Chapter 1

1. Define the following terms & explain with examples.
   i) Grammar    ii) Language

2. Mention the difference between DFA, NFA and εNFA.

3. What is the need for an NFA?

4. Give DFA’s accepting the following languages over the alphabet \{0,1\}.
   i) The set of all strings ending in 00.
   ii) The set of all strings with three consecutive 0’s (not necessarily at the end).

5. Define distinguishable and non distinguishable states.

6. Give a general procedure to convert an NFA into DFA.

7. Construct DFA for the language \( L = \{ w / w \text{ has odd number of 1’s and is followed by even number of 0’s} \} \). Completely define DFA and transition functions.

8. Describe two applications of DFA with transition diagrams.

9. Design a NFA to recognize the following set of strings.
   i) abc, abd and aacd. Assume the alphabet is \{a,b,c,d\}
   ii) 0101, 101, 011. Assume the alphabet is \{0,1\}

10. Give a general procedure to minimize the states of DFA.

11. Consider the following εNFA.

<table>
<thead>
<tr>
<th></th>
<th>Ε</th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>{q,r}</td>
<td>-</td>
<td>q</td>
<td>r</td>
</tr>
<tr>
<td>q</td>
<td>-</td>
<td>p</td>
<td>r</td>
<td>{p,q}</td>
</tr>
<tr>
<td>r*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

12. Give a DFA that accepts the decimal strings divisible by 3.

13. Convert the below εNFA to DFA.
Chapter 2

1. Define a regular expression. And find the regular expression for the following languages on \{0,1\}.
   i) Strings with an odd number of 1’s.
   ii) Strings which is having exactly three number of 1’s.
   iii) Strings containing even number of 0’s.
   iv) Strings not ending with 001.
   v) Strings containing exactly one 0.
   vi) Strings containing not more than three 0’s.

2. Obtain the regular expressions for the following languages on \{a,b\}.
   i) \(L=\{a^{2n} b^{2m} ; n\geq 0 , m\geq 0\}\)
   ii) \(L=\{w : |w| \mod 3 =0\}\)
   iii) \(L=\{a^n b^n ; m\geq 0, n\geq 0 , (m+n) \text{ is even}\}\)
   iv) \(L= \{ a^n b^m ; m\geq 1, n\geq 1, mn\geq 3\}\)

3. Prove that a language is regular if and only if it is accepted by finite automata.

4. What does the following regular expressions imply.
   i) \(a (a+b)^* ab\)
   ii) \((0*1 + 1*0)^*0\)
   iii) \(0^* + (01+0)^*\)

5. How can NFA be obtained from regular expression? Give the procedure.

6. Obtain NFA and DFA for the regular expression \((a+b)^* aa (a+b)^*\).

7. What are the applications of regular expression.

8. Obtain regular expression to identify an identifier.

Chapter 3

1. Discuss the closure properties of regular languages with examples.
2. State and prove the pumping lemma for regular languages.
3. Prove that the following languages are not regular.
   i) \(\{0^n 1^n 2^n / n, m\geq 1\}\)
   ii) \(\{0^n 1^n / n\geq 1\}\)
4. Show that if \(L \& M\) are regular languages , then so is \(L\cap M\).
5. Discuss the decision properties of regular languages with examples.
6. Explain the table filling method used to minimize the states of DFA & Find the minimized DFA for the following.

<table>
<thead>
<tr>
<th>State</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>B</td>
</tr>
<tr>
<td>D*</td>
<td>D</td>
<td>A</td>
</tr>
</tbody>
</table>
7. Show that the regular languages are closed under
   i) Union
   ii) Concatenation
   iii) Star closure
8. Show that the regular languages are closed under homomorphism.
9. Show that the following languages are not regular.
   i) \( L = \{ww^R \mid w \text{ is in } (0+1)^*\} \)
   ii) \( L = \{a^n b^n \mid n \geq 0\} \)
   iii) \( L = \{a^n \mid n \text{ is prime}\} \)

**Chapter 4**

1. What are the limitations of regular languages.
2. What is a context free grammar? Explain with example.
3. Construct CFG for the following languages:
   i. \( L = \{a^n w w^R b^n \mid w \in \{a,b\}^*\} \)
   ii. \( L = \{xy \mid x \in (a+b)^* y \in (ab \text{ or } ba)\} \)
      Show the rightmost derivation for the string aabbba along with derivation tree.
4. Let \( G = (V, T, P, S) \) be a CFG where
   \[ V = \{S\} \]
   \[ T = \{a, b\} \]
   \[ P = \{ S \rightarrow aSa / bSb / \varepsilon \} \]
   S is the start symbol.
   What is the language generated by this grammar?
5. Given the following GFG
   \[ E \rightarrow E + T / T \]
   \[ T \rightarrow T \ast F / F \]
   \[ F \rightarrow (E) / a / b / c \]
   Draw parse tree for the following sentences and also derive rightmost derivations
   for i) \((a+b)^* c\)       ii) \((a) + b^* c\)
6. What is meant by ambiguity? How to test the ambiguity of a grammar? Show that the
   following grammar is ambiguous.
   \[ S \rightarrow AB / aaB \]
   \[ A \rightarrow a / Aa \]
   \[ B \rightarrow b \]
   Construct an unambiguous grammar equivalent to above grammar.
7. Show the following grammar is ambiguous & give an equivalent unambiguous grammar.
   \[ S \rightarrow S + S / S \ast S / a \]
8. What is dangling else grammar? Show that the dangling else grammar is ambiguous. And
   construct an equivalent unambiguous grammar.
9. Describe leftmost and rightmost derivations with examples.

**Chapter 5**

1. What are the demerits of DFA when compared with PDA.
2. What is a PDA? Explain with an example.
3. Design PDA to accept the language \( L = \{ w / w \text{ is in } \{a,b\}^* \text{ and } n_a(w)=n_b(w) \} \)
4. Convert the following grammar to a PDA.
   
   \[
   \begin{align*}
   &I \rightarrow a / b \\
   &S \rightarrow aA \\
   &A \rightarrow aABC / bB / a \\
   &B \rightarrow b \\
   &C \rightarrow c
   \end{align*}
   \]
5. What does each of the following transitions represent?
   
   i) \( \delta(p, a, Z) = (q, aZ) \)
   
   ii) \( \delta(p, a, Z) = (q, \varepsilon) \)
   
   iii) \( \delta(p, a, Z) = (q, r) \)
   
   iv) \( \delta(p, \varepsilon, Z) = (q, r) \)
   
   v) \( \delta(p, \varepsilon, Z) = (q, Z) \)
   
   vi) \( \delta(p, \varepsilon, Z) = (q, \varepsilon) \)
6. Check whether the PDA from the following language is deterministic or non-deterministic.
   
   \( L = \{ a^n b^n ; n \geq 1 \} \)

**Chapter 6**

1. Prove that the family of context free languages are closed under union, concatenation and reversal operations.
2. State and prove pumping lemma for context free languages.
3. What are the applications of pumping lemma?
4. Show that \( L = \{ a^n b^n c^n / n \geq 0 \} \) is not context free.
5. What is CNF & GNF?
6. Eliminate unit productions from the following:
   
   i) \( S \rightarrow A0 / B \\
   \quad B \rightarrow A / 11 \\
   \quad A \rightarrow 0 / 12 / B \)
   
   ii) \( S \rightarrow Aa / B / Ca \\
   \quad B \rightarrow aB / b \\
   \quad C \rightarrow Db / D \\
   \quad D \rightarrow E / d \\
   \quad E \rightarrow ab \)
7. What is the general procedure to convert a grammar into its equivalent GNF notation? Explain.
8. Discuss the applications of CNF and GNF.
9. Obtain the following grammar in CNF.
   
   \( S \rightarrow aBa / abba \)
A → ab / AA  
B → aB / a  

10. Simplify the following CFG into CNF.  
S → ASB / ε  
A → aAS / a  
B → SbS / A / bb  

Chapter 7  

1. Define Turing machine and Instantaneous description(ID) of TM.  
2. Explain various forms of Turing machine.  
3. Define move of a TM.  
4. Design a Turing machine to accept L = \{ww^R / w is in (a+b)^*\}. And show the sequence of moves made by the turing machine for the string abba.  
5. Design a Turing machine to accept the language L= \{0^n 1^n / n>=1\}. Give the graphical representation. Also give the sequence of ID’s for the input 000111.  
6. Discuss how to use computer to simulate a turing machine and compare the running times of computer and turing machine.  
7. Prove that every language accepted by a multi tape Turing machine is recursively enumerable.  
8. Explain the general structure of multi tape and non deterministic Turing machine. And show that those are equivalent to basic Turing machine.  

Chapter 8  

1. Write short notes on post correspondence problem.  
2. Solve the post correspondence problem given below.  

<table>
<thead>
<tr>
<th>i</th>
<th>Wi</th>
<th>Xi</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>111</td>
</tr>
<tr>
<td>2</td>
<td>10111</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

3. Define the following:  
   i) Recursively enumerable (RE) language  
   ii) Recursive language  
   iii) Universal language  
4. Prove that if L is recursive language then L’ is also a recursive language.  
5. Prove that Universal language is Recursively enumerable but not recursive.  
6. Write an explanatory note on NP complete problems.