AFD Definições

Ao responder as questões abaixos indique a referência bibliográfica usada (livro).

1) Defina formalmente um AFD. Como eles podem ser representados?

2) Como cadeias são geradas por um AFD?

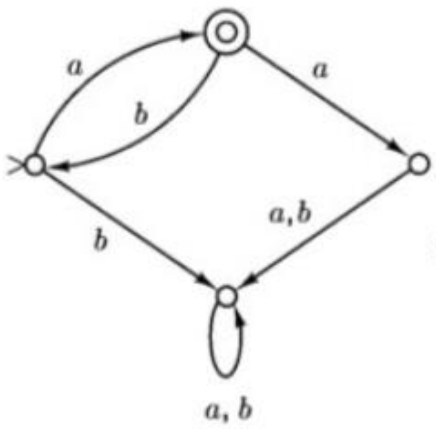
3) Defina o conceito de configuração quando do processamento de um AFD. Dê exemplos.

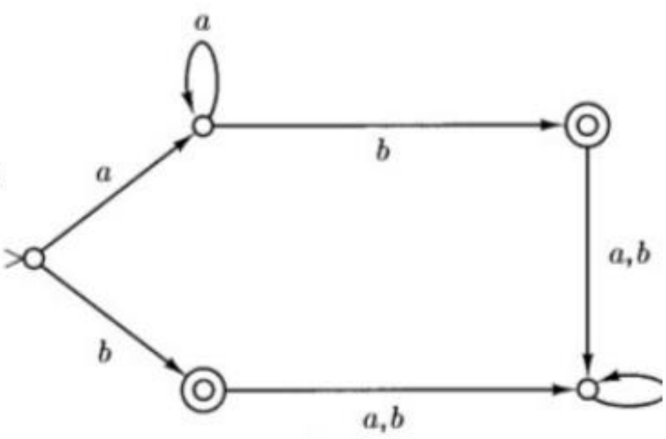
4) Quando uma cadeia é aceita/reconhecida por um AFD?

5) Dê 5 exemplos de AFDs e ilustre o processo de geração de cadeias.

6) O que significa a expressão a configuração (q,w) produz/deriva em um passo a configuração (p,v) quando do processamento de uma cadeia por um AFD? Notação para produção ⊢⊢. (Observação o afd produz = the dfa yields)
7) Defina função de transição estendida

8) Considere os AFDs abaixo. Mostre a derivação de 3 cadeias aceitas e 3 cadeias não aceitas pelos AFDS.





9) Instale o software Jautomata e processe as cadeias do exemplo anterior nos respectivos AFDs.

10) Usando o Jautomata desenvolva AFDs para reconhecer datas, valores monetários, e cadeias delimitadas por aspas.

Respostas:

1) Um autômato finito determinístico — também chamado máquina de estados finita determinística (AFD) — é uma Máquina de estados finita que aceita ou rejeita cadeias de símbolos gerando um único ramo de computação para cada cadeia de entrada.

2) O modo de geração tem como objetivo produzir uma lista de todas as cadeias de caracteres dentro da linguagem. Em vez de seguir uma única transição de cada estado, o modo de criação segue todas elas. Na prática, isso pode ser alcançado por um paralelismo maciço (fazer com que o programa se ramifique em dois ou mais processos cada vez que este se deparar com uma decisão) ou então por Recursão. Como antes, a computação começa no estado inicial e procede a seguir cada transição disponível, sem perder de vista que processos foram tomados. Toda vez que o autômato se encontra em um estado de aceitação este sabe que a sequência de processos tomada forma uma cadeia de caracteres válida na linguagem e adiciona essa cadeia na lista que o programa está gerando.

3) A configuração se refere a um par de um estado e a cadeia a ser processada (E, Cadeia)

ex. no início: (1, aabaa)

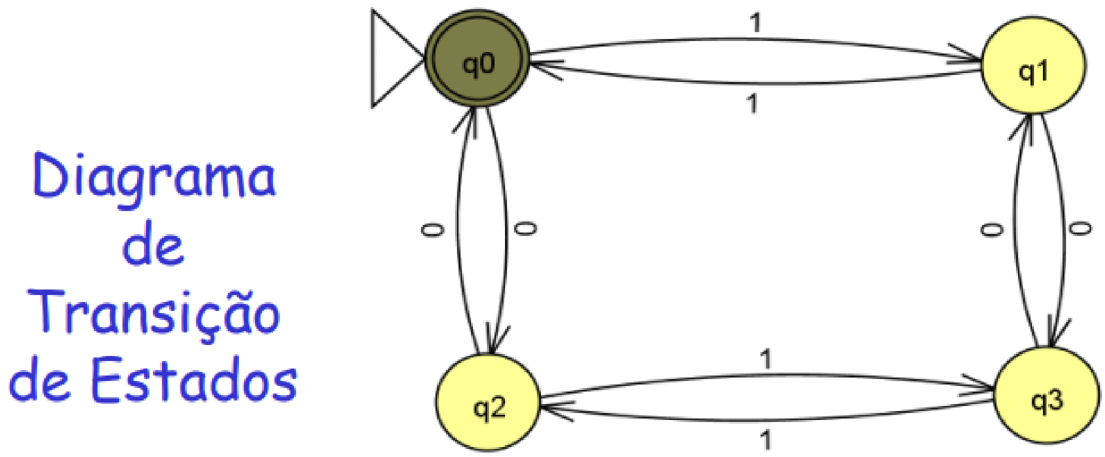
(2, abaa)

Cada par da forma estado atual, cadeia a ser lida, é uma configuração e mostra como está o processamento de uma máquina a ser implementada. Uma configuração é denotada pelo par (q,w), em que q pertence a Q e w pertence a Sigma*. Esta notação informa que atualmente a máquina se encontra no estado q e a cadeia a ser processada é w.

4) Quando ao iniciar o processamento de w no estado inicial, o AFD percorre uma série de configurações que o levam a uma configuração da forma (qf, epsilon) tal que qf é um estado final. Caso contrário, diz-se que a cadeia w não é aceita.

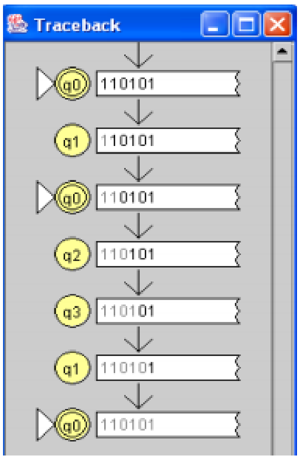
5)

EXEMPLO 1) Fazer um AFD M que aceita L(M) = {w | w possui um nro par de 0´s e de 1´s }



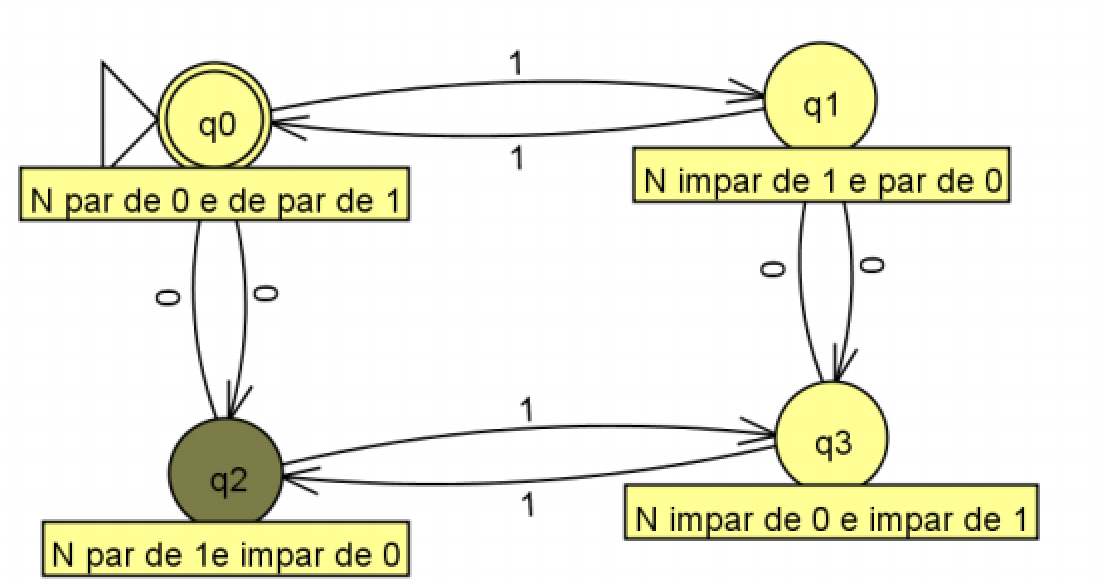
Testando

110101

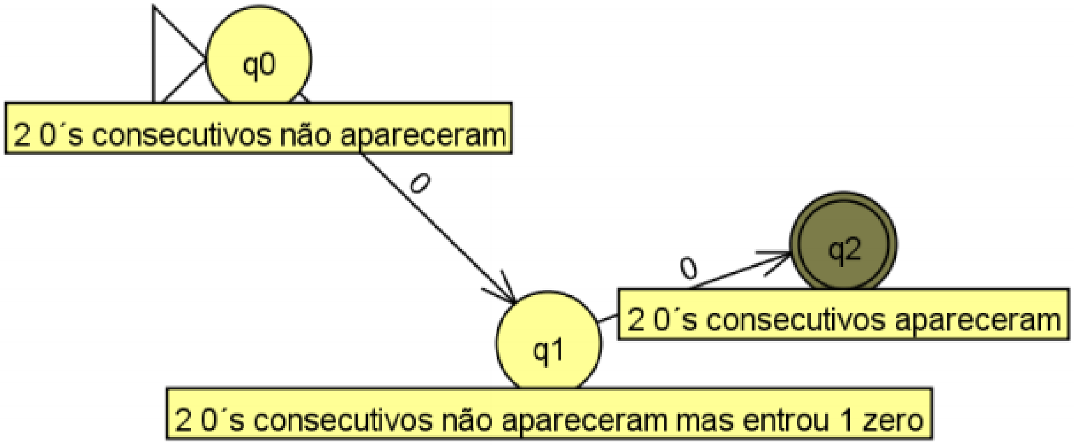


Testando

110

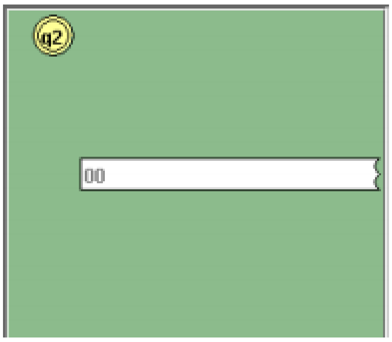


EXEMPLO 2) Fazer um AFD M que aceita L(M) = {w ∈ {0,1}* |w possua pelo menos dois 0´s consecutivos}



Testando

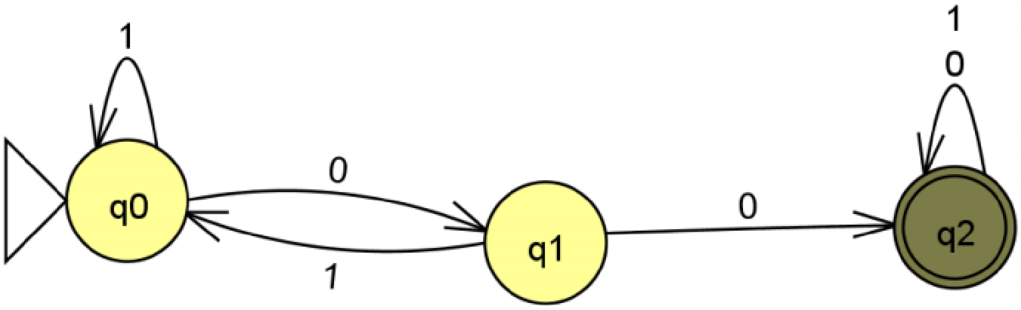
00



Fundo verde = aceito.

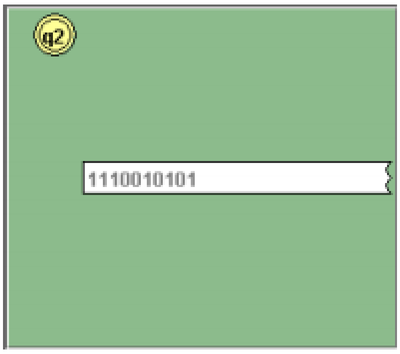
EXEMPLO 3) Completando o AFD M e reconhecendo uma cadeia de L(M) M = ({q0,q1,q2}, {0,1}, δ, q0,{q2})

δ:



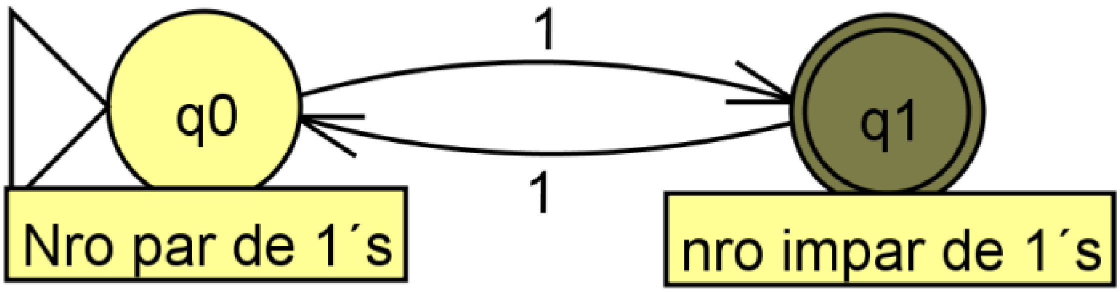
Testando

1110010101

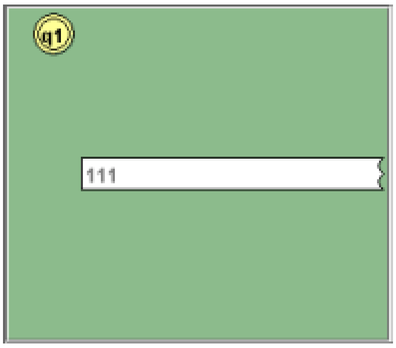


Fundo verde = aceito.

EXEMPLO 4) Fazer um AFD M que aceita L(M) = {w $\in$ {0,1}* |w possua um número impar de 1´s}



Testando 111

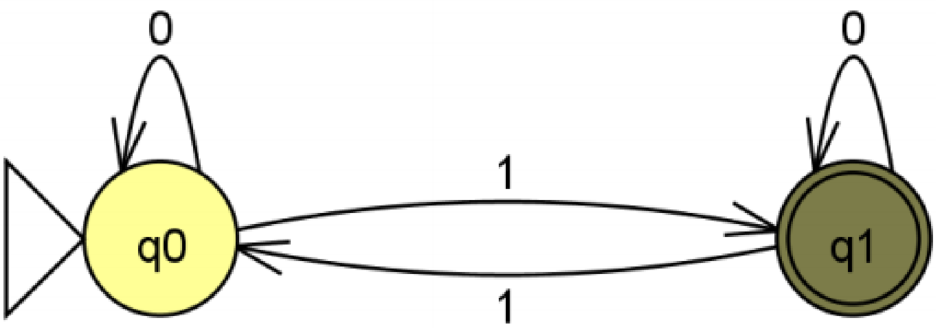


Fundo verde = aceito.

EXEMPLO 5)

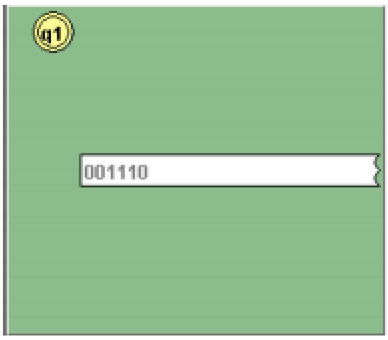
Completando o AFD M e reconhecendo uma cadeia de L(M) M = ({q0,q1}, {0,1}, δ, q0,{q1})

δ:



Testando

001110



Fundo verde = aceito.

TESTES FEITOS USANDO O SOFTWARE JFLAP.

6) A notação (q,w) |- (q',w') indica que o AFD n vai da configuração (q,w) para a configuração (q',w') (q primo, w primo) consumindo apenas um símbolo da cadeia de entrada. Com isto, tem-se que:

a) w pertence a Sigma*

b) w = aw', onde a é um símbolo (pertence a Sigma) e w' pertence a Sigma*

c) (q,a) -> q' Pertence a delta (é uma transição do AFD)

7) Seja um AFN M=(Q, Σ, δ, I, F). A função de transição estendida δ, é uma função de ℘(Q) x Σ* para ℘(Q), definida recursivamente como:

$$\hat{\delta}(\phi, w) = \phi \quad \text{para todo } w \in \Sigma^*$$

$$\hat{\delta}(A, \varepsilon) = A \quad \text{para todo } A \subseteq Q$$

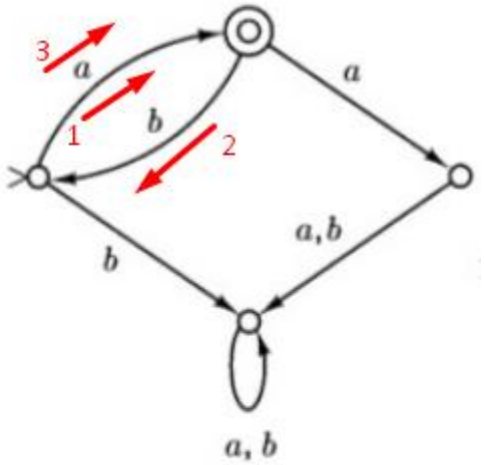
$$\hat{\delta}(A, ay) = \hat{\delta}(\cup_{q \in A} \delta(q, a), y) \quad \text{para todo } A \subseteq Q \text{ , } a \in \Sigma \text{ e } y \in \Sigma^*.$$

8)

AFD 1:

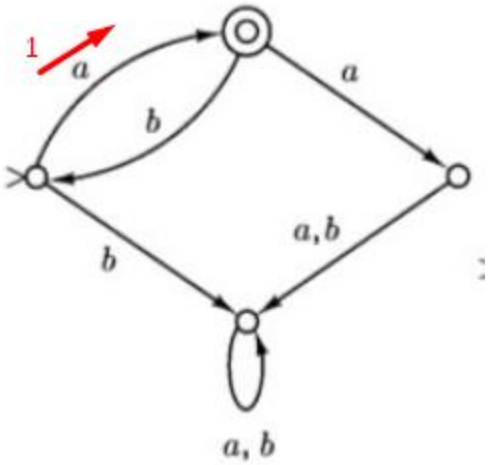
Cadeias: aba, a, ababa, ab, aaa, aab

Aba



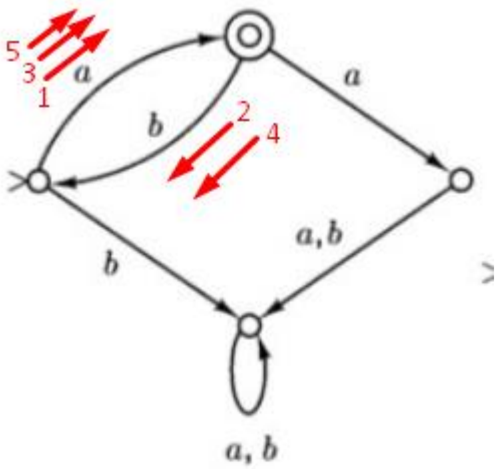
aceita.

A



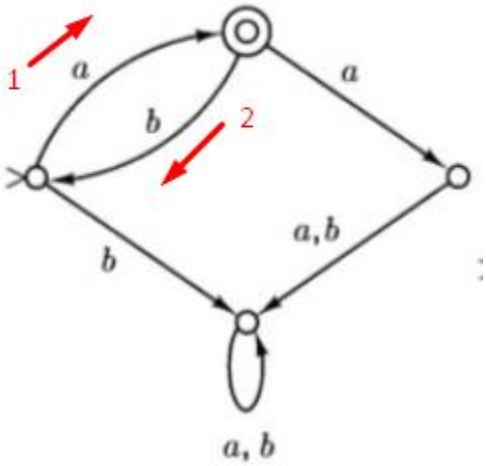
aceita.

Ababa



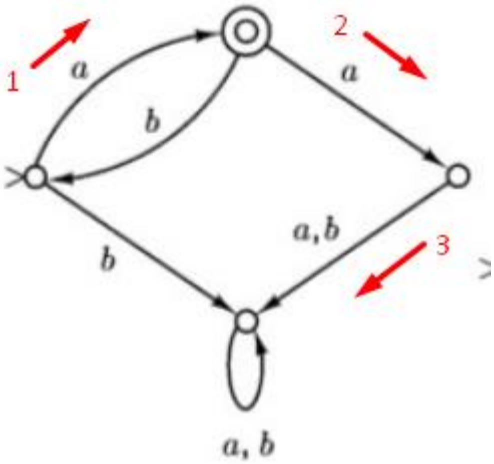
aceita.

Ab



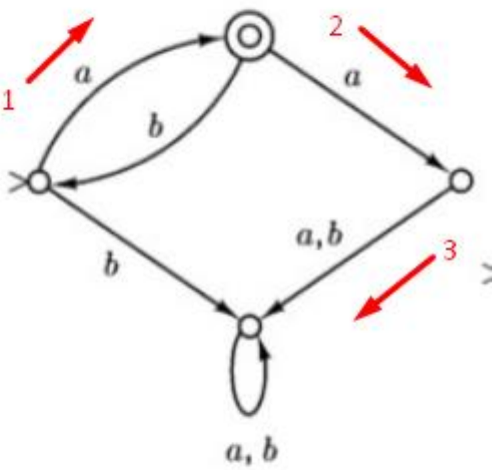
não aceita, pois não termina em um estado final.

Aaa



não aceita, pois não termina em um estado final.

aab

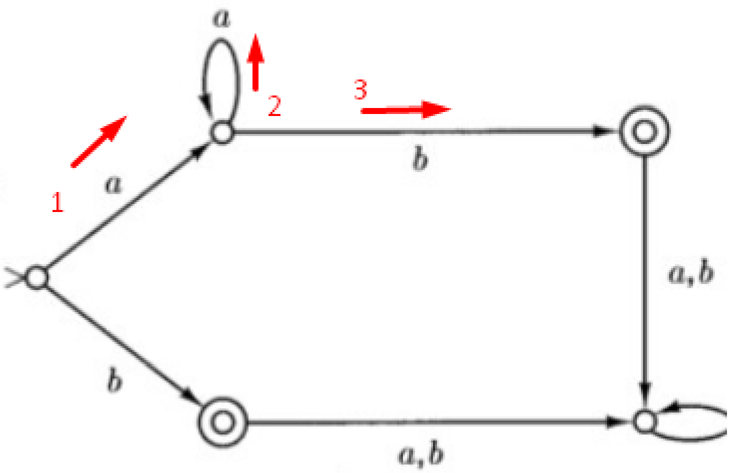


não aceita, pois não termina em um estado final.

AFD 2:

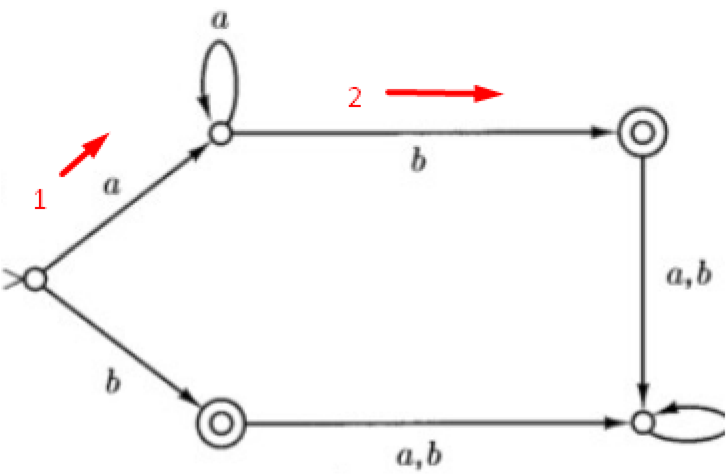
Cadeias: aab, ab, b, aa, abb, aabb

Aab



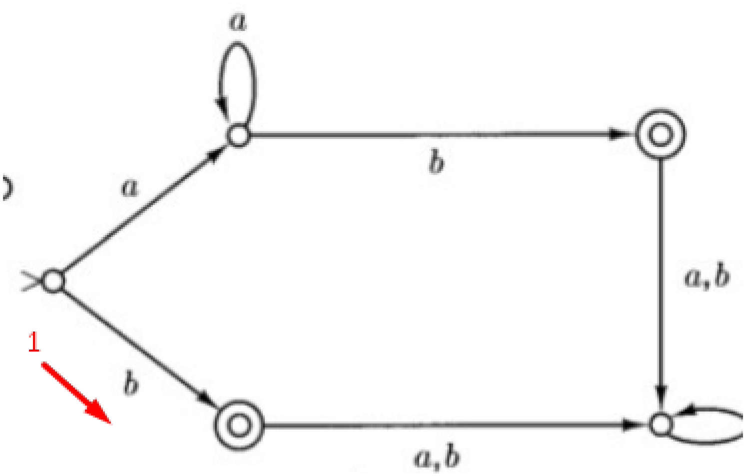
aceita.

Ab



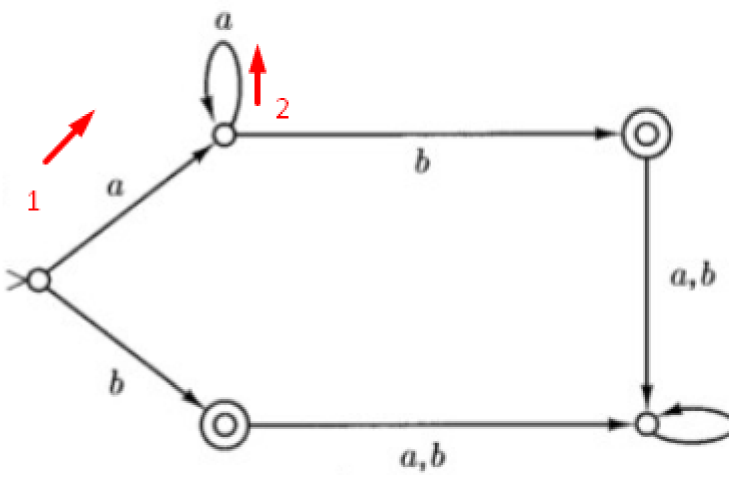
aceita.

B



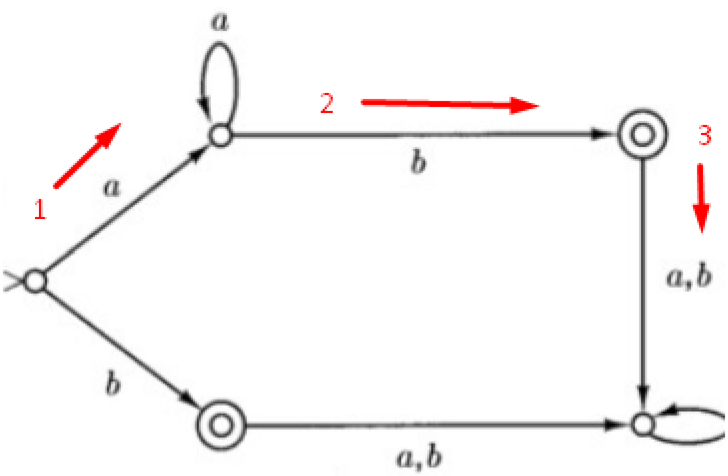
aceita.

Aa



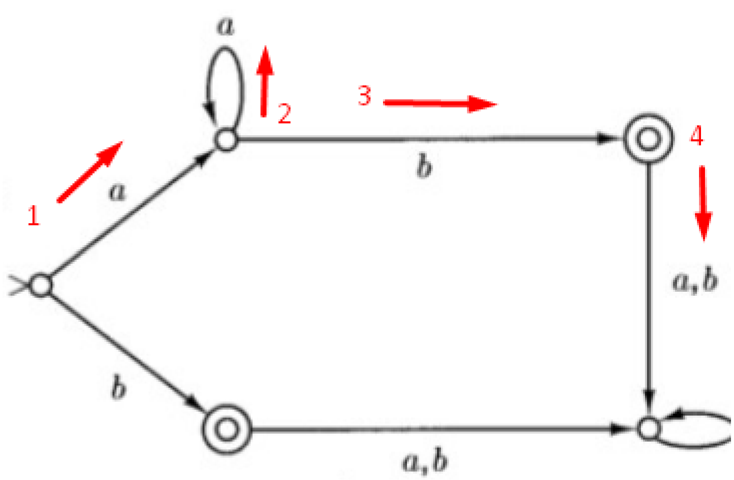
não aceita, pois não termina em um estado final.

Abb



não aceita, pois não termina em um estado final.

Aabb



não aceita, pois não termina em um estado final.