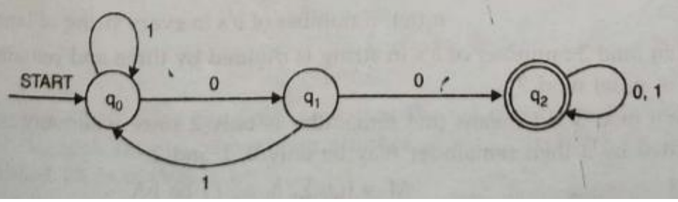
Instructions:

- **Students are required to write the complete answer for the first question of below question category. For other remaining, just draw the state diagram (i.e necessary figures only). Write necessary steps only so as to save time for this assignment completion.**
- **Scan it and make a single PDF and send it in the google classroom.**

DFA:

1. Design a FA that accepts set of strings such that every string ends in 00, over alphabet
2. Example 2.2. Construct a FA that accepts set of strings where the number of 0's in every string is multiple of three over alphabet $\Sigma = \{0, 1\}$.
3. Example 2.3. Design a FA which accepts set of strings containing exactly four 1's in every string over alphabet $\Sigma = \{0, 1\}$.
4. Example 2.4. Design a FA that accepts strings containing exactly 1 over alphabet $\{0, 1\}$
5. Example 2.5. Design a FA which accepts the language $L = \{w \in (0, 1)^* / \text{second symbol of } w \text{ is '0' and fourth input is } 1\}$.
6. Example 2.6. For the given FA write the language and also give the transition table
7. Example 2.6. For the given FA write the language and also give the transition table



8. Example 2.7. Design FA for the language $L = \{(01)^i 1^{2j} \mid i \geq 1, j \geq 1\}$.
9. Example 2.8. Design DFA for the language $L = \{w \in (a, b)^* / n_b(w) \bmod 3 > 1\}$.
10. Example 2.9. Design a FA over alphabet $\Sigma = \{0, 1\}$, which accepts the set of strings either start with 01 or end with 01.
11. Example 2.9. Design a FA over alphabet $\Sigma = \{0, 1\}$, which accepts the set of strings either start with 01 or end with 01.
12. Example 2.10. Give the DFA accepting the set of strings over alphabet $\Sigma = \{0, 1\}$, such that in each string number of 0's is divisible by five and number of 1's is divisible by 3.
13. Example 2.11. Design a FA which accepts the language $L = \{w / w$ has both an even number of 0's and an even number of 1's over alphabet $\Sigma = \{0, 1\}\}$.
14. Example 2.23. Design a DFA for the language $L = \{w : n_a = 1, w \in (a, b)^*\}$
15. Example 2.24. Design a DFA for the language $L = \{w : n_a(w) \geq 1, w \in (a, b)^*\}$.
16. Example 2.25. Design the determinstic finite automata for the given language $L = \{w : n_a(w) \leq 3, w \in \{a, b\}^*\}$.

17. Example 2.26. Design a DFA for the given language
$L = \{w : n_a(w) \geq 1, n_b(w) = 2, w \in \{a, b\}^*\}$.

18. Example 2.27. Design a DFA for the language
$L = \{w : n_a(w) = 2, n_b(w) > 2, w \in \{a, b\}^*\}$.

19. Example 2.28. Given DFA is

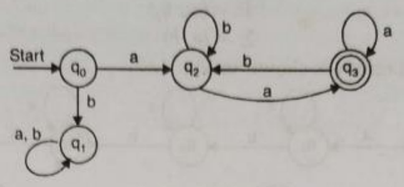


Fig. 2.64.

(i) Give the language of DFA shown in Fig. 2.64.

(ii) Prove that there exist a DFA for $\bar{L}$ if L is the language of given DFA.

20. Example 2.30. Design a DFA for the language
$L = \{ab^5 w b^4 : w \in \{a, b\}^*\}$.

21. Example 2.31. Design a DFA for the language
$L = \{w_1 ab w_2 : w_1, w_2 \in (a, b)^*\}$.

22. Example 2.33. Design a determinstic finite automation over alphabet $\Sigma = \{a, b\}$, such that every string, accepted by automation contains no runs of length less than four.

23. Example 2.34. Design a DFA for the language
$L = \{w$: every run of a's has length either two or three$\}$.

24. Example 2.35. Design a DFA, which accepts strings, in which every 00 is followed immediately by a 1. For example, the strings 001, 0010, 00100111001 are in the language, but 0001 and 00100 are not.

25. Example 2.36. Design a DFA for the language, contains strings in which left most symbol differ from right most symbol. Σ is given $\{0, 1\}$.

26. Example 2.37. Design a determinstic finite automation which accepts set of strings such that every string containing 00 as a substring but not 000 as sub-string.

27. Example 2.38. Construct a DFA that accepts strings on $\{0, 1\}$, if and only if the value of the string, interpreted as a binary representation of an integer, is zero module five. For example, 0101 and 1111, representing the integers 5 and 15, respectively, are to be accepted.

28. Example 2.40. Here DFA given in Fig. 2.81 for the language L find the DFA for L^2.

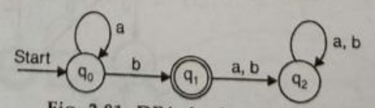


Fig. 2.81. DFA for language L.

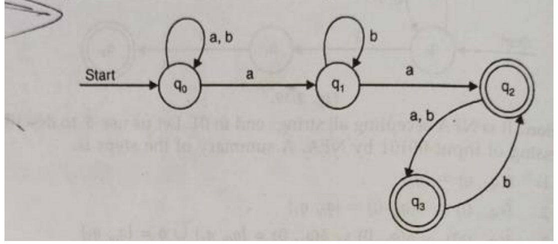
NFA:

29. Find NFA with four state for the language L= { (a^n : n ≥ 0) U (b^na : n ≥1)}

30. Construct an NFA for (ab/ba)*ab

NFA to DFA Conversion:

31. Example 2.17. Convert the following NFA in to DFA.

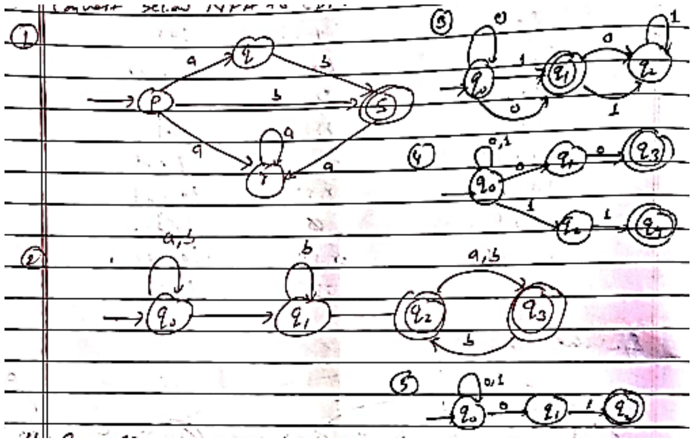


32. Example 2.17. Given NFA is

δ/Σ	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_2\}$
q_1	$\{q_0\}$	$\{q_1\}$
$^*q_2$	ϕ	$\{q_0, q_1\}$

Convert it in to DFA.

33. Convert below NFA to DFA:



ε -NFA

34. **Example 2.18.** Consider the NFA with ϵ-transition $M = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b, c\}$ and ϵ moves.

Initial state $= \{q_0\}$

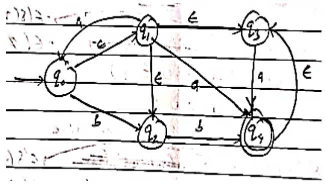
$F = \{q_2\}$

Transition Table

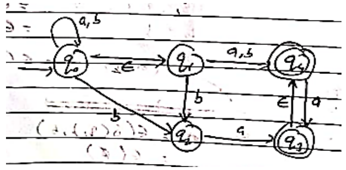
δ/Σ	a	b	c	ϵ
$\rightarrow q_0$	$\{q_0\}$	$\{\phi\}$	$\{\phi\}$	$\{q_1\}$
q_1	$\{\phi\}$	$\{q_2\}$	$\{\phi\}$	$\{q_2\}$
$^*\ q_2$	$\{\phi\}$	$\{\phi\}$	$\{q_2\}$	$\{\phi\}$

35. Convert below ε-NFA to its equivalent DFA.

a)



b)

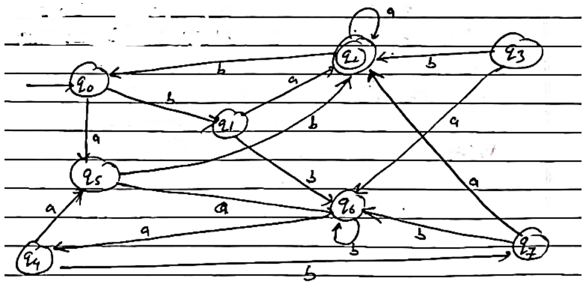


State Minimization (DFA Minimization)

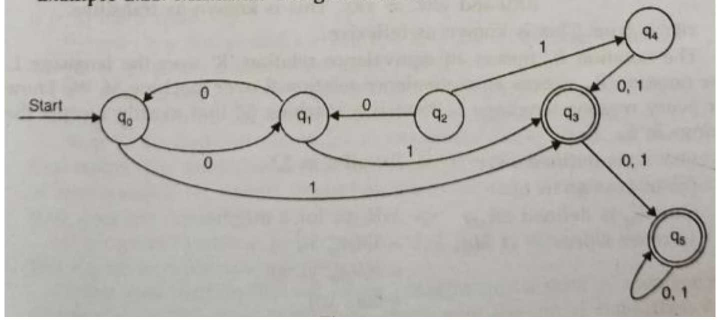
36. Minimize following DFA by using State Minimization method, where: → represents initial state and * represents final state.

δ/Σ	0	1
$\rightarrow q_0$	q_1	q_2
$*q_1$	q_1	q_3
q_2	q_2	q_2
$*q_3$	q_5	q_2
$*q_4$	q_4	q_2
$*q_5$	q_4	q_2
q_6	q_5	q_6
q_7	q_6	q_6

37. Minimize below DFA:



38. Example 2.21. Minimise the given DFA.



Regular Expressions and Regular Language

39. Design a Finite Automata from the given RE [ab + (b + aa)b* a].

40. Design an NFA from the given RE [a (a* ba* ba*)*].

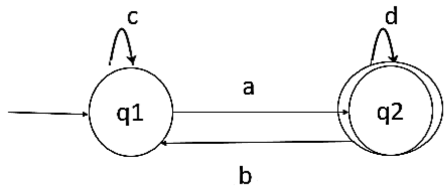
41. Construct the FA for regular expression 0*1 + 10.

42. Construct finite automata for this regular expression RE 10+(0+11)0*

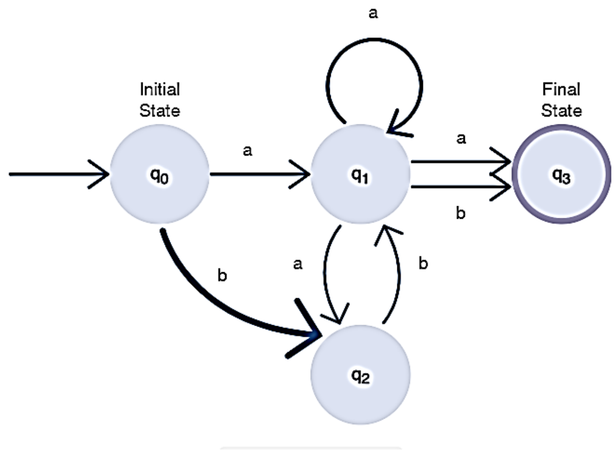
43. Create a ∈-NFA for regular expression: (a/b)*a

44. Example 4.12. One the basis of above discussion find the automation for regular expression $a \cdot (a + b)^* \cdot b \cdot b$.

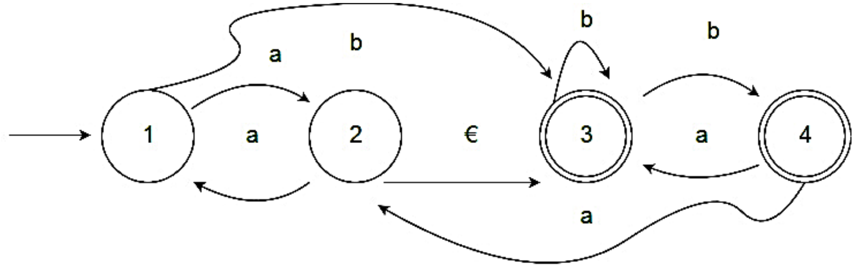
45. Convert the below FA to RE:



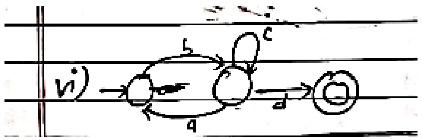
46. Convert below finite automaton to RE.



47. Deduce the RE from below FA.

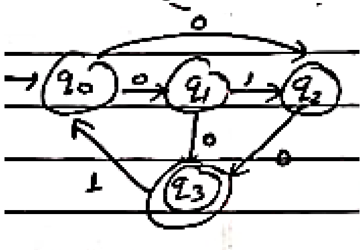


48. Convert below From FA to Regex

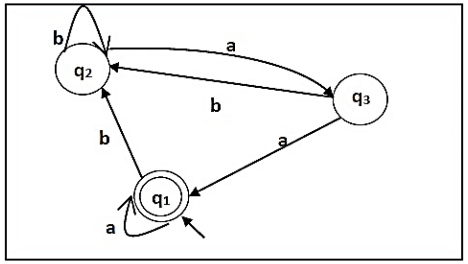


Arden's Theorem

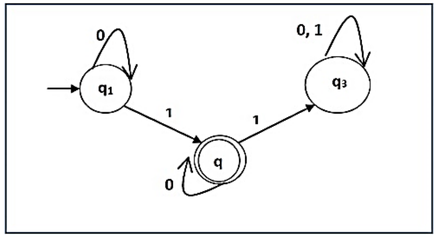
49. Construct Regex for below FA.



50. Construct a regular expression to the automata given below:



51. Construct a regular expression corresponding to the automata given below:



Pumping Lemma for Regular Language

52. State and prove Pumping Lemma for Regular Language.

53. Show that the language L={$a^n b^n$, n>=1} is not regular.

54. Show that the language A={yy | y belongs to $\{0,1\}^*$} is not regular.

55. Show that the language L={$a^{2n} b^n$, n>=1} is not regular.

Decision Properties and Closure properties of Regular Language

56. State the decision properties of Regular language

57. State the closure properties of Regular language.

58. Show that if L is regular, then complement of L(i.e. L') is also regular.

***