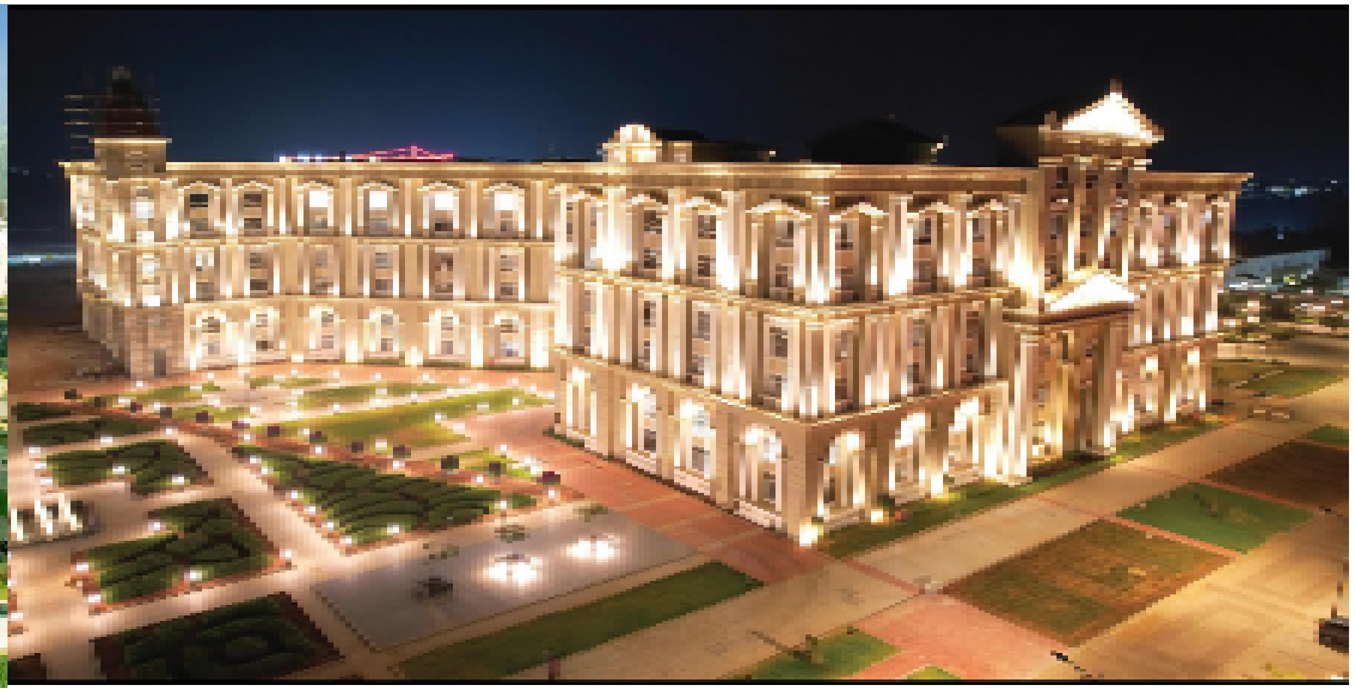
DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Deterministic Finite Automata

By

Gaurav Kumar

Divisible by k problems

$$\boldsymbol{\delta(q_i, a) = q_j \text{ where } j = (r * i + d) \bmod k}$$

- r is the radix of input. For binary $r = 2$
- i is the remainder obtained after dividing by k
- d represent digits. For binary $d = \{0, 1\}$
- k is divisor

The steps to be followed to find FA using these types of problems is shown below:

Step 1: Identify the radix, input alphabets and the divider k.

Step 2: Compute the possible remainders: These remainders represent the states of DFA.

Step 3: Find the transitions using $\delta(q_i, a) = q_j$ where $j = (r * i + d) \bmod k$.

Step 4: Construct the DFA using the transitions obtained in step 3.

▮ Example 1: Now, let us "Construct a DFA which accepts strings of 0's and 1's where the value of each string is represented as a binary number. Only the strings representing zero modulo five should be accepted. For example, 0000, 0101, 1010, 1111, etc. should be accepted".

Solution: The DFA can be obtained as shown below:

Step 1: Identify the radix, input alphabets and the divisor k: In this case, r = 2:

$d = \{0, 1\}, k = 5$

Step 2: Compute the possible remainders: After dividing by k, the possible remainders are:

$i = 0, 1, 2, 3, 4$

Step 3: Compute transitions: The transitions can be computed using the following relation:

$$\boldsymbol{\delta(q_i, a) = q_j} \text{ where } \boldsymbol{j = (r * i + d) \bmod k}$$

with $r = 2$ and $k = 5$

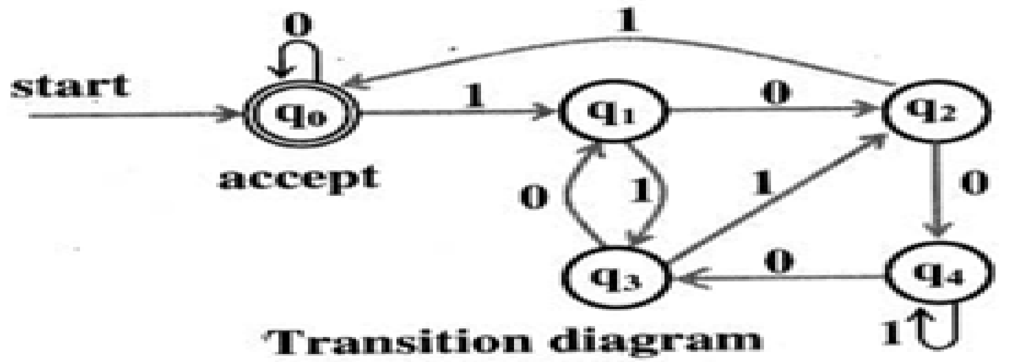
So, $\boldsymbol{j = (2i + d) \bmod 5}$

remainder	d	(2* i +d) mod 5 = j	$\delta(q_i,d) = q_j$
i = 0	0	(2*0 + 0) mod 5 = 0	$\delta(q_0,0) = q_0$
	1	(2*0 + 1) mod 5 = 1	$\delta(q_0,1) = q_1$
i = 1	0	(2*1 + 0) mod 5 = 2	$\delta(q_1,0) = q_2$
	1	(2*1 + 1) mod 5 = 3	$\delta(q_1,1) = q_3$
i = 2	0	(2*2 + 0) mod 5 = 4	$\delta(q_2,0) = q_4$
	1	(2*2 + 1) mod 5 = 0	$\delta(q_2,1) = q_0$
i = 3	0	(2*3 + 0) mod 5 = 1	$\delta(q_3,0) = q_1$
	1	(2*3 + 1) mod 5 = 2	$\delta(q_3,1) = q_2$
i = 4	0	(2*4 + 0) mod 5 = 3	$\delta(q_4,0) = q_3$
	1	(2*4 + 1) mod 5 = 4	$\delta(q_4,1) = q_4$

Transitions of resulting DFA

Step 4: The DFA can be defined as $\mathbf{M = (Q, \Sigma, \delta, q_0, F)}$ where

- $\mathbf{Q} = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\mathbf{q_0}$ is the start state
- $\mathbf{F} = \{q_0\}$
- δ is shown below using the transition diagram and transition table as shown below:



Transition diagram

δ	0	1
$\rightarrow *q_0$	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

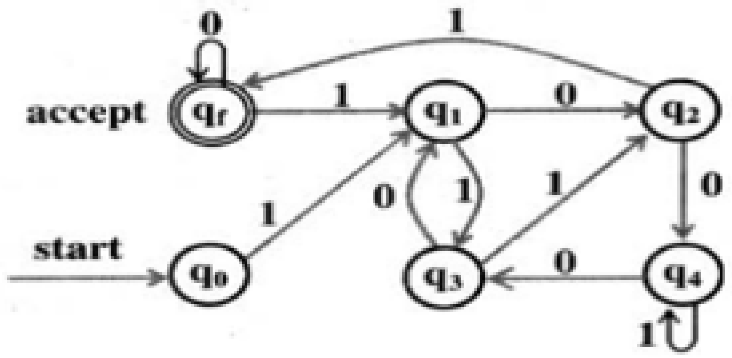
Transition Table

Note: Observe that for modulo 5, number of states of DFA will be 5.

In general, for modulo k, number of states of DFA will be k.

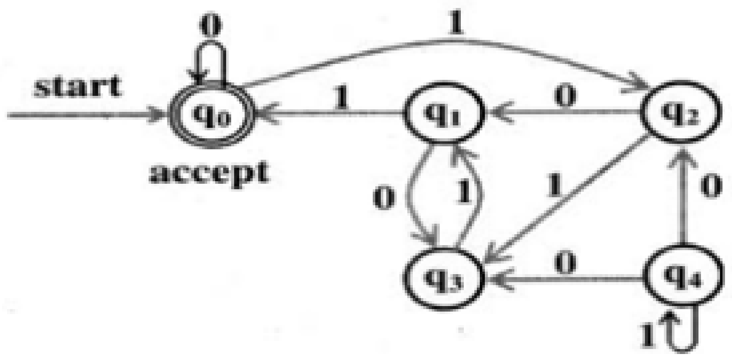
▮ **Example 2:** Now, let us "obtain a DFA which accepts the set of all strings beginning with a 1 that when interpreted as a binary integer, is a multiple of 5. For example, 101, 1010, 1111 etc are multiples of 5. Note that 0101 is not beginning with 1 and it should not be accepted".

Solution: The solution to this problem is same as above problem. But, the number always should start with a 1. If a binary number starts with a 0, the number should never be accepted. So, let us rename the final state as q_f and have the new start state q_0 and from this symbol on 1, enter into state q_1. The resulting DFA is shown using the transition diagram as shown below:



▮ **Example 3:** Now, let us "obtain a DFA that accepts set of all strings that, when interpreted in reverse as a binary integer, is divible by 5. Examples of strings in the language are 0, 10011, 1001100 and 0101".

Solution: The solution remains same as Example 1. But, reverse the direction of all arrow marks (except the arrow labeled with start). The resulting DFA is shown below:



▮ **Example 4:** Now, let us "Draw a DFA to accept decimal strings divisible by 3".

Solution: The DFA can be obtained as shown below:

Step 1: Identify the radix, input alphabets and the divisor k: In this case, r = 10:

Step 2: Compute the possible remainders: After dividing any decimal number by 3, results in following remainders:

$i = 0, 1, 2$ which implies q_0, q_1 and q_2 are the states of DFA

Step 3: Compute transitions: The transitions can be computed using the following relation: $\boldsymbol{\delta(q_i, a) = q_j}$ where $\boldsymbol{j = (r * i + d) \bmod k}$

$$\text{with } r = 10 \quad \text{and} \quad k = 3$$

$$\text{So, } \boldsymbol{j = (2i + d) \bmod 3}$$

Note: For the sake of convenience, let us group the digits from 0 to 9 based on the remainders we get after dividing by 3 as shown below:

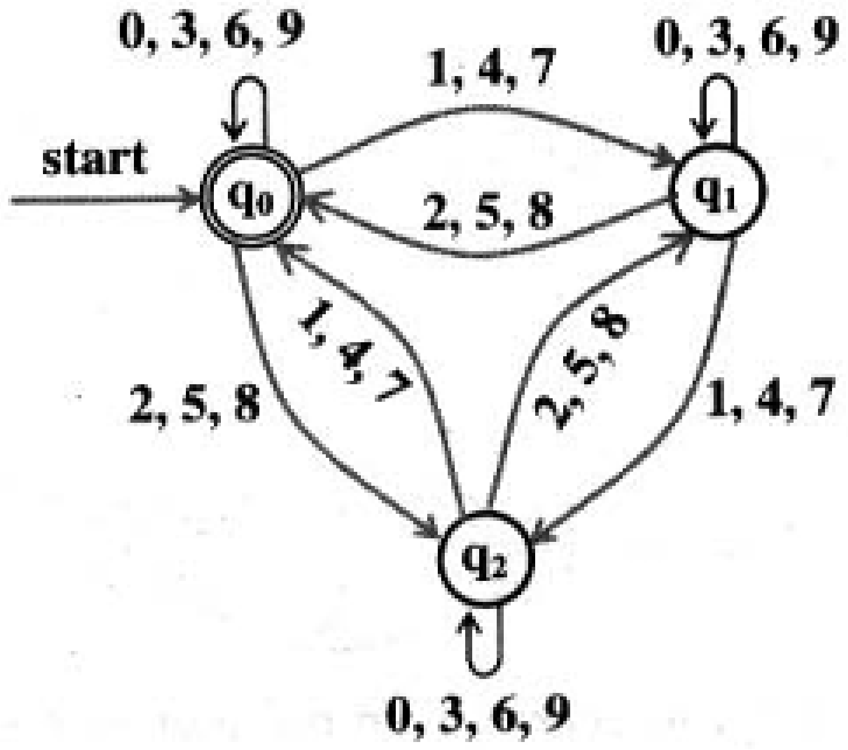
- $\{0, 3, 6, 9\}$ with 0 as the remainder. So, δ from $\{0, 3, 6, 9\} \Rightarrow \delta$ from $\{0\}$
- $\{1, 4, 7\}$ with 1 as the remainder. So, δ from $\{1, 4, 7\} \Rightarrow \delta$ from $\{1\}$
- $\{2, 5, 8\}$ with 2 as the remainder. So, δ from $\{2, 5, 8\} \Rightarrow \delta$ from $\{2\}$

remainder	d	**(2* i +d) mod 3 = j**	$\boldsymbol{\delta(q_i,d) = q_j}$		Transitions of DFA
$i = 0$	0	$(10*0 + 0) \bmod 3 = \mathbf{0}$	$\delta(q_0,0) = q_0$	$\Rightarrow$	$\boldsymbol{\delta(q_0,\{0,3,6,9)\} = q_0}$
	1	$(10*0 + 1) \bmod 3 = \mathbf{1}$	$\delta(q_0,1) = q_1$	$\Rightarrow$	$\boldsymbol{\delta(q_0,\{1,4,7\}) = q_1}$
	2	$(10*0 + 2) \bmod 3 = \mathbf{2}$	$\delta(q_0,2) = q_2$	$\Rightarrow$	$\boldsymbol{\delta(q_0,\{2,5,8\}) = q_2}$
$i = 1$	0	$(10*1 + 0) \bmod 3 = \mathbf{1}$	$\delta(q_1,0) = q_1$	$\Rightarrow$	$\boldsymbol{\delta(q_1,\{0,3,6,9)\} = q_1}$
	1	$(10*1 + 1) \bmod 3 = \mathbf{2}$	$\delta(q_1,1) = q_2$	$\Rightarrow$	$\boldsymbol{\delta(q_1, \{1,4,7\}) = q_2}$
	2	$(10*1 + 2) \bmod 3 = \mathbf{0}$	$\delta(q_1,2) = q_0$	$\Rightarrow$	$\boldsymbol{\delta(q_1,\{2,5,8\}) = q_0}$
$i = 2$	0	$(10*2 + 0) \bmod 3 = \mathbf{2}$	$\delta(q_2,0) = q_2$	$\Rightarrow$	$\boldsymbol{\delta(q_2,\{0,3,6,9)\} = q_2}$
	1	$(10*2 + 1) \bmod 3 = \mathbf{0}$	$\delta(q_2,1) = q_0$	$\Rightarrow$	$\boldsymbol{\delta(q_2, \{1,4,7\}) = q_0}$
	2	$(10*2 + 2) \bmod 3 = \mathbf{1}$	$\delta(q_2,2) = q_1$	$\Rightarrow$	$\boldsymbol{\delta(q_2,\{2,5,8\}) = q_1}$

Step 4: The DFA can be defined as $\mathbf{M = (Q, \Sigma, \delta, q_0, F)}$ where

- $\mathbf{Q} = \{q_0, q_1, q_2\}$
- $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $\mathbf{q_0}$ is the start state

- $\mathbf{F} = \{q_0\}$
- δ is shown below using the transition diagram and transition table as shown below:



Modulo-k counter Problems

1.6.3. Modulo k Counter Problems

Example 1: Let us "Obtain a DFA to accept strings of even number of a's". The DFA can be constructed by following the steps one by one as shown below:

Solution: It is given that L = {w : w has even number of a's}.

Identify the number of states: The string w may have even number of a's or odd number of a's and results in following two cases (two states):

- **Case 0:** Strings that accepts Even number of a's is denoted by: E
- **Case 1:** Strings that accepts Odd number of a's is denoted by: O

Identify the start state and final state: Before reading any of the input symbols, number of a's will be zero which represent Even a's. So, the state E with even number of a's is the start state.

Since, it is required to accept Even number of a's the state E which accepts even number of a's is the final state.

Design: Once the start state and final states are identified the transitions can be easily obtained as shown below:

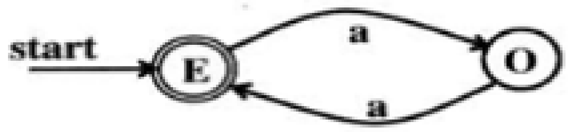
- From a state E, on reading *a* results in odd number of a's and hence change to odd state O. The transition is:

- From a state O, on reading *a* results in even number of a's and hence change to even state E. The transition is:

  δ(O, a) = E

So, the DFA $M = (Q, \Sigma, \delta, q_0, F)$ is defined as:

- $Q = \{E, O\}$
- $\Sigma = \{a\}$
- δ = shown in transition diagram



DFA to accept even number of a's

- $q_0 = E$ is the start state
- $F = \{E\}$

Note: The DFA to obtain odd number of a's can be obtained by making O state (odd state) as the final state and E state as the start state as shown below:

DFA to accept odd number of a's

▮ Example 2: Let us "Obtain a DFA to accept the language $L = \{ w : |w| \bmod 3 = 0\}$ where $\Sigma = \{a\}$".

Solution: It is given that $L = \{w : |w| \bmod 3 = 0\}$ where $\Sigma = \{a\}$ } which indicates that the language consists of strings of multiples of 3 a's.

Identify the number of states: The language can be interpreted as strings of a's such that the number of a's in string is divisible by 3. Note that $|w| \bmod 3$ results in three cases:

- **Case 0:** Results in remainder 0: The state is identified as q_0. q_0
- **Case 1:** Results in remainder 1. The state is identified as q_1. q_1
- **Case 2:** Results in remainder 2. The state is identified as q_2. q_2

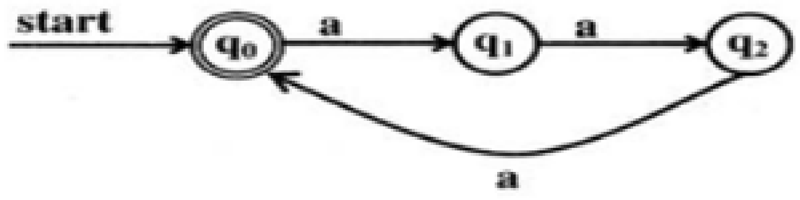
Identify the start state and final state: Before reading any inputs, number of a's will be zero. So, 0 mod 3 = 0 which results in case 0 . Hence, state q0 is start state. Since, it is required to accept string of a's such that |w| mod 3 = 0, which results in case 0, q_0 is considered as final state.

Design: Once the start state and final states are identified the transitions can be easily obtained as shown below:

- From a state with remainder 0 (case 0) denoted by q_0, on reading *a* results in remainder 1 (case 1) denoted by q_1. So, $\boldsymbol{\delta(q_0, a) = q_1}$
- From a state with remainder 1 (case 1) denoted by q_1, on reading *a* results in remainder 2 (case 2) denoted by q_2. So, $\boldsymbol{\delta(q_1, a) = q_2}$
- From a state with remainder 2 (case 2) denoted by q_1, on reading *a* results in remainder 0 (case 0) denoted by q_0. So, $\boldsymbol{\delta(q_2, a) = q_0}$

So, the DFA $M = (Q, \Sigma, \delta, q0, F)$ is defined as:

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{a\}$
- δ : the transitions are shown using transition diagram as shown below:



- q_0 = start state
- $F = \{q_0\}$

The given language accepted by above DFA can also be represented as:

- $L = \{w : |w| \bmod 3 = 0\}$ where $\Sigma = \{a\}$ }

  or

- $L = \{w: n_a(w)$ are divisible by 3 where $\Sigma=\{a\}\}$

  or

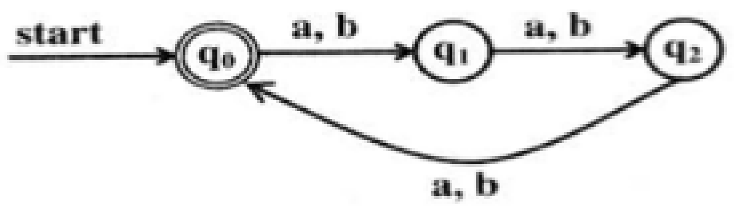
- $L = \{a^{3n} : n \geq 0\}$

Example 3: Let us "Obtain a DFA to accept the language $L = \{ w : |w| \bmod 3 = 0\}$ on $\Sigma = \{a, b\}$".

Solution: The language consists of strings of a's and b's whose length is a multiple of 3 and can be represented as:

$$L = \{\varepsilon, aaa, aab, aba, abb, baa, bab, bba, bbb, \ldots\}$$

The design is similar to that of the previous problem. But, add one extra label b for each edge as shown below:

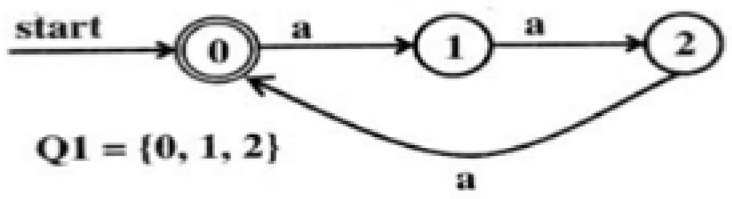


Example 4: Now, let us "Obtain a DFA to accept the following language L = {w such that

a) $|w| \bmod 3 \geq |w| \bmod 2$ where $w \in \Sigma^*$ and $\Sigma = \{a\}$

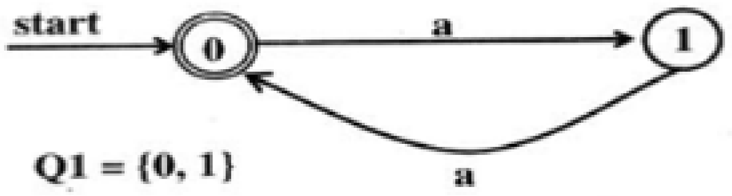
b) $|w| \bmod 3 \neq |w| \bmod 2$ where $w \in \Sigma^*$ and $\Sigma = \{a\}$

Solution: The DFA to accept a string w such that $|w| \bmod 3 = 0$ can be written as shown below (see Example 2):



M1

On similar lines, the DFA to accept a string w such that $|w| \bmod 2 = 0$ can be written as shown below (see Example 1):



M2

Transitions: The transitions of DFA which has strings of w with $|w| \bmod 3$ and $|w| \bmod 2$ can be obtained by taking the cross product of Q1 and Q2 as shown below:

$$Q1 \times Q2 = \{ (0,0), (0,1), (1,0), (1,1), (2,0), (2,1) \}$$

The transitions on each of the pair (x, y) can be obtained as shown below:

Transitions from M1 — Transitions from M2

$\delta((0, 0), a) = (\delta(0, a), \ \delta(0, a))$
$\quad q_0 \qquad\qquad 1 \qquad\qquad 1 = (1, 1)$ **(final state)**

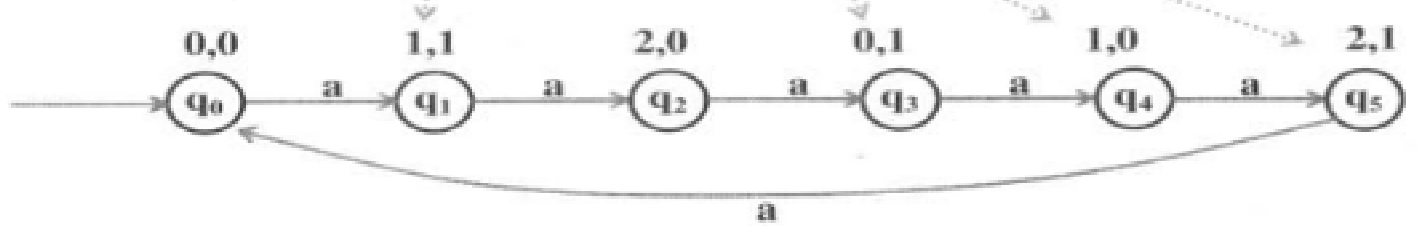
$\delta((1,1), a) = (\delta(1, a), \ \delta(1,a))$
$\quad q_1 \qquad\qquad 2 \qquad\qquad 0 = (2,0)$

$\delta((2,0), a) = (\delta(2, a), \ \delta(0, a))$
$\quad q_2 \qquad\qquad 0 \qquad\qquad 1 = (0, 1)$

$\delta((0, 1), a) = (\delta(0, a), \ \delta(1, a))$
$\quad q_3 \qquad\qquad 1 \qquad\qquad 0 = (1, 0)$

$\delta((1, 0), a) = (\delta(1, a), \ \delta(0, a))$
$\quad q_4 \qquad\qquad 2 \qquad\qquad 1 = (2, 1)$

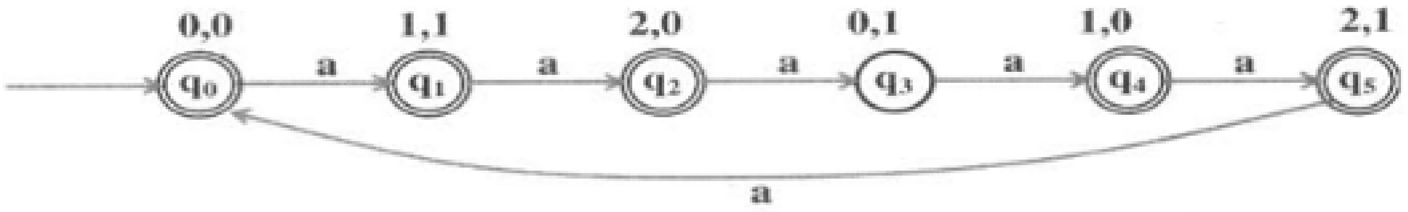
$\delta((2, 1), a) = (\delta(2, a), \ \delta(1, a))$
$\quad q_5 \qquad\qquad 0 \qquad\qquad 0 = (0,0)$



Case 1: To accept strings of *w* such that $|w| \bmod 3 \geq |w| \bmod 2$, the pairs (x, y) such that $x \geq y$ are final states. So, in the above DFA, the final states are:

$$F = \{ (0, 0), (1, 1), (2, 0), (1, 0), (2, 1) \}$$

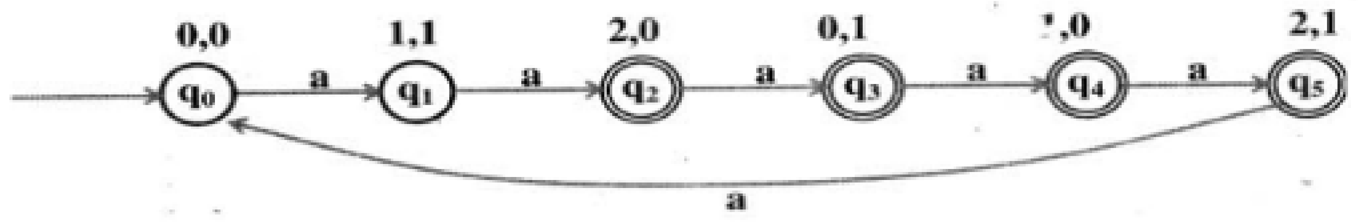
So, the DFA to accept the given language is shown below:



Case 2: To accept strings of *w* such that $|w| \bmod 3 \neq |w| \bmod 2$, $w \in \Sigma^*$ and $\Sigma = \{a\}$ the pairs (x, y) such that $x \neq y$ are final states. So, the final states in the DFA are:

$$F = \{ (2, 0), (0,1), (1, 0), (2, 1) \}$$

So, the DFA to accept the given language is shown below:



Example 5: Now, let us "Obtain a DFA to accept the following language L = {w such that

a) |w| mod 3 ≥ |w| mod 2 where w ∈ Σ* and Σ = {a, b}

b) |w| mod 3 ≠ |w| mod 2 where w ∈ Σ* and Σ = {a,b }

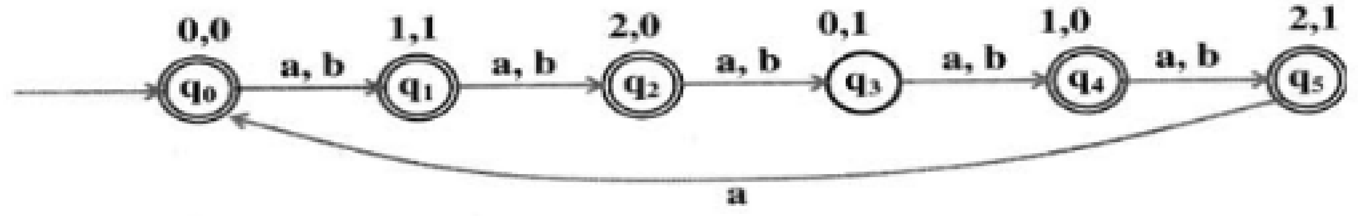
Solution: The solution is exactly similar to the above, but, the extra label *b* should be added along with *a* as shown below:

Case 1: To accept strings of *w* such that |w| mod 3 ≥ |w| mod 2, w ∈ Σ* and Σ = {a,b}.

The pairs (x, y) such that x >= y are final states. So, in the above DFA, the final states are:

F = { (0, 0), (1, 1), (2, 0), (1, 0), (2, 1) }

So, the DFA to accept the given language is shown below:

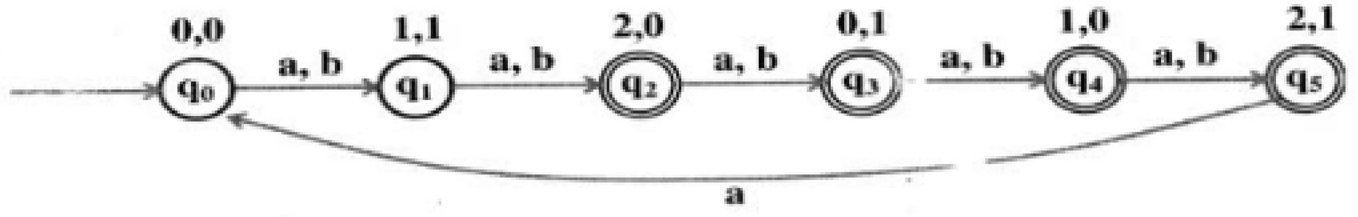


Case 2: To accept strings of *w* such that |w| mod 3 ≠ |w| mod 2, w ∈ Σ* and Σ = {a,b}.

The pairs (x, y) such that x ≠ y are final states. So, the final states in the DFA are:

F = { (2, 0), (0,1), (1, 0), (2, 1) }

So, the DFA to accept the given language is shown below:



Example 6: Let us "Obtain a DFA to accept the language L = { w : |w| mod 5 $\neq$ 0 } on $\Sigma = \{a\}$".

Solution: It is given that L = {w : |w| mod 5 $\neq$ 0} where $\Sigma = \{a\}$ } which indicates that the language consists of strings of a's which are not multiples of 5 a's.

Identify the number of states: The given language can be interpreted as strings of only a's such that the number of a's in the string is not divisible by 5. Note that

|w| mod 5

results in remainder 0, 1, 2, 3 and 4 which results in five cases as shown below:

- **Case 0:** Results in remainder 0: The state is identified as q_0 q_0
- **Case 1:** Results in remainder 1. The state is identified as q_1 q_1
- **Case 2:** Results in remainder 2. The state is identified as q_2 q_2
- **Case 3:** Results in remainder 3. The state is identified as q_3 q_3
- **Case 4:** Results in remainder 4. The state is identified as q_4 q_4

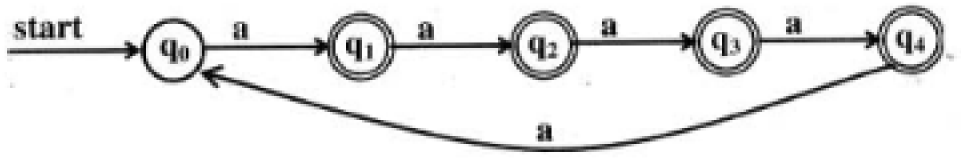
Identify the start state and final state: Before reading any of the input symbols, number of a's will be zero. So, 0 mod 5 = 0 which results in case 0 . Hence, state q0 is start state. Since, it is required to accept string of a's such that |w| mod 5 $\neq$ 0 (i.e., remainder is not zero), other than q_0 all other states can be the final states. So, the states q1, q_2, q_3 and q_4 are final states.

Design: Once the start state and final states are identified, the transitions can be easily obtained as shown below:

- From a state with remainder 0 (case 0) denoted by q_0, on reading *a* results in remainder 1 (case 1) denoted by q_1. So, $\boldsymbol{\delta(q_0, a) = q_1}$
- From a state with remainder 1 (case 1) denoted by q_1, on reading *a* results in remainder 2 (case 2) denoted by q_2. So, $\boldsymbol{\delta(q_1, a) = q_2}$
- From a state with remainder 2 (case 2) denoted by q_1, on reading *a* results in remainder 3 (case 3) denoted by q_3. So, $\boldsymbol{\delta(_q2, a) = q_3}$
- From a state with remainder 3 (case 3) denoted by q_3, on reading *a* results in remainder 4 (case 4) denoted by q_4. So, $\boldsymbol{\delta(q_3, a) = q_4}$
- From a state with remainder 4 (case 4) denoted by q_4, on reading *a* results in remainder 0 (case 0) denoted by q_0. So, $\boldsymbol{\delta(q_2, a) = q_0}$

So, the DFA $M = (Q, \Sigma, \delta, q_0, F)$ is defined as:

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a\}$
- δ = shown in transition diagram as shown below:

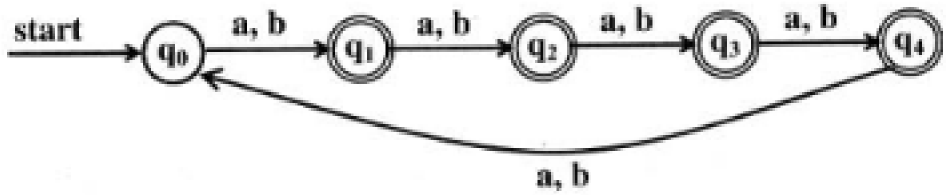


- q_0 = start state
- $F = \{q_1, q_2, q_3, q_4\}$

Example 7: Let us "Obtain a DFA to accept the language $L = \{ w : |w| \bmod 5 \neq 0 \}$ o
$\Sigma = \{a,b\}$".

Solution: The design is exactly similar to the previous problem but each arrow is labeled with *a*
b. So, the DFA $M = (Q, \Sigma, \delta, q_0, F)$ is defined as:

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- δ = shown in transition diagram as shown below:



- q_0 = start state
- $F = \{q_1, q_2, q_3, q_4\}$

Example 8: Let us "Obtain a DFA to accept strings of a's and b's having even number of a'
and even number of b's".

Solution: It is given that L = {w : w has even number of a's and even number of b's.}

Identify the number of states: The string w made up of a's and b's may have:

- *Even number of a's* denoted by $\mathbf{E_a}$
- *Even number of b's* denoted by $\mathbf{E_b}$
- *Odd number of a's* denoted by $\mathbf{O_a}$
- *Odd number of b's* denoted by $\mathbf{O_b}$

So, even/odd a's along with even/odd b's results in following four cases (four states):

- **Case 0:** Strings having Even a's and Even b's is denoted by: q_0 

- **Case 1:** Strings having Even a's, Odd b's is denoted by: q_1 

- **Case 2:** Strings having Odd a's, Even b's is denoted by: q_2 

- **Case 3:** Strings having Odd a's, Odd b's is denoted by: q_3 

Identify the start state and final state: Before reading any of the input symbols, number of a's and number of b's will be zero which represent <u>E</u>ven a's and <u>E</u>ven b's (case 0). So, the state q_0 is the start state.

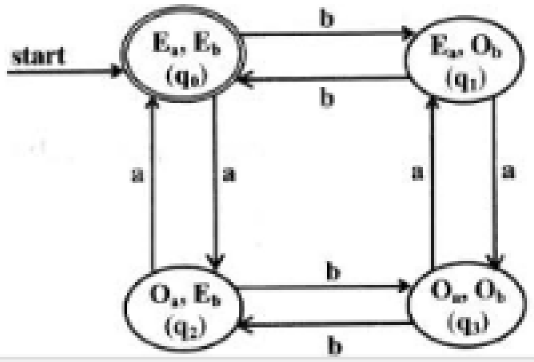
Since, it is required to accept Even a's and Even b's, the state q0 with even a's and even b's is the final state.

Design: Once the start state and final states are identified the transitions can be easily obtained as shown below:

- From a state E_a with even number of a's, on reading input symbol *a*, results in odd number of a's denoted by O_a. So, $\boldsymbol{\delta(E_a, a) = O_a}$
- From a state O_a with odd number of a's, on reading input symbol *a*, results in even number of a's denoted by E_a. So, $\boldsymbol{\delta(O_a, a) = E_a}$
- From a state E_b with even number of b's, on reading input symbol *b*, results in odd number of b's denoted by O_b. So, $\boldsymbol{\delta(E_b, b) = O_b}$
- From a state O_b with odd number of b's, on reading input symbol *b*, results in even number of b's denoted by E_b. So, $\boldsymbol{\delta(O_b, b) = E_b}$

So, the DFA M = (Q, Σ, δ, q0, F) is defined as:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{a, b\}$
- δ = shown in transition diagram



DFA to accept even number of a's and even number of b's

- q_0 = start state
- $F = \{q_0\}$

Note: The language accepted by above DFA can be written as:

$$L = \{w : w \text{ has even number of a's and even number of b's.}\}$$

or

$$L = \{w : \text{Both } N_a(w) \text{ and } N_b(w) \text{ are divisible by } 2\}$$

or

$$L = \{w : \text{Both } N_a(w) \text{ and } N_b(w) \text{ are multiples of } 2\}$$

or

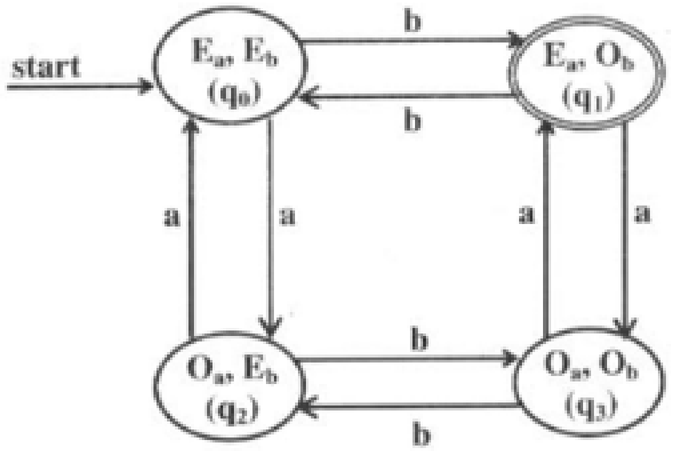
$$L = \{w \mid N_a(w) \bmod 2 = 0 \text{ and } N_b(w) \bmod 2 = 0\}$$

Note: $N_a(w)$ is the total number of a's in the string *w* and $N_b(w)$ is the total number of b's in the string *w*.

Note: The DFA to accept even number of a's and odd number of b's can be obtained by making:

E_a, O_b
(q_1)

as the only final state as shown below:

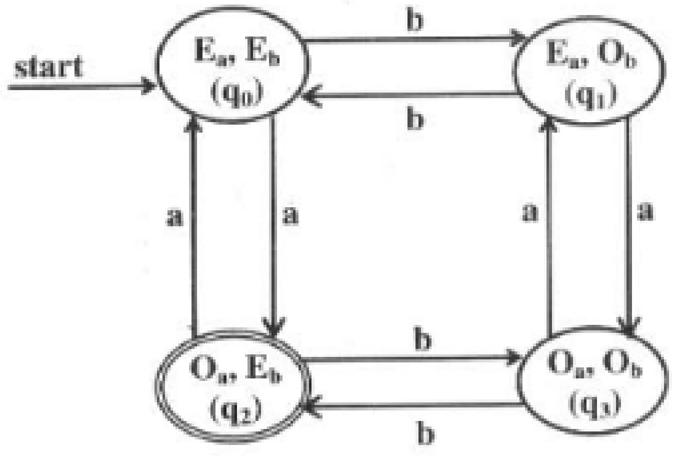


DFA to accept even number ofa's and odd number of b's

Note: The DFA to accept odd number of a's and even number of b's can be obtained by making

O_a, E_b
(q_2)

as the only final state as shown below:

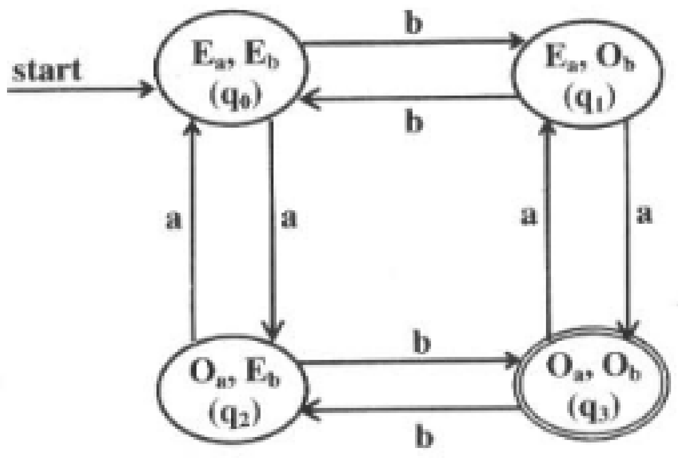


DFA to accept odd number of a's and even number of b's

Note: The DFA to accept odd number of a's and odd number of b's can be obtained by making:

O_a, O_b
(q_3)

as the only final state as shown below:



DFA to accept odd number of a's and odd number of b's

Example 9: Obtain a DFA to accept strings of a's and b's such that

$$L = \{w \mid w \in (a+b)^* \text{ such that } N_a(w) \bmod 3 = 0 \text{ and } N_b(w) \bmod 2 = 0\}\ \}$$

Solution: The $N_a(w)$ mod 3 gives the remainder after dividing number of a's by 3.

The possible remainders are $\{\ 0,\ 1,\ 2\}$ and can be represented as:

$$\begin{array}{l} \{\ 0,\ \ 1,\ \ 2\} \\ \ \ \downarrow\ \ \ \downarrow\ \ \ \downarrow \\ Q_1 = \{\ A_0,\ A_1,\ A_2\} \end{array}$$

The $N_b(w)$ mod 2 gives the remainder after dividing number of b's by 2.

The possible remainders are {0, 1} and can be represented as:

$$Q_2 = \{B_0, B_1\} \quad (2)$$

Identify the states of DFA: Since each state of DFA should keep track of Na(w) mod 3 and Nb(w) mod 2, the possible states of the DFA can be obtained by Q1xQ2 (cross product) and can be represented as shown below:

Q1 x Q2 = {(A0, B0), (A0, B1), (A1, B0), (A1, B1),(A2, B0), (A2, B1)}

where

- A0 indicates that $N_a(w) \bmod 3 = 0$
- A1 indicates that $N_a(w) \bmod 3 = 1$
- A2 indicates that $N_a(w) \bmod 3 = 2$
- B0 indicates that $N_b(w) \bmod 2 = 0$
- B1 indicates that $N_b(w) \bmod 2 = 1$

Identify the start state: Before reading any of the input symbols, number of a's and number of b's will be zero. So, Na(w) mod 3 = 0 and Nb(w) mod 2 = 0 which can be denoted by the state (A0, B0). So, (A0, B0) is the start state.

Identify the final state: Since, it is required to accept the language

$$L = \{w : \underbrace{N_a(w) \bmod 3 = 0}_{A0} \text{ and } \underbrace{N_b(w) \bmod 2 = 0}_{B0}\}$$

the state (A0, B0) is the final state.

Design: Once the start state and final states are identified, the transitions for the input symbol *a* can be obtained as shown below:

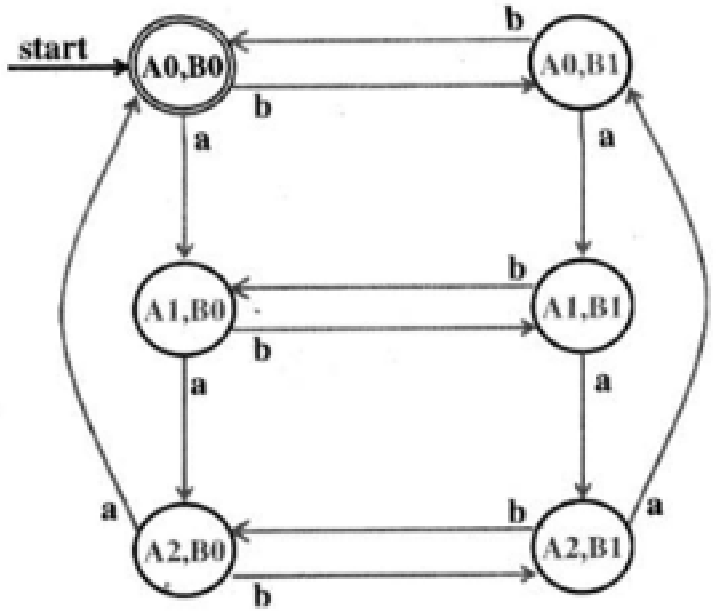
- From state A0, on reading input symbol *a*, the machine should change the state to A1. So, $\boldsymbol{\delta(A_0, a) = A_1}$
- From state A1, on reading input symbol *a*, the machine should change the state to A2. So, $\boldsymbol{\delta(A_1, a) = A_2}$
- From state A2, on reading input symbol *a*, the machine should change the state to A0. So, $\boldsymbol{\delta(A_2, a) = A_0}$

Similarly, the transitions for the input symbol *b* can be obtained as shown below:

- From state B0, on reading input symbol *b*, the machine should change the state to B1. So, $\boldsymbol{\delta(B_0, b) = B_1}$
- From state B1, on reading input symbol *b*, the machine should change the state to B0. So, $\boldsymbol{\delta(B_1, b) = B_0}$

So, the DFA $M = (Q, \Sigma, \delta, q_0, F)$ is defined as:

- $Q = \{(A0,B0), (A0, B1), (A1, B0), (A1, B1), (A2,B0), (A2, B1)\}$
- $\Sigma = \{a, b\}$
- $q_0 = \{(A0, B0)$ is the start state
- $F = \{(A0, B0)\}$
- δ = shown in transition diagram



Note: By changing the final states of DFA various languages can be accepted by DFAas shown below:

- To accept the language $L = \{w \mid w \in (a+b)^* \; N_a(w) \bmod 3 = \underset{\substack{\downarrow \\ \mathbf{A1}}}{1} \text{ and } N_b(w) \bmod 2 = \underset{\substack{\downarrow \\ \mathbf{B0}}}{0}\}$

  make the state (A1, B0) as the final state.

- To accept the language $L = \{w \mid w \in (a+b)^* \; N_a(w) \bmod 3 = \underset{\substack{\downarrow \\ \mathbf{A2}}}{2} \text{ and } N_b(w) \bmod 2 = \underset{\substack{\downarrow \\ \mathbf{B0}}}{0}\}$

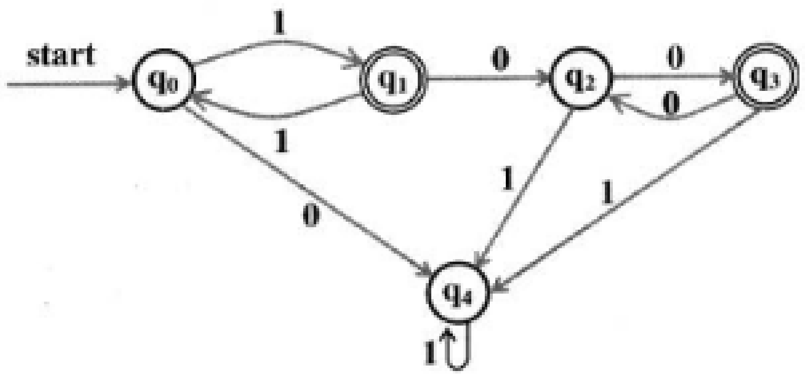
  make the state (A2, B0) as the final state.

- To accept the language $L = \{w \mid w \in (a+b)^* \; N_a(w) \bmod 3 = \underset{\substack{\downarrow \\ \mathbf{A2}}}{2} \text{ and } N_b(w) \bmod 2 = \underset{\substack{\downarrow \\ \mathbf{B1}}}{1}\}$

  make the state (A2, B1) as the final state and so on.

Example 10: Now, let us "Draw a DFA to accept the language: L = {w: w has odd number of 1's and followed by even number of 0's) Completely define DFA and transition function:

Solution: The DFA to accept strings of 0's and 1's such that the string has odd number of 1's and followed by even number of 0's is shown below:



Example 11: Now, let us "Obtain a DFA to accept strings of a's and b's such that the number of a's is divisible by 5 and number of b's is divisible by 3".

Note: Observe that the problem is similar to that the previous problem with more number of states.

Solution: The given language can be interpreted as shown below:

$$L = \{w \mid w \in (a+b)^* \ N_a(w) \bmod 5 = 0 \text{ and } N_b(w) \bmod 3 = 0\}$$

The $N_a(w)$ mod 5 gives the remainder after dividing number of a's by 5.

The possible remainders are { 0, 1, 2, 3, 4} and can be represented as:

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{array}$$

$$Q_1 = \{A0, A1, A2, A3, A4\} \quad (1)$$

The $N_b(w)$ mod 3 gives the remainder after dividing number of b's by 3.

The possible remainders are {0, 1, 2} and can be represented as:

$$\begin{array}{ccc} 0 & 1 & 2 \\ \downarrow & \downarrow & \downarrow \end{array}$$

$$Q_2 = \{B0, B1, B2\} \quad (2)$$

Identify the states of DFA: Since each state of DFA should keep track of Na(w) mod 3 and Nb(w) mod 2, the possible states of the DFA can be obtained by Q1xQ2 (cross product) and can be represented as shown below:

$$
\begin{aligned}
Q1 \times Q2 = \{&(A0, B0), (A0, B1), (A0, B2),\\
&(A1, B0), (A1, B1), (A1, B2),\\
&(A2, B0), (A2, B1), (A2, B2),\\
&(A3, B0), (A3, B1), (A3, B2),\\
&(A4, B0), (A4, B1), (A4, B2)\\
\}&
\end{aligned}
$$

where

- A0 indicates that $N_a(w) \bmod 5 = 0$
- A1 indicates that $N_a(w) \bmod 5 = 1$
- A2 indicates that $N_a(w) \bmod 5 = 2$
- A3 indicates that $N_a(w) \bmod 5 = 3$
- A4 indicates that $N_a(w) \bmod 5 = 4$
- B0 indicates that $N_b(w) \bmod 3 = 0$
- B1 indicates that $N_b(w) \bmod 3 = 1$
- B2 indicates that $N_b(w) \bmod 3 = 2$

Identify the start state: Before reading any of the input symbols, number of a's and number of b's will be zero. So, $N_a(w) \bmod 5 = 0$ and $N_b(w) \bmod 3 = 0$ which can be denoted by the state (A0, B0). So, (A0, B0) is the start state.

Identify the final state: Since, it is required to accept the language

$$L = \{w : \underset{\textbf{A0}}{\underset{\downarrow}{N_a(w) \bmod 5 = 0}} \text{ and } \underset{\textbf{B0}}{\underset{\downarrow}{N_b(w) \bmod 3 = 0}}\}$$

the state (A0, B0) is the final state.

Design: Once the start state and final states are identified, the transitions for the input symbol *a* can be obtained as shown below:

- From state A0, on reading input symbol *a*, the machine should change the state to A1. So, $\boldsymbol{\delta(A_0, a) = A_1}$
- From state A1, on reading input symbol *a*, the machine should change the state to A2. So, $\boldsymbol{\delta(A_1, a) = A_2}$
- From state A2, on reading input symbol *a*, the machine should change the state to A3. So, $\boldsymbol{\delta(A_2, a) = A_3}$
- From state A3, on reading input symbol *a*, the machine should change the state to A4. So, $\boldsymbol{\delta(A_3, a) = A_4}$
- From state A4, on reading input symbol *a*, the machine should change the state to A0. So, $\boldsymbol{\delta(A_4, a) = A_0}$

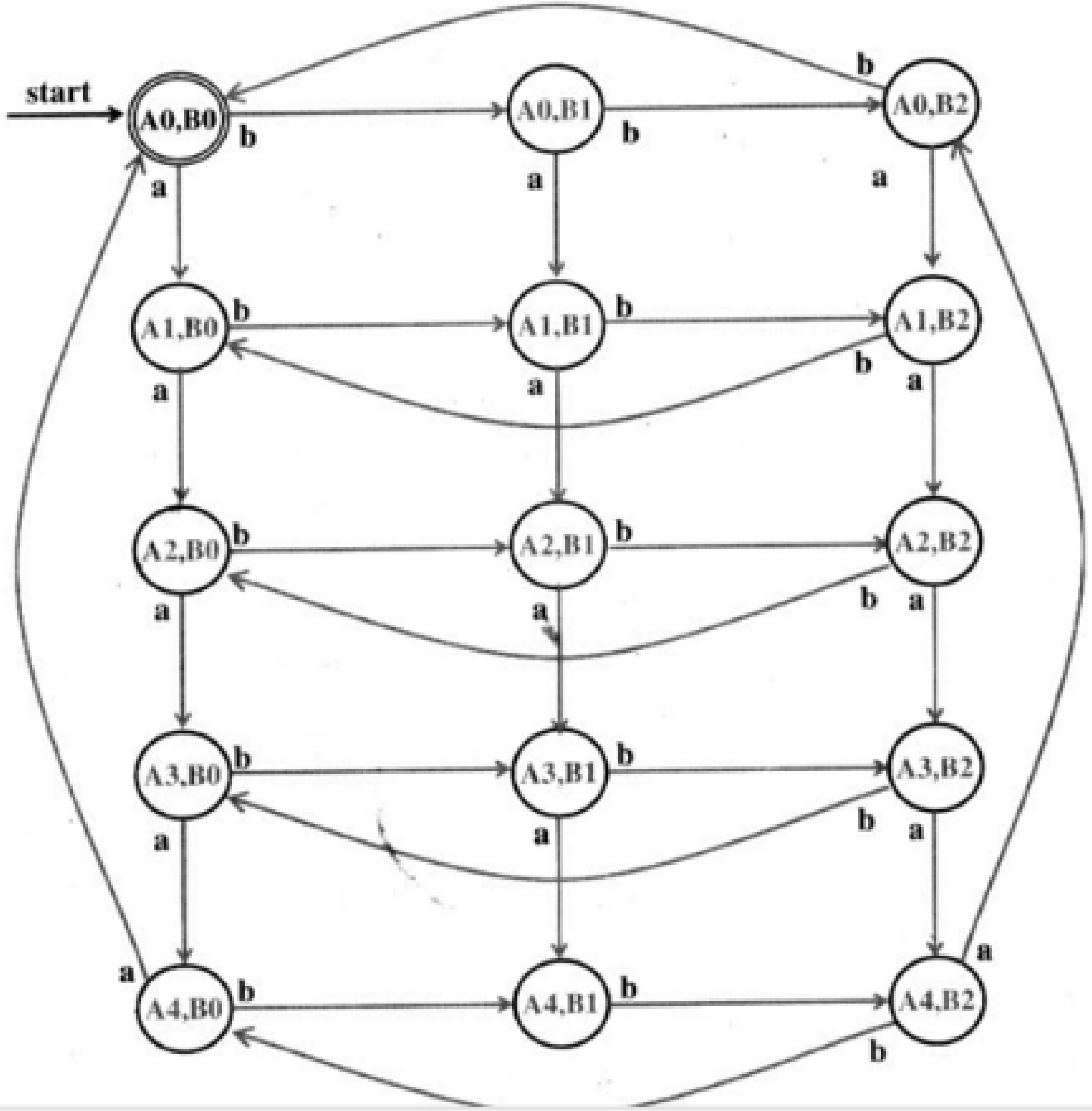
Similarly, the transitions for the input symbol *b* can be obtained as shown below:

- From state B0, on reading input symbol *b*, the machine should change the state to B1. So, $\boldsymbol{\delta(B_0, b) = B_1}$
- From state B1, on reading input symbol *b*, the machine should change the state to B2. So, $\boldsymbol{\delta(B_1, b) = B_2}$

- From state B2, on reading input symbol *b*, the machine should change the state to B0. So, $\delta(\mathbf{B_2}, \mathbf{b}) = \mathbf{B_0}$

So, the DFA M = (Q, Σ, δ, q0, F) is defined as:

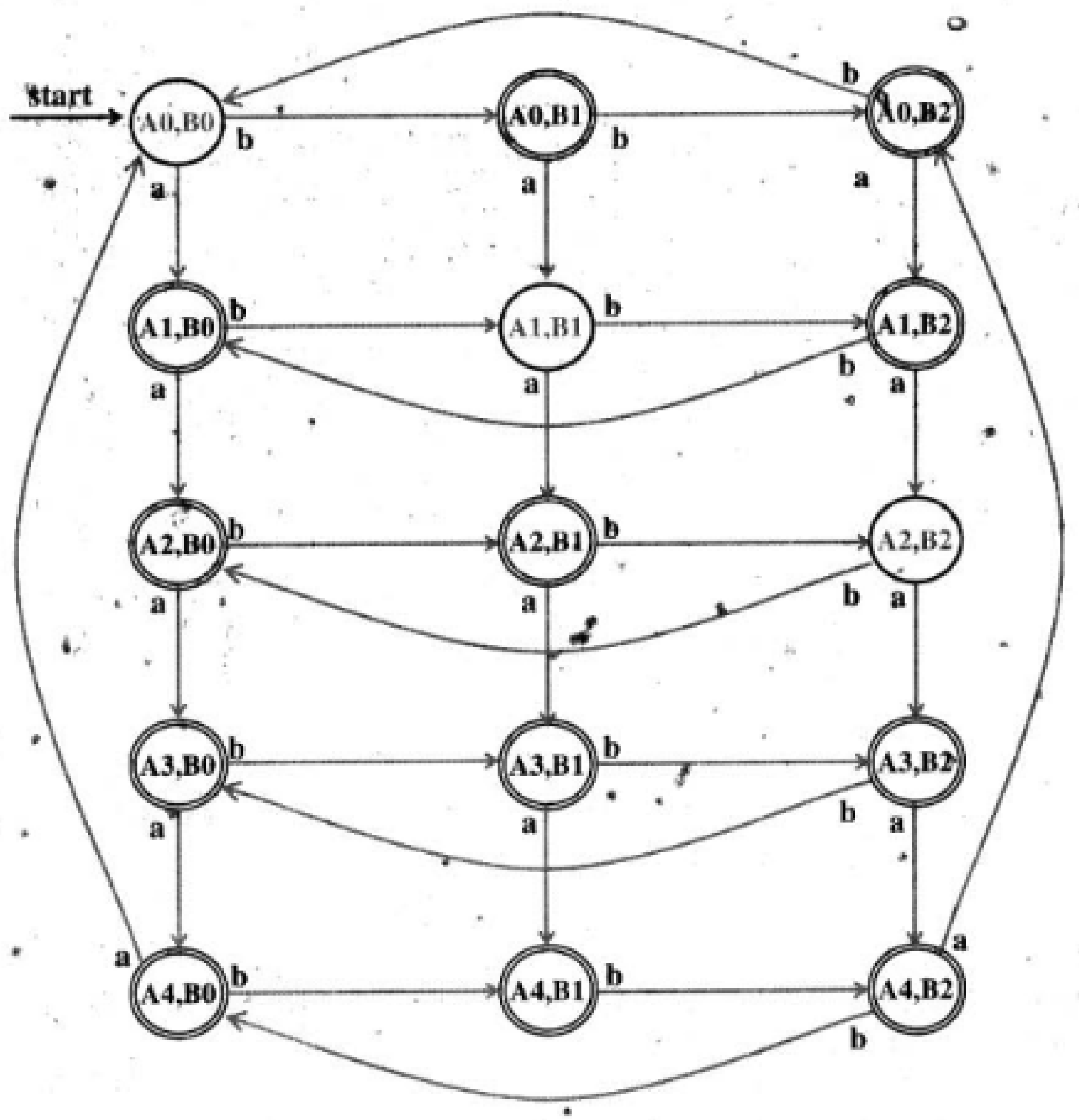
- Q = {(A0, B0), (A0, B1), (A0, B2),
  (A1, B0), (A1, B1), (A1, B2),
  (A2, B0), (A2, B1), (A2, B2),
  (A3, B0), (A3, B1), (A3, B2),
  (A4, B0), (A4, B1), (A4, B2) }
- Σ = {a, b}
- q_0 = (A0, B0) is the start state
- F = {(A0, B0)}
- δ = shown below using the transition diagram



Now, let us "Construct a DFA to accept the following language":

$L = \{w : n_a(w) \bmod 5 \neq n_b(w) \bmod 3\}$

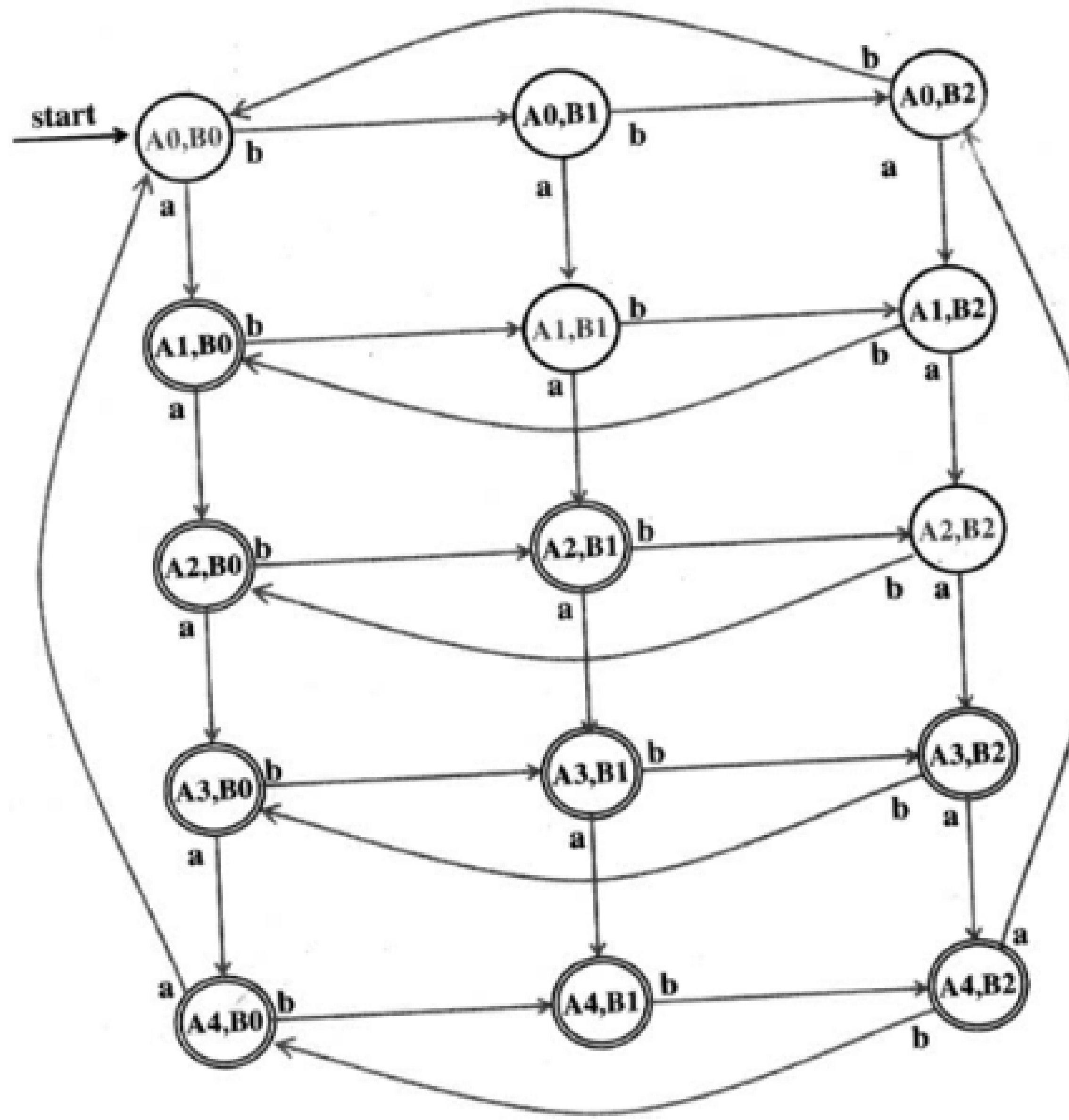
Note: The design is exactly same as the above problem, Except the states (A0, B0), (A1, B1) and (A2, B2), the rest of the states are final states. The DFA to accept the given language is shown below:



Now, let us "Construct a DFA to accept the following language":

$L = \{w : n_a(w) \bmod 5 > n_b(w) \bmod 3\}$

Note: The design is exactly similar to the above problem. But, in all possible states (A_i, B_j), the index i should be greater than index j. So, the final DFA to accept the above language is shown below:



Disadvantages of DFA

- Constructing DFA is difficult.
- The DFA can not guess input.
- DFA is not very powerful.
- At any point of time, the DFA is in only one state. So, any point of time DFA can not be in several states.

Why NFA?

- Very easy to construct.
- A “non-deterministic” Finite Automaton has ability to guess its input.
- A NFA is more powerful than DFA.
- It has power to stay several state at once.
- An NFA is efficient mechanism to describe complicated language concisely.