Cairo University
Faculty of Computers and Artificial Intelligence

SCS432-Theory of Computations
Assignment - 2
[Deterministic Finite Automaton & Nondeterministic Finite Automaton]

Supervised By:
DR.Samar Hesham
TA.Ibrahim Adel

Prepared & Implemented By:

Mootaz Medhat Ezzat Abdelwahab (20206074)

➤ Problem 1

Design a DFA that accepts all strings which do not contain the substring ba over {a, b}

➥ M = ({q0, q1, q2}, {a, b}, δ, q0, {q0, q1}), Where:

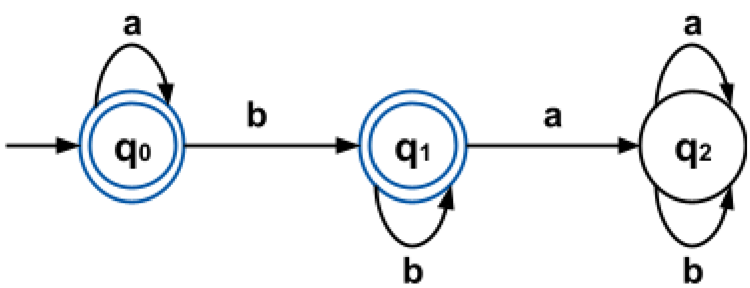
- DFA's states: {q0, q1, q2}
- alphabet: {a, b}
- δ is:

δ(q0, a) = q0 δ(q0, b) = q1

δ(q1, a) = q2 δ(q1, b) = q1

δ(q2, a) = q2 δ(q2, b) = q2

- start state: q0
- set of accepting states: {q0, q1}



➢ Problem 2

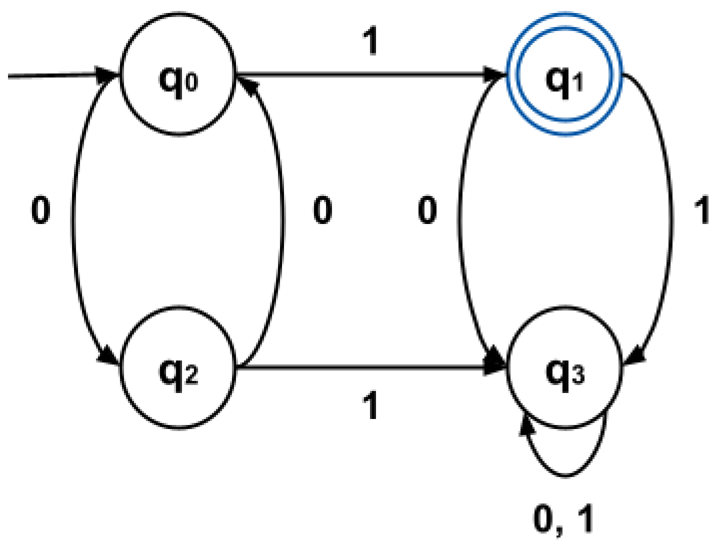
Design a DFA that accepts all strings that contains even number of 0's followed by single 1 over {0, 1}

➥ **M = ({q0, q1, q2, q3}, {0, 1}, δ, q0, {q1}), Where:**

- **DFA's states: {q0, q1, q2, q3}**
- **alphabet: {0, 1}**
- **δ is:**

δ(q0, 0) = q2	δ(q0, 1) = q1
δ(q1, 0) = q3	δ(q1, 1) = q3
δ(q2, 0) = q0	δ(q2, 1) = q3
δ(q3, 0) = q3	δ(q2, 1) = q3

- **start state: q0**
- **set of accepting states: {q1}**



➢ Problem 3

Design a DFA that accepts all strings that contain odd number of x's over {x, y}.

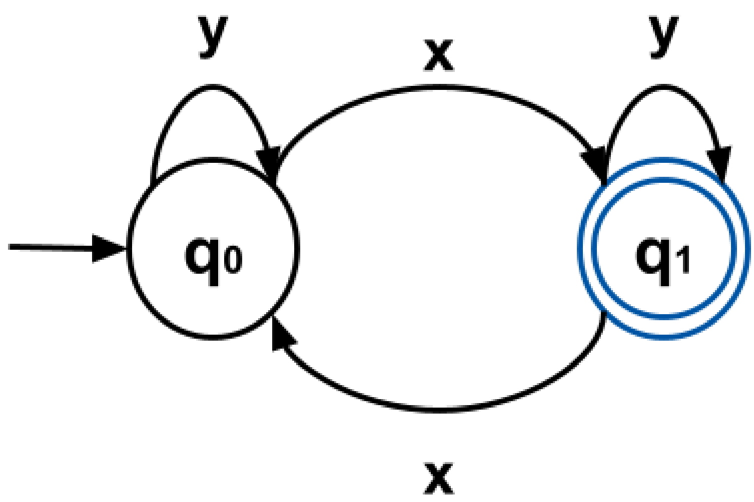
➥ **M = ({q0, q1}, {x, y}, δ, q0, {q1}), Where:**

- **DFA's states: {q0, q1}**
- **alphabet: {x, y}**
- **δ is:**

δ(q0, x) = q1 **δ(q0, y) = q0**

δ(q1, x) = q0 **δ(q1, y) = q1**

- **start state: q0**
- **set of accepting states: {q1}**



➢ Problem 4

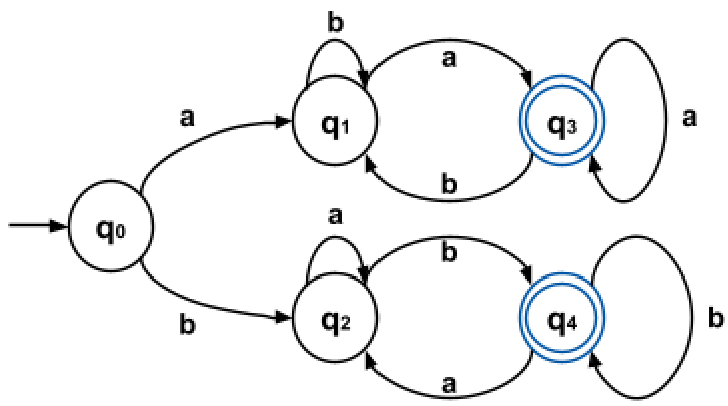
Design a DFA that accepts strings starting and ending with the same characters over {a,b}

➥ **M = ({q0, q1, q2, q3, q4}, {a, b}, δ, q0, {q3, q4}), Where:**

- **DFA's states: {q0, q1, q2, q3, q4}**
- **alphabet: {a, b}**
- **δ is:**

δ(q0, a) = q1	**δ(q0, b) = q2**
δ(q1, a) = q3	**δ(q1, b) = q1**
δ(q2, a) = q2	**δ(q2, b) = q4**
δ(q3, a) = q3	**δ(q3, b) = q1**
δ(q4, a) = q2	**δ(q4, b) = q4**

- **start state: q0**
- **set of accepting states: {q3, q4}**



➢ Problem 5

Design a DFA that accepts all the strings that binary integers divisible by 4 over {0,1}

➥ **M = ({q0, q1, q2, q3}, {0, 1}, δ, q0, {q3}), Where:**

■ **DFA's states: {q0, q1, q2, q3}**

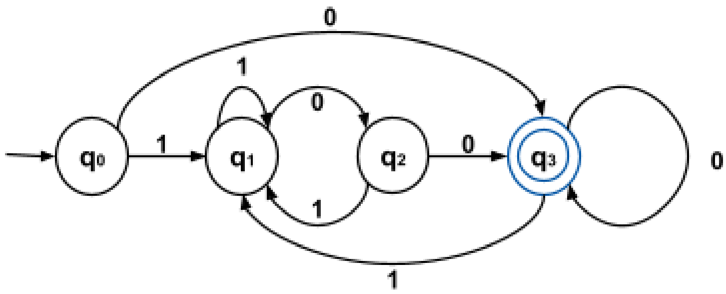
■ **alphabet: {0, 1}**

■ **δ is:**

δ(q0, 0) = q3	δ(q0, 1) = q1
δ(q1, 0) = q2	δ(q1, 1) = q1
δ(q2, 0) = q3	δ(q2, 1) = q1
δ(q3, 0) = q3	δ(q3, 1) = q1

■ **start state: q0**

■ **set of accepting states: {q3}**



➤ Problem 6

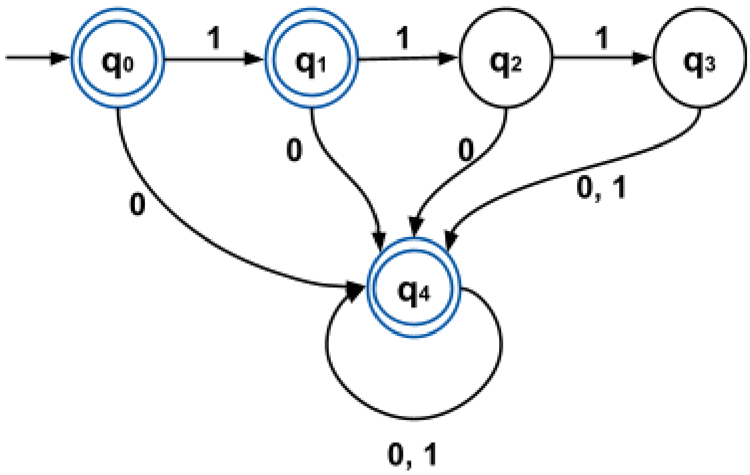
Construct an DFA that accepts all strings {W | W is any string except 11 and 111}

➥ **M = ({q0, q1, q2, q3, q4}, {0, 1}, δ, q0, {q0, q1, q4}), Where:**

- **DFA's states: {q0, q1, q2, q3, q4}**
- **alphabet: {0, 1}**
- **δ is:**

δ(q0, 0) = q4	**δ(q0, 1) = q1**
δ(q1, 0) = q4	**δ(q1, 1) = q2**
δ(q2, 0) = q4	**δ(q2, 1) = q3**
δ(q3, 0) = q4	**δ(q3, 1) = q4**
δ(q4, 0) = q4	**δ(q4, 1) = q4**

- **start state: q0**
- **set of accepting states: {q0, q1, q4}**



➢ Problem 7

Construct an NFA that accepts all strings over the alphabet {0, 1} containing an equal number of occurrences of '01' and '10'

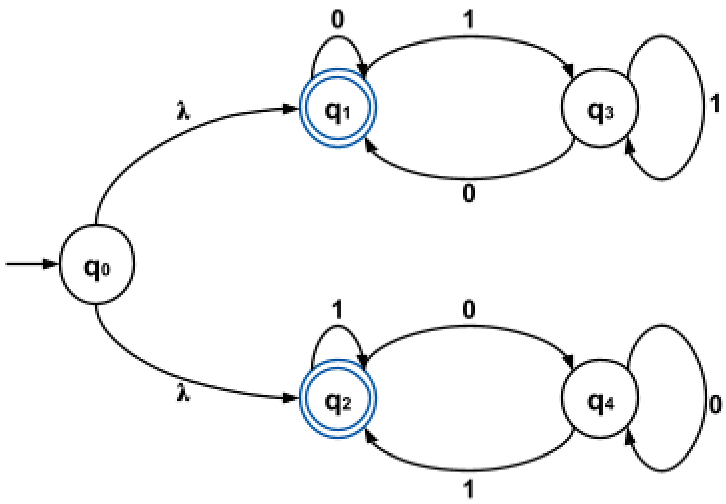
➥ **M = ({q0, q1, q2, q3, q4}, {0, 1}, δ, q0, {q1, q2}), Where:**

- **NFA's states: {q0, q1, q2, q3, q4}**
- **alphabet: {0, 1}**
- **δ is:**

δ(q0, λ) = {q1, q2}

δ(q1, 0) = {q1}	**δ(q1, 1) = {q3}**
δ(q2, 0) = {q4}	**δ(q2, 1) = {q2}**
δ(q3, 0) = {q1}	**δ(q3, 1) = {q3}**
δ(q4, 0) = {q4}	**δ(q4, 1) = {q2}**

- **start state: q0**
- **set of accepting states: {q1, q2}**



➢ Problem 8

Design an NFA that accepts all strings over the alphabet {0, 1} that contain the substring "101" or "010"

➥ **M = ({q0, q1, q2, q3, q4, q5, q6, q7, q8}, {0, 1}, δ, q0, {q4, q8}), Where:**

- **NFA's states: {q0, q1, q2, q3, q4, q5, q6, q7, q8}**
- **alphabet: {λ, 0, 1}**
- **δ is:**

δ(q0, λ) = {q1, q5}

δ(q1, 0) = {q1} **δ(q1, 1) = {q1, q2}**

δ(q2, 0) = {q3}

δ(q3, 1) = {q4}

δ(q4, 0) = {q4} **δ(q4, 1) = {q4}**

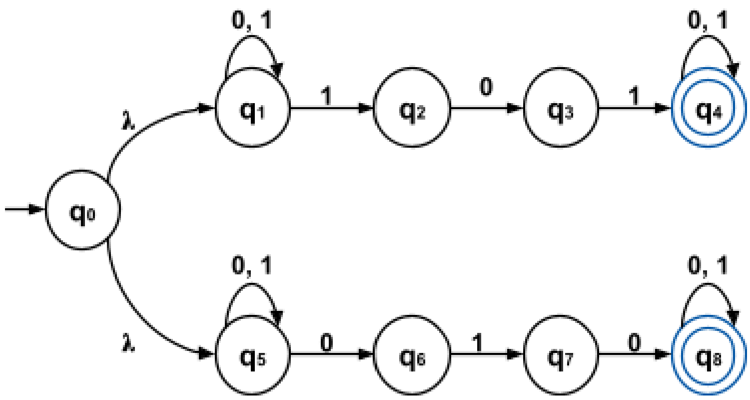
δ(q5, 0) = {q5, q6} **δ(q5, 1) = {q5}**

δ(q6, 1) = {q7}

δ(q7, 0) = {q8}

δ(q8, 0) = {q8} **δ(q8, 1) = {q8}**

- **start state: q0**
- **set of accepting states: {q4, q8}**



➢ Problem 9

Design an NFA that accepts all strings over the alphabet {0, 1} where no two consecutive characters are the same

➥ **M = ({q0, q1, q2, q3, q4}, {0, 1}, δ, q0, {q1, q2, q3, q4}), Where:**

- **NFA's states: {q0, q1, q2, q3, q4}**
- **alphabet: {λ, 0, 1}**
- **δ is:**

δ(q0, 0) = {q1}   δ(q0, 1) = {q2}

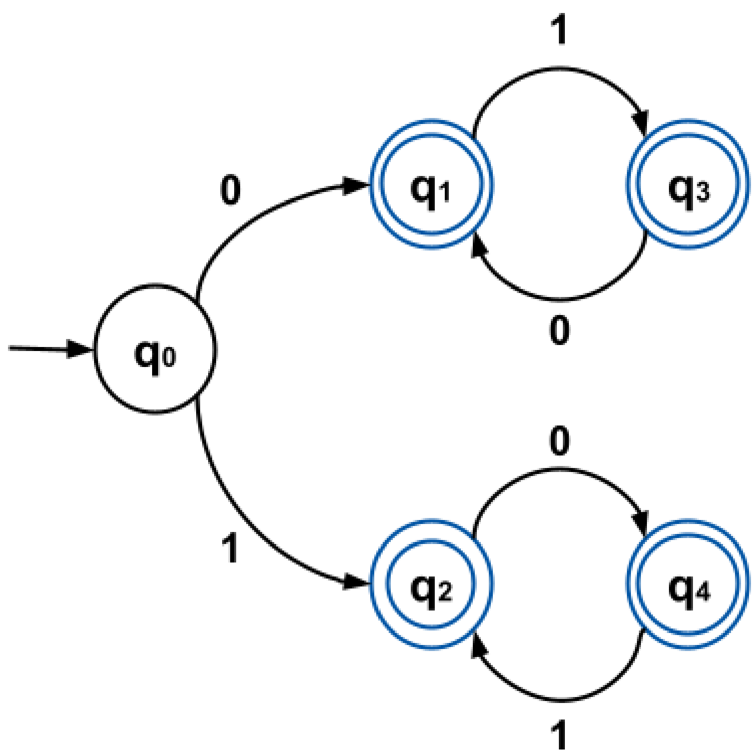
δ(q1, 1) = {q3}

δ(q2, 0) = {q4}

δ(q3, 0) = {q1}

δ(q4, 1) = {q2}

- **start state: q0**
- **set of accepting states: {q1, q2, q3, q4}**



➢ Problem 10

Design an NFA that recognizes strings over the alphabet {0, 1} where every '0' is followed by at least one '1'

➥ **M = ({q0, q1}, {0, 1}, δ, q0, {q0}), Where:**

- **NFA's states: {q0, q1}**
- **alphabet: {0, 1}**
- **δ is:**

δ(q0, 0) = {q1}   δ(q0, 1) = {q10}

δ(q1, 0) = {q0}

- **start state: q0**
- **set of accepting states: {q0}**