BLG 311E
Formal Languages and Automata

Recitation-2

Ali Azmoudeh
Erdi Sarıtaş
Mehmet Selahaddin Şentop

Deterministic and Non-Deterministic Finite Automata

Question

Design a DFA over the alphabet (a, b) that accepts strings of the form:

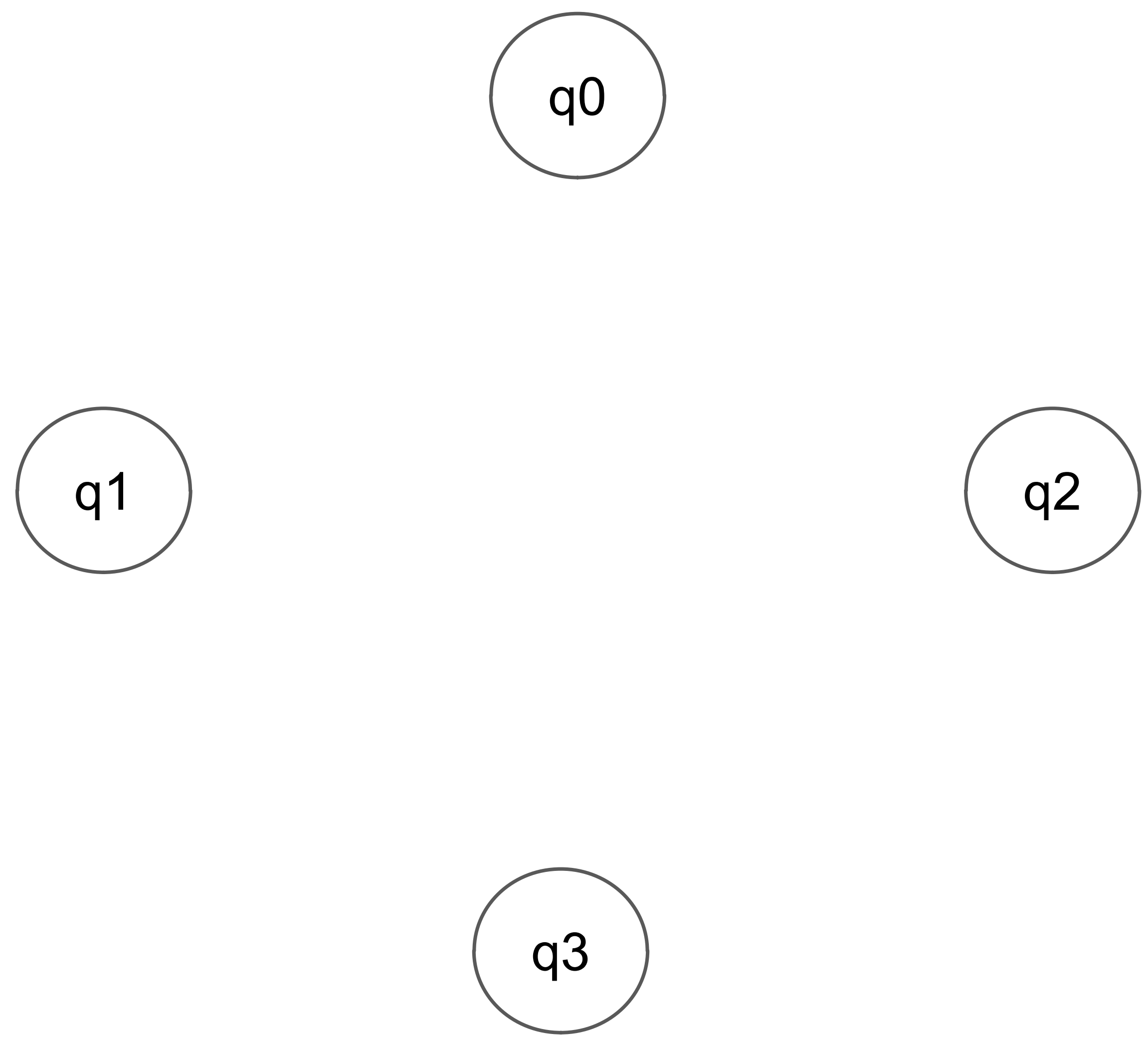
$$ab^{2k+1}\ ,\ \ k \geq 0$$

Solution

ab^{2k+1} , $k \geq 0$

States: q0, q1, q2, q3

Alphabets: a, b

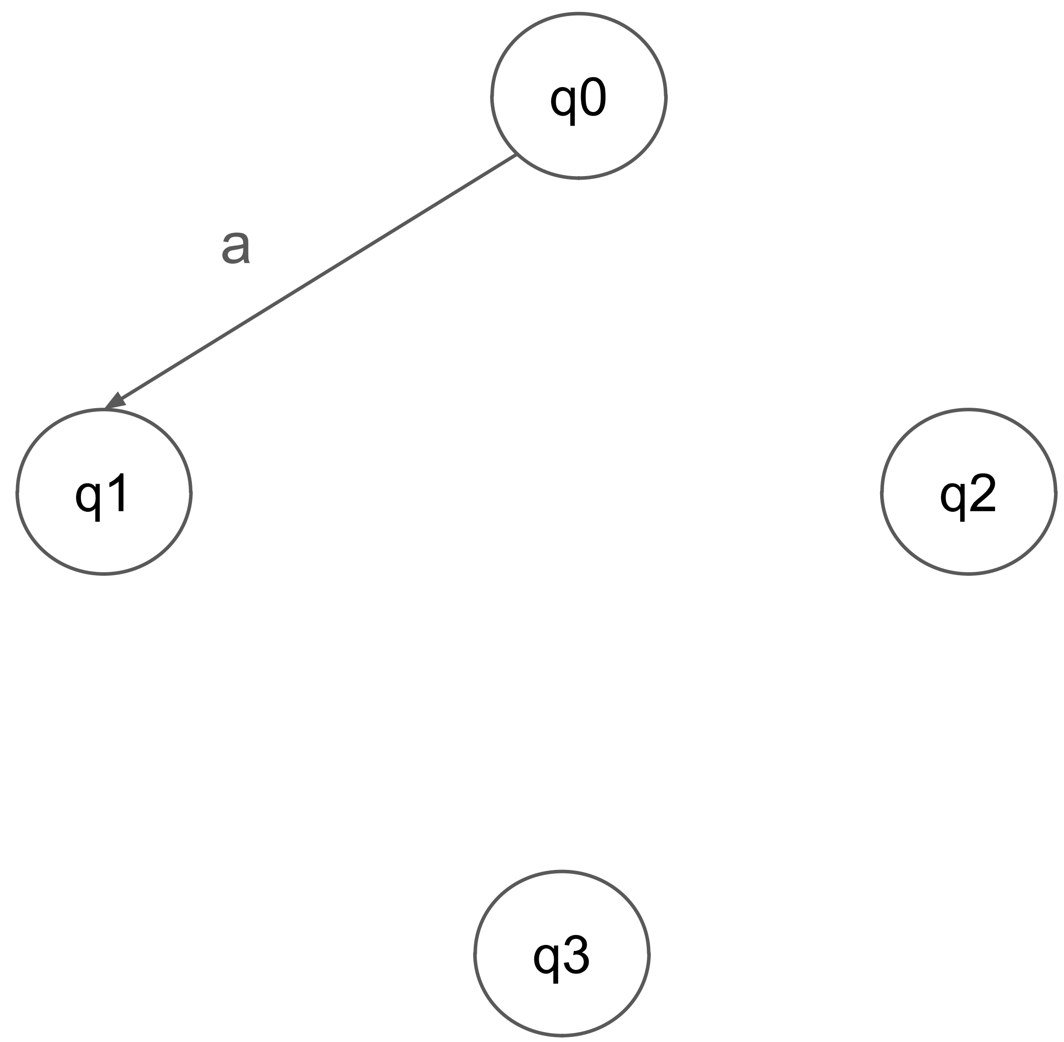


Solution

ab^{2k+1} , $k \geq 0$

States: q0, q1, q2, q3

Alphabets: {a, b}

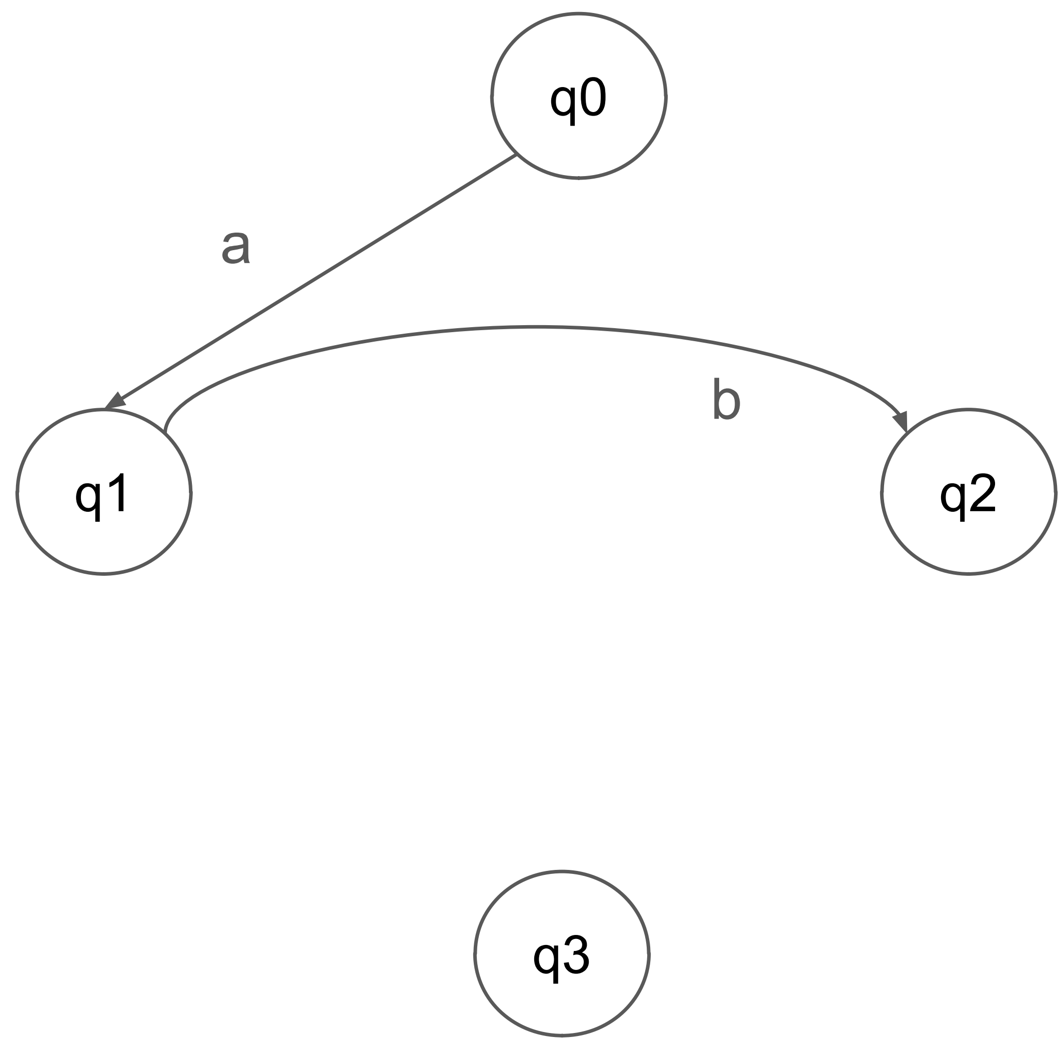


Solution

ab^{2k+1} , $k \geq 0$

States: Q0, Q1, Q2, Q3

Alphabets: a, b

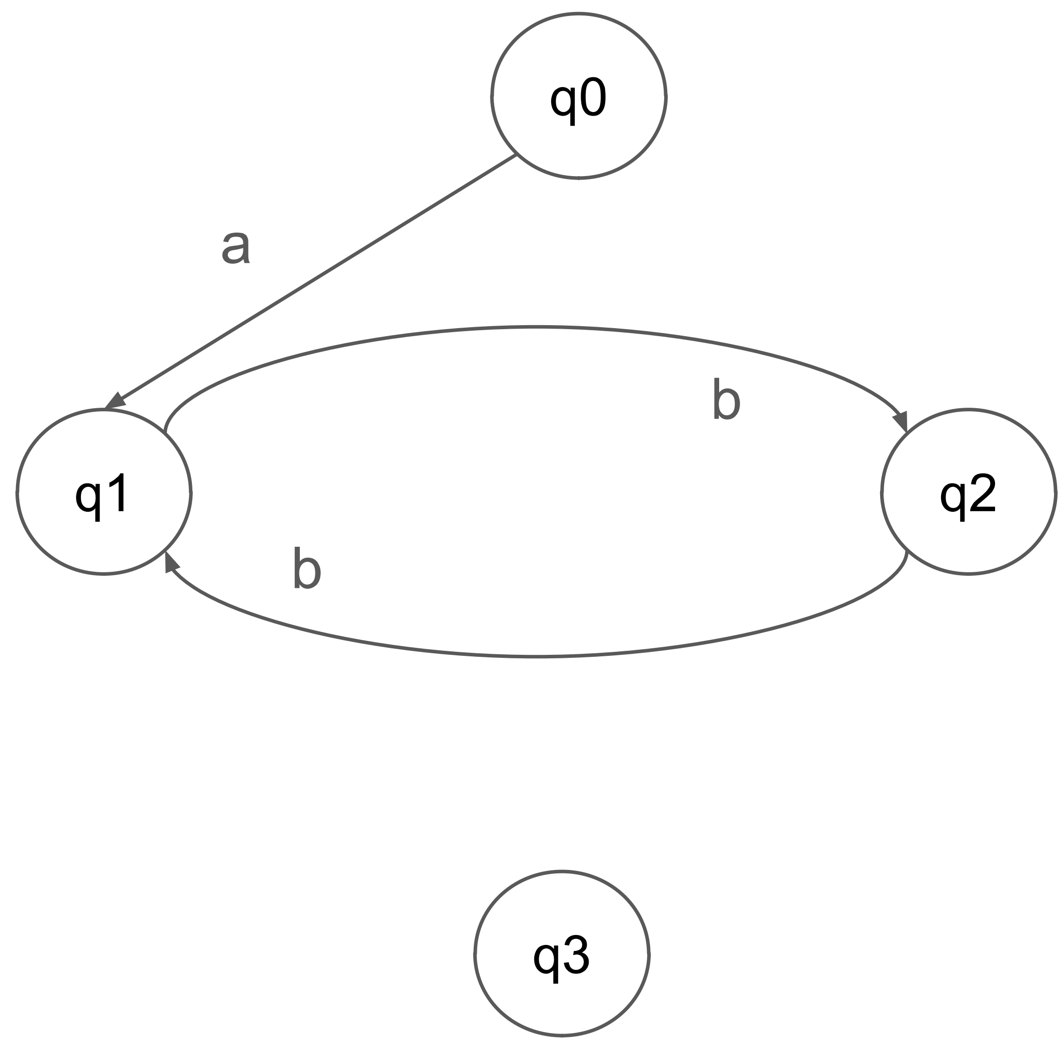


Solution

ab^{2k+1} , $k \geq 0$

States: Q0, Q1, Q2, Q3

Alphabets: a, b

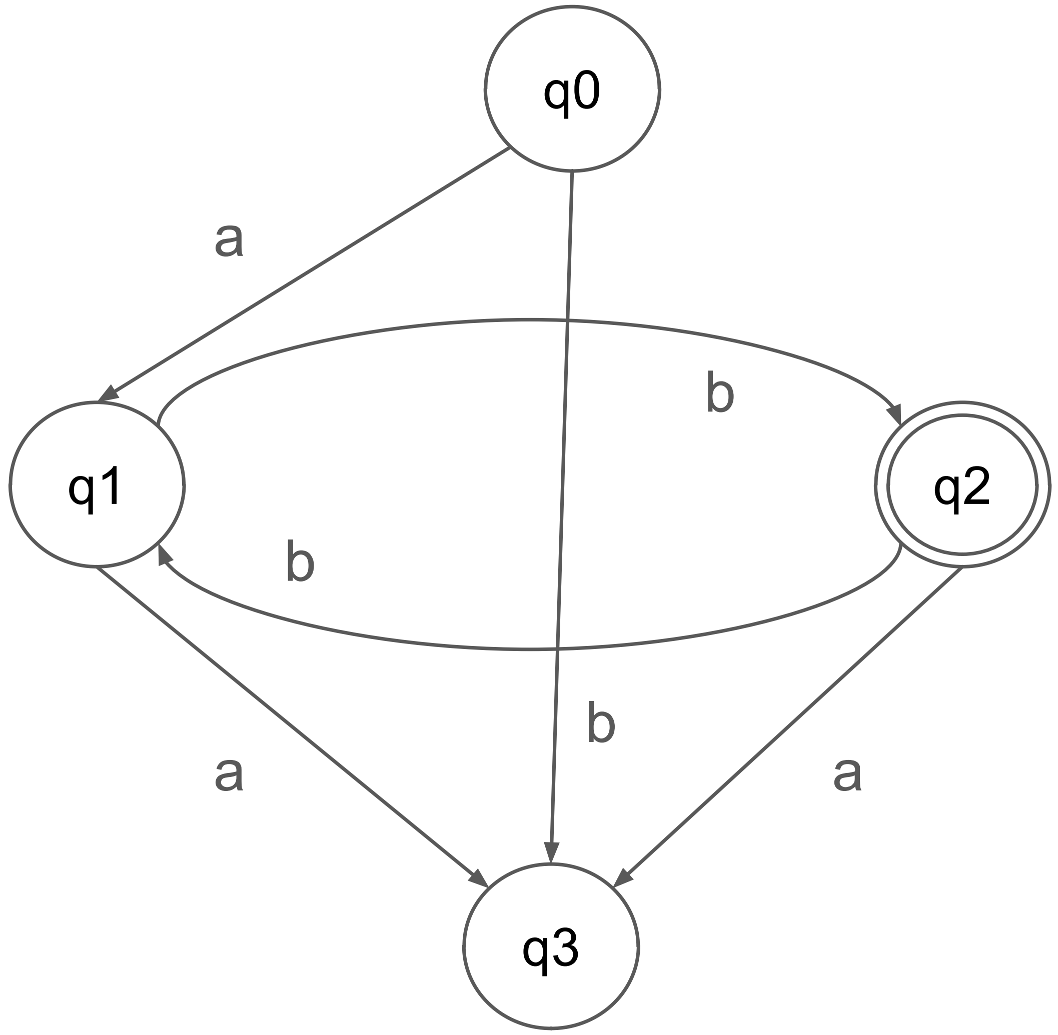


Solution

$$ab^{2k+1}\ ,\ \ k\geq 0$$

States: Q0, Q1, Q2, Q3

Alphabets: a, b

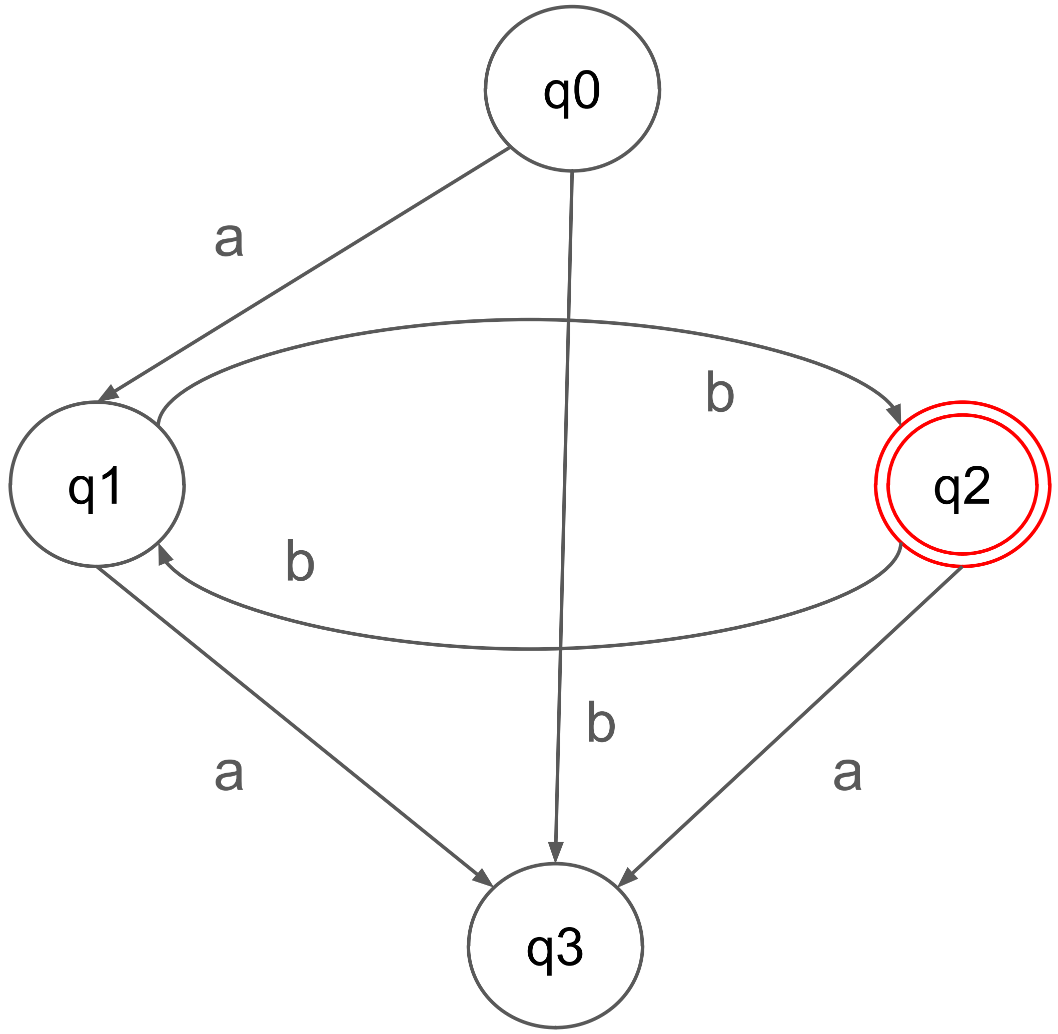


Solution

ab^{2k+1} , $k \geq 0$

States: Q0, Q1, Q2, Q3

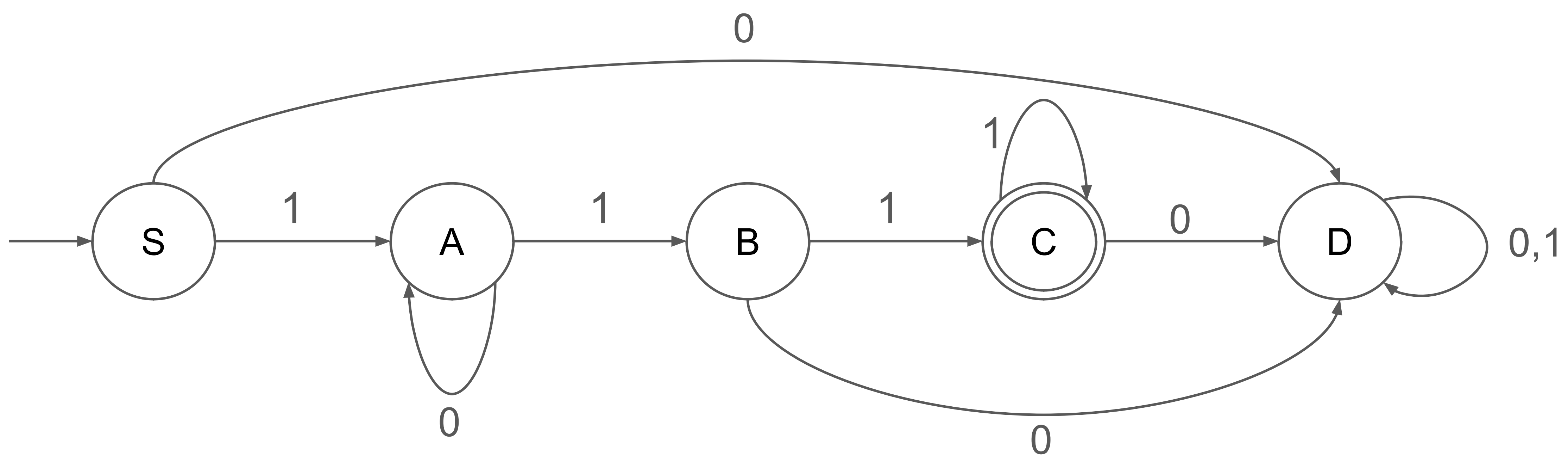
Alphabets: a, b



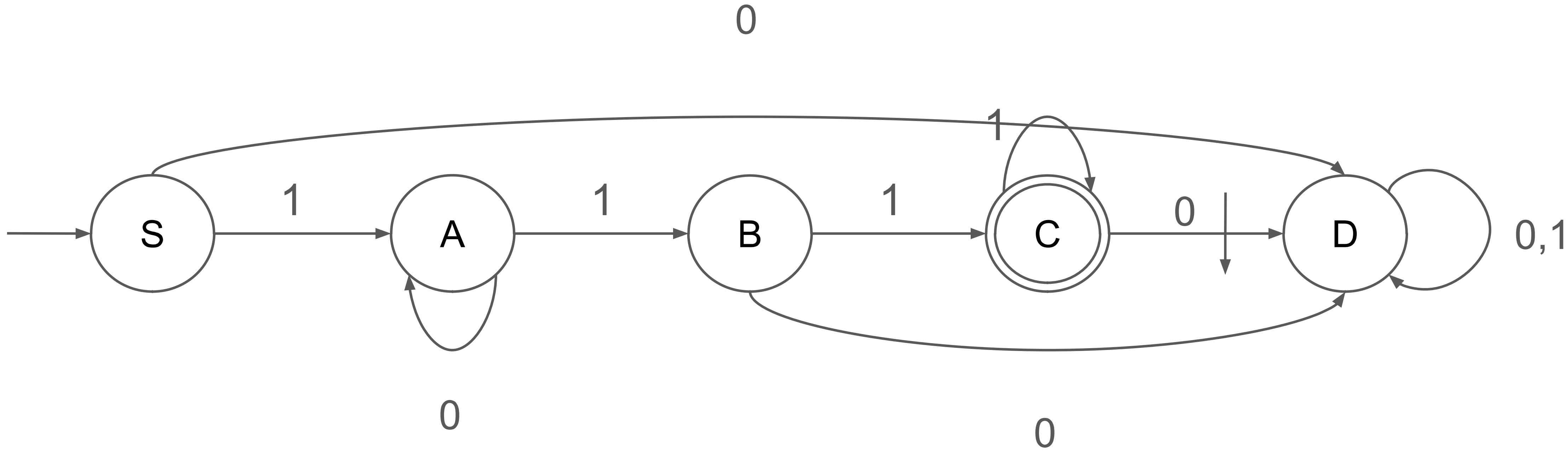
ab(bb)*

Question

Find the Grammar and regular expression of the given DFA automata below.



Solution



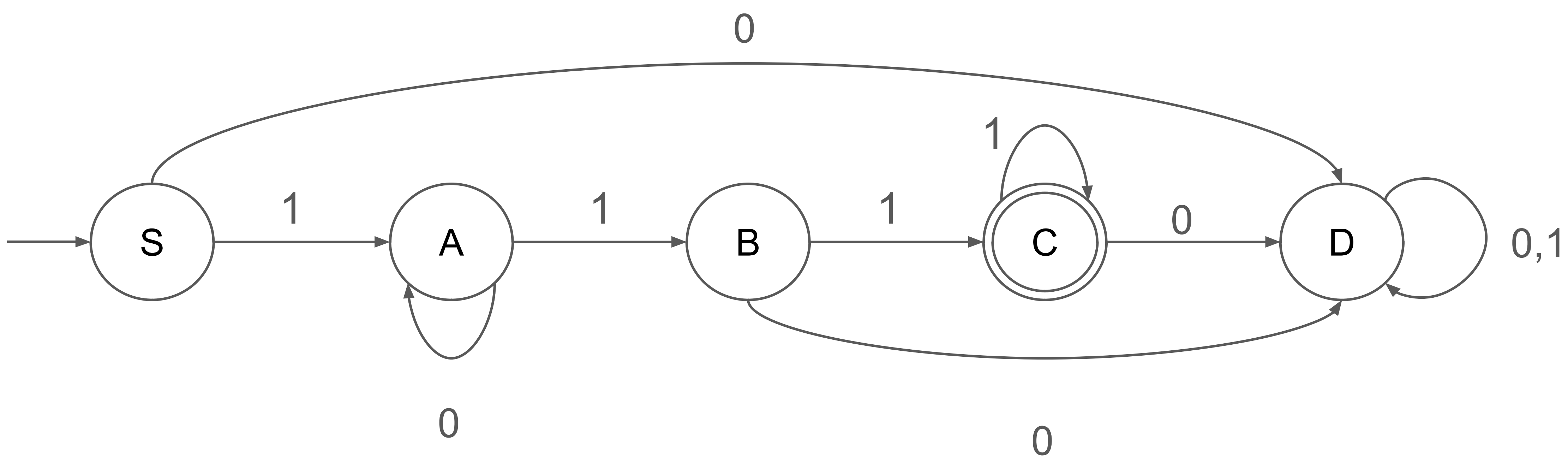
S→1A

A→0A|1B

B→1C

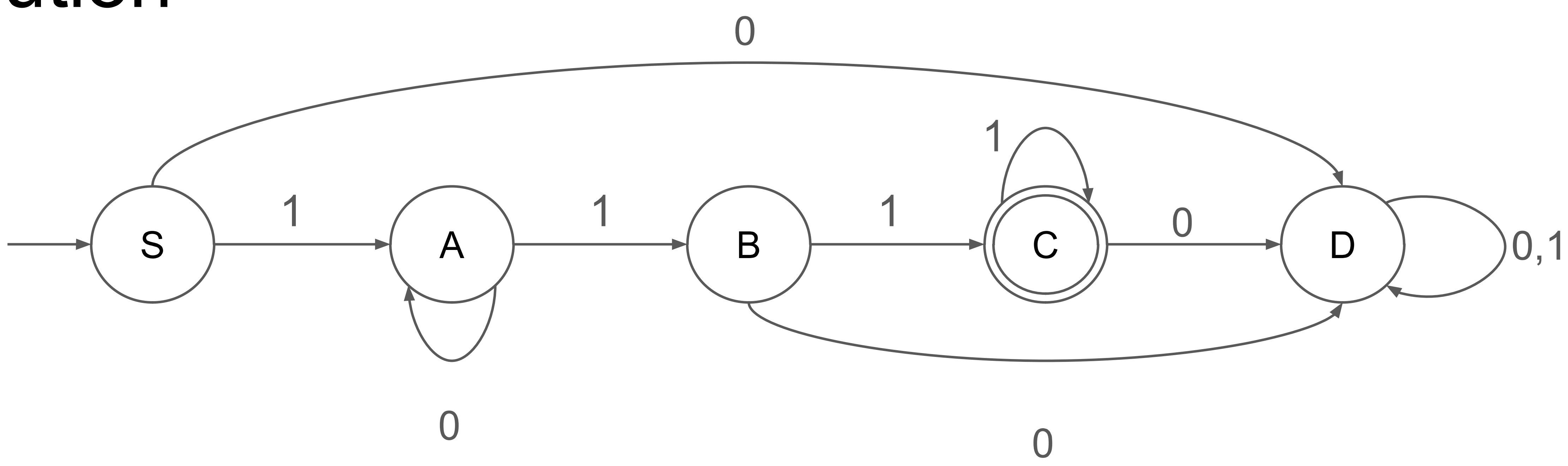
C→1C

Solution



S→1A
A→0A|1B
B→1C
C→1C

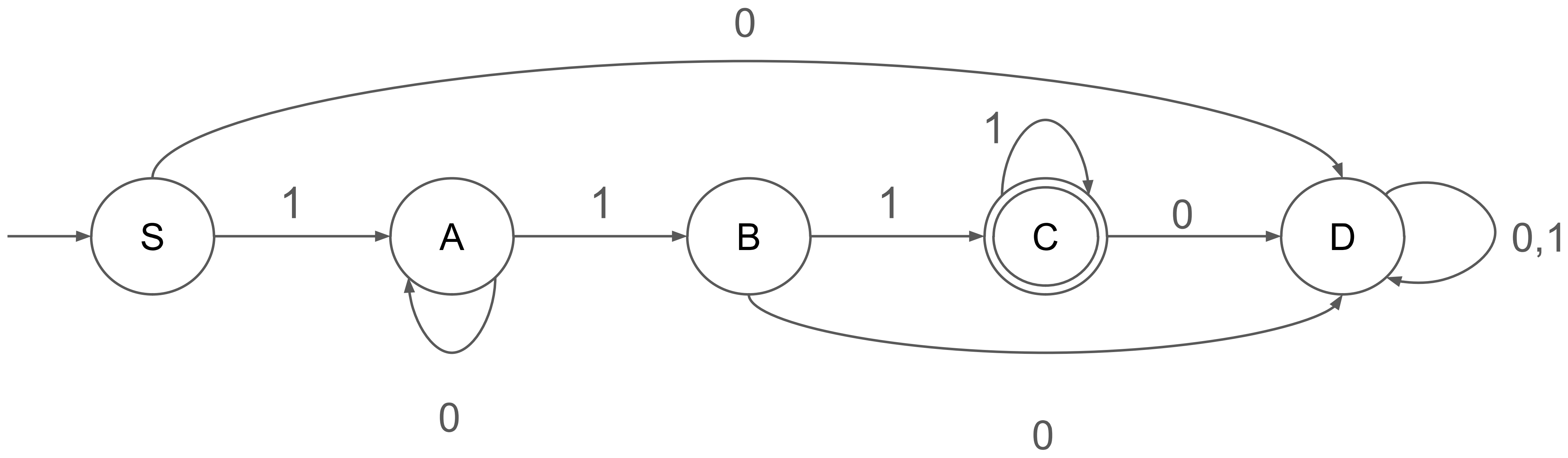
Solution



S→1A
A→0A|1B
B→1C
C→1C

Regular Expression?

Solution



S→1A
A→0A|1B
B→1C
C→1C

Regular Expression:
10*111*

Question

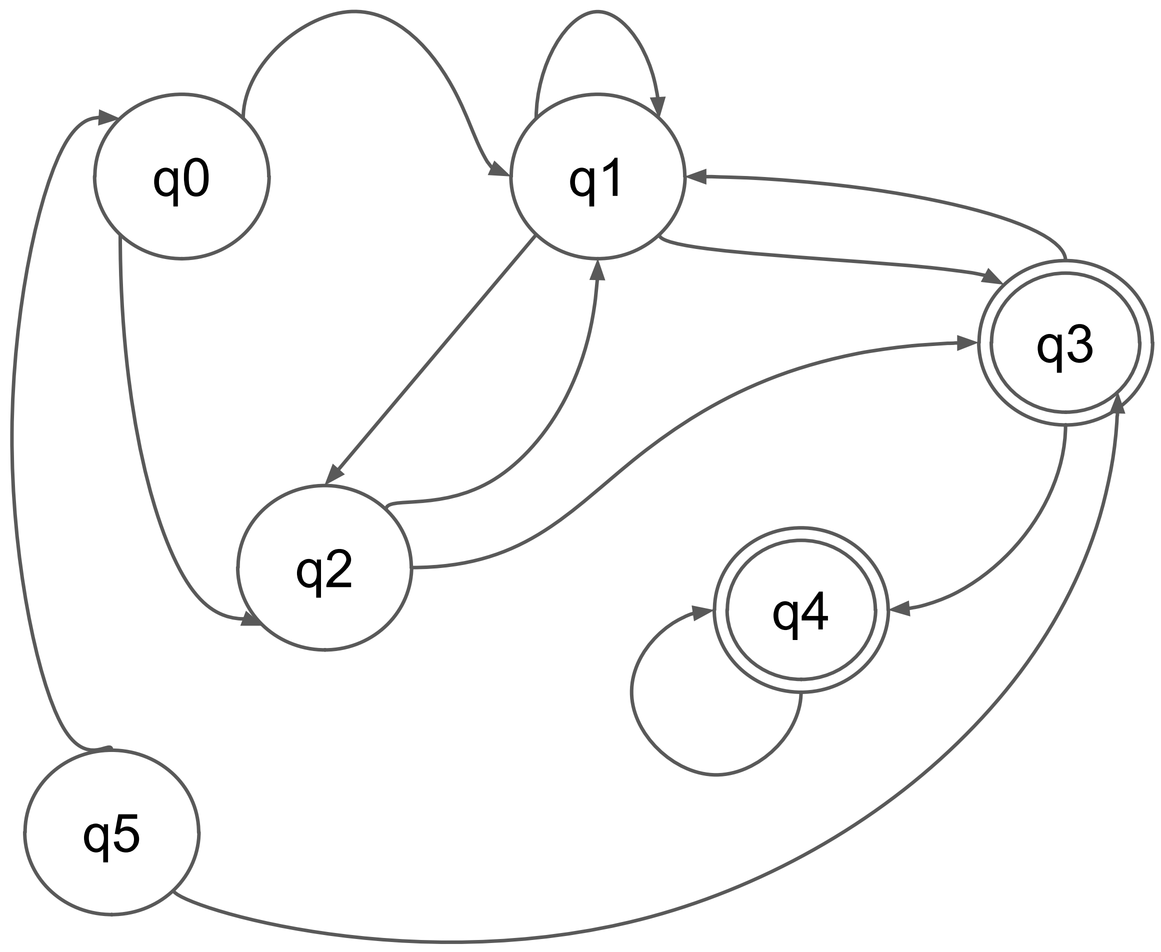
For the given transition table:

A. Draw the Deterministic Finite Automaton (DFA) based on the transitions and output.
B. Simplify the dependency table. Show all steps clearly.

State	Input a	Input b	Output
Q0	Q1	Q2	0
Q1	Q3	Q2	0
Q2	Q1	Q3	0
Q3	Q4	Q1	1
Q4	Q4	Q4	1
Q5	Q0	Q3	0

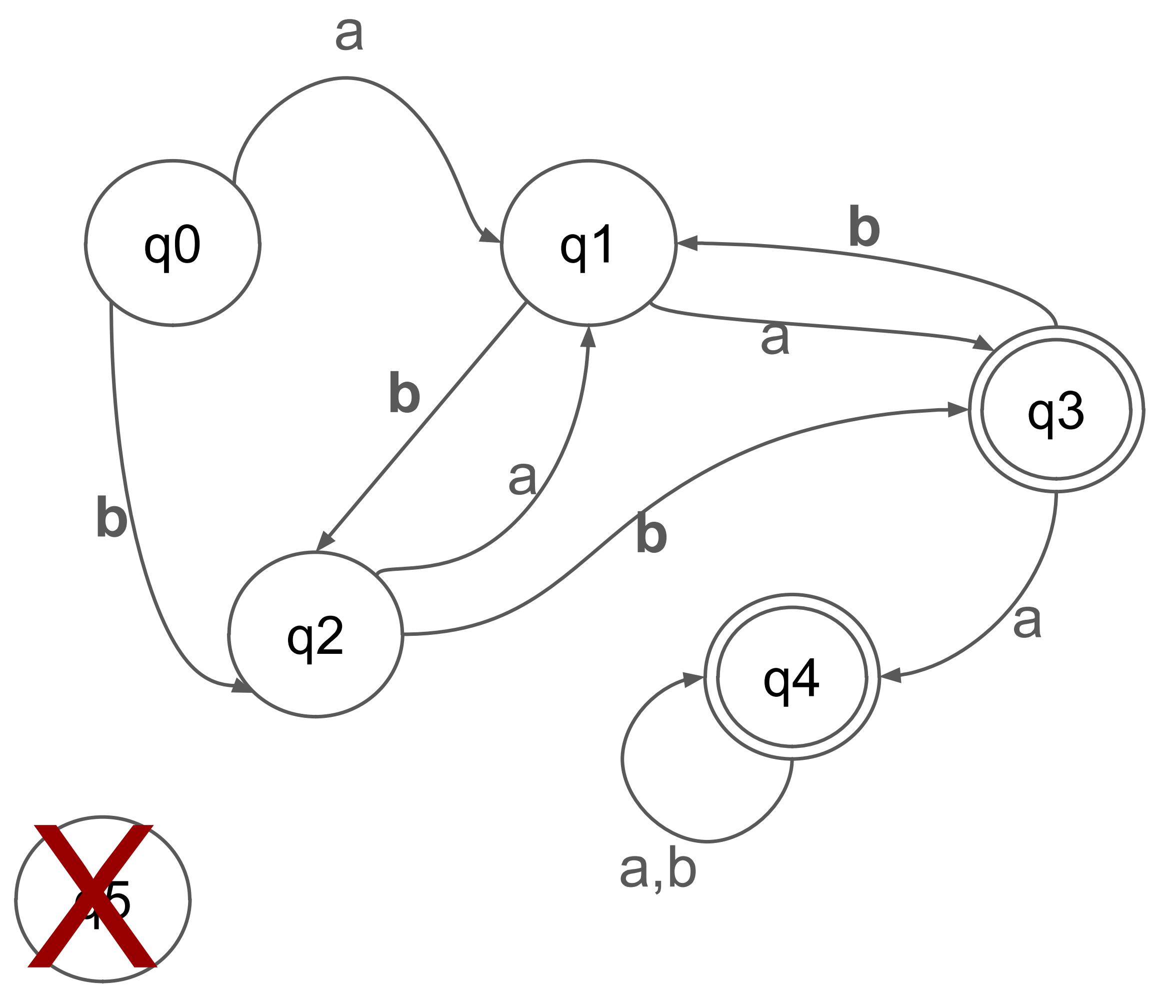
Solution

State	Input a	Input b	Output
Q0	Q1	Q2	0
Q1	Q3	Q2	0
Q2	Q1	Q3	0
Q3	Q4	Q1	1
Q4	Q4	Q4	1
Q5	Q0	Q3	0

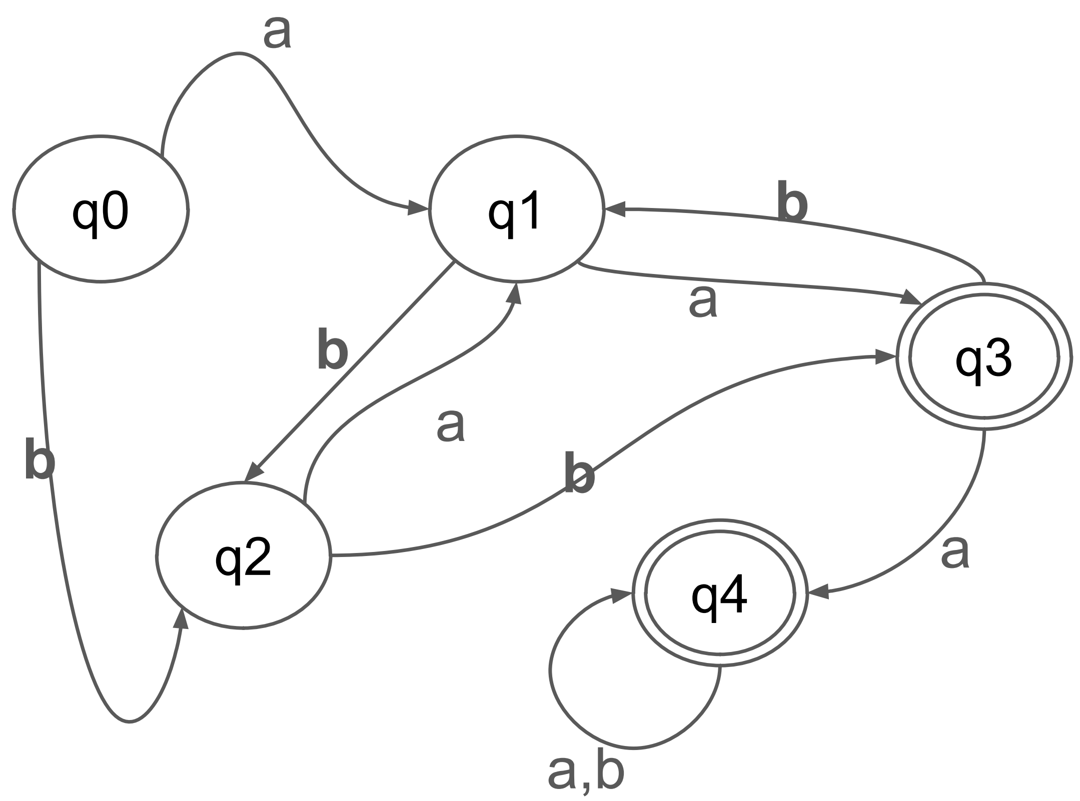


Solution

State	Input a	Input b	Output
Q0	Q1	Q2	0
Q1	Q3	Q2	0
Q2	Q1	Q3	0
Q3	Q4	Q1	1
Q4	Q4	Q4	1
Q5	Q0	Q3	0



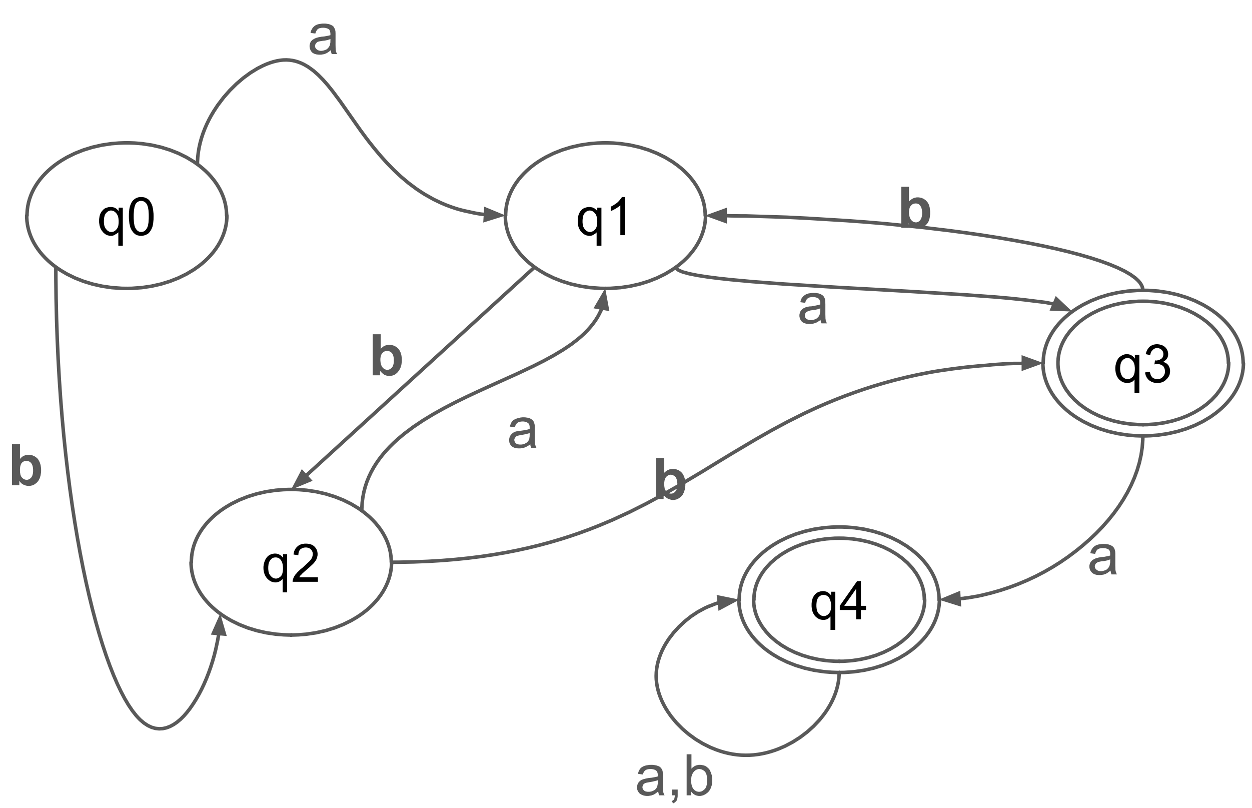
Solution



State	Input a	Input b	Output
Q0	Q1	Q2	0
Q1	Q3	Q2	0
Q2	Q1	Q3	0
Q3	Q4	Q1	1
Q4	Q4	Q4	1
Q5	Q0	Q3	0

	Q0	**Q1**	**Q2**	**Q3**	**Q4**
Q0	—				
Q1		—			
Q2			—		
Q3				—	
Q4					—

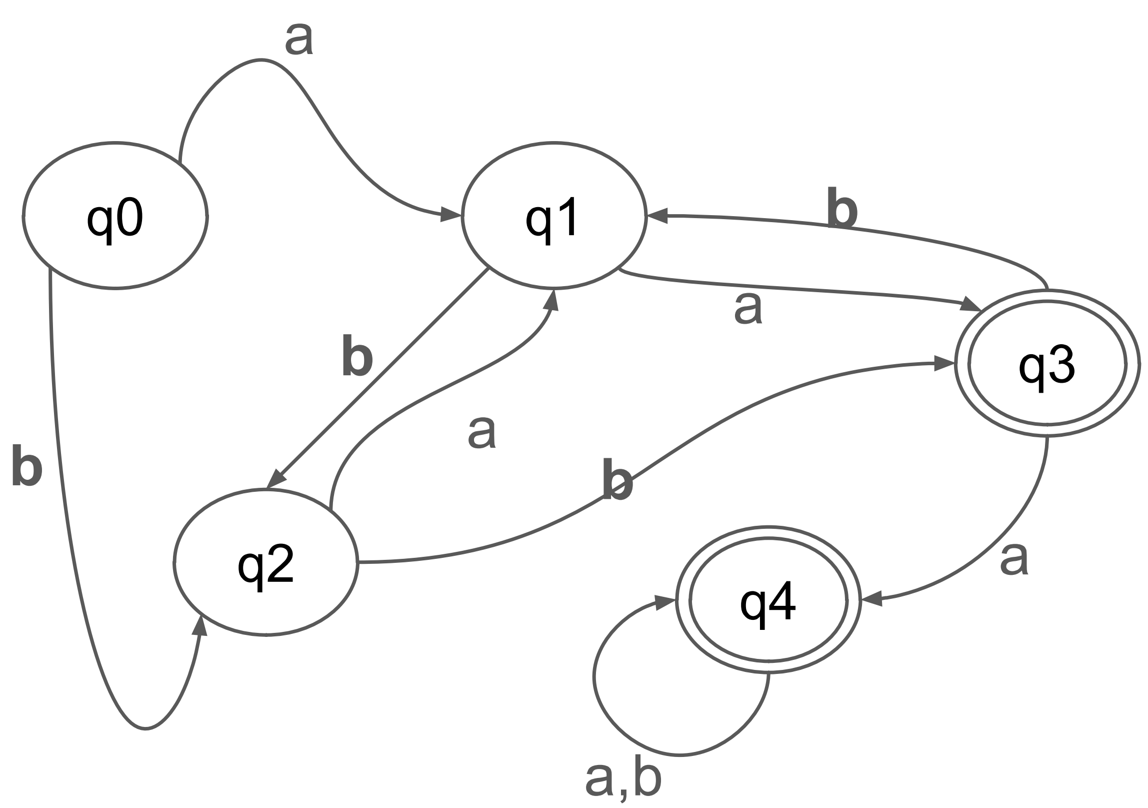
Solution



State	Input a	Input b	Output
Q0	Q1	Q2	0
Q1	Q3	Q2	0
Q2	Q1	Q3	0
Q3	Q4	Q1	1
Q4	Q4	Q4	1
~~Q5~~	~~Q0~~	~~Q3~~	~~0~~

	Q0	Q1	Q2	Q3	Q4
Q0	—	✗	✗	✗	✗
Q1		—	✗	✗	✗
Q2			—	✗	✗
Q3				—	✗
Q4					—

Solution



State	a	b	Output
Q0	Q1	Q2	0
Q1	Q3	Q2	0
Q2	Q1	Q3	0
Q3	Q4	Q1	1
Q4	Q4	Q4	1

	Q0	Q1	Q2	Q3	Q4
Q0	—	✗	✗	✗	✗
Q1		—	✗	✗	✗
Q2			—	✗	✗
Q3				—	✗
Q4					—

Question

Design an NFA over the alphabet {a, b} that accepts all strings which:

- Contain at least three a,

  or

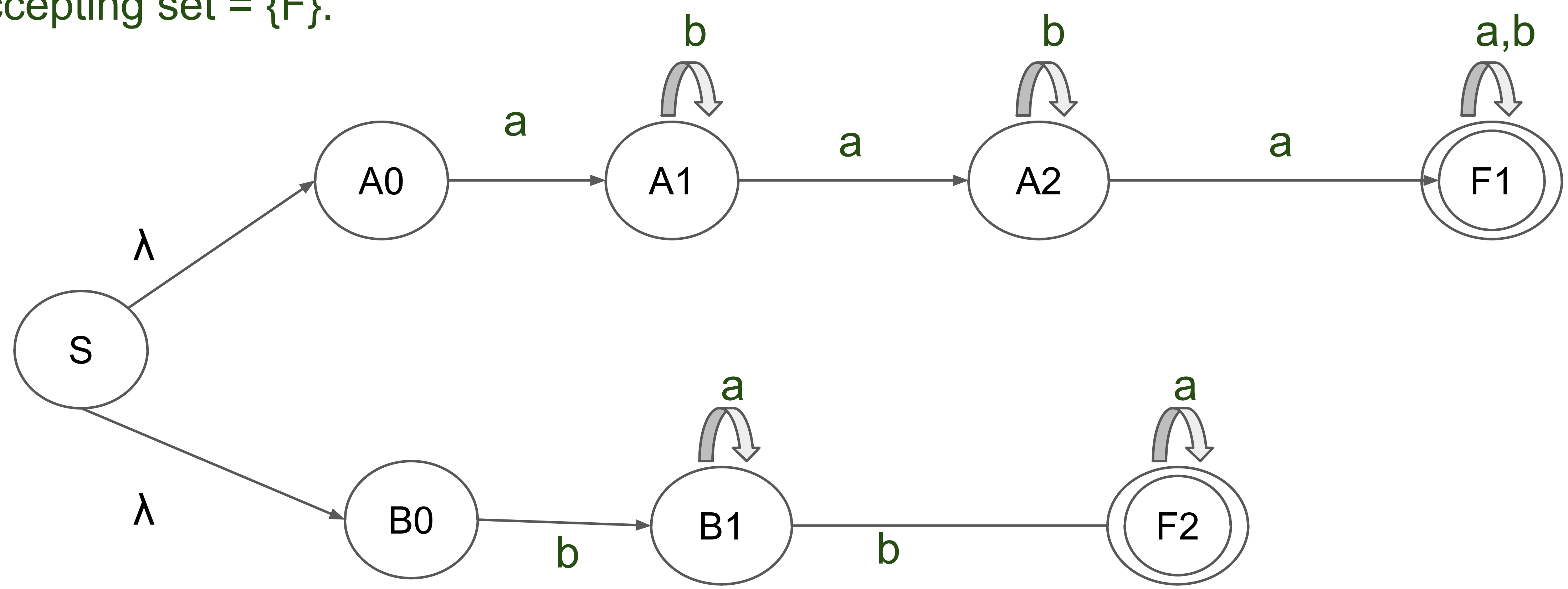
- Contain exactly two b.

Solution

Contain at least three “a” or Contain exactly two “b”.

Alphabet Σ = {a, b}.

Accepting set = {F}.



Question

Consider the following right-linear grammar over the alphabet {a, b}:

$S \rightarrow aA \mid bB \mid a$

$A \rightarrow aS \mid b$

$B \rightarrow aA \mid bS$

A. Design a Non-deterministic Finite Automaton (NFA) that accepts the same language.
B. Derive the regular expression corresponding to this language. Show your reasoning.

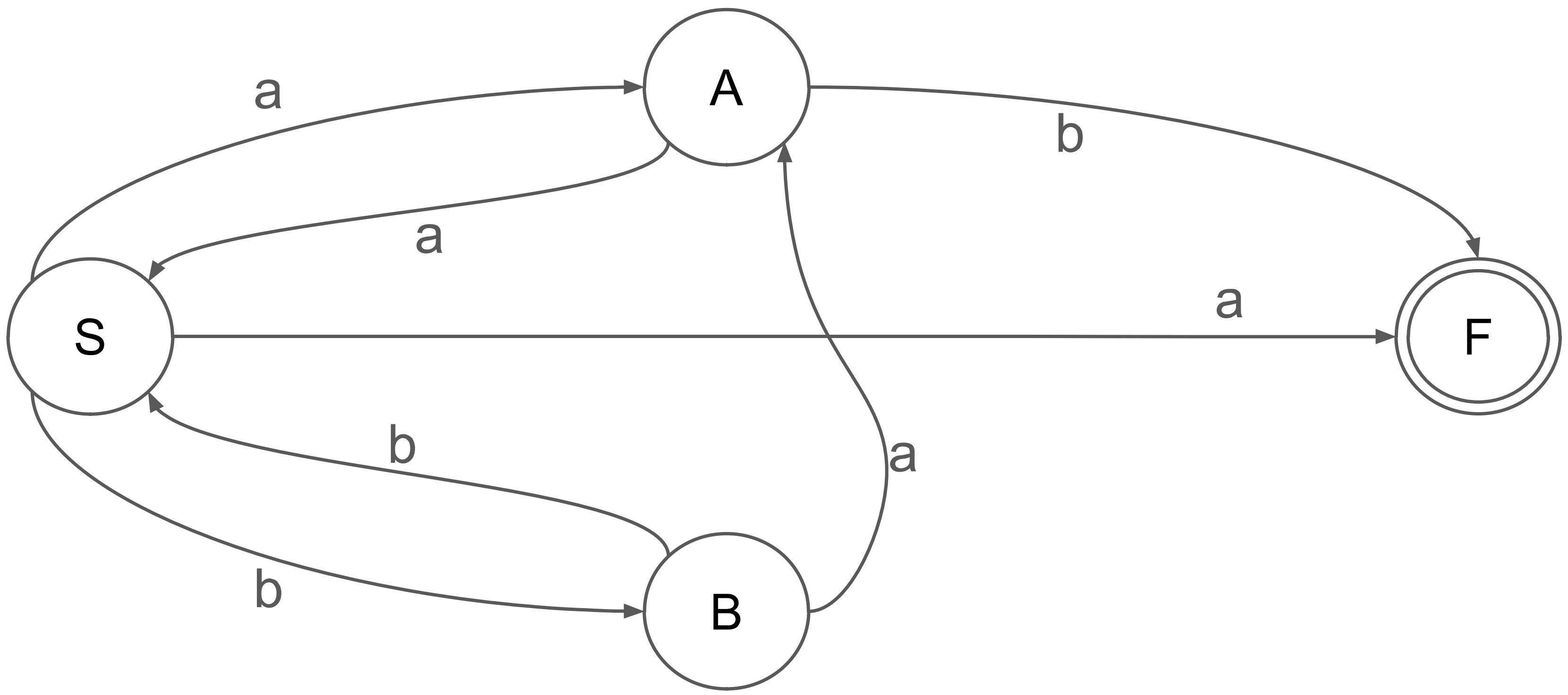
Solution

S → aA | bB | a

A → aS | b

B → aA | bS

A. Design a Non-deterministic Finite Automaton (NFA) that accepts the same language.



Solution

$S \rightarrow aA \mid bB \mid a$

$A \rightarrow aS \mid b$

$B \rightarrow aA \mid bS$

B. Derive the regular expression corresponding to this language. Show your reasoning.

$S = aA \cup bB \cup a$

$A = aS \cup b$

$B = aA \cup bS$

Solution

$S \rightarrow aA \mid bB \mid a$

$A \rightarrow aS \mid b$

$B \rightarrow aA \mid bS$

B. Derive the regular expression corresponding to this language. Show your reasoning.

Eliminating A and B

$S = aA \cup bB \cup a$

$A = aS \cup b$

$B = aA \cup bS$

$A = aS \cup b \Rightarrow B = a(aS \cup b) \cup bS = aaS \cup ab \cup bS$

$\Rightarrow S = a(aS \cup b) \cup b(aaS \cup ab \cup bS) \cup a$

$= (aa \cup baa \cup bb)S \cup (a \cup ab \cup bab)$

Solution

$S \rightarrow aA \mid bB \mid a$

$A \rightarrow aS \mid b$

$B \rightarrow aA \mid bS$

B. Derive the regular expression corresponding to this language. Show your reasoning.

Eliminating A and B

$S = aA \cup bB \cup a$

$A = aS \cup b$

$B = aA \cup bS$

$A = aS \cup b \Rightarrow B = a(aS \cup b) \cup bS = aaS \cup ab \cup bS$

$\Rightarrow S = a(aS \cup b) \cup b(aaS \cup ab \cup bS) \cup a$

$= (aa \cup baa \cup bb)S \cup (a \cup ab \cup bab)$

Apply Arden's rule

$R = (aa \cup baa \cup bb)$, $P = (a \cup ab \cup bab)$

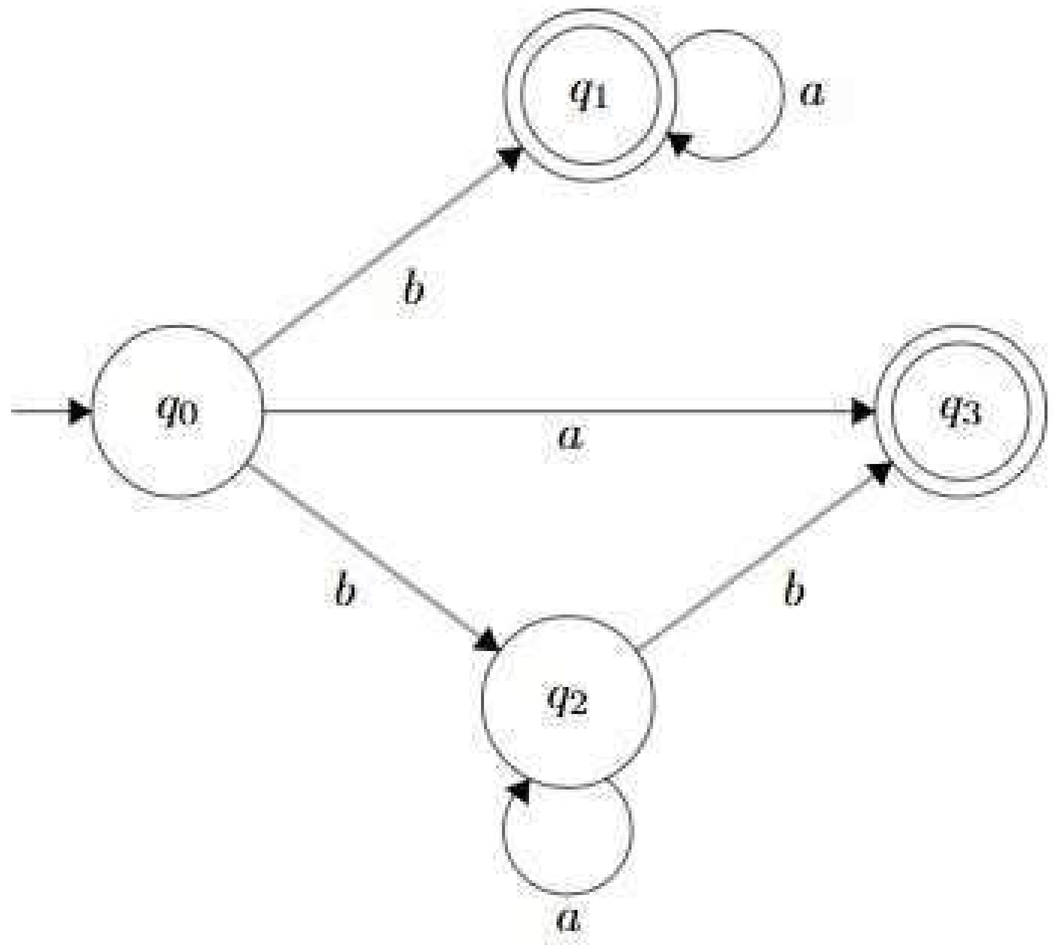
$\varepsilon \notin R$, then $\Rightarrow S = R^*P = (aa \cup baa \cup bb)^*(a \cup ab \cup bab)$

$\Rightarrow$ $(aa|bb|baa)^*(a|ab|bab)$

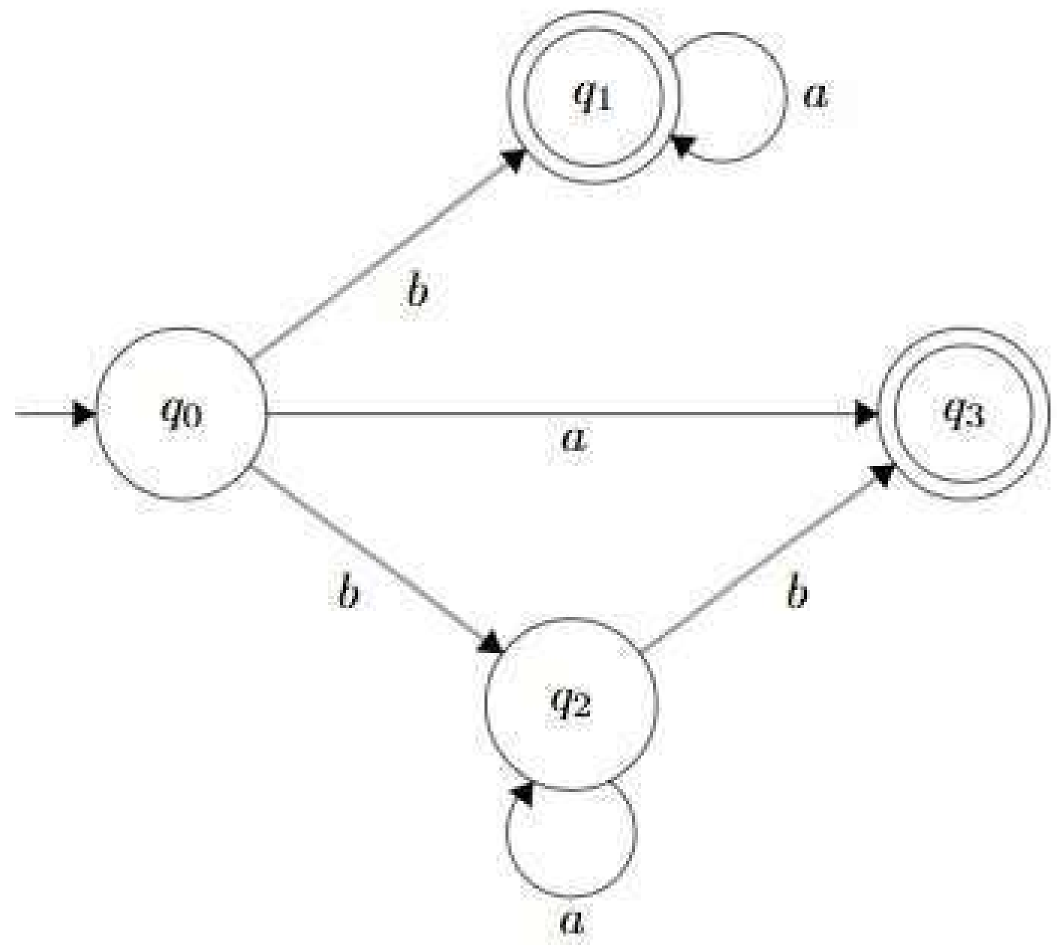
DFA-NFA Equivalency

Question

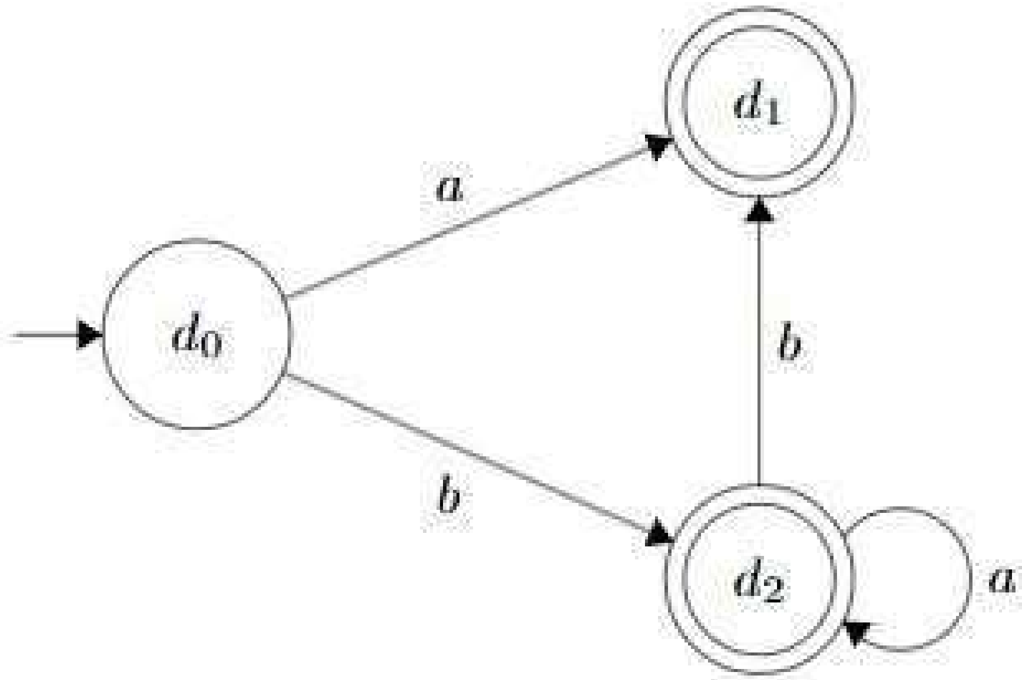
Find the DFA equivalent of the given NFA below.



New State	Next State with ‘a’	Next State with ‘b’
d_0 $\{q_0\}$	d_1 $\{q_3\}$	d_2 $\{q_1, q_2\}$
d_1 $\{q_3\}$	X	X
d_2 $\{q_1, q_2\}$	d_2 $\{q_1, q_2\}$	d_1 $\{q_3\}$



New State	Next State with ‘a’	Next State with ‘b’
d_0 $\{q_0\}$	d_1 $\{q_3\}$	d_2 $\{q_1, q_2\}$
d_1 $\{q_3\}$	X	X
d_2 $\{q_1, q_2\}$	d_2 $\{q_1, q_2\}$	d_1 $\{q_3\}$



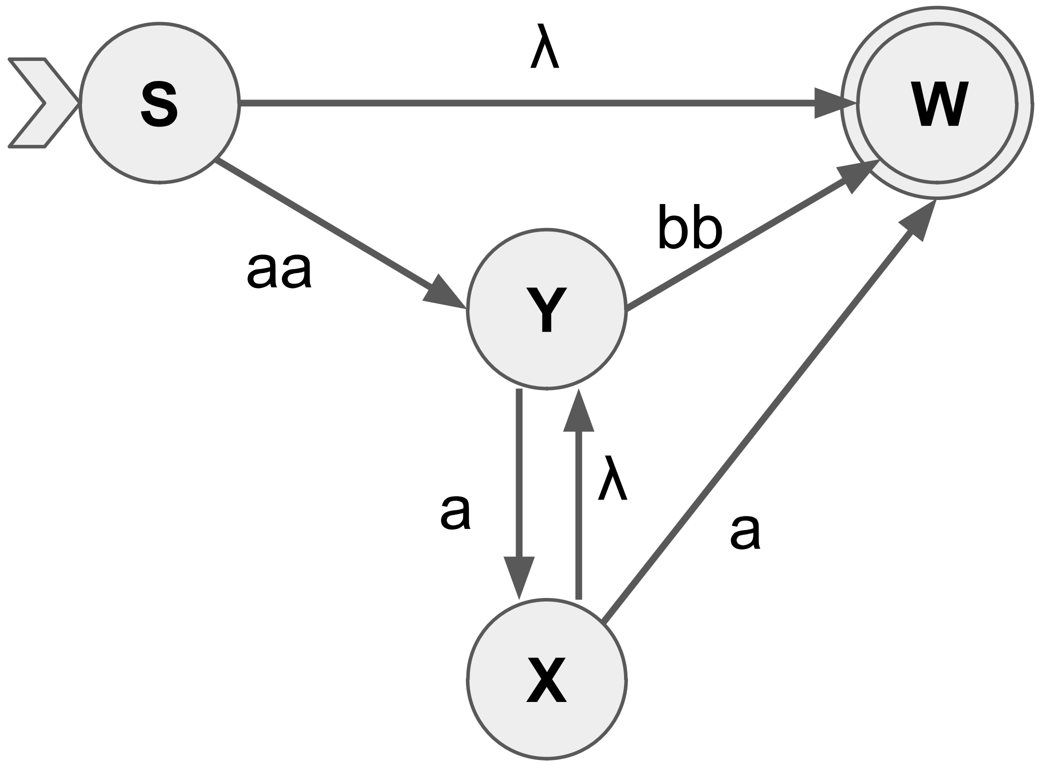
Question

S → aaY | W
X → Y ∣ aW
Y → aX | bbW
W → λ

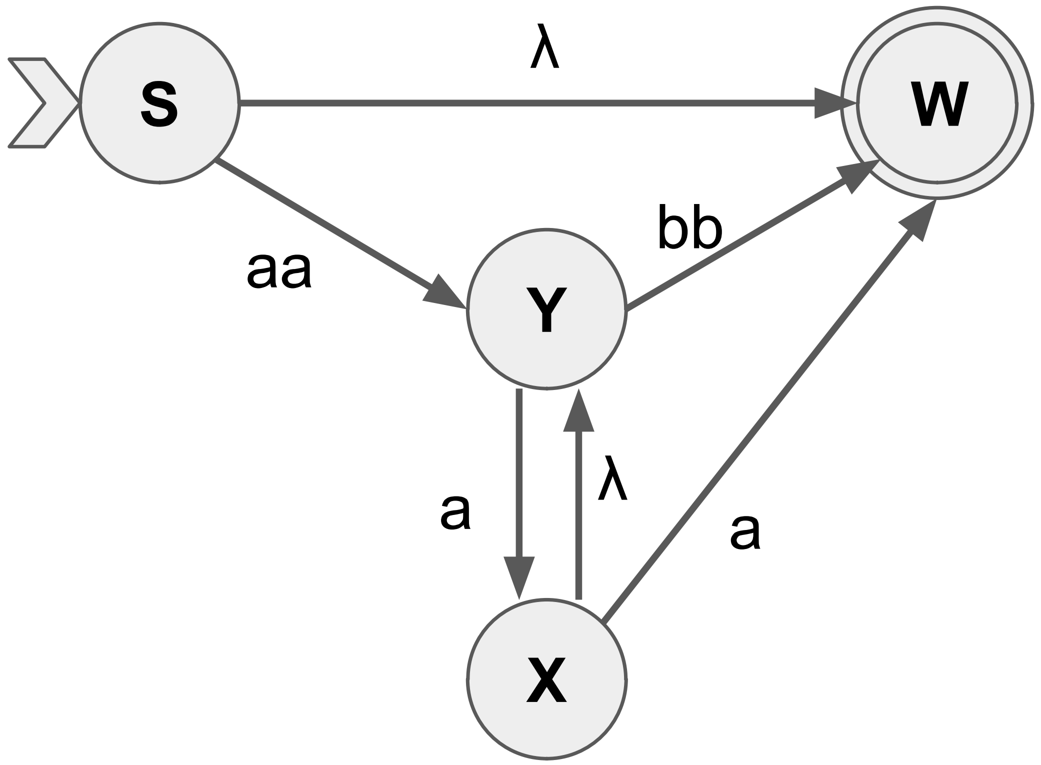
For the given grammar above;

- Draw the NFA that accepts the same language.
- Heuristically, find the regular expression.
- Find the DFA equivalent.

S → aaY | W
X → Y ∣ aW
Y → aX | bbW
W → λ

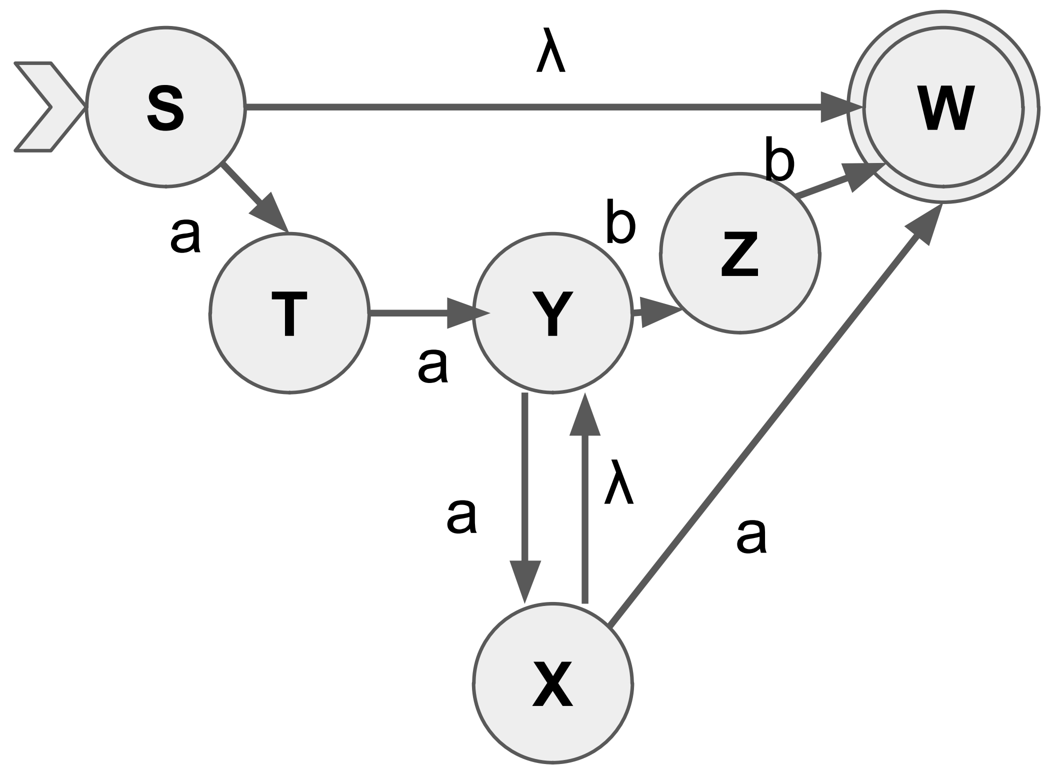


S → aaY | W
X → Y | aW
Y → aX | bbW
W → λ

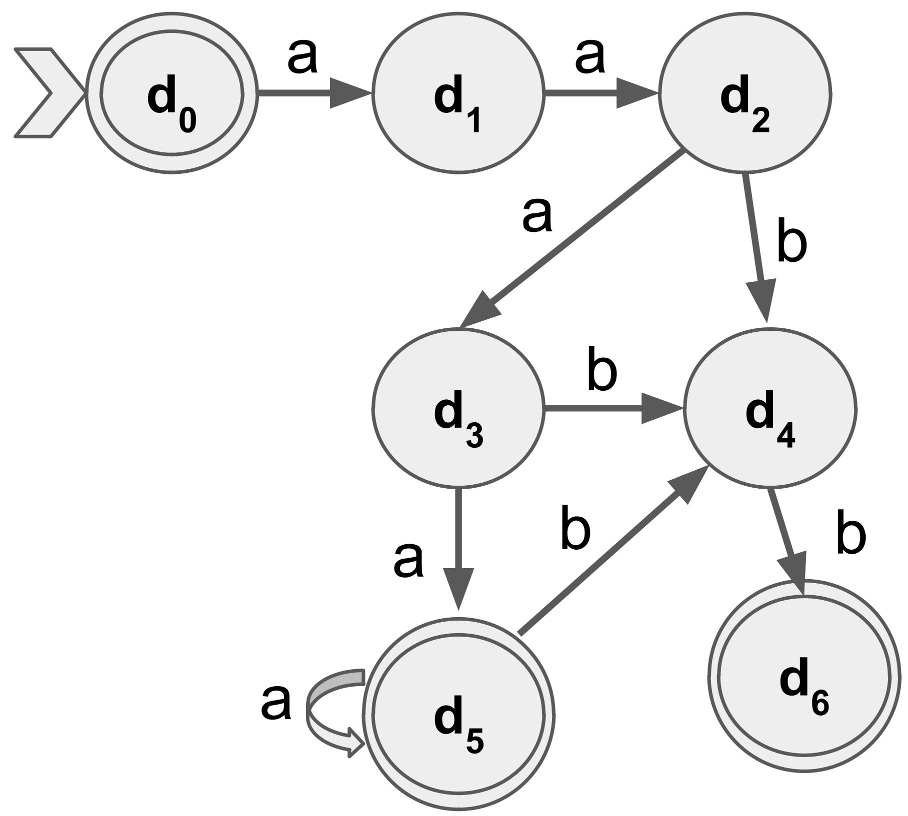


$\boxed{\lambda \mid aaa^*(aa|bb)}$

New State	Next State with ‘a’	Next State with ‘b’
d_0 {S,W}	d_1 {T}	X
d_1 {T}	d_2 {Y}	X
d_2 {Y}	d_3 {X,Y}	d_4 {Z}
d_3 {X,Y}	d_5 {X,Y,W}	d_4 {Z}
d_4 {Z}	X	d_6 {W}
d_5 {X,Y,W}	d_5 {X,Y,W}	d_4 {Z}
d_6 {W}	X	X



New State	Next State with ‘a’	Next State with ‘b’
d_0 {S,W}	d_1 {T}	X
d_1 {T}	d_2 {Y}	X
d_2 {Y}	d_3 {X,Y}	d_4 {Z}
d_3 {X,Y}	d_5 {X,Y,W}	d_4 {Z}
d_4 {Z}	X	d_6 {W}
d_5 {X,Y,W}	d_5 {X,Y,W}	d_4 {Z}
d_6 {W}	X	X



Question

S $\rightarrow$ 0A | 1B | 0S
A $\rightarrow$ 1B | 0S | 1
B $\rightarrow$ 1B | 0C | 0 | 1
C $\rightarrow$ 1C | 1

For the given grammar above;

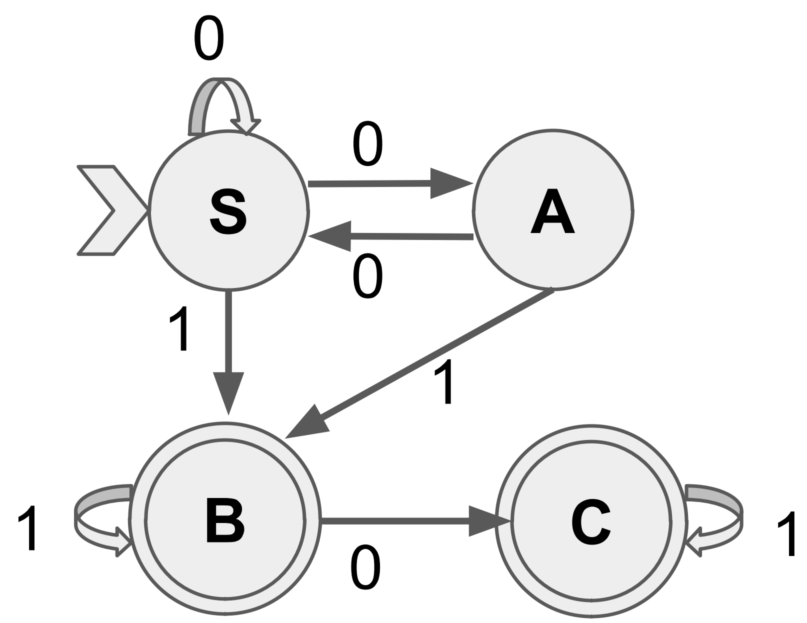
- Draw the NFA that accepts the same language.
- Find the DFA equivalent.

S → 0A | 1B | 0S

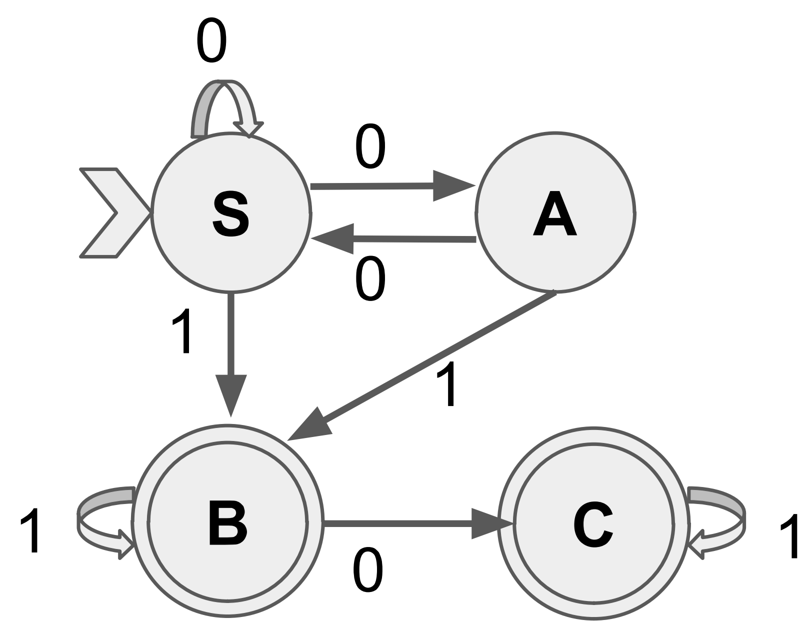
A → 1B | 0S | 1

B → 1B | 0C | 0 | 1

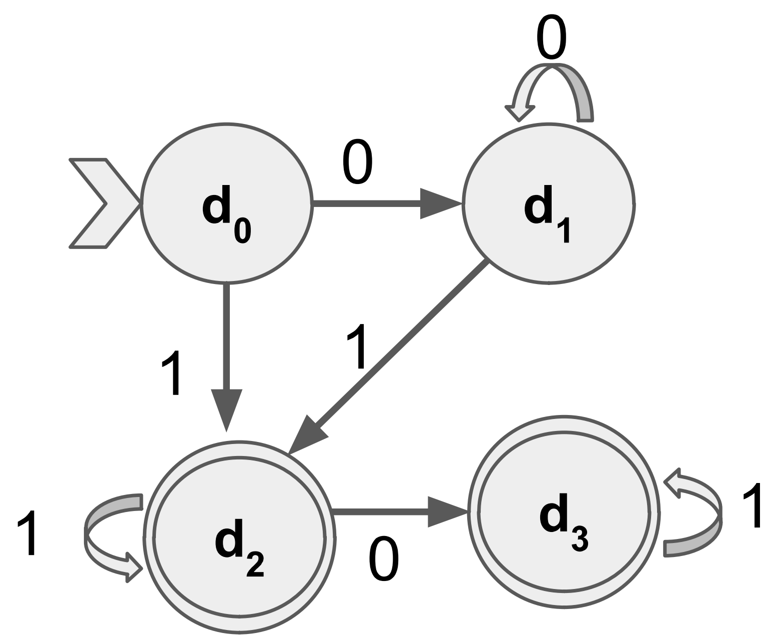
C → 1C | 1



New State	Next State with '0'	Next State with '1'
d_0 {S}	d_1 {S,A}	d_2 {B}
d_1 {S,A}	d_1 {S,A}	d_2 {B}
d_2 {B}	d_3 {C}	d_2 {B}
d_3 {C}	X	d_3 {C}



New State	Next State with ‘0’	Next State with ‘1’
d_0 {S}	d_1 {S,A}	d_2 {B}
d_1 {S,A}	d_1 {S,A}	d_2 {B}
d_2 {B}	d_3 {C}	d_2 {B}
d_3 {C}	X	d_3 {C}

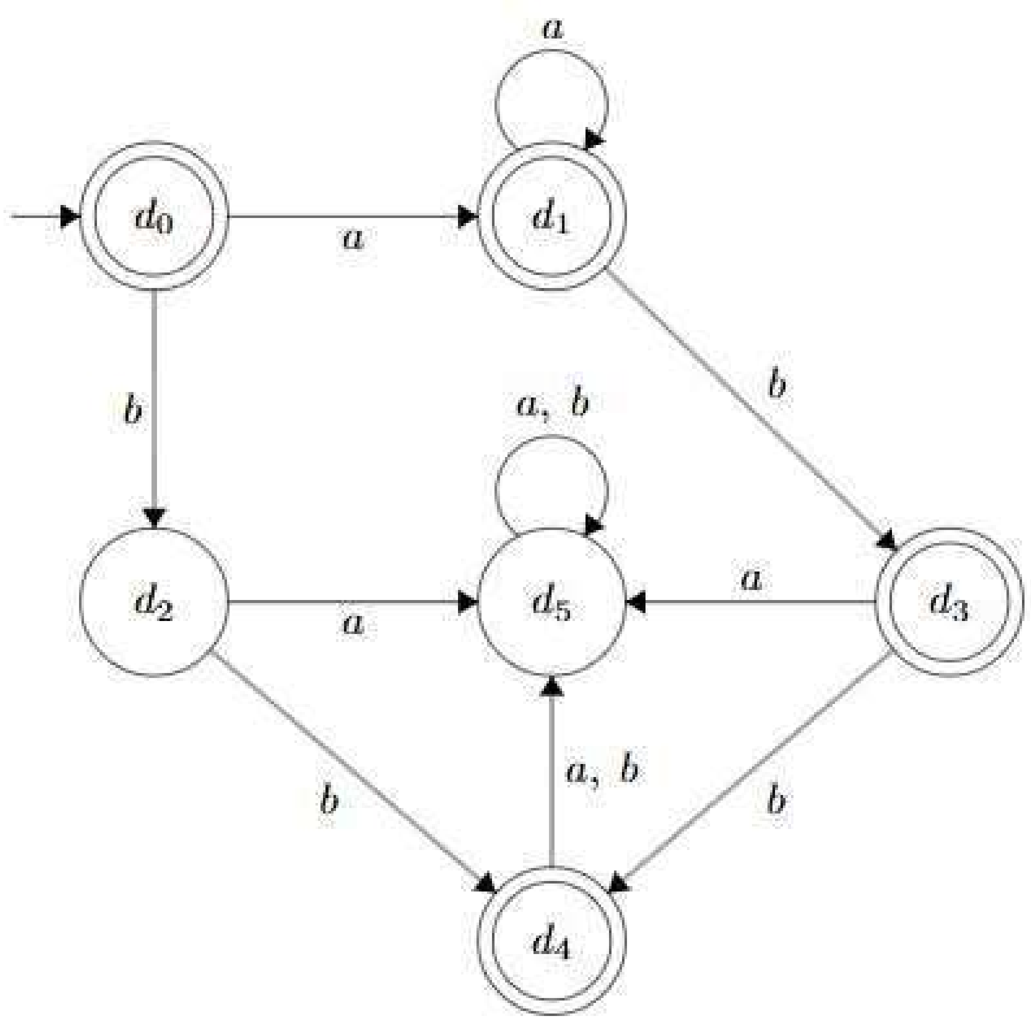


Recognizing Regular Expressions

Question

For the given automata;

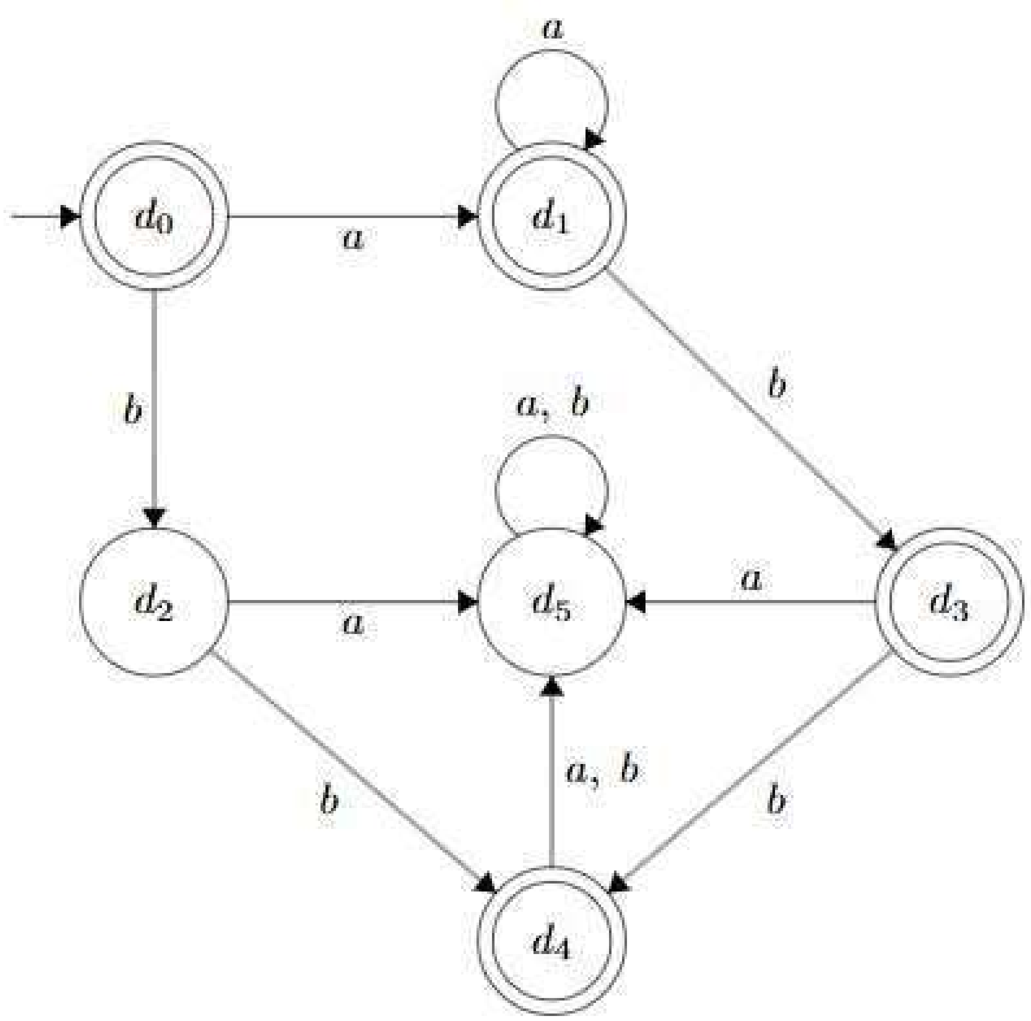
- Heuristically, find it’s regular expression.
- Systematically create the regular expression and compare it to the previously defined expression.



No loop, we can check paths to accepting states

- For $d_0 \rightarrow \lambda$
- For $d_1 \rightarrow a^+$
- For $d_3 \rightarrow a^+b$
- For $d_4 \rightarrow$ bb or a^+bb

Thus we can state: RegEx $\rightarrow a^* \vee a^+b \vee a^*bb$

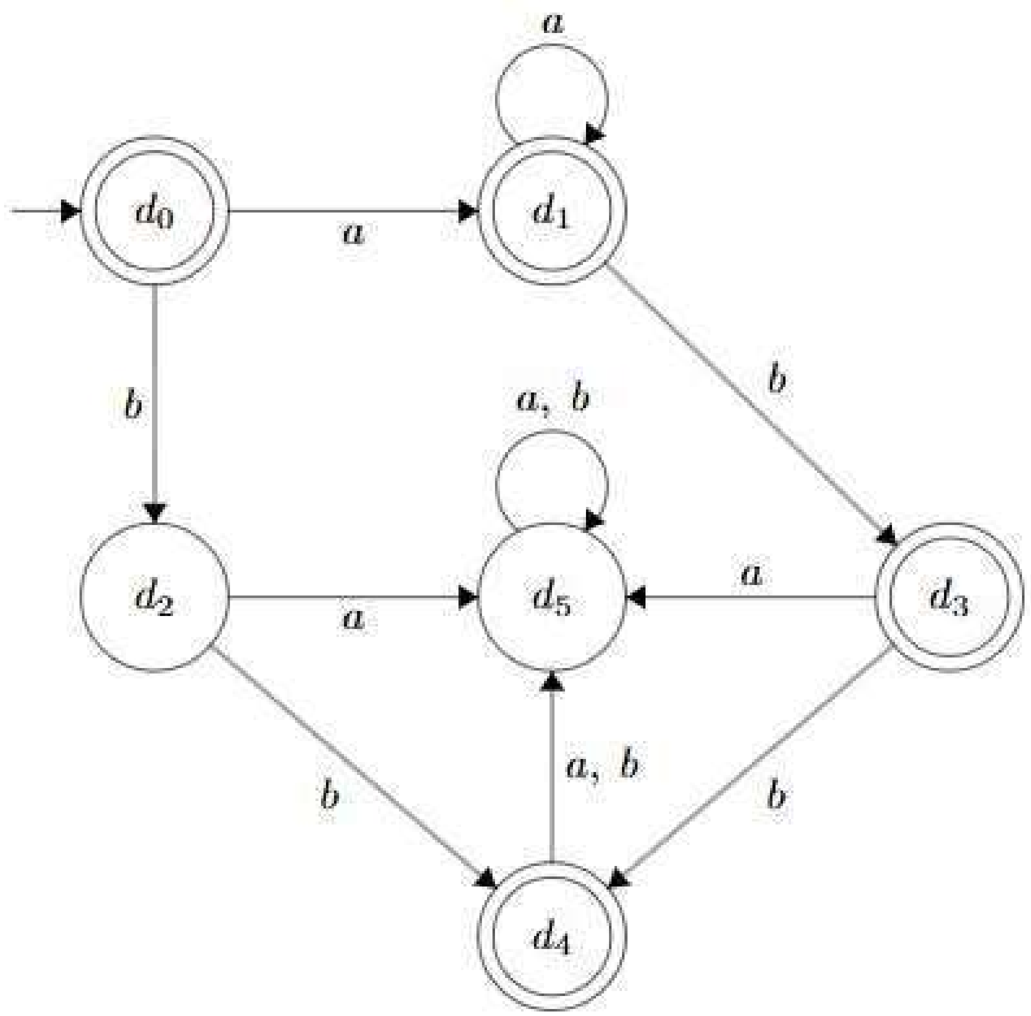


- $d_0 = \lambda$
- $d_1 = d_0 a \vee d_1 a$
- $d_2 = d_0 b$
- $d_3 = d_1 b$
- $d_4 = d_2 b \vee d_3 b$

ReGex will be $d_0 \vee d_1 \vee d_3 \vee d_4$

- $d_0 \rightarrow \lambda = \lambda$
- $d_1 \rightarrow a \vee d_1 a = aa^*$
- $d_2 \rightarrow d_0 b = b$
- $d_3 \rightarrow d_1 b = aa^*b$
- $d_4 \rightarrow d_2 b \vee d_3 b = bb \vee aa^*bb$

$d_0 \vee d_1 \vee d_3 \vee d_4 = \lambda \vee a^+ \vee a^+b \vee bb \vee a^+bb = a^* \vee a^+b \vee a^*bb$

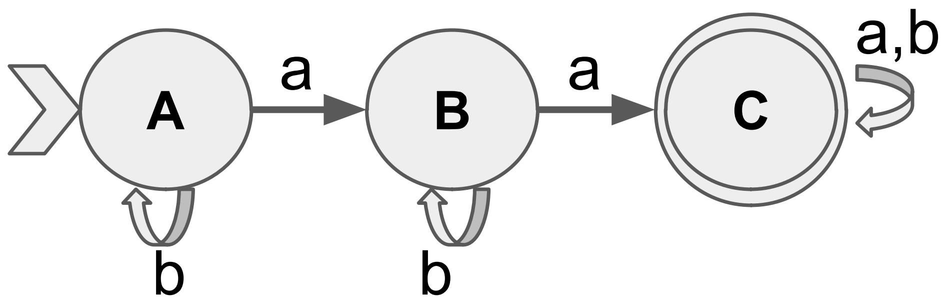


Question

L= {w: w has at least two a's **OR** an odd number of b's} is given.

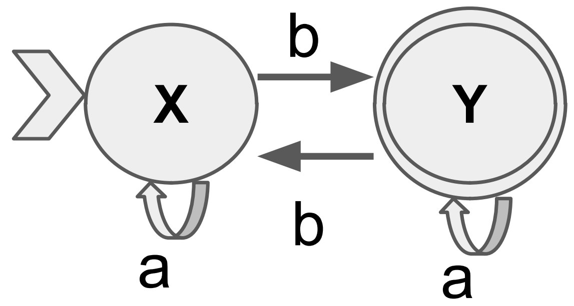
- Design a DFA recognizing both parts of the given language separately.
- Systematically combine two DFAs.

L1= {w: w has at least two a's}



Current State	Next State with ‘a’	Next State with ‘b’
A	B	A
B	C	B
C	C	C

L2= {w: w has an odd number of b's}



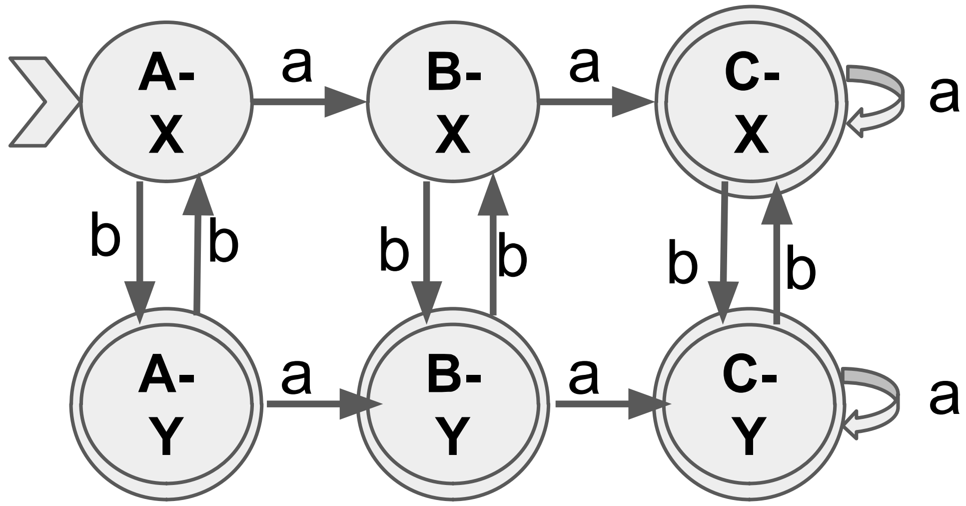
Current State	Next State with ‘a’	Next State with ‘b’
X	X	Y
Y	Y	X

Current State	Next State with ‘a’	Next State with ‘b’
A	B	A
B	C	B
C	C	C

Current State	Next State with ‘a’	Next State with ‘b’
X	X	Y
Y	Y	X

New State	Next State with ‘a’	Next State with ‘b’
A-X	B-X	A-Y
B-X	C-X	B-Y
C-X	C-X	C-Y
A-Y	B-Y	A-X
B-Y	C-Y	B-X
C-Y	C-Y	C-X

New State	Next State with ‘a’	Next State with ‘b’
A-X	B-X	A-Y
B-X	C-X	B-Y
C-X	C-X	C-Y
A-Y	B-Y	A-X
B-Y	C-Y	B-X
C-Y	C-Y	C-X



ba(a+aa*(a+$+b(a+ba*b)*(b+ba*(a+$))))
+(a+bab)(bb)*(b(a+aa*(a+$))+(a+baa*b)(
a+ba*b)*(b+ba*(a+$)))+($+bb)(bb)*(ba(a
+aa*(a+$+b(a+ba*b)*(b+ba*(a+$))))+(a+
bab)(bb)*(b(a+aa*(a+$))+(a+baa*b)(a+b
a*b)*(b+ba*(a+$))))

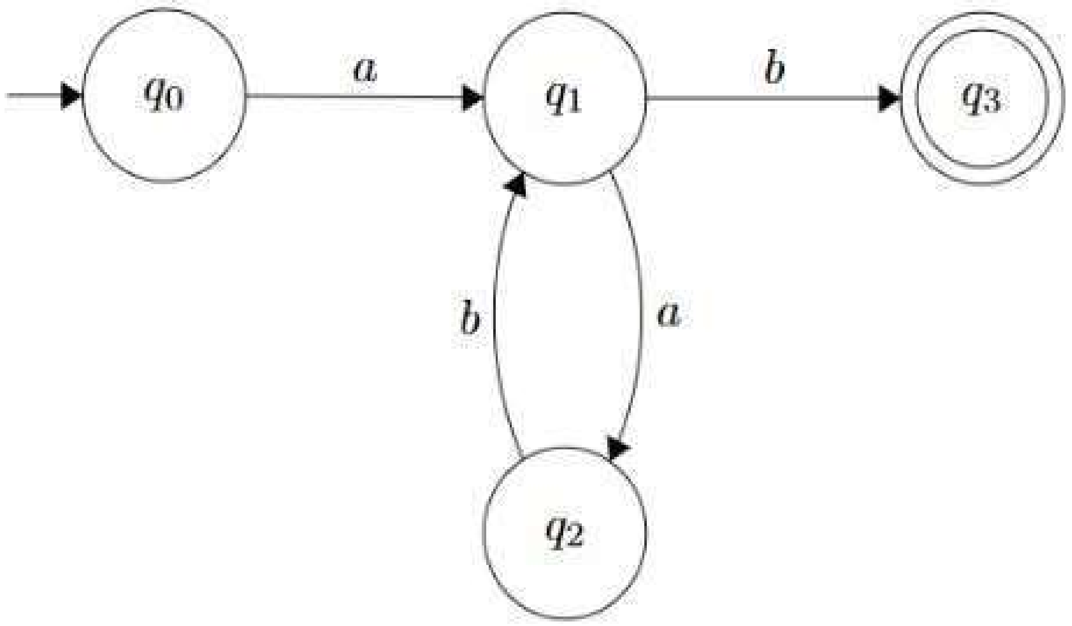
Question

Using a systematic approach, check whether each of the following regular expression couples are disjoint, and if they are not, find the shortest non-empty (not λ) string that matches both of them

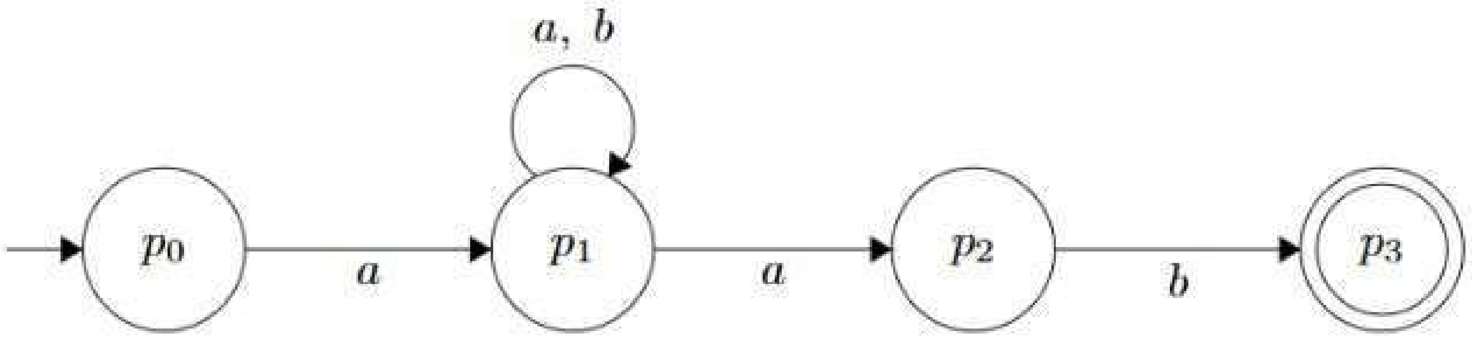
A. L1 = a(ab)*b and L2 = a(a v b)*ab

B. L1 = a(ab)*a and L2 = a(a v b)*ba

L1 = a(ab)*b and L2 = a(a v b)*ab



L1



L2

Current State	Next State with ‘a’	Next State with ‘b’
q0	q1	X
q1	q2	q3
q2	X	q1
q3	X	X

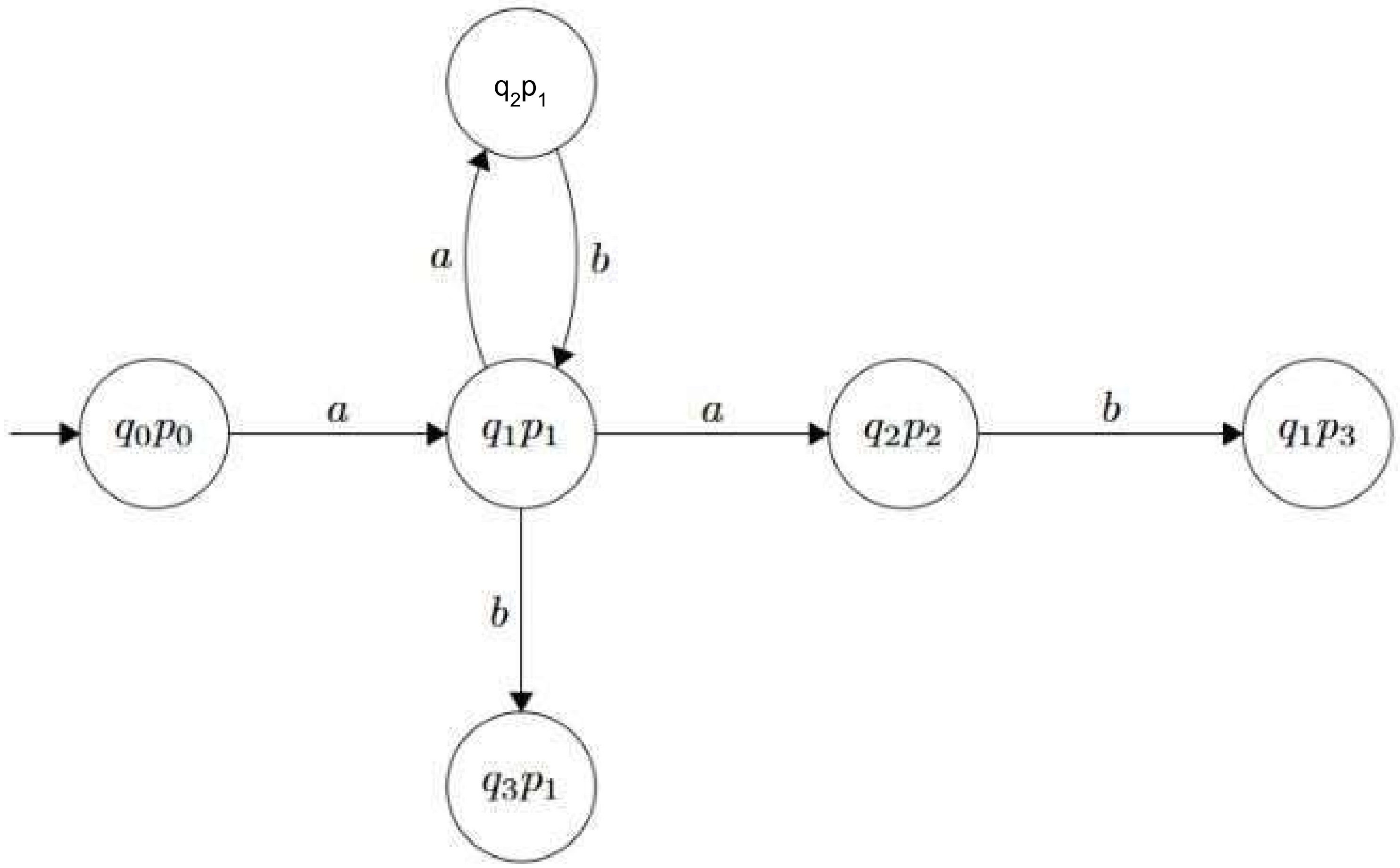
Current State	Next State with ‘a’	Next State with ‘b’
p0	p1	X
p1	p1,p2	p1
p2	X	p3
p3	X	X

Current State	Next State with ‘a’	Next State with ‘b’
q0	q1	X
q1	q2	q3
q2	X	q1
q3	X	X

Current State	Next State with ‘a’	Next State with ‘b’
p0	p1	X
p1	p1,p2	p1
p2	X	p3
p3	X	X

Current State	Next State with ‘a’	Next State with ‘b’
q0-p0	q1-p1	X
q1-p1	q2-p1,q2-p2	q3-p1
q2-p1	X	q1-p1
q2-p2	X	q1-p3
q3-p1	X	X
q1-p3	X	X

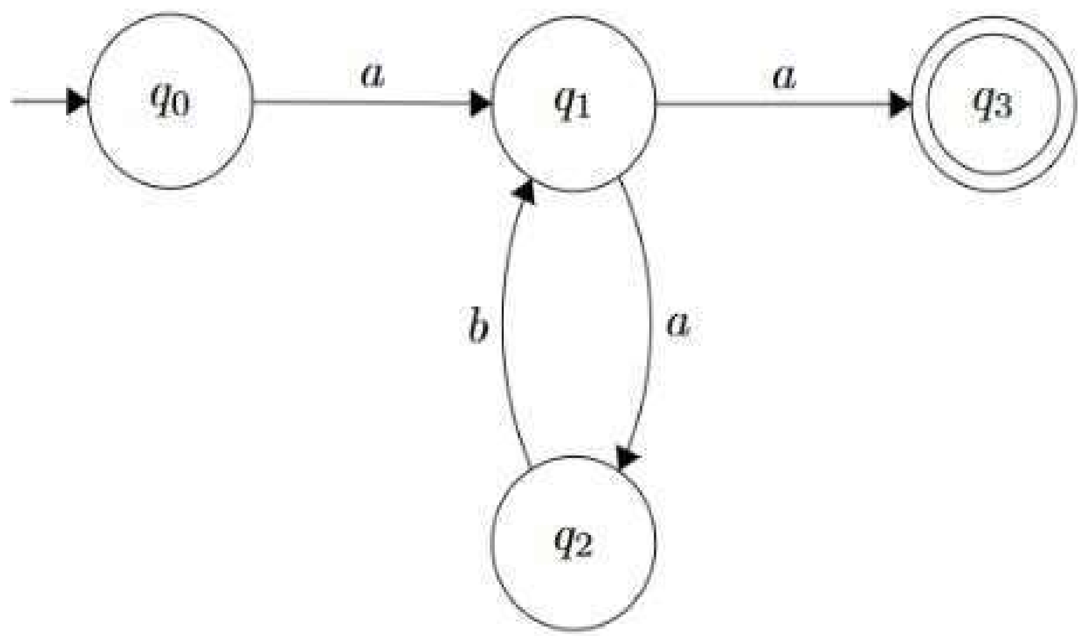
Current State	Next State with ‘a’	Next State with ‘b’
q0-p0	q1-p1	X
q1-p1	q2-p1,q2-p2	q3-p1
q2-p1	X	q1-p1
q2-p2	X	q1-p3
q3-p1	X	X
q1-p3	X	X



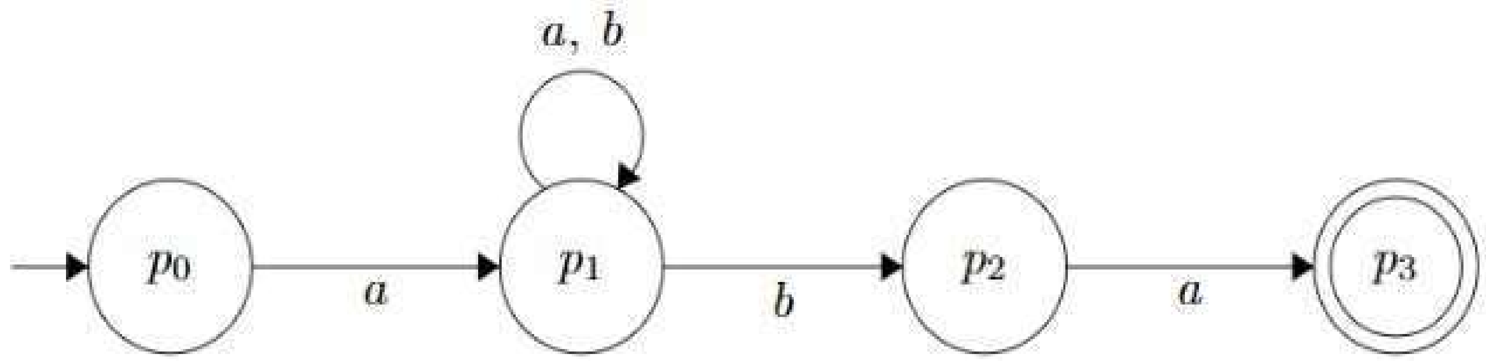
L1 and L2

As there is no reachable final state, we can not find a string

L1 = a(ab)*a and L2 = a(a v b)*ba



L1



L2

Current State	Next State with ‘a’	Next State with ‘b’
q0	q1	X
q1	q2,q3	X
q2	X	q1
q3	X	X

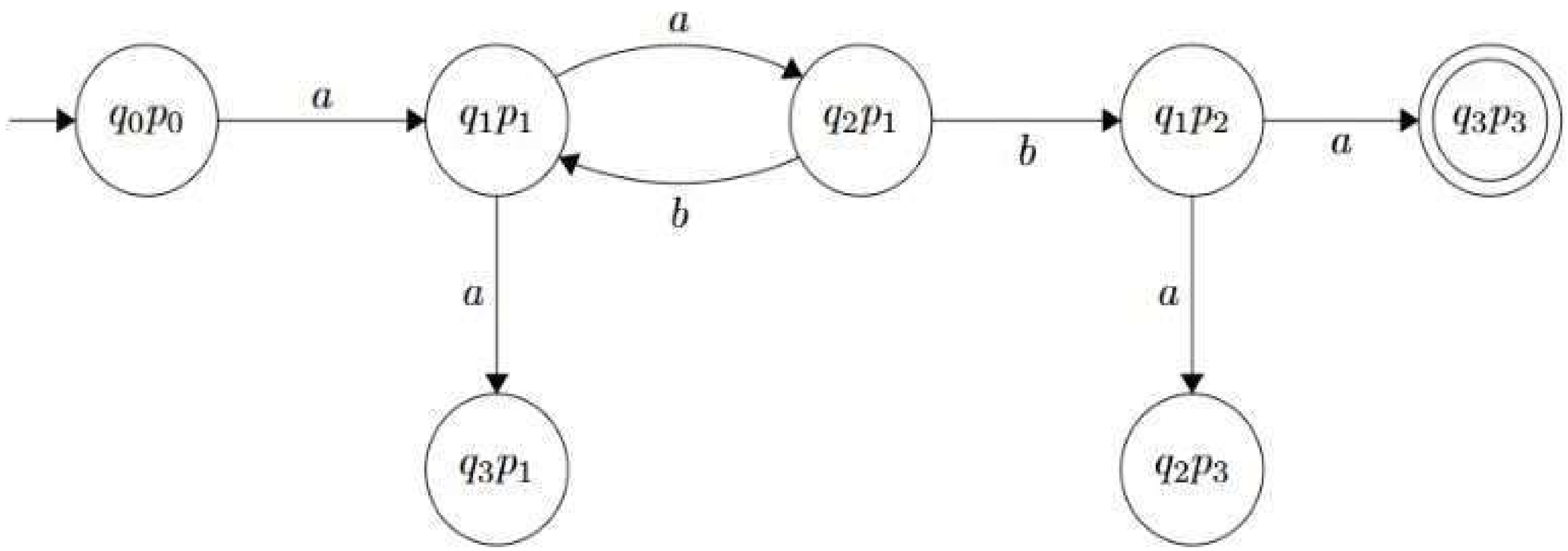
Current State	Next State with ‘a’	Next State with ‘b’
p0	p1	X
p1	p1	p1,p2
p2	p3	X
p3	X	X

Current State	Next State with ‘a’	Next State with ‘b’
q0	q1	X
q1	q2,q3	X
q2	X	q1
q3	X	X

Current State	Next State with ‘a’	Next State with ‘b’
p0	p1	X
p1	p1	p1,p2
p2	p3	X
p3	X	X

Current State	Next State with ‘a’	Next State with ‘b’
q0-p0	q1-p1	X
q1-p1	q2-p1,q3-p1	X
q2-p1	X	q1-p1,q1-p2
q3-p1	X	X
q1-p2	q2-p3,q3-p3	X
q2-p3	X	X
q3-p3	X	X

Current State	Next State with ‘a’	Next State with ‘b’
q0-p0	q1-p1	X
q1-p1	q2-p1,q3-p1	X
q2-p1	X	q1-p1,q1-p2
q3-p1	X	X
q1-p2	q2-p3,q3-p3	X
q2-p3	X	X
q3-p3	X	X



L1 and L2

Shortest string: aaba