CSE 2233 Theory of Computing

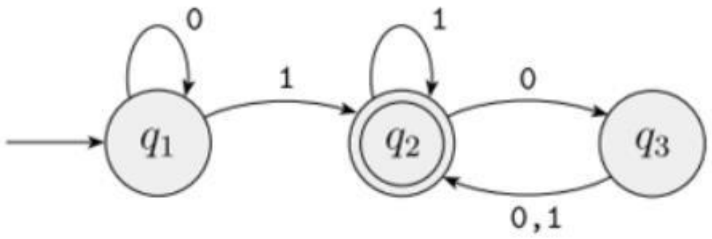
Shekh. Md. Saifur Rahman

Lecturer, CSE Department

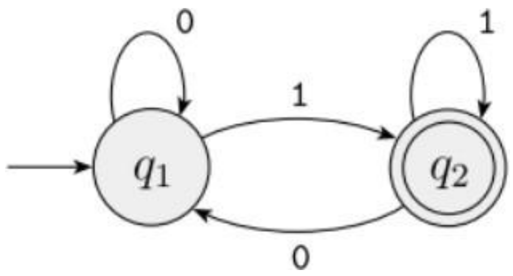
UIU

Deterministic Finite Automaton (DFA)

The term “deterministic finite automata” refers to the fact that on each input there is one and only one state to which the automaton can transition from its current state.



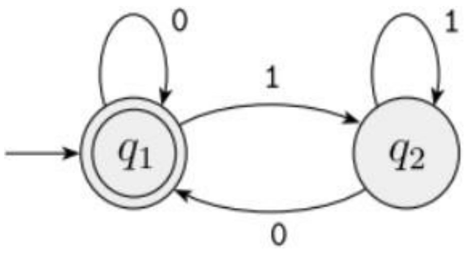
Two state Finite Automaton $M1$



Defined by 5-tuple (Q, Σ, δ, Start state, F)

- Q = ?
- Σ = ?
- δ = ?
- Start state = ?
- Accept state, F = ?
- L($M1$) = ?
- Acceptability Check : 01011

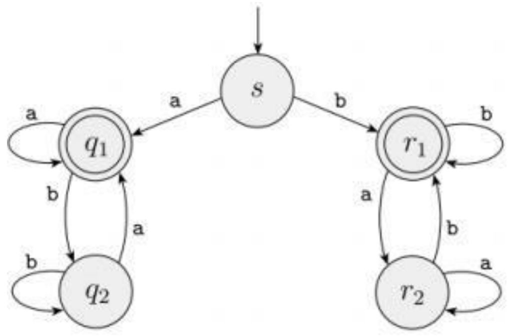
Two state Finite Automaton $M2$



Defined by 5-tuple (Q, Σ, δ, Start state, F)

- Q = ?
- Σ = ?
- δ = ?
- Start state = ?
- Accept state, F = ?
- L($M2$) = ?
- Acceptability Check :

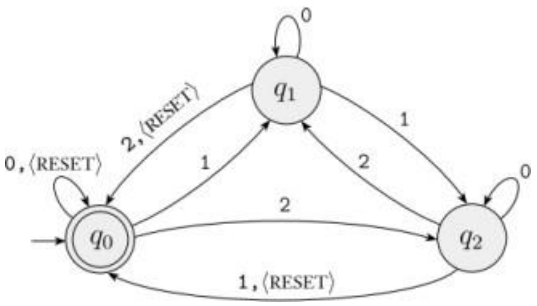
Finite Automaton $M3$



Defined by 5-tuple (Q, Σ, δ, Start state, F)

- Q = ?
- Σ = ?
- δ = ?
- Start state = ?
- Accept state, F = ?
- L($M3$) = ?
- Acceptability Check :

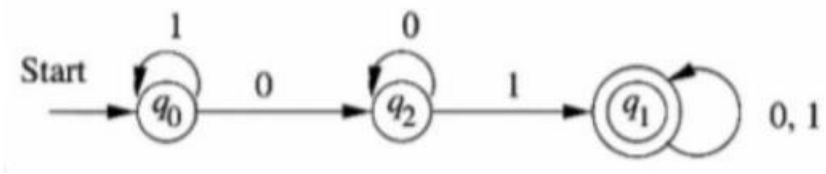
Finite Automaton $M4$



Defined by 5-tuple (Q, Σ, δ, Start state, F)

- Q = ?
- Σ = ?
- δ = ?
- Start state = ?
- Accept state, F = ?
- L($M4$) = ?
- Acceptability Check :

Finite Automaton $M5$



Defined by 5-tuple (Q, Σ, δ, Start state, F)

- Q = ?
- Σ = ?
- δ = ?
- Start state = ?
- Accept state, F = ?
- L($M5$) = ?
- Acceptability Check :

Designing Finite Automata

Machine, $E1$

Give a DFA for $\Sigma = \{ 0, 1 \}$ that accepts any language consists of all strings with an odd number of 1s.

Do we need to remember the entire string seen so far?

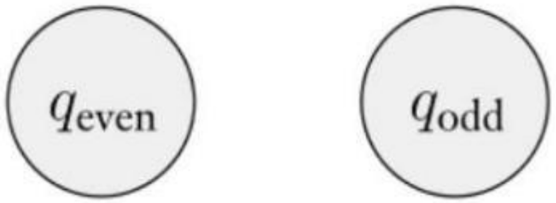


Fig: The two states q_{even} and q_{odd}

Simply remember whether the number of 1s seen so far is even or odd and keep track of this information as you read new symbols. If you read a 1, flip the answer; but if you read a 0, leave the answer as is.

Apply Transitions

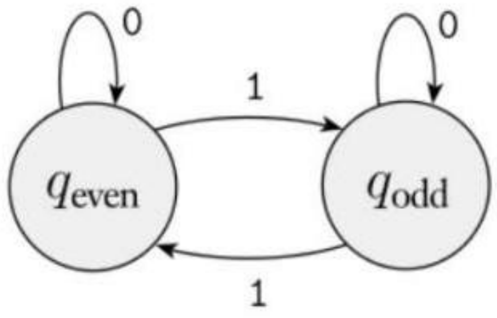


FIGURE 1.19
Transitions telling how the possibilities rearrange

Add accept and reject states

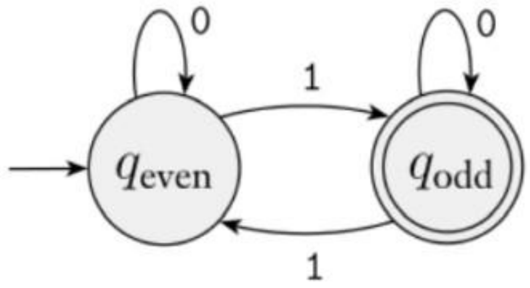


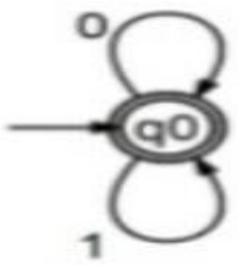
FIGURE 1.20
Adding the start and accept states

Machine, $E2$

Give a DFA for Σ = { 0, 1 } that accepts any string containing any number of 0’s, 1’s or empty string.

Machine, $E2$

Give a DFA for Σ = { 0, 1 } that accepts any string containing any number of 0's, 1's or empty string.

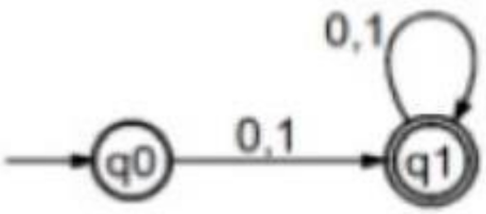


Machine, $E3$

Give a DFA for $\Sigma = \{ 0, 1 \}$ that accepts any string containing any number of 0's, 1's except empty string.

Machine, $E3$

Give a DFA for Σ = { 0, 1 } that accepts any string containing any number of 0's, 1's except empty string.

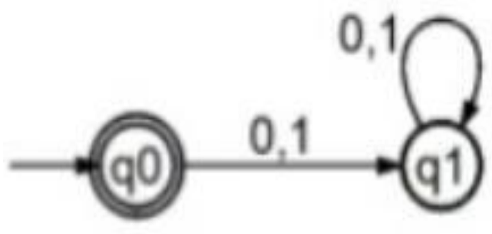


Machine, $E4$

Give a DFA for $\Sigma = \{ 0, 1 \}$ that accepts only the empty string.

Machine, $E4$

Give a DFA for Σ = { 0, 1 } that accepts only the empty string.

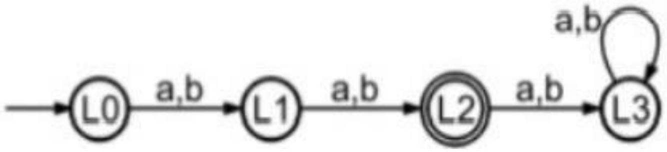


Machine, $E5$

Give a DFA for $\Sigma = \{ a, b \}$ and strings that have exact length 2.

Machine, $E5$

Give a DFA for Σ = { a, b } and strings that have exact length 2.

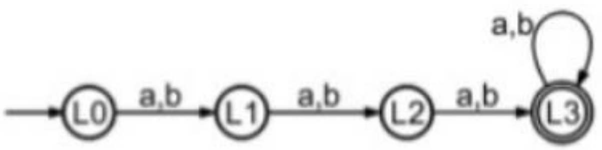


Machine, $E6$

Give a DFA for $\Sigma = \{ a, b \}$ and strings that have length at least 3.

Machine, $E6$

Give a DFA for Σ = { a, b } and strings that have length at least 3.

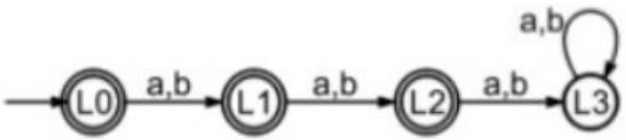


Machine, $E7$

Give a DFA for $\Sigma = \{ a, b \}$ and strings that have length at most 2.

Machine, $E7$

Give a DFA for Σ = { a, b } and strings that have length at most 2.



Machine $E_{8.1}$ -
What about length at least 7

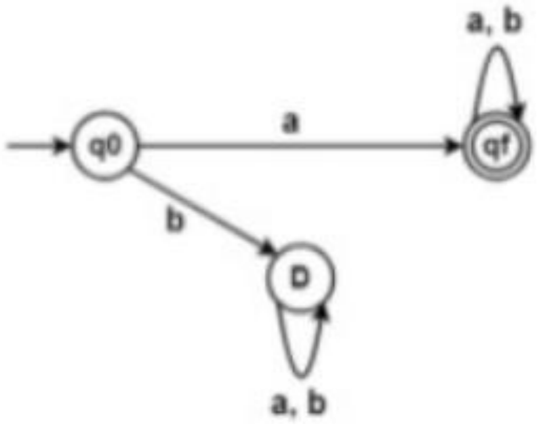
Machine $E_{8.2}$ -
What about length at most 5

Machine, $E8$

Draw a DFA for the language accepting strings starting with ‘a’ over input alphabets $\Sigma = \{ a, b \}$.

Machine, $E8$

Draw a DFA for the language accepting strings starting with ‘a’ over input alphabets Σ = { a, b }.

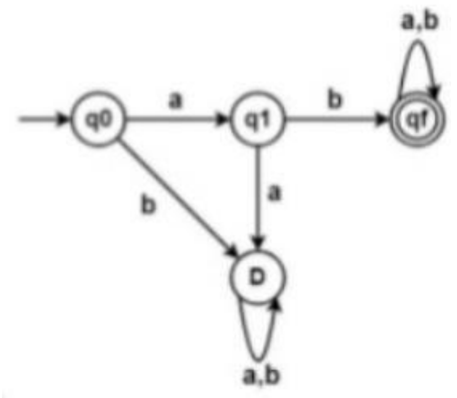


Machine, $E9$

Draw a DFA for the language accepting strings starting with ‘ab’ over input alphabets $\Sigma = \{ a, b \}$.

Machine, $E9$

Draw a DFA for the language accepting strings starting with 'ab' over input alphabets Σ = { a, b } .



Machine, $E10$

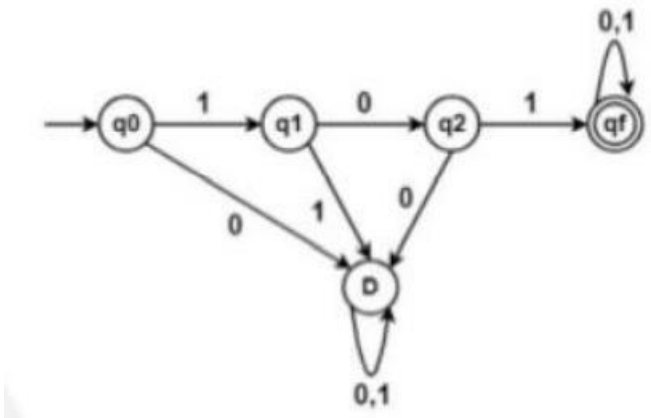
Draw a DFA for the language accepting strings starting with ‘101’ over input alphabets

$\Sigma = \{ 0, 1 \}$.

Machine, $E10$

Draw a DFA for the language accepting strings starting with '101' over input alphabets

$\Sigma = \{ 0, 1 \}$.

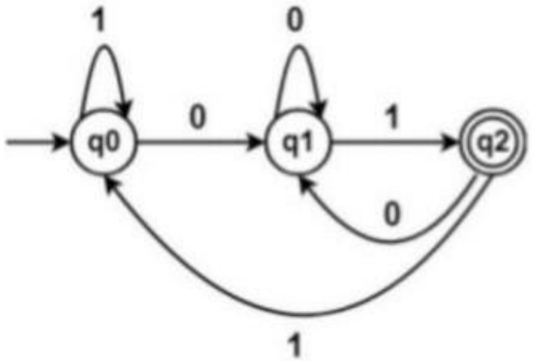


Machine, $E11$

Draw a DFA for the language accepting strings ending with ‘01’ over input alphabets $\Sigma = \{ 0, 1 \}$

Machine, $E11$

Draw a DFA for the language accepting strings ending with '01' over input alphabets Σ = { 0, 1 }

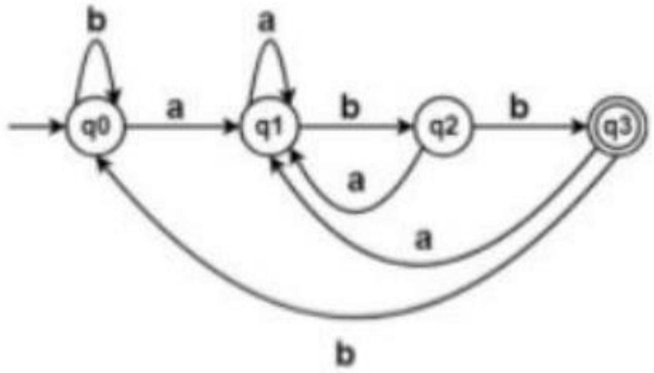


Machine, $E12$

Draw a DFA for the language accepting strings ending with ‘abb’ over input alphabets $\Sigma = \{ a, b \}$

Machine, $E12$

Draw a DFA for the language accepting strings ending with 'abb' over input alphabets Σ = { a, b }



Machine, $E13$

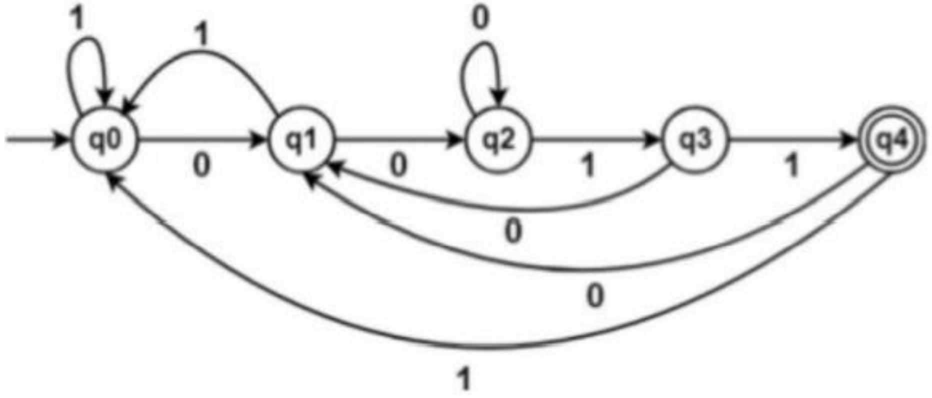
Draw a DFA for the language accepting strings ending with ‘0011’ over input alphabets

$\Sigma = \{ 0, 1 \}$.

Machine, $E13$

Draw a DFA for the language accepting strings ending with ‘0011’ over input alphabets

$\Sigma = \{ 0, 1 \}$.



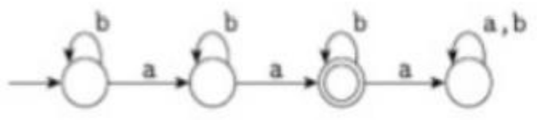
Machine, $E14$ and $E15$

- Give a DFA for Σ = { a, b } and strings that have exactly 2 a’s.

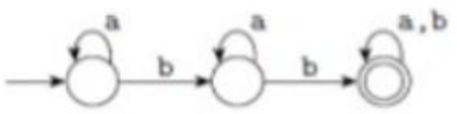
- Give a DFA for Σ = { a, b } and strings that have at least 2 b’s.

Machine, $E14$ and $E15$

- Give a DFA for Σ = { a, b } and strings that have exactly 2 a's.



- Give a DFA for Σ = { a, b } and strings that have at least 2 b's.

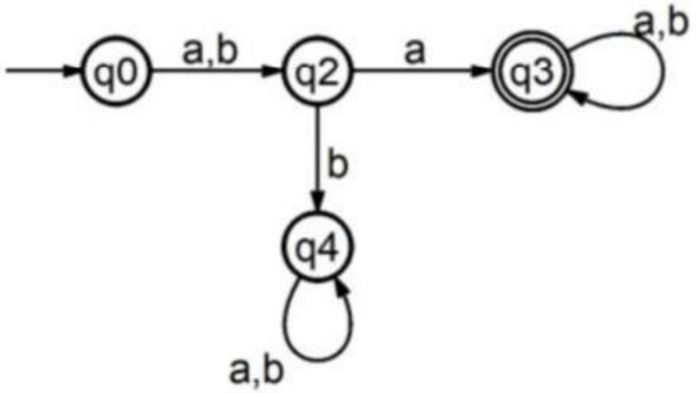


Machine, $E16$

Give a DFA for Σ = { a, b } and the second symbol from the left side is ‘a’.

Machine, $E16$

Give a DFA for Σ = { a, b } and the second symbol from the left side is ‘a’.

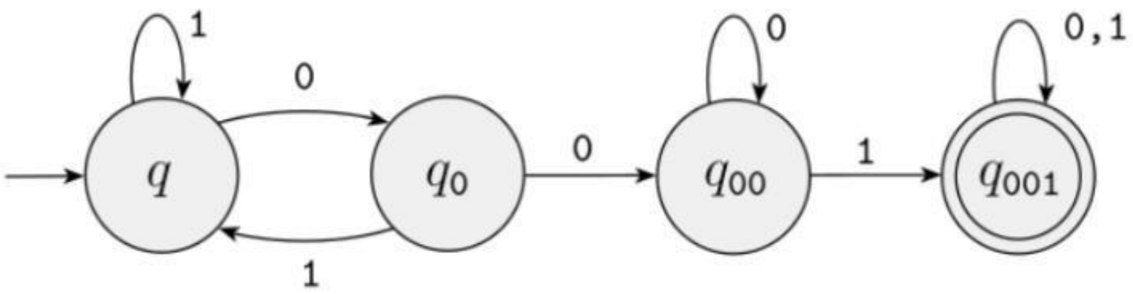


Machine, $E17$

- Design a finite automaton to recognize the regular language of all strings that contain the string 001 as a substring.
- For example, 0010, 1001, 001, and 11111110011111 are all in the language, but 11 and 0000 are not.

Machine, $E17$

- Design a finite automaton to recognize the regular language of all strings that contain the string 001 as a substring.
- For example, 0010, 1001, 001, and 11111110011111 are all in the language, but 11 and 0000 are not.



Machine, $E18$ and $\boldsymbol{E19}$

- Design a finite automaton to recognize the regular language of all strings that contain the string abca as a substring.

- Design a finite automaton to recognize the regular language of all strings that contain the string abababa as a substring.

Machine, $E18$ and $\boldsymbol{E19}$

- Design a finite automaton to recognize the regular language of all strings that contain the string abca as a substring.

- Design a finite automaton to recognize the regular language of all strings that contain the string abababa as a substring.

Machine, $E20$, $E21$ and $E22$

- Draw a DFA for the language accepting strings that do not start with 'ab' over input alphabets $\Sigma = \{ a, b \}$.

- Draw a DFA for the language accepting strings that do not end with 'abb' over input alphabets $\Sigma = \{ a, b \}$.

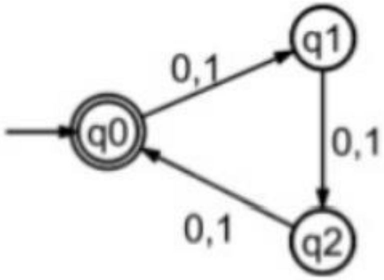
- Design a finite automaton to recognize the regular language of all strings that do not contain the string 001 as a substring.

Machine, $E23$

Give a DFA for $\Sigma = \{ 0, 1 \}$ that accepts only those strings (including empty string) whose length is a multiple of 3.

Machine, $E23$

Give a DFA for Σ = { 0, 1 } that accepts only those strings (including empty string) whose length is a multiple of 3.



Machine $E_{5.1}$ -
L(M) = { w | the length of w mod 3 = 1 }

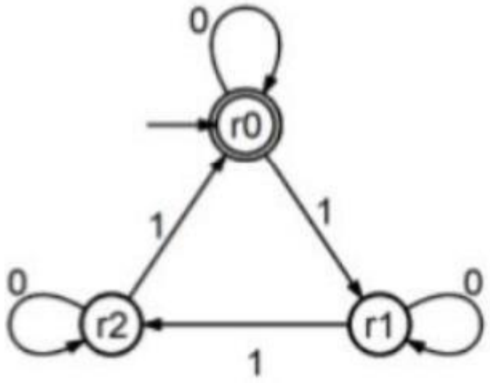
Machine $E_{5.2}$ -
L(M) = { w | the length of w mod 4 = 0 }

Machine, $E24$

Give a DFA for Σ = { 0, 1 } and strings that contain any number of 0’s and the total number of 1’s is a multiple of 3.

Machine, $E24$

Give a DFA for Σ = { 0, 1 } and strings that contain any number of 0's and the total number of 1's is a multiple of 3.



Machine $E_{9.1}$ -
What about strings with number of 0's is a multiple of 4 and any number of 1's

Machine, $E25$

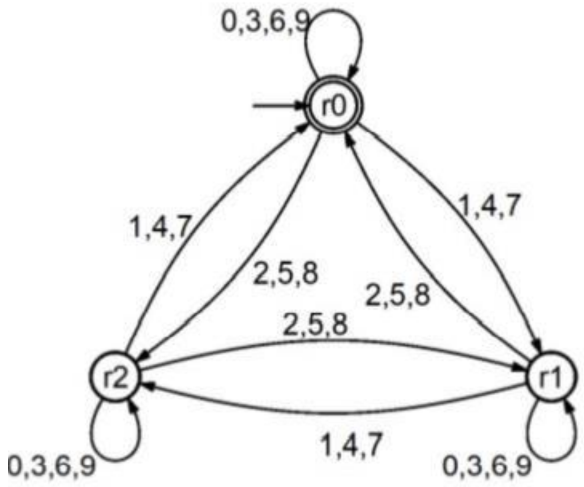
Give a DFA for Σ = { 0, 1 } and only accepts binary strings those are a multiple of 3.

Machine, $E26$

Give a DFA for $\Sigma = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$ that accepts strings whose decimal equivalent is a multiple of 3

Machine, $E26$

Give a DFA for Σ = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 } that accepts strings whose decimal equivalent is a multiple of 3

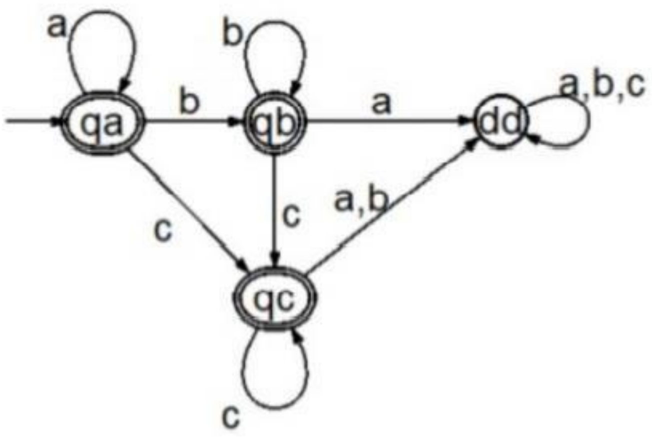


Machine, $E27$

Give a DFA for Σ = { a, b, c } and recognizes strings having patterns = $\{a^n b^m c^l$ | n,m,l >= 0 }

Machine, $E27$

Give a DFA for Σ = { a, b, c } and recognizes strings having patterns = $\{a^n b^m c^l$ | n,m,l >= 0 }

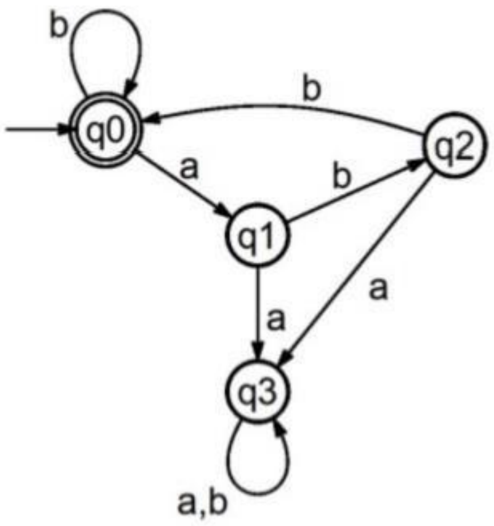


Machine, $E28$

Give a DFA for Σ = { a, b } and contains strings where each ‘a’ is followed by ‘bb’

Machine, $E28$

Give a DFA for Σ = { a, b } and contains strings where each ‘a’ is followed by ‘bb’



Try yourself => Machine, $E29$ and Machine, $E30$

- Give a DFA for Σ = { a, b } and recognizes strings having patterns = $\{a^n b^m |$ n+m = even }.
- Give a DFA for Σ = { a, b } and recognizes strings having patterns = $\{a^n b^m |$ n+m = odd }.

Thanks