^*ab|bb)^*$

$((ba|a)(aa|b)^*|\varepsilon)|a(aa|b)^*$