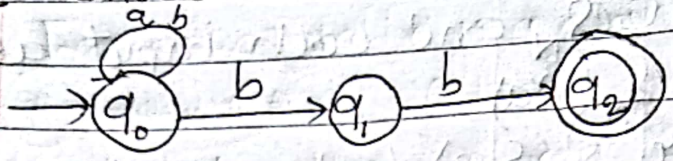
practice

Example: Convert NFA to DFA



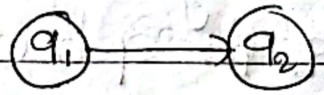
Soln>

Notations for DFA's (Methods for describing transition function)

There are two preferred notations for describing this class of automata;

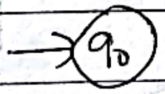
(i) Transition Diagram (State Diagram)

It is a graphical representation in which states are represented by circles, transitions are represented by arrows with input symbols as labels,

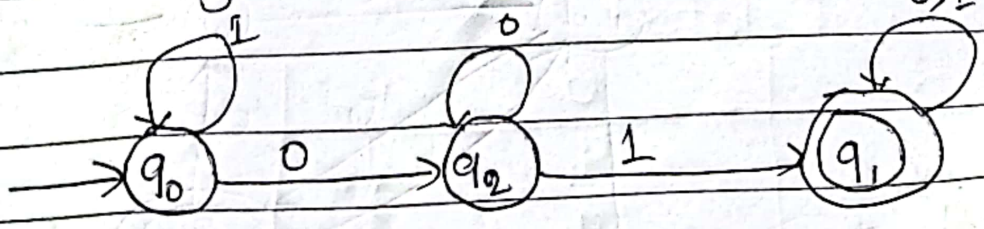


and accepting state are designated by double instead of single circles.

The initial state will have an arrow pointing to it that doesn't come from another state



Example:- The transition diagram represents for the DFA accepting all strings with a substring 01 is shown in figure below:



(ii). Transition Table

Transition table is a conventional, tabular representation of the transition function δ that takes the arguments from $Q \times \Sigma$ and returns a value which is one of the states of the automation. The row of the table corresponds to the states while column corresponds to the input symbol. The starting state in the table is represented by $\rightarrow$ followed by the state i.e. $\rightarrow q$, for q being start state, whereas final states as q, for q being final state.

The transition table for the DFA accepting all strings over $\{0,1\}$ having substring 01 is given below:

δ:	0	1
$\rightarrow q_0$	q_2	q_0
q_1	q_1	q_1
q_2	q_2	q_1

1.5. Minimization of DFA

DFA minimization stands for converting a given DFA to its equivalent DFA with minimum number of states. DFA minimization is also called as optimization of DFA and uses partitioning algorithm.

Steps to minimize the DFA:

step1: Remove all the states that are unreachable from the initial state via any set of the transition of DFA.

step2: Draw the transition table for all pair of states.

step3: Now split the transition table into two tables T1 and T2. T1 contains all final states and T2 contains non-final states.

step4: Find similar rows from T1 such that:

$$\delta(q, a) = p$$
$$\delta(r, a) = p$$

That means, find the two states which have the same value of a and b and remove one of them.

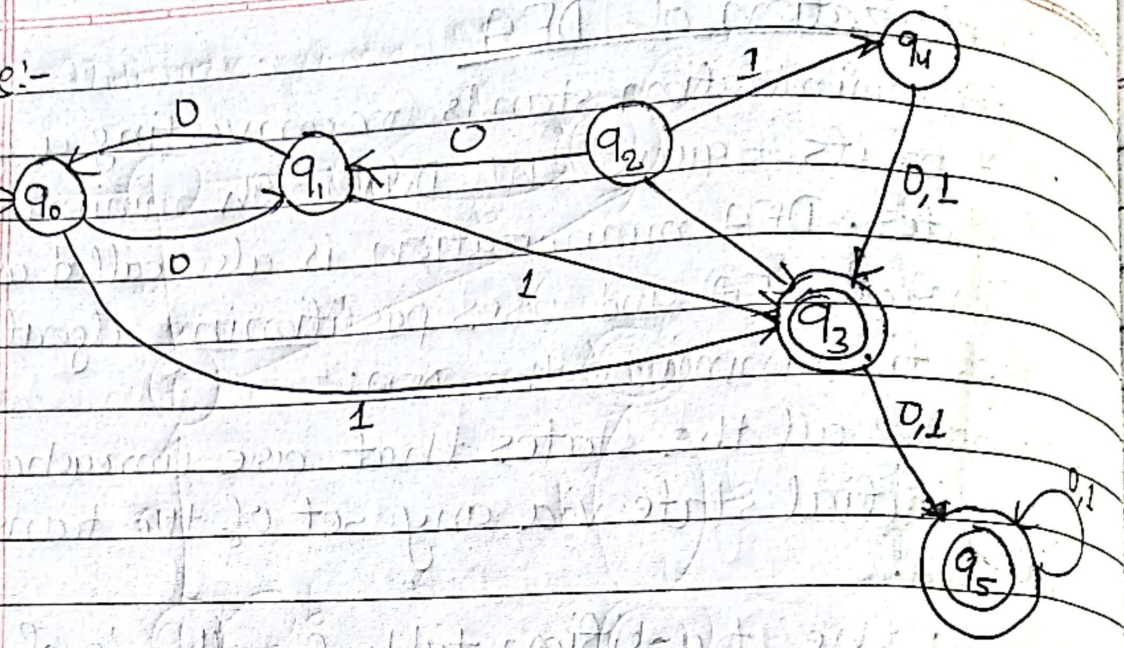
step5: Repeat step3 until we find no similar rows available in the transition table T1.

step6: Repeat step3 and step4 for table T2 also.

step7: Now combine the reduced T1 and T2 tables. The combined transition table is the transition table of ~~min~~ minimized DFA.

HK-27

Example:-



Solution, Step1: In the given DFA, q_2 and q_4 are the unreachable states so remove them.

Step2: Draw the transition table for the rest of the states

States	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_0	q_3
q_3	q_5	q_5
q_5	q_5	q_5

Step3: Now divide rows of transition table into two state as:

(i) One set contains those rows, which start from non-final states:

State	0	1
q_0	q_1	q_3
q_1	q_0	q_3

(ii) Another set contains those rows, which starts from final states:

State	0	1
q_3	q_5	q_5
q_5	q_5	q_5

Step4: Set 1 has no similar rows so set 1 will be the same.

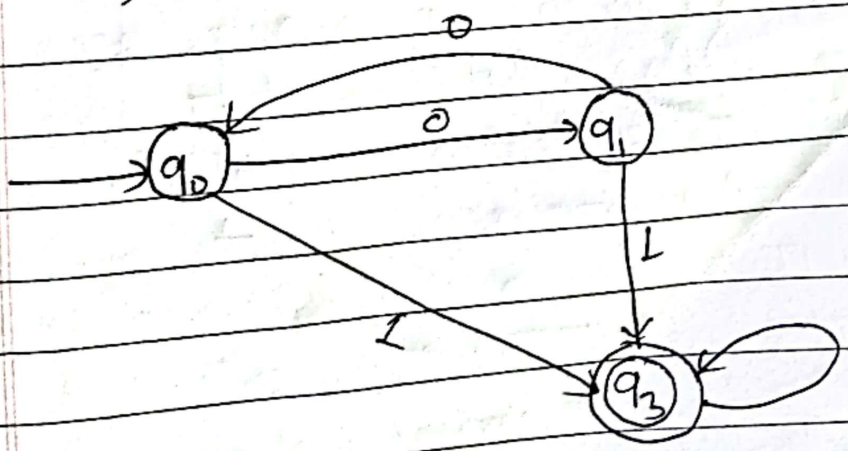
Step5: In set 2, row1 and row2 are similar since q_3 and q_5 transit to the same state on 0 and 1. So skip q_5 and then replace q_5 by q_3 in the rest.

State	0	1
q_3	q_3	q_3

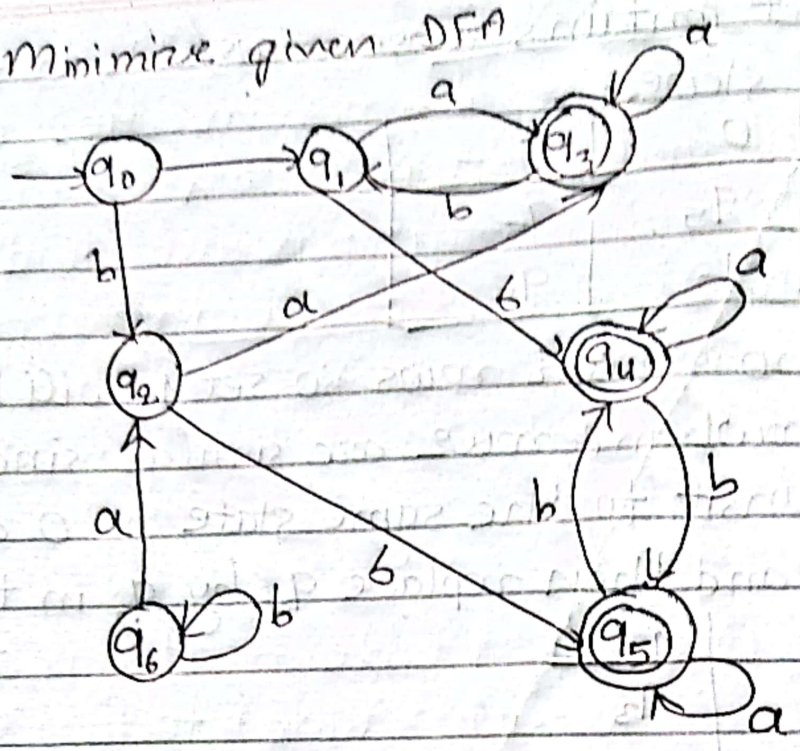
Step6: Now combine set1 and set2 as:

State	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_0	q_3
q_3	q_3	q_3

Now, it is transition table of minimized DFA



Example: Minimize given DFA



states	a	b
q_0	q_1	q_2
q_1	q_3	q_4
q_2	q_3	q_5
q_3	q_3	q_1
q_4	q_4	q_5
q_5	q_5	q_4
q_6	q_2	q_6

1.6 Regular Expressions

The language accepted by finite automata is described by simple expression called regular expression. The notation involves combination of strings of symbols from some alphabet Σ, parenthesis ( ) and the operators $+, \cdot, *$.

Here, Σ is the set of symbols in the language.

$+$ represents the union/positive closure

$\cdot$ represents the concatenation.

$*$ star closur/Kleen Closure

Closure Representation

Σ^+ = It is a positive closure that represents a set of all strings except Null or ε strings

$\Sigma^+ = g^+ = \{g, gg, ggg, gggg, \ldots\}$

→ Here, + means one or more occurrences.

Σ^* = It is "Kleen Closure" that represents the occurrence of certain alphabets for given language alphabets from zero to the infinite number of times. In which ε-string is also included. Example:

$\Sigma^* = g^* = \{\varepsilon, g, gg, ggg, gggg, \ldots\}$

→ Here, * means 0 or more occurrences

A regular expression can be formally defined as follows:

1. ϕ is a regular expression denoting the empty language. $(L(\phi) = \{\})$
2. ε or λ is a Regular Expression denoting the language contents and empty string.
   $(L(\varepsilon) = \{\varepsilon\})$
3. X is a regular expression which indicates the language contents only $L = \{x\}$
4. If X is a Regular Expression, denoting the language $L(X)$ and Y is a Regular Expression denoting the language $L(Y)$, then
   - $X+Y$ is a regular expression corresponding to the language $L(X) \cup L(Y)$ where
     $L(X+Y) = L(X) \cup L(Y)$
   - $X \cdot Y$ is a Regular Expression corresponding to the language $L(X) \cdot L(Y)$ where $L(X \cdot Y) = L(X) \cdot L(Y)$
   - R^* is a Regular Expression corresponding to the language $L(R^*)$ where $L(R^*) = (L(R))^*$
5. The expression obtained by applying any of the rules from 1 to 4 are regular expression.

Note:-

→ The languages accepted by some regular expressions are referred to as Regular languages.

→ A regular expression can also be described as a sequence of patterns that defines a string.

→ Regular expression ~~can also be described as a sequence of pattern~~ are used to match character combinations in strings. String searching algorithm used this pattern to find the operations on a string.

Example

Regular expression	Meaning
$(00)^*$	Regular expression containing even number of 0's excluding null string.
$(11)^*1$	Regular expression containing odd number of 1's.
$(0+1)^*011$	Regular expression containing set of strings ending with 011.
$ab(a+b)^*$	Regular expression containing set of strings ~~start~~ a's and b's and starting with the symbol ab.
$(aa)^*(bb)^*b$	Regular expression containing set of strings having even num of a's must be followed by od number of b's.
$(a+b)^*$	Set of strings of a's and b's o any length including the NULL st

$aa^*bb^*cc^*$ ⟶ Regular expression containing at least one a, followed by at least one b followed by at least one c.

$a^*b^*c^*$ ⟶ Regular expression containing n number of a's followed by n number of b's and followed by n number of c's when n≥0

$a^+b^+c^+$ ⟶ Set of strings consisting of at least one 'a' followed by string consisting of at least one 'b' followed by string consisting of at least one 'c'.

$(a+b)^*(a+bb)$ → Set of string of a's and b's ending with either a or bb.

✡ Regular Expressions and Regular set

Regular Expressions	Regular Set
$(0+10^*)$	$L=\{0,1,10,100,1000,10000,\ldots\}$
(0^*10^*)	$L=\{1,01,10,010,0010,\ldots\}$
$(0+\varepsilon)(1+\varepsilon)$	$L=\{\varepsilon,0,1,01\}$
$(a+b)^*$	Set of strings of a's and b's of any length including the null string. So $L=\{\varepsilon,a,b,aa,ab,bb,ba,aaa,\ldots\}$
$(a+b)^*abb$	Set of strings of a's and b's

HK-34

ending with the string abb. So,
L={abb, aabb, babb, aaabb, ababb, ...}

$(11)^*$ Set of consisting of even number of 1's including empty string, so
L={ε, 11, 1111, 111111, ...}

$(a|a)^*(bb)^*b$ Set of strings consisting of even no. of a's followed by odd number of b's, so, L={b, aab, aabbb, aabbbbb, aaaab, aaaabbb, ...}

$(aa+ab+ba+bb)^*$ String of a's and b's of even length can be obtained by concatenating any combination of the strings aa, ab, ba and bb including null, So
L={aa, ab, ba, bb, aaab, aaba, ...}