Unit – I

short Answers

1. Define Quintuple Specification of Automata
2. Extended transition function of FA
3. What is state and write its types
4. What is transition table and state diagram
5. Process Acceptance of String by Finite Automata
6. Write transition function of DFA, NFA, & NFA- ε
7. Write Extended transition function of NFA
8. Write Extended transition function of NFA- ε
9. Define ε-closure
10. Define Language Acceptance of DFA
11. Define Language Acceptance of NFA- ε
12. Write Applications of Finite Automata
13. Define String
14. Explain Different operations on Strings
15. Explain Components of Finite Automata
16. Elements or Characteristics of DFA
17. Draw a DFA for the language accepting strings ending with '00' over input alphabets ∑= {0, 1}?
18. Draw a DFA for the language accepting odd binary numbers strings over input alphabets ∑ = {0, 1}?

Long Questions:

1. Draw a DFA for the language accepting strings containing neither '00', nor '11' as substring over input alphabets ∑ = {0, 1} ?
2. construct DFA for binary integer divisible by 3
3. Construct DFA accepting set of all strings containing even no. of a's and even no. of b's over input alphabet {a,b}.
4. Construct DFA a DFA for the language accepting strings with '0' and '1' only over input alphabets ∑={0, 1}
5. Compare **NFA and DFA** in terms of **structure, expressiveness, and efficiency**.
6. Construct a DFA, the language recognized by the Automaton being L={w/ w does not contain the substring ab}.Draw the transition table.
7. Discuss various Differences between DFA and NFA.
8. Construct a NFA over {a,b} which accepts string starts with 'a' and ends with 'b'.
9. Conversion of DFA from given NFA(prepare examples)
10. Conversion of DFA from given NFA- ε (prepare examples)

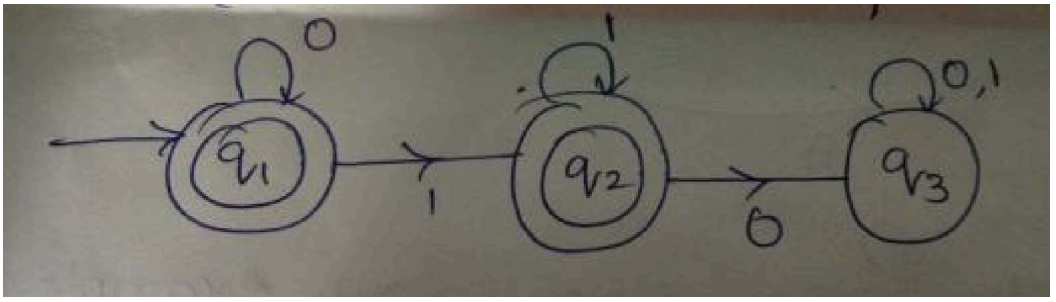
Unit – II

short Answers

1. Define CFG and its Quad Tuple
2. Define forms of Regular expression
3. What are Components of RE
4. Write Language associate with RE
5. What are basic operations and properties of RE
6. Write Applications of RE
7. What is Derivation
8. What is Yield of a Tree
9. Define Sub tree
10. Identify given grammar and derive the given string
11. Construction of RE (examples)
12. Define Ambiguous grammar
13. Mention any two applications of Context Free Grammar

Long Questions:

1. Construct a regular expression for the language consisting of all strings over {0,1} that end with "01"
2. Show how to convert the given **finite automaton into an equivalent regular expression(Take examples)**
3. Discuss **applications of regular expressions** in computer science
4. Prove using the **pumping lemma** that L = $\{a^n b^n \mid n \geq 0\}$ is **not regular**.
5. Prove that L = $\{ a^p \mid p \text{ is prime } \}$ is not regular.
6. Apply the **pumping lemma** to show that the language L = $\{ w \in \{0,1\}^* \mid$ w has equal number of 0s and 1s $\}$ is not regular
7. Define CFG with an example and explain how **derivations** are performed using it.
8. Explain with an example what is meant by **ambiguity in a grammar**. Suggest how ambiguity can be removed.
9. Discuss Identity rules. Simplify the Regular Expression or Algebraic expressions
10. Find the regular expression for the following DFA



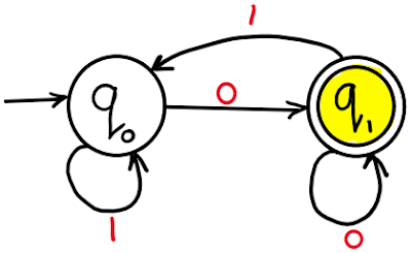
11. Construct Leftmost Derivation. , Rightmost Derivation, Derivation Tree for the following grammar with respect to the string aaabbabbba.

    S□ aB | bA
    A□aS| bAA|a
    B□bS | aBB | b
12. Construct leftmost and rightmost derivations for the strings, if the language is given as S□AS| ε
    A□aa|ab|ba|bb
    Strings: a) aabbba b) baabab c) aaabbb
13. Show that the following CFG ambiguous. S□iCtS | iCtSeS | a,
    C□b.
14. Derive the parse tree E□ E+E | E*E |id for the stringw=id+id*id

Sample DFA Examples:

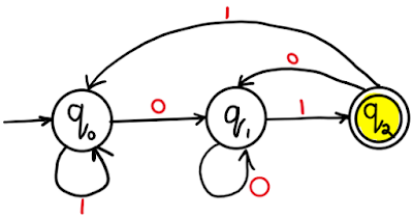
Example 1: Draw a DFA for the language accepting strings ending with '0' over input alphabets ∑={0, 1} ?

Solution:



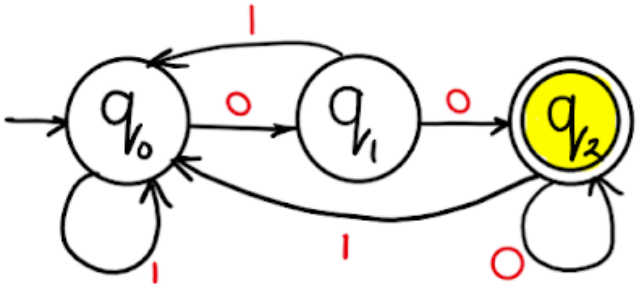
Example 2: Draw a DFA for the language accepting strings ending with '01' over input alphabets ∑={0, 1} ?

Solution:



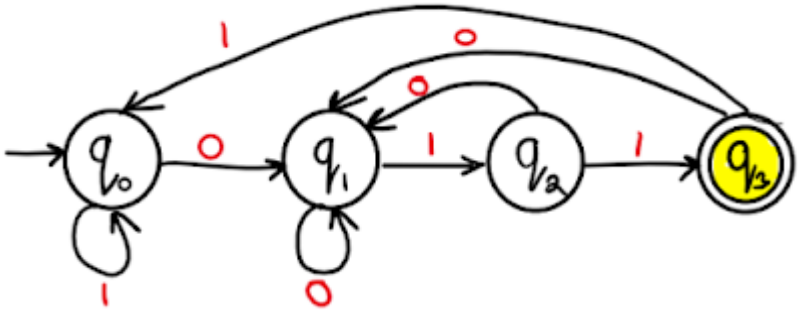
Example 3: Draw a DFA for the language accepting strings ending with '00' over input alphabets ∑={0, 1} ?

Solution:



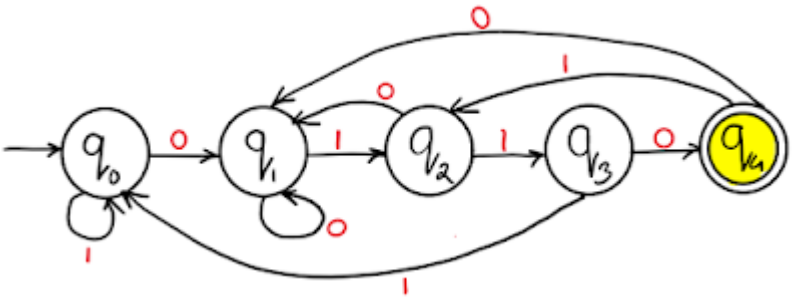
Example 4: Draw a DFA for the language accepting strings ending with '011' over input alphabets ∑ = {0, 1} ?

Solution:



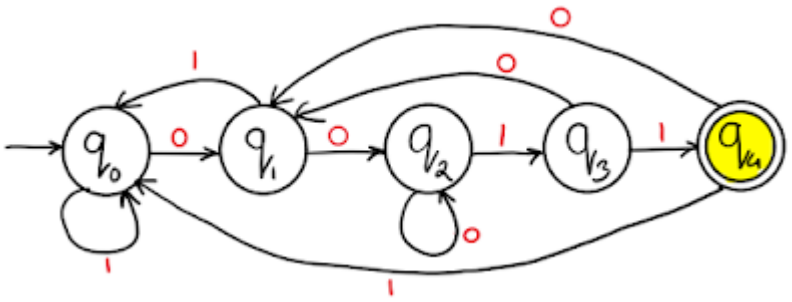
Example 5: Draw a DFA for the language accepting strings ending with '0110' over input alphabets ∑ = {0, 1} ?

Solution:



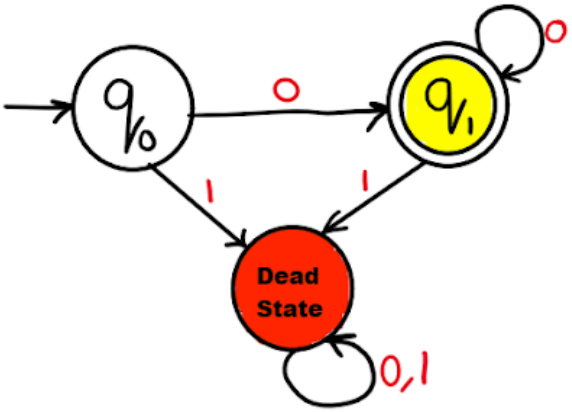
Example 6: Draw a DFA for the language accepting strings ending with '0011' over input alphabets ∑ = {0, 1} ?

Solution:



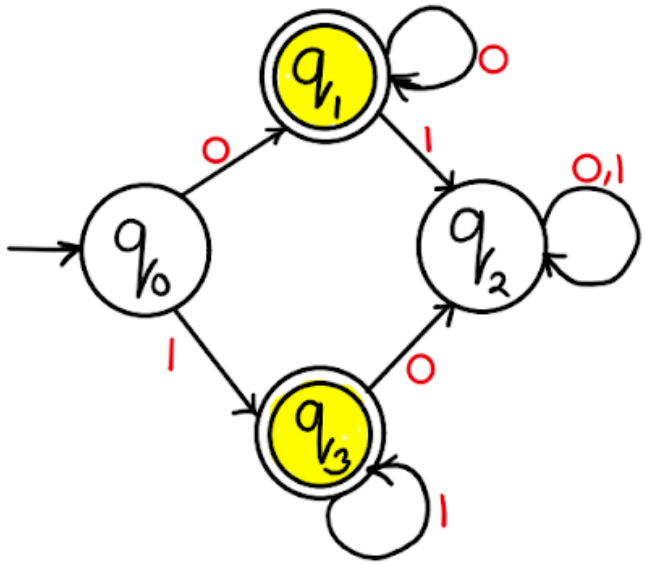
Example 7: Draw a DFA for the language accepting strings with ‘0’ only over input alphabets ∑={0, 1} ?

Solution:



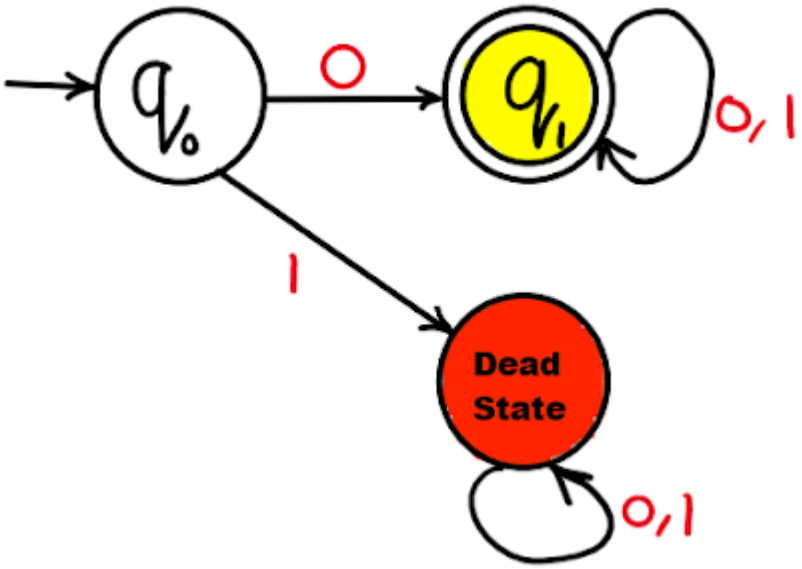
Example 8: Draw a DFA for the language accepting strings with ‘0’ and ‘1’ only over input alphabets ∑={0, 1} ?

:



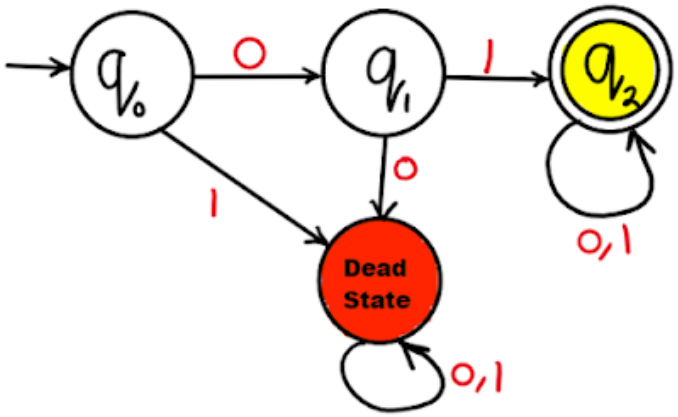
Example 9: Draw a DFA for the language accepting strings starting with '0' over input alphabets ∑={0, 1} ?

Solution:



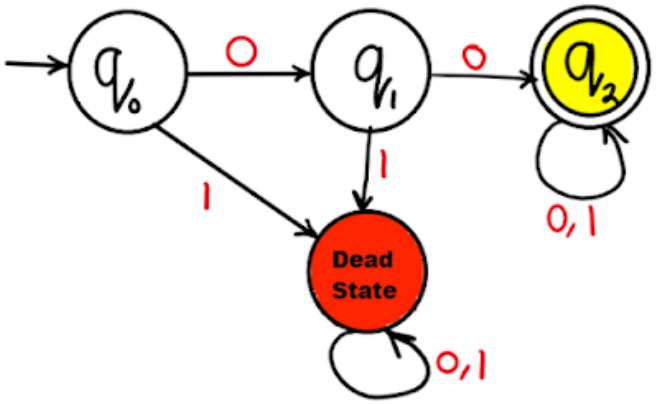
Example 10: Draw a DFA for the language accepting strings starting with '01' over input alphabets ∑={0, 1} ?

Solution:



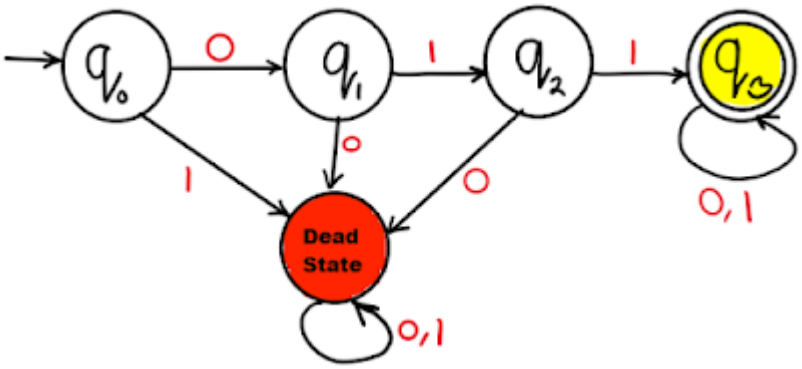
Example 11: Draw a DFA for the language accepting strings starting with '00' over input alphabets ∑={0, 1} ?

Solution:



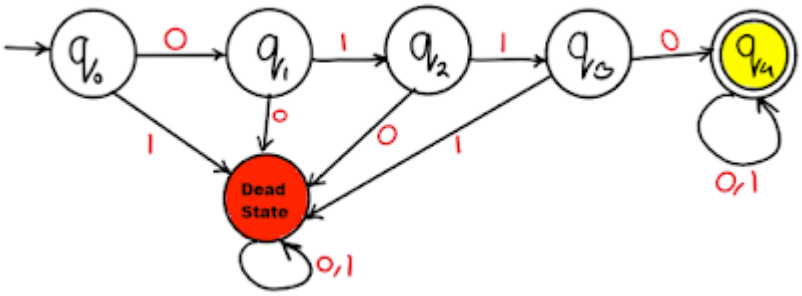
Example 12: Draw a DFA for the language accepting strings starting with '011' over input alphabets ∑ = {0, 1} ?

Solution:



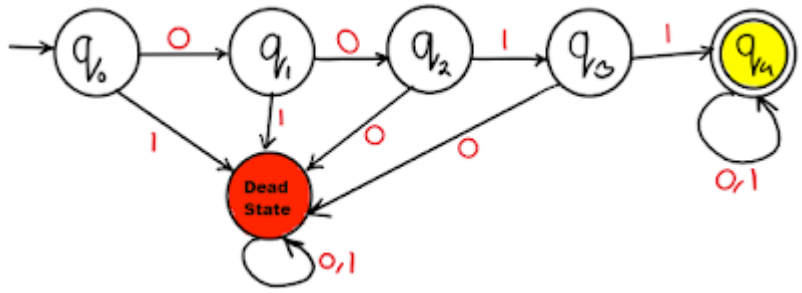
Example 13: Draw a DFA for the language accepting strings starting with '0110' over input alphabets ∑ = {0, 1} ?

Solution:



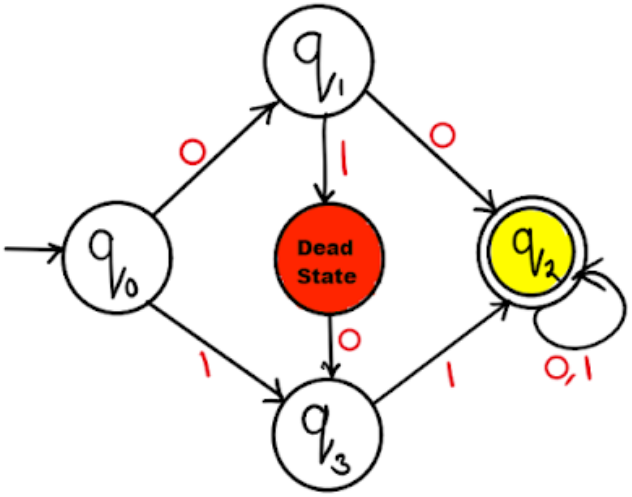
Example 14: Draw a DFA for the language accepting strings starting with '0011' over input alphabets ∑ = {0, 1} ?

Solution:



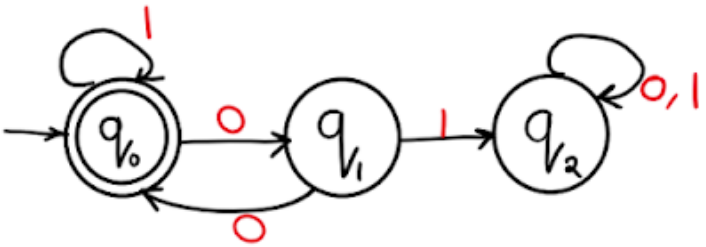
Example 15: Draw a DFA for the language accepting strings starting with '00' or '11' over input alphabets ∑ = {0, 1} ?

Solution:



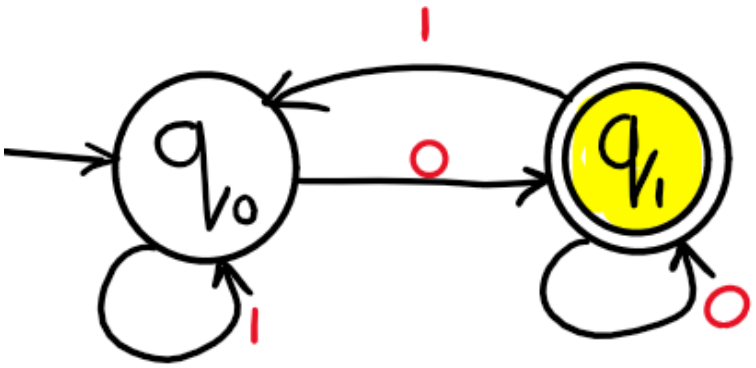
Example 16: Draw a DFA for the language accepting strings without substring '00' over input alphabets ∑ = {0, 1} ?

Solution:



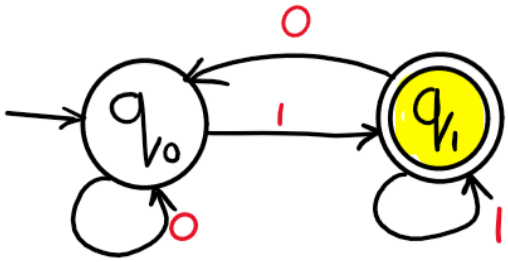
Example 17: Draw a DFA for the language accepting even binary numbers strings over input alphabets ∑ = {0, 1} ?

Soluntion:



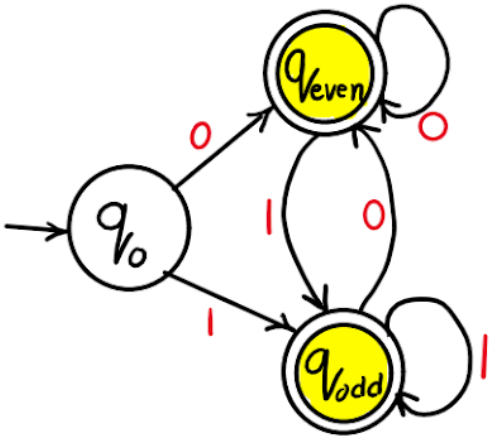
Example 18: Draw a DFA for the language accepting odd binary numbers strings over input alphabets ∑ = {0, 1} ?

Solution:



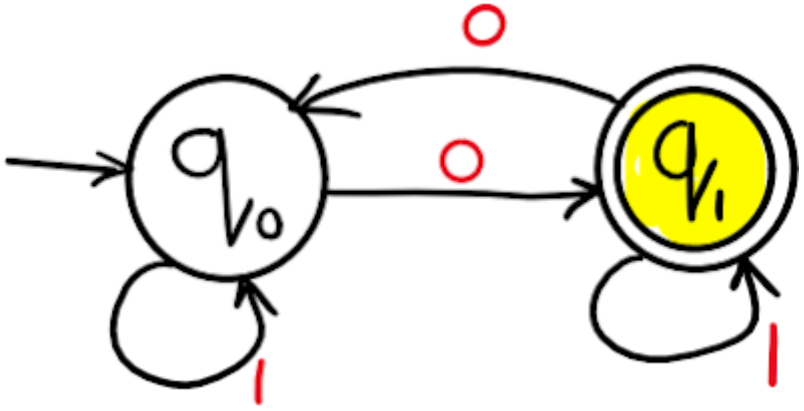
Example 19: Draw a DFA for the language accepting odd or even binary numbers strings over input alphabets ∑ = {0, 1} ?

Solution:



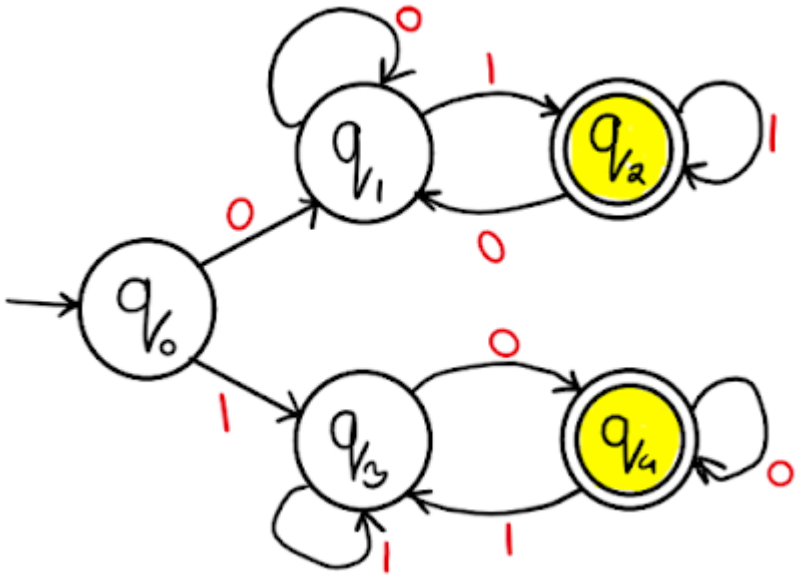
Example 20: Draw a DFA for the language accepting strings containg even number of total zeros over input alphabets ∑ = {0, 1} ?

Solution:



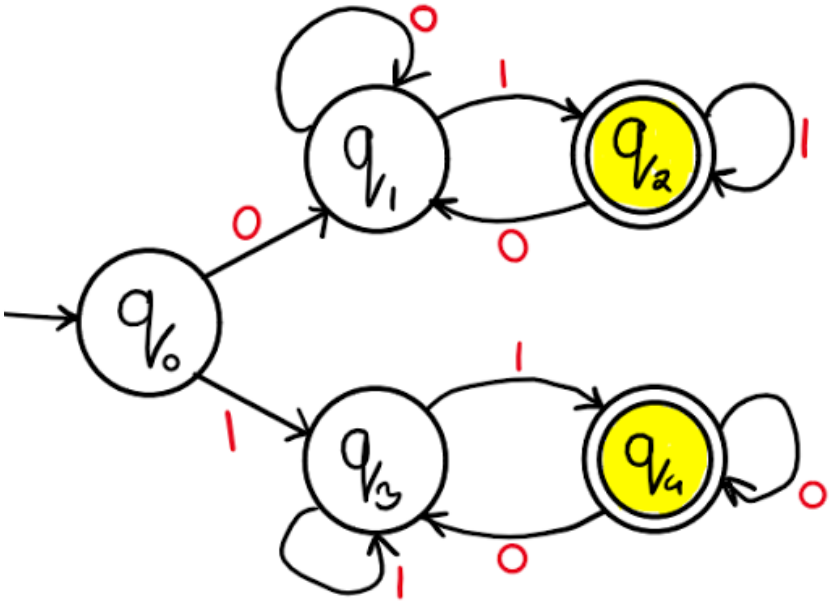
Example 21: Draw a DFA for the language accepting strings starting and ending with different characters over input alphabets ∑ = {0, 1} ?

Soluiton:



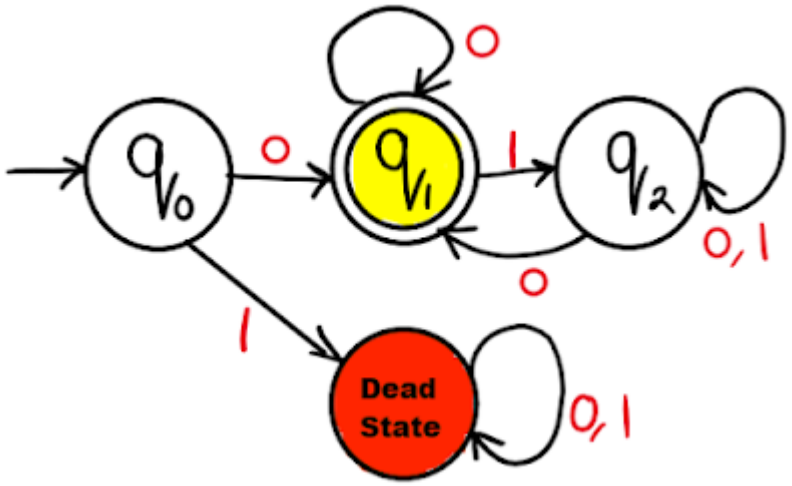
Example 22: Draw a DFA for the language accepting strings starting and ending with same character over input alphabets ∑ = {0, 1} ?

Solution:



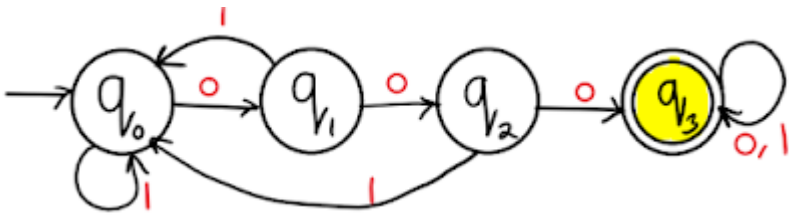
Example 23: Draw a DFA for the language accepting strings starting and ending with ‘0’ always over input alphabets ∑ = {0, 1} ?

Solution:



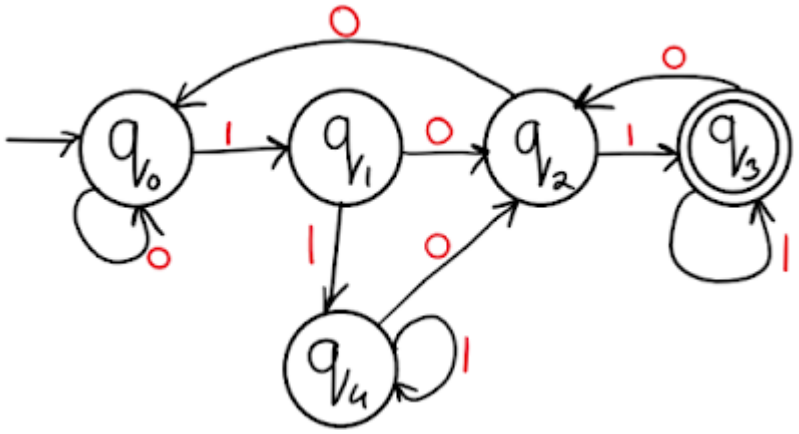
Example 24: Draw a DFA for the language accepting strings containing three consecutives ‘0’ always over input alphabets ∑ = {0, 1} ?

Solution:



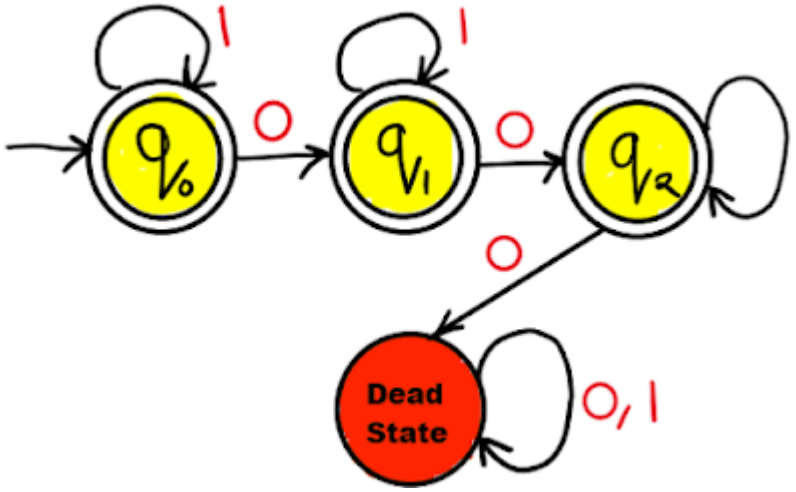
Example 25: Draw a DFA for the language accepting strings such that each ‘0’ is immediately preceded and followed by ‘1’ over input alphabets ∑ = {0, 1} ?

Solution:



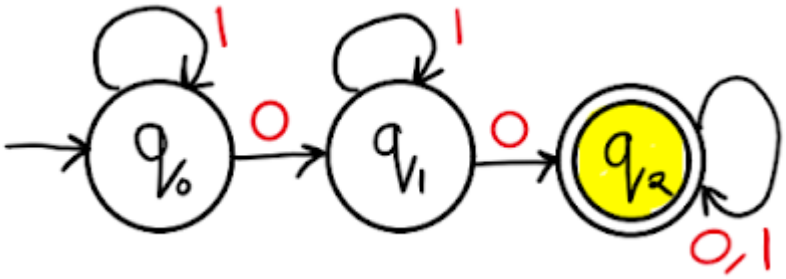
Example 26: Draw a DFA for the language accepting strings containing at most two '0' over input alphabets ∑ = {0, 1} ?

Solution:



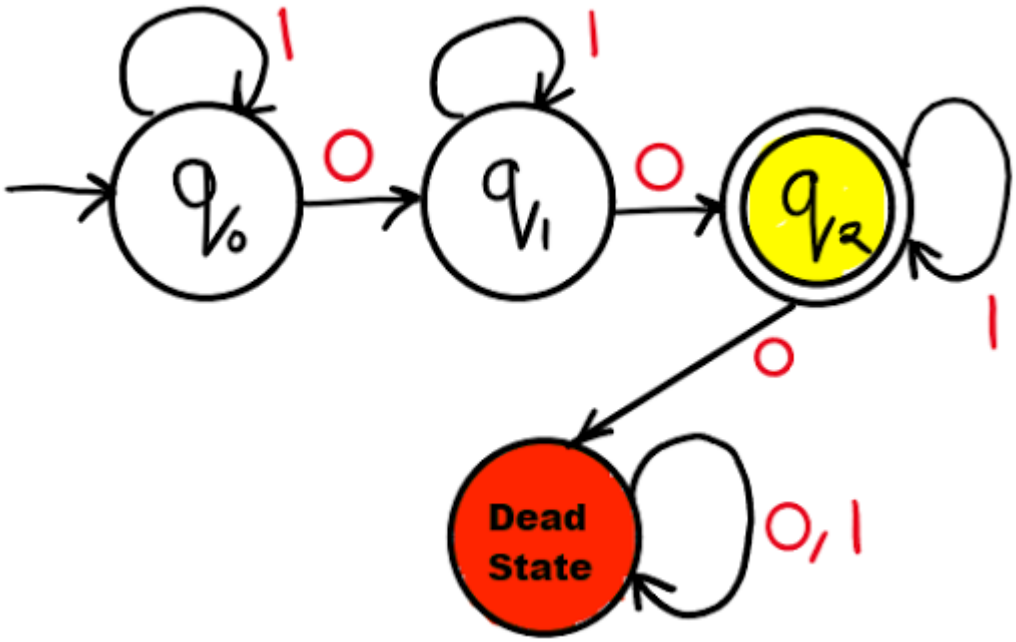
Example 27: Draw a DFA for the language accepting strings containing at least two '0' over input alphabets ∑ = {0, 1} ?

Solution:



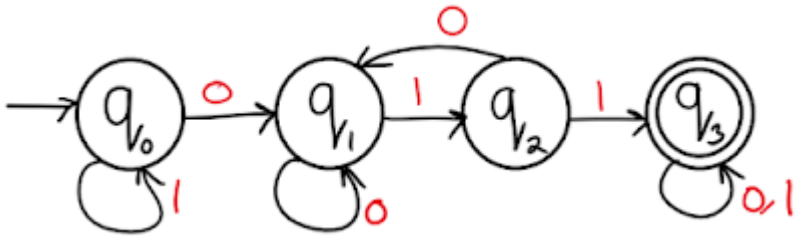
Example 28: Draw a DFA for the language accepting strings containing exactly two ‘0’ over input alphabets ∑ = {0, 1} ?

Solution:



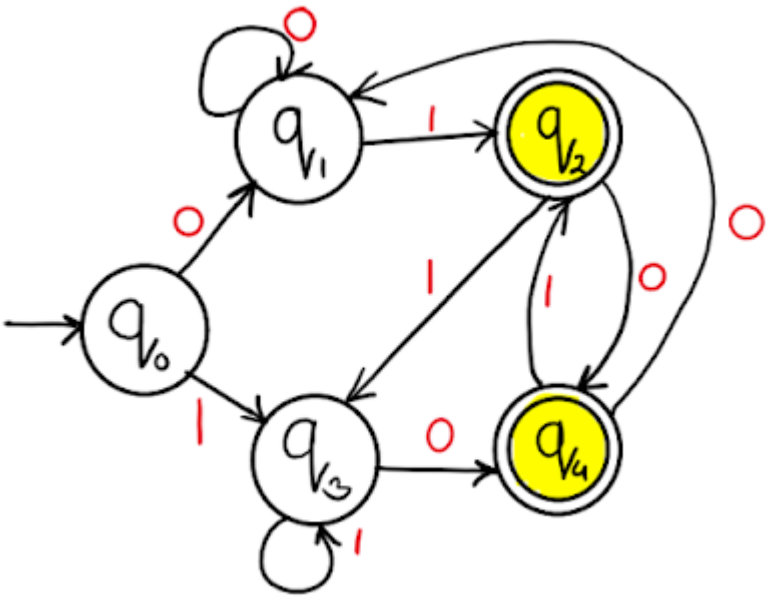
Example 29: Draw a DFA for the language accepting strings with ‘011’ as substring over input alphabets ∑ = {0, 1} ?

Solution:



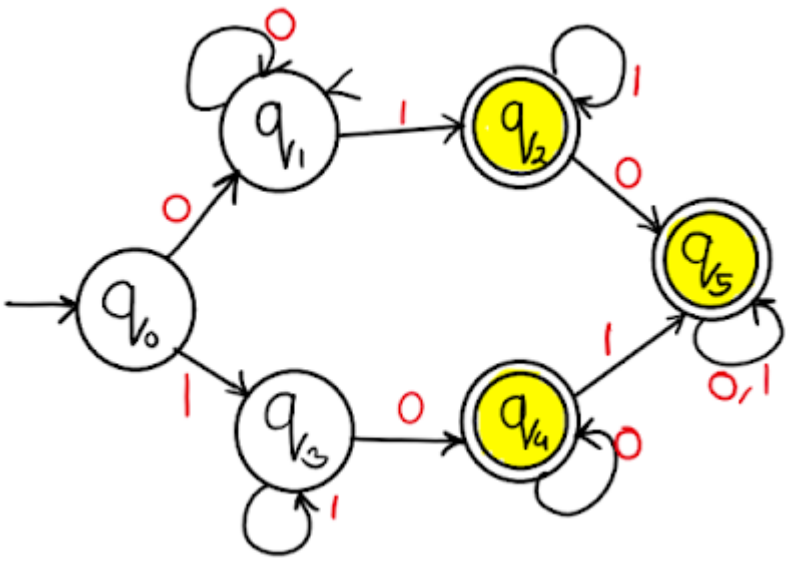
Example 30: Draw a DFA for the language accepting strings ending in either ’01’, or ’10’ over input alphabets ∑ = {0, 1} ?

Solution:

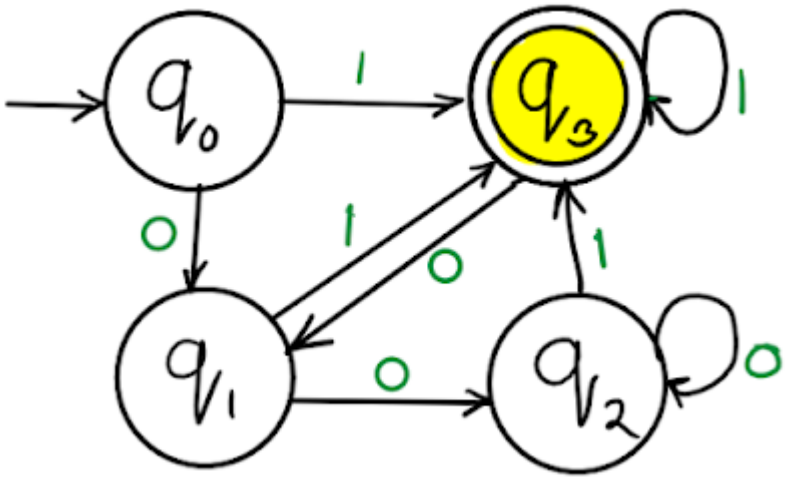


Example 31: Draw a DFA for the language accepting strings containing ’01’, or ’10’ as substring over input alphabets ∑ = {0, 1} ?

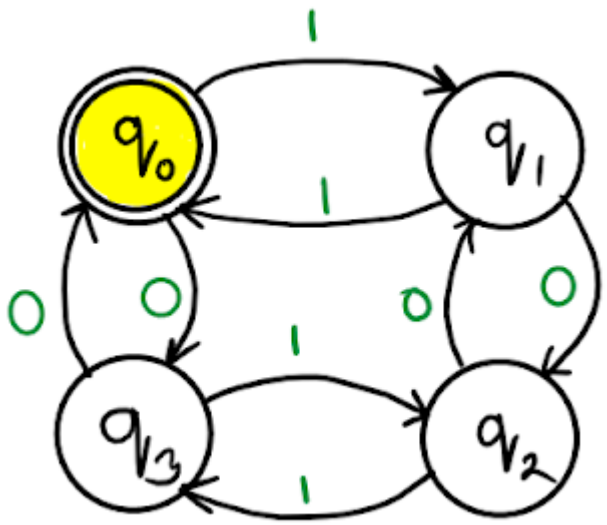
Solution:



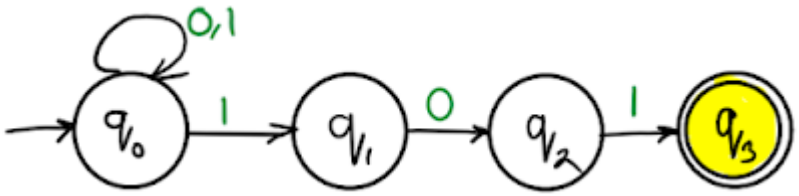
Example 32: Draw DFA that accepts any string which ends with 1 or it ends with an even number of 0's following the last 1. Alphabets are {0,1}.
Solution:



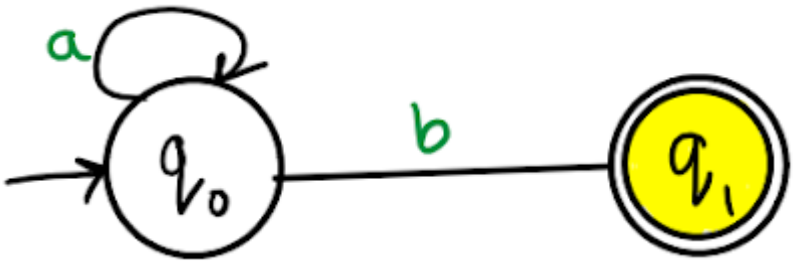
Example 33: Construct DFA accepting set of all strings containing even no. of a's and even no. of b's over input alphabet {a,b}.
Solution:



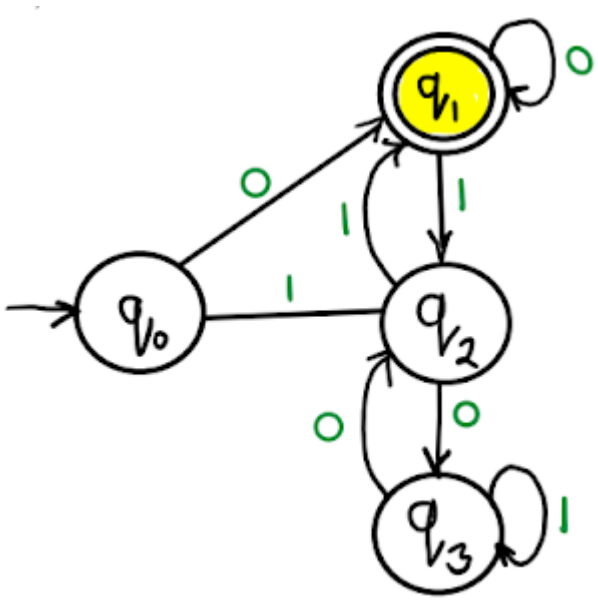
Example 34: Give DFA accepting the language over alphabet {0,1} such that all strings of 0 and 1 ending in 101.
Solution:



Example 35: Construct DFA for anb | n>=0.
Solution:

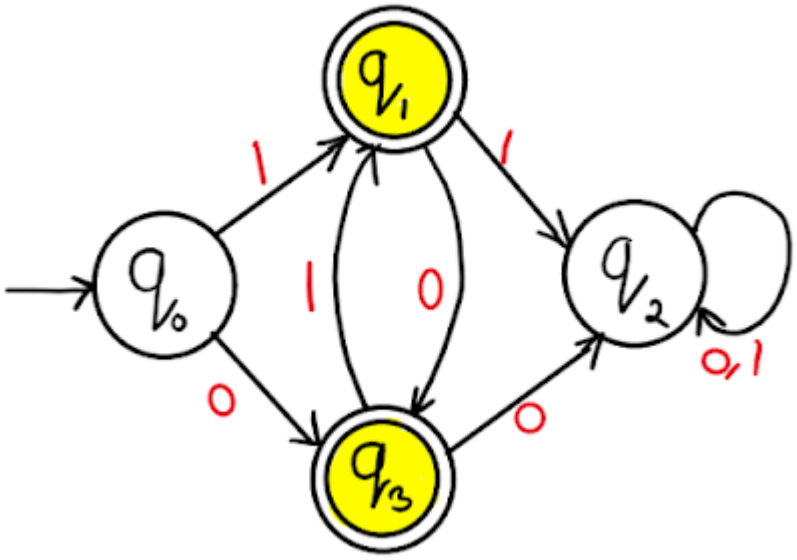


Example 36: construct DFA for binary integer divisible by 3 ?
Solution:



Example 37: Draw a DFA for the language accepting strings containing neither ’00’, nor ’11’ as substring over input alphabets ∑ = {0, 1} ?

Solution:



Some Examples On Regular Expressions

Example 1: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contains only a's or only b's of length zero or more.

Solution: r = a* + b*

Example 2: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w is of length one or more and contains only a's or only b's. r = a+ + b+

Solution: $r = a^+ + b^+$

Example 3: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contains zero or more a's followed by zero or more b's

Solution: r = a*b*

Example 4: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length even

Solution: r = [(a + b) (a + b)]*

Example 5: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length odd

Solution: r = (a + b) [(a + b) (a + b)]*

Example 6: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length three

Solution: r = (a + b) (a + b) (a + b)

Example 7: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length atmost three

Solution: r = (a + b + ∈) (a + b + ∈) (a + b + ∈)

Example 8: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length odd containing only b's

Solution: r = (bb)* b

Example 9: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w starting with a always

Solution: r = a(a + b)*

Example 10: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w starting and ending with b and having only a's in between.

Solution: r = b a* b

Example 11: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w starting and ending with same double letter

Solution: r = {(aa (a + b)* aa) | (bb (a + b)* bb)

Example 12: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with starting and ending with different letters

Solution: r = (a(a+b)* b) | (b (a + b)* a)

Example 13: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with at least two occurrence of a

Solution: r = (a + b)* a (a + b)* a (a + b)*

Example 14: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with exactly two occurrence of a

Solution: r = b* a b* a b*

Example 15: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with at most two occurrence of a

Solution: r = b* (a + ∈) b* (a + ∈) b*

Example 16: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with begin or end with aa or bb

Solution: r = ((aa + bb) (a + b)*) + ((a + b) * (aa + bb))

Example 17: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with begin and end with aa or bb

Solution: r = ((aa + bb) (a + b)* (aa + bb)) + aa + bb

Example 18: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with total length multiple of 3 always

Solution: r = [(a + b) (a + b) (a + b)]*

Example 19: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w containing total a's as multiple of 3 always

Solution: r = [b* a b* a b* a b*]*

Example 20: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with exactly two or three b's

Solution: r = a* b a* b a* (b + ∈) a*

Example 21: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with number of a's even

Solution: r = b* + (b* a b* a b*)*

Example 22: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w in which b is always tripled

Solution: r = (a + bbb)*

Example 23: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with at least one occurrence of substring aa or bb

Solution: r = (a + b)* (aa + bb) (a + b)*

Example 24: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with at the most one occurrence of sub-string bb

Solution: r = (a + ba)* (bb + ∈) (a + ab)*

Example 25: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w without sub-string ab

Solution: r = b* a*

Example 26: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w without sub-string aba

Solution: r = (a + ∈) (b + aa+)* (a + ∈)

Example 27: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w in which 3rd character from right end is always a

Solution: r = (a + b)* a (a + b) (a + b)

Example 28: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w always start with ‘a’ and the strings in which each ‘b’ is preceded by ‘a’.

Solution: (a + ab)*

Example 29: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contains atleast one ‘a’.

Solution: (a + b)* a (a + b)*

Example 30: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contain atleast two ‘a’s or any number of ‘b’s.

Solution: (a* a b* a b*) + b*

Example 31: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contain atleast one ‘a’ followed by any number of ‘b’s followed by atleast one ‘c’.

Solution: a^+ b* c^+