Ques Explain the concept of Automation.

Ans The word automata itself, closely related to the word "automation", denotes automatic processes carrying out the production of specific processes. Simply stated, automata theory deals with the logic of computation with respect to simple machines, referred to as automata.

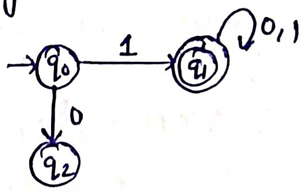
Ques Explain

(a) Transition Diagram

A transition diagram or state transition diagram is a directed graph which can be constructed as follows:

* There is a node for each state in Q, which is represented by the circle.
* There is a directed graph edge from node q to node p labeled if $\delta(q,a) = p$.
* In start state, there is an arrow with no source.
* Accepting states or final states are indicating by a double circle.

Example

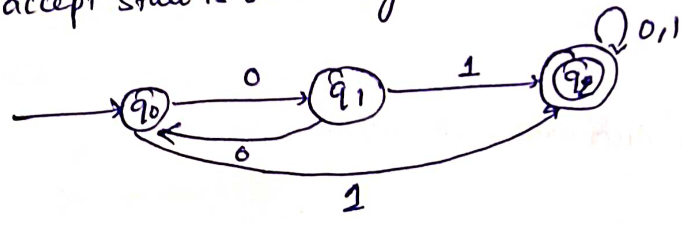


(b) Transition Table

The transition table is basically a tabular representation of the transition function. It takes two arguments (a state and a symbol) and returns state (the next state).

A transition table is represented by the following things:

* Columns correspond to input symbols.
* Rows correspond to states.
* Entries correspond to the next state.
* The start state is denoted by an arrow with no source.
* The accept state is denoted by a star.



Transition table :-

Present State	Next State for Input 0	Next State for Input 1
$\rightarrow q_0$	q_1	q_2
q_1	q_0	q_2
$*q_2$	q_2	q_2

Ques Difference between DFA and NFA.

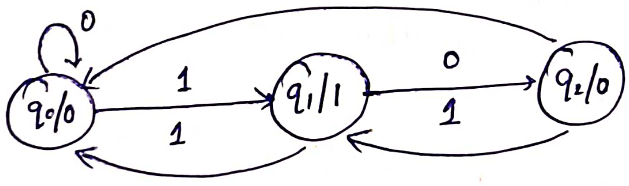
Ans

DFA	NFA
1. DFA Stands for Deterministic Finite Automata.	NFA stands for Nondeterministic Finite Automata.
2. For each symbolic representation of the alphabet, there is only one state transition in DFA.	No need to specify how does the NFA react according to some symbol.
3. DFA cannot use Empty string transition.	NFA can use Empty string transition.
4. It can be understood as one machine.	It can be understood as multiple little machines computing at the same time.
5. In DFA, the next possible state is distinctly set.	In NFA, each pair of state and input symbol can have many possible next states.
6. $\delta : Q \times M \rightarrow Q$ i.e. next possible state belongs to Q.	$\delta : Q \times M \rightarrow 2^Q$ i.e. Next possible state belongs to power set of Q.

Ques Explain the concept of Automata with output with example.

As Finite automata can also be used as an output device. A finite automata with output is similar to finite automata except that the additional capacity of producing output. In a formal way it is also known as Finite State Machine or Transducer.

Example



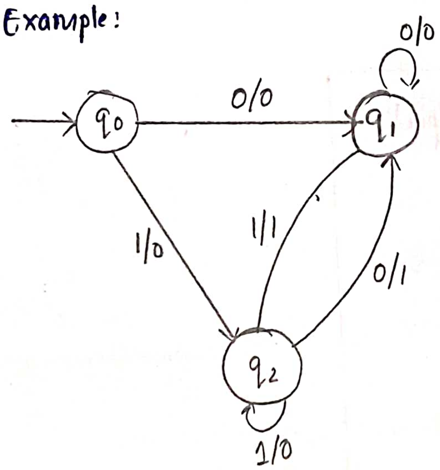
There are two machines for Automata with output

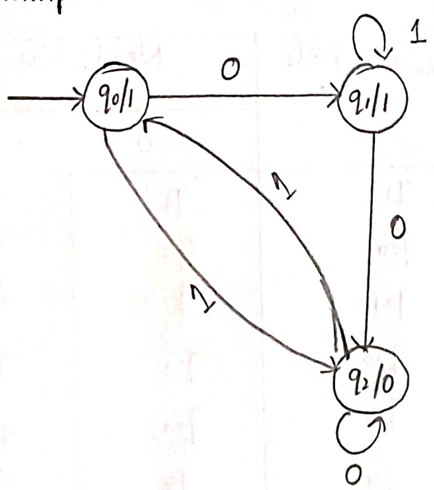
1. Mealy Machine
2. Moore Machine

Que What is difference between Mealy and Moore Machine?

Ans

	Mealy Machine	Moore Machine
1.	Output depends upon present state as well as present Input.	Output depends only upon the present state.
2.	If input changes, output also changes.	If input changes, output changes not.
3.	Less number of states are required.	More states are required.
4.	Asynchronous output generation.	Synchronous output and state generation.
5.	Output is placed on transition.	Output is placed on state.
6.	It is difficult to design.	It is easy to design.
7.	Example:	Example:





Que Explain the procedure for transforming a mealy machine into moore machine.

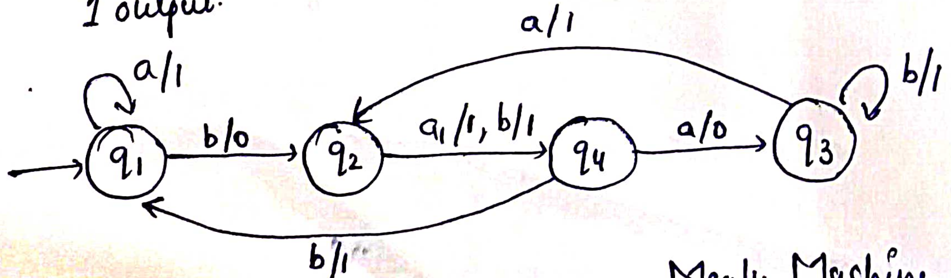
Ans The following steps are used for converting Mealy machine to the Moore machine:

Step 1: For each state (Qi), calculate the number of different outputs that are avilable in the transition table of the Mealy machine.

Step 2: Copy state Q, if all outputs of Qi are the same. Break qi into n states as Qin, if it has n distinct outputs n=0,1,2...

Step 3: If the output of initial state is 0, insert a new initial state which gives 1 output.

Ex



Mealy Machine

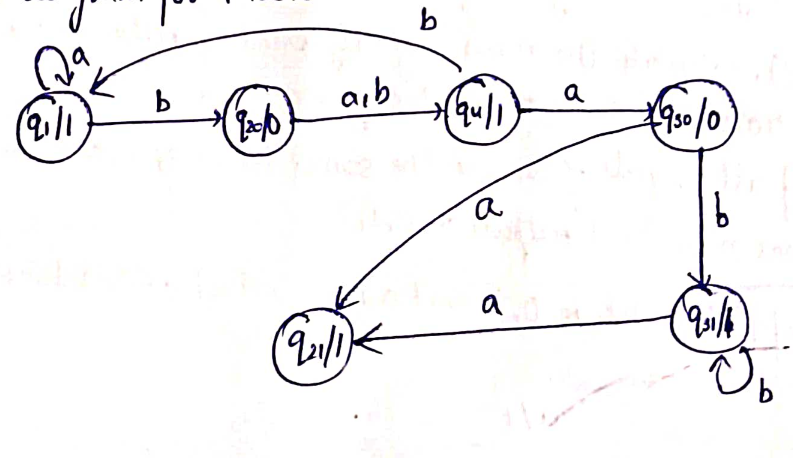
Transition Table for Mealy Machine

Present State	Next State			
	a		b	
	State	Output	State	Output
q_1	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	1	q_3	1
q_4	q_3	0	q_1	1

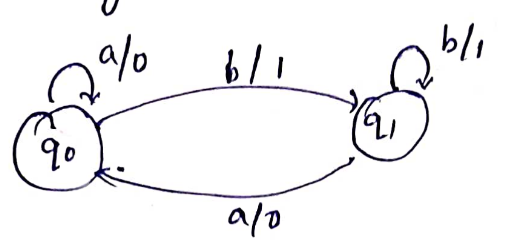
Transition Table for Moore Machine

Present State	Next state		Output
	a = 0	a = 1	
q_1	q_1	q_2	1
q_{20}	q_4	q_4	0
q_{21}	ϕ	ϕ	1
q_{30}	q_{21}	q_{31}	0
q_{31}	q_{21}	q_{31}	1
q_4	q_3	q_4	1

Transition diagram for Moore Machine



Equivalent Mealy Machine



Ques Explain Construction of DFA from NDFA.

Ans An NFA can have zero, 1 or more than one move from a given state on a given input symbol. An NFA can also have NULL moves. On the other hand, DFA has one and only one move from a given state on a given input symbol.

CONVERSION NFA → DFA

Suppose there is an NFA, N< $Q, \Sigma, q_0, \delta, F$ > which recognizes a language L. Then the DFA D< $Q', \Sigma, q_0, \delta', F'$ > can be considered for language L as:

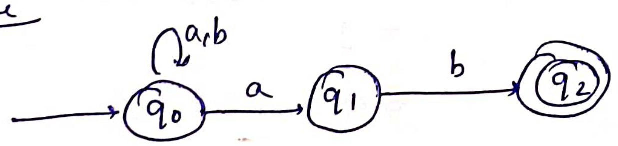
Step 1: Initially $Q' = \phi$

Step 2: Add q_0 to Q'.

Step 3: For each state in Q', find the possible set of states for each input symbol using transition function of NFA. In this set of states is not in Q', add it to Q'.

Step 4: Final state of DFA will all states with contain F.

Example



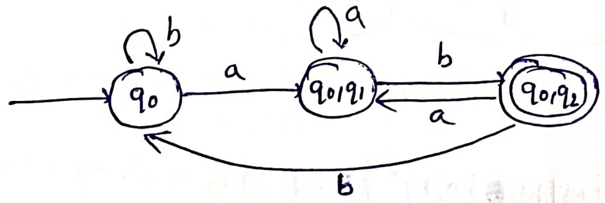
Transition table for given NFA:

State	a	b
q_0	q_0, q_1	q_0
q_1	–	q_2
q_2	–	–

Transition Table for DFA

State	a	b
q_0	$\{q_0, q_1\}$	q_0
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	q_0

Hence required DFA



Ques Write procedure for construction of Minimum Automata or State Minimization of DFA.

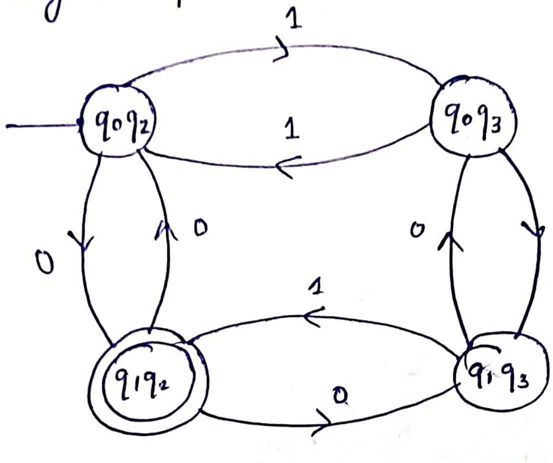
Ans Minimization of DFA means reducing the number of States from given FA. Thus we get the Finite State Machine with redudant states after minimizing the FSM.

We have to follow the various steps to minimize the DFA.

1. Remove all the states that are unreachable from initial state via any set of transition of DFA.
2. Draw the transition table for all pair of states.
3. Now, split the transition table into two tables T1 and T2. T1 contains all final states and T2 contains non final states.
4. Find similar rows such that $S(q,a) = p$, $S(r, a = p)$
   That means, find the two states which have the same value of a and b and remove one of them.
5. Repeat step 3 untill we find no similar rows avilable in the transition table T1.
6. Repeat step 3 and step 4 for table T2 also.
7. Now combine the reduced T1 and T2 tables. The combined trancition table is the transition table of minimized DFA.

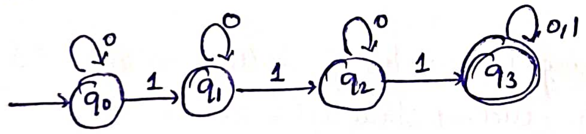
Ques Design a DFA to accept odd and even numbers represented using binary notations.

Ans This machine accept the language which contains odd no. of 0's and even no. of 1's. As we know that q_1 indicate odd no. of 0's and q_2 indicates even no. of 1's. So, the final states of the required DFA will contain both q_1 and q_2.

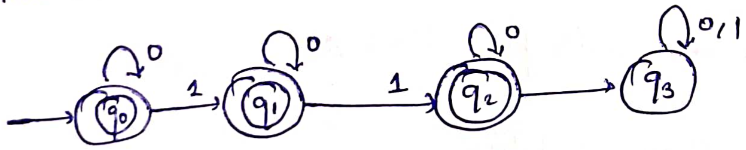


Ques Design DFA which accept $L(M) = \{w | w \in \{0,1\}^*\}$ and w is a string that does not contain consecutive 1's.

Ans When three consecutive 1's occur the DFA will be



Here two consecutive 1's or single 1 is acceptable, hence



The stages q_0, q_1, q_2 are the final states. The DFA will generate the strings that do not contain consecutive 1's like 10, 110, 101, ... etc.

Ques Construct a DFA the language recognized by the automaton being $L = \{0^m 1^n | m \geq 0 \text{ and } n \geq 1\}$.

Ans

Ques Differentiate between strings and words of language using example.

Ans A string is any combination of the letters of an alphabet where as the words of a language are the strings that always made according to certain rules used to define that language. For example

Alphabet $\Sigma = \{a, b\}$. Here a, b are letters of the alphabet

We can make a lot of string from these letters a and b. For ex. a, b, aa, ab, ba...

But when we define a languge over this alphabet having no a's and only odd no. of b's. The the words of this language would have only those strings that have only odd number of b's and no a's.

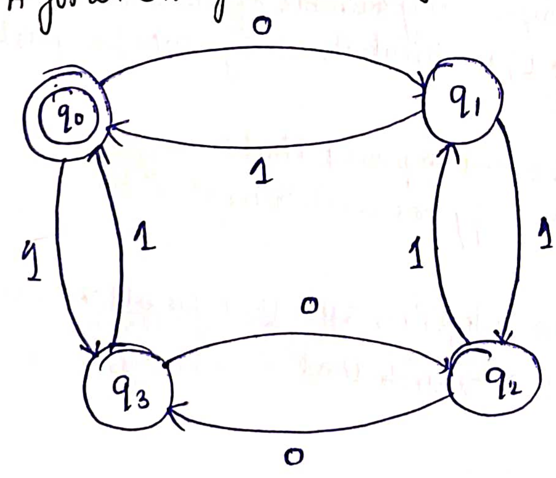
Ques Differentiate between kleen closure and positive closure.

Ans

Kleen Closure	Positive Closure
* Infinite set of all possible strings of all possible lengths including ϵ.	Infinite set of all possible lengths excluding ϵ.
* Denoted by Σ^*	Denoted by Σ^+
* $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots$	$\Sigma^+ = \Sigma^* - \{\epsilon\}$

Ques Draw DFA for all strings over $\{0,1\}$ consisting of even number of 0's and 1's.

Ans



Ques Draw a finite automaton that accepts all binary strings where 0's and 1's are alternative.

Ans Regular Expression of set of all strings of 0's and 1's starting with two zeroes:

$00(0+1)^*$

Regular Expression of set of all strings of 0's and 1's having even no. of 0's followed by odd numbers of 1's:

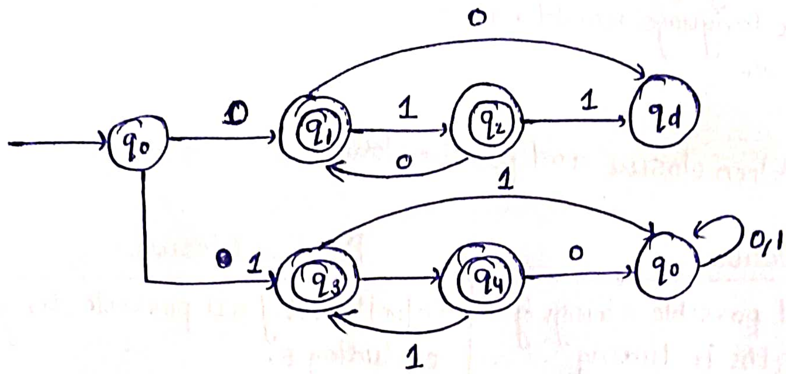
$(00)^* 1 (11)^*$

Regular expression of set of all strings of 0's and 1's containing at least one 0 and atleast two 1's :

$00^*11(0+1)^* + 0111^*(0+1)^*$

Final Expression is :-

$(01)^* + (10)^* + 0(10)^* + 1(01)^*$



Finite Automata of Regular Expression for alternate 0's and 1's

<u>Ques</u> State and prove pumping lemma theorem for Context Free Languages. By using Pumping lemma prove that $L = \{a^n b^n a^n \mid n > 0\}$ is not context free language.

<u>Ans</u> Pumping Lemma for CFL states that for any Context Free Language L, it is possible to find two substrings that can be 'pumped' any number of times and still be in the same language. For any language L, we break its strings into five parts and pump second and fourth substring.

Pumping Lemma, here also, is used as a tool to proved that a language is not CFL. Because, if any one string does not satisfy one conditions, then the language is not CFL.

Thus, if L is a CFL, there exists an integer n, such that for all $x \in L$ with $|x| \geq n$, there exists $u, v, w, x, y \in \Sigma^*$, such that $x = uvwxy$, and

1. $|vwx| \leq n$
2. $|vx| \geq 1$
3. for all $i \geq 0$: $uv^iwx^iy \in L$

Let $L = \{a^n b^n a^n \mid n > 0\}$. By using pumping lemma show that L is not CFL.

<u>Step 1:</u> Let L is a CFL and we will get contradiction. Let n be a natural number obtained by pumpping lemma.

<u>Step 2:</u> Let $w = a^n b^n a^n$ where $|w| >= n$. By using pumping lemma we can write $w = uvwxy$ with $|vy| >= 1$ and $|vxy| <= n$.

Context-free languages are not closed under :-

* Intersection
* Complement

Ques 18 Find a regular expression corresponding to each of the following subset $\{0,1\}$:

(i) The language of all strings contain atleast two 0's

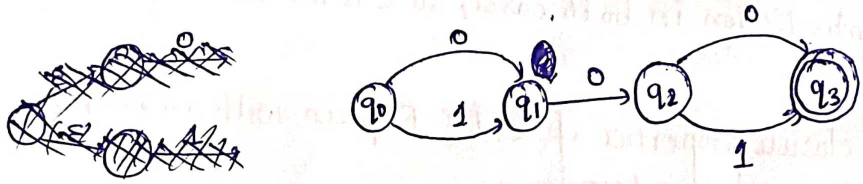
Ans $r = (0+1)^* 0 (0+1)^* 0 (0+1)^*$

(ii) The language of all strings containing atmost two 0's

Ans $r = 1^* + 1^* 0 1^* + 1^* 0 1^* 0 1^*$

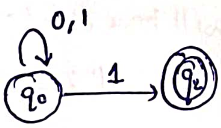
Ques Convert $(0+1)^* 0 (0+1)$ to NFA.

Ans

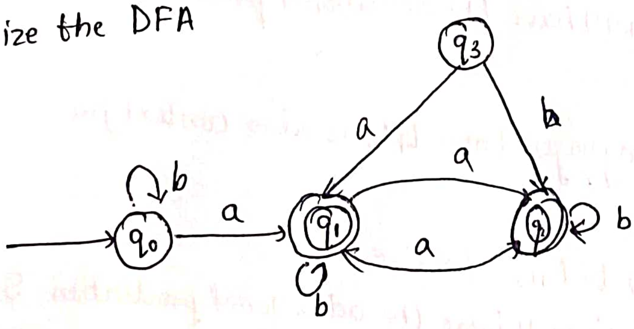


Ques Convert $(0+1)^* 1$ to NFA.

Ans



Ques Minimize the DFA



(a)

Ans Here, q_3 is unreachable state. So, we have to remove this state.

Transition Table :-

State	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_1	q_2
q_3	q_1	q_2

Step 3 Case 1:

When both v and y contains equal number of a's and b's.

Let $i = 2$

Then $uv^2xy^2z = a^{n+k}b^{n+k}a^n \notin L$

Case 2:

All words in $a^nb^na^n$ have one occurence of substring ab or ba no matter what n is.

Let $i = 2$

Then uv^2xy^2z will have more than one substring ab or ba, so it cannot be in the form $a^nb^na^n$.

Hence, $uv^2xy^2z \notin L$.

There is contradiction in both cases, so L is not context free language.

Ques What are different closure properties of CFL? Explain with an example.

Ans Context-free languages are closed under –

1. Union

Let L_1 and L_2 be two context free languages. Then $L_1 \cup L_2$ is also context free.

Example

Let $L_1 = \{a^nb^n, n>0\}$. Corresponding Grammar G_1 will have $P: S1 \rightarrow aAb|ab$

Let $L_2 = \{c^md^m, m \geq 0\}$. G_2 $P: S_2 \rightarrow cBb|\epsilon$

Union of L_1 and L_2, $L = L_1 \cup L_2 = \{a^nb^n\} \cup \{c^md^m\}$

The corresponding grammar G will have the additional production $S \rightarrow S1|S2$

2. Concatenation

If L_1 and L_2 are context free languages, then L_1L_2 is also context free.

Example

Union of Languages L_1 and L_2, $L = L_1L_2 = \{a^nb^nc^md^m\}$

The corresponding grammar G will have the additional production $S \rightarrow S1S2$

3. Kleene Star

If L is a context free language, then L^* is also context free.

Example

Let $L = \{a^nb^n, n \geq 0\}$.

Corresponding Grammar G will have $P: S \rightarrow aAb|\epsilon$

Kleen Star $L_1 = \{a^nb^n\}^*$

The corresponding grammar G_1 will have additional production

$S1 \rightarrow SS1|\epsilon$

Now we have to find if these states are equivalent classes.

Accepting states — Non-accepting states

$\pi_0 = \{q_1, q_2\}$ $\quad$ $\{q_0, q_3\}$

G_1 $\quad$ G_2

Here q_1 and q_2 are not equivalent hence put in other groups.

$\pi_1 = \{q_1\}$ $\quad$ $\{q_2$ $\quad$ $\{q_0, q_3\}$

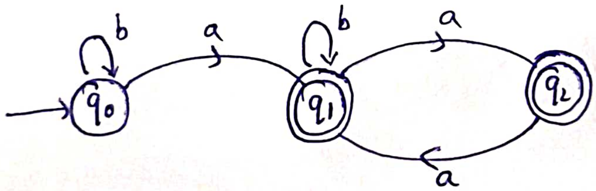
Now, check for q_0 and q_3

q_0 and q_3 are not equivalent. Hence

$\pi_2 = \{q_1\}$ $\quad$ $\{q_2\}$ $\quad$ $\{q_0\}$ $\quad$ $\{q_3$

Hence, states are

$\pi_2 = \{q_1\} \{q_2\} \{q_0\}$



This is the required minimized DFA.