Module 1: Introduction to theory of computation and FSM

Objective: Upon the completion of this chapter you will be able to

- Define Finite automata,
- Basic principle of computation, formal languages
- Applications of finite automata i.e. DFA & NFA.

1. Define language accepted by DFA
2. Define a regular language
3. Give the formal definition of NFA
4. Define extended transition function for NFA
5. Define language accepted by a NFA
6. Define dead configuration in case of NFA
7. What are the advantages of non-deterministic FA
8. Give the formal definition of NFA
9. Define the terms prefix and suffix of a string, productions, sententential form.
10. Compare NFA & DFA
11. Define the terms alphabet, string, prefix, suffix, language give examples to each.
12. Define an automata for serial binary adder
14. Write a note on applications of formal languages and automata.
15. Explain the operation of a Deterministic Finite Acceptor (DFA) with a diagram.
16. Distinguish between NFA & DFA.
17. Define the equivalence between two finite acceptors?
18. Define distinguishable and indistinguishable states.
19. Derive the DFA that accepts the language \( L = \{a^n b : n \geq 0\} \)
20. Find the DFA that recognizes the set of all strings on \( \Sigma = \{a, b\} \) starting with the prefix “ab”
21. Find the DFA that accepts all strings on alphabet \( \{0, 1\} \) except those containing substring 001.
22. Give the procedure to reduce number of states in DFA.
23. Give Nondeterministic finite Automata accepting the following Language
   The set of strings in \((0+1)^*\) such that some two 0’s are separated by a string whose length is 4i, for some \(i \geq 0\).
24. Give a description about FA with empty moves
25. Construct DFA for the set of all strings beginning with a 1 which interpreted as the binary representation of an integer, is congruent to zero modulo 5
26. Construct DFA accepting the following language The set of all strings such that the 10th symbol from the right end is 1.
27. Explain different units of automata. Explain the terms
   1) Configuration 2) Move 3) Transition functions
   Show that the language \( L = \{a^m b^m : m \geq 0\} \) is regular. Also show that \( L^2 \) is regular?
28. Construct a DFA & NFA to accept all string in \( \{a, b\} \) such that every “a” has one “b” immediately to its right?
29. Define: a) Symbol or element b) Alphabet(\( \Sigma \)) c) String\( (w, u, v) \) d) Concatenation of strings e) Reverse of string f) length of string g) substring, prefix, suffix of a string h) \( w^n \) h) \( \Sigma^* \)
30. Define the language accepted by DFA, when is the language called regular. Show that the language \( L = \{a^m b^m w : m \geq 0\} \) is regular.
31. Draw NFA for transition table given below:

<table>
<thead>
<tr>
<th>States</th>
<th>Input</th>
</tr>
</thead>
</table>

   States: Input
32 Define 

a) Language (L) 

b) Sentence 

c) Complement (L') 

d) $L^\delta$ 

e) $L_1.L_2$ 

f) $L^*$ 

g) $L^+$ 

h) $L^+$ 

33 Give the formal definition of DFA? Explain transition graph? Give an example? Define extended transition function $(\delta^*)$? Define transition table? 

- Draw the transition table, transition diagram, transition function of DFA 
  a) which accepts strings which have odd number of a’s and b’s over the alphabet \{a,b\} 
  b) which accepts string which have even number of a’s and b’s over the alphabet \{a,b\} 
  c) which accepts all strings ending in 00 over alphabet \{0,1\} 
  d) which accepts all strings having 3 consecutive zeros 
  e) which accepts all strings having 5 consecutive ones 
  f) which accepts all strings having even number of symbols?

34 Give DFA & NFA which accept the language \{ (10)^n : n \geq 0 \}

35 Prove the equivalence between DFA & NFA 

OR 

Let $L$ be the language accepted by a NFA $M_N = (Q_N, \Sigma, \delta_N, Q_{N_0}, F_N)$. Then prove that there exists a deterministic finite acceptor $M_D = (Q_D, \Sigma, \delta_D, Q_{D_0}, F_D)$ such that $L = L(M_D)$.

36 Define grammar, proof techniques, language.

37 Convert the following NFA to DFA

1) Reduce the number of states in DFA

**Diagram:**

- DFA with states q0, q1, q2, transitions on a and b.
- NFA with states q0, q1, transitions on 0, 1.
- DFA with states q0, q1, q2, transitions on 0, 1.
Module 2: Regular Expressions and languages, Properties of Regular Languages.

Objective: upon completion of this chapter you will be able to define
- Regular expression and need for regular expression
- Regular languages, regular grammars
- Ways of representing regular languages and their applications
- Proving Languages not to be Regular
- Properties of Regular languages like Closure under Union, Complementation, and Intersection.

1. Give the formal definition of a regular expression with example. 2
2. Define a linear grammar. 2
3. Define unit production. 2
4. Define regular grammar with example. 2
5. How is language L(R) denoted by regular expression “R” defined? Give examples. 5
6. Find all strings in L((a+b)*b(a+ab)*). 5*
7. Show that the automaton generated by procedure reduce is deterministic 5*
8. Write the NFA which accepts L(r) where r = (a + bb)*(ba* + λ). 5*
9. Prove that “Language generated by a right linear grammar is a regular language” 5*
10. Define regular expression. Give a regular expression for L={a,b^n : n ≥ 4, m≤3}. 5*
11. Show that family of regular languages are closed under intersection. 5*
12. Define homomorphism and homomorphic image. Let ∑={a,b} and [a,b,c} and h is defined by h(a) = ab, h(b) = bbc, if w=aba what is h(w)? and if L={aa,aba}, what is h(L)? 5*
13. Define regular expression and language denoted by any regular expression 4*
14. Find regular expression for the language L = {w ∈ {0,1} : w has no pairs of consecutive zeros. 6*
15. Prove the following identities for regular expression r, s, and t where r=s means L(r)=L(s) r+s=s+r, (rs)t=r(st), (r+s)t=rt+st 6
16. Prove or disprove the following for regular expressions r, s, and t (rs+r)=(sr+r)* 6
17. Prove that class of Regular sets is closed under quotient with arbitrary sets 6
18. Prove that the class of regular sets is closed under Substitution 6
19. Prove that the class of Regular sets is closed under homomorphism and inverse homomorphism 6
20. Find the NFA that accepts the language L{ab*aa+bba*ab) 5*
21. Construct right and left linear grammar for the language L={a^n b^m : n≥ 2, m≥ 3} 7*
22. Let L1= L(a*bba*) and L2= L(aba*) find L1/L2 8*
23. Give the set notation of language L(R) denoted by regular expressions given below.
   a) a^* (a+b)
   b) (a+b)^* (a+bb)
   c) (aa)^* (bb)^* b

   Prove the following:
   If the states qa and qb are indistinguishable, and if qc and qa n ar distinguishable, then qb,qc must be indistinguishable.

24. Let r be a regular expression. Then prove that there is some NFA that accepts L(r) & hence L(r) is a regular language. 8
25 Let L be a regular language i.e., there is a NFA that accepts L. Then prove that there exists a regular expression “r” such that L = L(R).

26 Explain generalized transition graphs & how they are used for writing regular expression denoting same language as given NFA.

27 Construct the finite automaton that accepts the language generated by grammar (\{ V_0, V_1 \}, \{ a, b \}, \{ V_0 \}, \{ V_0 \rightarrow aV_1, V_1 \rightarrow abV_0 \} ).

28 P.T. “A language L is regular if and only if there exists a left linear grammar G such that L = L(G)”

29 P.T. “A language L is regular if and only if there exists a regular grammar G such that L = L(G)”

30 Let h be a homomorphism & L a regular language. Then prove that homomorphic image h(L) is also regular.

31 Prove that The set L = \{ 0^{2i} | I is an integer, I \geq 1 \} which consists of all strings of 0’s whose length is a perfect square, is not regular.

32 What are the Applications of Pumping Lemma

33 What are Decision Algorithms for Regular sets

34 Define Emptiness, Finiteness, and Infiniteness, Equivalence

35 Let L be any subset of 0*. Prove that L* is Regular.

36 Show that r = (1+01)^*(0+1*) denotes the language L = \{ w \in \{0,1\}^* : w has no pair of consecutive zeros \} find the other two expressions.

37 Give the set and explain in English the sets denoted by following regular expressions.
   a) (1+0)(00+1)
   b) (1+01+001)(0+00)
   c) (0+1)00(0+1)
   d) 012
   e) 00 11 22

38 Denote the regular languages defined by the following grammar as regular expressions.
   a) G1 = ( \{ S \}, \{ a,b \}, S, \{ S \rightarrow abS | a \} )
   b) G2 = ( \{ S,S1,S2 \}, \{ a,b \}, S, \{ S \rightarrow S1ab, S1 \rightarrow S1ab | S2, S2 \rightarrow a \} )

39 Show that the family of regular languages is closed under following operations
   a) union
   b) intersection
   c) concatenation
   d) complementation
   e) star-closure
   f) difference

40 Is the Class of Regular sets closed under infinite union

41 What is the relationship between the class of regular sets and the least class of languages closed under union, intersection and complement Containing all finite sets

42 Give a finite automaton construction to prove that class of regular sets is closed under substitution

43 Prove that if two finite automata are equivalent they accept the same language

44 Let L be the set of strings of 0’s and 1’s beginning with a 1 whose value treated as a binary number is prime, Prove that L is not regular.

45 What are the properties of Regular sets and prove that given L is not regular with an example

46 What are the closure properties of Regular sets

47 Write NFA & right linear grammar for L(aab*a)

48 Prove that the language L = \{ a^n b^n : n \geq 0 \} is not regular using pigeonhole principle. State and prove pumping lemma for regular languages? What is the application of pumping lemma.

49 Using pumping lemma, prove that following languages are not regular.
a) \( L = \{ a^n b^n : n \geq 0 \} \)

b) \( L = \{ ww^r : w \in \Sigma^* \} \Sigma = \{ a,b \} \)

c) \( L = \{ w \in \Sigma^* : n_a(w) < n_b(w) \} \Sigma = \{ a,b \} \)

d) \( L = \{ (ab)^n a^k : n > k , k \geq 0 \} \)

e) \( L = \{ a^n : n \geq 0 \} \)

f) \( L = \{ a^n b^k c^n k : n \geq 0 , k \geq 0 \} \)

g) \( L = \{ a^n b^l : n \neq l \} \)

50 Define regular expression. Construct an NFA for the \( L((a+b)^*abb) \)  

51 Show that if \( L \) is a regular language on alphabet \( \Sigma \) then there exists a right linear grammar \( G = (V, \Sigma, S, P) \) such that \( L = L(G) \).

52 Given the below NFA, write the corresponding regular expression using generalized transition graphs.

---

Module - 3 : Context free Grammars and PDA

Objective: upon completion of this chapter you will be able to define

- Context free grammars
- Way of describing sentence structure
- How the Ambiguity in grammars can be Reduced
- CFG application in the design of programming languages and in the construction of compilers

---

<table>
<thead>
<tr>
<th>Question</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define a ambiguous CFG.</td>
<td>2</td>
</tr>
<tr>
<td>Define CFG and What are its advantages</td>
<td>5</td>
</tr>
<tr>
<td>Define simple grammar or s-grammar? What are its applications?</td>
<td>5</td>
</tr>
<tr>
<td>Define leftmost and rightmost derivation with example,</td>
<td>5</td>
</tr>
<tr>
<td>Define derivation tree, partial derivation tree, yield,</td>
<td>5</td>
</tr>
<tr>
<td>Explain dependency graph &amp; its applications in CFG.</td>
<td>5</td>
</tr>
<tr>
<td>Write the regular expression for all pascal real numbers.</td>
<td>5</td>
</tr>
<tr>
<td>Explain exhaustive search parsing? What is the serious flaw in using exhaustive search parsing?</td>
<td>5</td>
</tr>
<tr>
<td>Prove the substitution rule of context free grammar?</td>
<td>5</td>
</tr>
<tr>
<td>Find the regular expression for pascal sets whose elements are integer numbers.</td>
<td>5</td>
</tr>
<tr>
<td>Let ( L_1 = L(a^<em>bba^</em>) ) and ( L_2 = L(aba^*) ). find ( L_1/L_2 ).</td>
<td>5</td>
</tr>
<tr>
<td>Define inherently ambiguous language and give an example ?</td>
<td>5</td>
</tr>
<tr>
<td>What are CFG's Give CFG for the language ( L = { a^n b^n</td>
<td>n &gt; 0 } )</td>
</tr>
<tr>
<td>Given the grammar ( G ) as follows: ( S \rightarrow aAS</td>
<td>a, A \rightarrow SB</td>
</tr>
<tr>
<td>Show that the grammar given below is ambiguous ( S \rightarrow E</td>
<td>E \rightarrow E+E )</td>
</tr>
<tr>
<td>What are the applications of CFG</td>
<td>6*</td>
</tr>
<tr>
<td>Give a CFG generating the following set that is the set of palindromes over alphabet ( { a,b } )</td>
<td>5</td>
</tr>
<tr>
<td>Give a CFG for the set of all strings of balanced parenthesis, each left parenthesis has a matching right parenthesis and pairs of matching parenthesis are properly nested</td>
<td>5</td>
</tr>
<tr>
<td>Define context free grammars formally. Give some examples.</td>
<td>5*</td>
</tr>
<tr>
<td>Let ( G ) be the grammar ( S \rightarrow aS</td>
<td>aSbS ) ( \epsilon ) prove that ( L(G) = { x</td>
</tr>
</tbody>
</table>
22 Write CFG which generates the following CFL’s L(G) = \{ w w^r : w \in \Sigma^* \} \Sigma = \{ a,b \}
   a) L(G) = \{ ab(baa)^n bba(ba)^n : n \geq 0 \}
   b) L = \{ a^n b^m : n \neq m \}
   c) L = \{ w \in \{ a,b \}^* : n_a(w) = n_b(w) and n_a(v) \geq n_b(v) \text{ where } v \text{ is any prefix of } w \}
   d) L = \{ a^n b^m : n \geq 0 \text{ m } \geq 0 \}

23 Let G = (V,T,S,P) be a CFG. Then prove that for every w \in L(G), there exists a derivation tree of G whose yield is w.

24 Prove that yield of any derivation tree is in L(G), where G is a CFG.

25 If L is a regular language, prove that the language \{ uv : U \subseteq \mathbb{L} \text{, } v \subseteq \mathbb{L}^R \text{ Is also regular} \}

26 Find DFA’s that accepts the following languages.
   a) L = \{ ab(baa)^n bba(ba)^n : n \geq 0 \}
   b) L = \{ (ab)^n + (a^n b^n) \}
   c) L = \{ a^n b^n : n \geq 0 \}

27 Construct parse tree for the following grammar S -> aAS | a
   A -> SbA | SS | ba

28 Let G=(V,T,P,S) be a CFG, then S=>a if and only if there is a derivation tree in grammar G with yield a

29 Construct Leftmost and Rightmost derivation tree for the following grammar
   S => aAS => aSbAS => aabAS => aabbaS => aabbaa

30 What are ambiguous grammar and inherently ambiguous grammar with an example

31 The grammar E -> E + E | E * E | (E) | id generates the set of arithmetic expressions with +,*, Parentheses and id. Construct an equivalent unambiguous grammar.

32 Show that every CFL without \epsilon is generated by a CFG all of whose productions are of the form A -> a and A -> aB

33 Construct a CFG for the following language:
   L = \{ a^n b^n c^n : n \geq 0 \}

34 Define a CFG which generates the following CFL’s
   L(G) = \{ w w^r : w \in \Sigma^* \} \Sigma = \{ a,b \}

35 Let G be the grammar
   S -> aB | bA, A -> a | aS | bAA, B -> b | bS | aBB
   Find a leftmost and rightmost derivation parse tree for the string aaabbabbba

36 Is the grammar given in q(42) unambiguous if it is prove it

37 What are linear grammar show that if all productions of a CFG are of the form A -> wB or A -> w then L(G) is a regular set

38 Can every CFL without \epsilon be generated by a CFG all of whose productions are of the forms A -> BCD and A -> a?

39 Construct a CFG for the set of all strings over the alphabet \{ a,b \} with exactly twice as many a’s and b’s.

40 Construct parse tree for the following grammar S -> aAS | a
   A -> SbA | SS | ba

41 What are the demerits of regular languages when compared to context free languages

42 Define Linear Context free grammar and write the Pumping lemma for Linear Languages.

43 Construct a CFG for the following Language with n>=0, m>=0
   L = \{ a^n b^m : W \in \{ a,b \}^* \}

44 Show that family of CFL is closed under union, concatenation and star closure

45 Show that the language L = \{ a^n b^n c^n : n \geq 1 \} is not a CFL

### Pushdown Automata

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Define the instantaneous description of a NPDA</td>
</tr>
<tr>
<td>2</td>
<td>Give the formal definition of DPDA and deterministic CFL.</td>
</tr>
<tr>
<td>3</td>
<td>Define Linear Context free grammar and write the Pumping lemma for Linear Languages.</td>
</tr>
<tr>
<td>4</td>
<td>Distinguish between DPDA and NPDA</td>
</tr>
<tr>
<td>5</td>
<td>What are the demerits of regular languages when compared to context free languages</td>
</tr>
</tbody>
</table>

---

**PEST-BSC, DEPT OF ISE**
6. What are the demerits of DFA (or NFA) when compared with PDA?
7. Why FAs are less powerful than the PDA’s?
8. How the Transition/move of a PDA defined?
9. State and prove pumping lemma for CFL. What is its application?
10. Give two reasons why finite automata cannot be used to recognize all CFL & why PDA is required for that purpose.
11. Explain the operations of a NPDA with diagram?
12. Write a NPDA that accepts the language L = \{a^n b^n : n \geq 0 \} U \{a\}.
13. When do we say a CFL is accepted by NPDA. Define
   a) acceptance by final state.
   b) Acceptance by empty stack.
14. Define PDA. Describe the acceptance by “final State” and acceptance by “empty Stack”.
15. What does each of the following transitions represent?
   a. \(\delta(p,a,Z) = (q,aZ)\)
   b. \(\delta(p,a,Z) = (q,\varepsilon)\)
   c. \(\delta(p,a,Z) = (q,r)\)
   d. \(\delta(p,\varepsilon,Z) = (q,r)\)
   e. \(\delta(p,\varepsilon,\varepsilon) = (q,Z)\)
   f. \(\delta(p,\varepsilon,Z) = (q,\varepsilon)\)
16. Give the formal definition of NPDA. Explain clearly the transition function?
17. If L is a CFL, then there exists a Pda M such that L= N(M).
18. If L is N(M1) for some PDA M1, then L is L(M2) for some PDA M2.
19. If L is L(M2) for some PDA M2, then L is N(M1) for some PDA M1.
20. When the PDA is Deterministic and when it is called nondeterministic.
21. Is the PDA to accept the Language L(M)={wCw^R | W ∈ (a+b)^*} is deterministic.
22. Construct a NPDA for the following languages
   a) \(L = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}\)
   b) \(L = \{ww^r : w \in \{a,b\}^+\}\)
23. Show that the language L = \{a^n b^n : n \geq 0, n \neq 100\} is context free.
24. Prove that for any CFL L(specified as CFG without \(\lambda\) productions), there exists a NPDA M such that L = L(M).
25. Obtain a PDA to accept the language L(M)=\{W | W ∈ (a+b)^* and na(W)=nb(W) i.e the number of a’s in string w should be equal to number of b’s in w.
26. What is an instantaneous description? Explain with respect to PDA.
27. Construct a NPDA that accepts the language generated by grammar with productions
   a) \(S \rightarrow aA\)
   b) \(S \rightarrow Aabc \mid bB \mid a\)
   c) \(B \rightarrow b\)
   d) \(C \rightarrow c\)
28. Obtain a PDA to accept the Language L*(M)=\{wCw^R | W ∈ (a+b)^*\}. Where \(W^R\) is reverse of W. Show the sequence of moves made by the PDA for the string aabCbaa, aabCbab.
29. If L = L(M) for some NPDA M, then prove that L is CFL.
30. Write the CFG for language accepted by NPDA whose transitions are given below :-
   \(\delta(q0,a,z) = \{(q0,Az)\}\)
   \(\delta(q0,a,A) = \{(q0,A)\}\)
   \(\delta(q0,b,A) = \{(q1,\lambda)\}\)
   \(\delta(q1,\lambda,z) = \{(q2,\lambda)\}\)
31. Obtain a PDA to accept a string of balanced Parentheses. The parentheses to be considered are(,),[],{ and }.
32. Show that following languages are not context free using pumping lemma
   a) \(L = \{a^n b^n c^n : n \geq 0\}\)
   b) \(L = \{ww : w \in \{a,b\}^+\}\)
   c) \(L = \{a^n : n \geq 0\}\)
   d) \(L = \{a^n b^n : n = j^2\}\)
Define linear CFL. State pumping lemma for Linear CFL.
33 Obtain a PDA to accept the Language \( L = \{ w \mid w \in (a,b)^* \text{ and } na(w) > nb(w) \} \)

34 Construct an npda that accepts the language generated by the grammar
\[
S \rightarrow aABB \mid aAA \\
A \rightarrow aBB \mid a \\
B \rightarrow bBB \mid A
\]

35 Construct the NPDA Corresponding to the grammar
\[
S \rightarrow aA, A \rightarrow aABC \mid bB \mid a, B \rightarrow b, C \rightarrow c
\]
Derive the string for the grammar and show the sequence of moves made by NPDA in Processing the same string.

36 Show that language \( L = \{ w \mid na(w) = nb(w) \} \) is not linear.

37 Design PDA for the language \( L = \{ a^n b^n \mid n \geq 0 \} \) give the trace for the input aaabbb

38 Define an NPDA. Discuss about the language accepted by a Push down automata. Design an NPDA for the Language \( L = \{ W : na(W) = nb(w) + 1 \} \)

39 Construct an NPDA that accepts the Language accepted by the grammar \( S \rightarrow aA,A \rightarrow aABC/bB/a, B \rightarrow b, C \rightarrow c \)

40 Design a PDA for the following language \( L = \{ a^n b^n \mid n \geq 0 \} \). give the trace for the input aaabbb

41 Construct an NPDA Corresponding to the grammar
\[
S \rightarrow aA, A \rightarrow aABC \mid bB \mid a \\
B \rightarrow b \\
C \rightarrow c
\]

42 Obtain NPDA for the language \( L = \{ wwR : w \in (0+1)^* \} \)
Show that accessible instantaneous description for the string 001100

43 Construct the PDA equivalent to the following grammar
\[
S \rightarrow aAA, A \rightarrow aS \mid bS \mid a
\]

44 Show that if \( L \) is a CFL, then there is a PDA \( M \) accepting \( L \) by final state such that \( M \) has at most two states and makes no \( \varepsilon \) moves

45 If \( L \) is \( N(M) \) for some PDA \( M \), then \( L \) is a Context-free Language

46 For the Grammar
\[
S \rightarrow aABB \mid aAA \\
A \rightarrow aBB \mid a \\
B \rightarrow bBB \mid A \\
C \rightarrow a
\]
Obtain the Corresponding PDA

47 For the grammar
\[
S \rightarrow aABC \\
A \rightarrow aB \mid a \\
B \rightarrow bA \mid b \\
C \rightarrow a
\]
Obtain the Corresponding PDA

48 What is the Procedure to convert a CFG to PDA

49 What is application of GNF notation of a CFG? Is the PDA to accept the language consisting of balanced parentheses is deterministic

50 What is the general procedure used to convert from PDA to CFG

---

**Module 4 Properties of Context-Free Languages.**

**Objective:** Upon Completion of this chapter you will be knowing about
- Different kinds of Normal forms for CFG’s that is CNF & GNF.
- How the CFL can be proved as not Context free(Pumping Lemma for CFG)
- Closure Properties of CFL’s i.e Closure Properties under Union, Intersection, Complementation.

<p>| 1 | What is a normal form &amp; why is it required? | 4 |
| 2 | Explain the method of Substitution with examples | 4 |
| 3 | What is Left Recursion? How it can be Eliminated | 4 |
| 4 | What is the need for simplifying a Grammar | 4 |
| 5 | Define CNF of a CFG. | 6 |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 6 | Convert the following CFG into CNF  
S → bA | aB  
A → bAA | aS | a  
B → aBB | bS | b |   |
| 7 | Eliminate Left Recursion from the following grammar  
E → E + T | T  
T → T* F | F  
F → (E) | id |   |
| 8 | Eliminate Left Recursion from the following Grammar  
S → Ab | a  
A → Ab | Sa |   |
| 9 | Is the following Grammar ambiguous  
S → aSh | SS | ε |   |
| 10 | Define Greibach normal form convert the following grammar  
S → Abb | a, A → aaA | B, B → bAb into the Greibach normal form |   |
| 11 | Convert the grammar with productions S → Aba, A → aab, B → Ac to CNF. |   |
| 12 | Obtain the following grammar in CNF  
S → aA | a | B | C  
A → aB | ε  
B → aA  
C → εCD  
D → abd |   |
| 13 | Obtain the following grammar in GNF  
S → aA | a | B | C  
A → aB | ε  
B → aA  
C → εCD  
D → abd |   |
| 14 | Define CNF and GNF Convert the following grammar to CNF  
S → S | [s ≥ S] | p | q (S being the only variable.) |   |
| 15 | Prove the family of CFL’s are not closed under intersection and Complementation |   |
| 16 | What are ambiguous grammars and inherently ambiguous grammars, give an example for each |   |
| 17 | Prove that family of CFL is closed under union, concatenation and star closure. |   |
| 18 | Prove that family of CFL is not closed under intersection and complementation. |   |
| 19 | Let L1 be a CFL and L2 be a regular language. Then prove that L1 INTERSECTION L2 is context free. |   |
| 20 | Show that the language L = { w ∈ {a,b,c}* : n_a(w) = n_b(w) = n_c(w) } is not context free. |   |
| 21 |   |   |
| 22 | What is CNF and GNF form Explain with an Example? |   |
| 23 | Prove that for every CFG we can have an equivalent grammar using CNF notations where a language does not contain ε |   |
| 24 | What is the general Procedure to convert a grammar into its equivalent GNF notation |   |
| 25 | Convert the following grammar into GNF  
S → AB1 | 0  
A → 00A | B  
B → 1A1 |   |
| 26 | Convert the following grammar into GNF  
A → BC  
B → CA | b  
C → AB | a |   |
| 27 | State and prove Pumping Lemma for Context free Languages |   |
| 28 | What are the Applications of pumping Lemma |   |
| 29 | Show that L = {a^n b^n c^n | n ≥ 0} is not Context free |   |
30 Prove that CFLs are not closed under intersection and Complementation
31 Prove that CFL's are closed under Union, Concatenation, and star closure
32 Show that L= \{Ww \mid W \in \{a, b\}^*\} is not Context free
33 Show that L= \{a^p b^q \mid p=q^2\} is not context free
34 Show that L=\{an^i \mid n>0\} is not Context free

Module 5: Turing machines, Decidability, Complexity and Computability

Objective: upon completion of this chapter you will be able to define
- Turing machines
- Limitations of computation
- Fundamental concept of Turing machine
- How Turing machine acts as transducers
- Nondeterministic Turing machine

1 Define computations of a TM? 5
2 Explain with diagram the operation of Turing machines? Give formal definition of Turing machine. 5*
3 Explain what is meant by instantaneous description of a TM? 5
4 For \(\Sigma = \{a,b\}\) design a TM that accepts \(L = \{ a^nb^n : n \geq 1 \}\) 5*
5 Design a TM that accepts \(L = \{ a^nb^nc^n : n \geq 1 \}\) 5*
6 Define language accepted by TM? 5
7 When do we say that a language is not accepted by TM? 5*
8 Define formally non-deterministic TM. 5
9 On what basis we say that TM is transducer 5
10 Define the operation of TM as transducers? Define a Turing computable function? 5

11 What is Turing Computable 5
12 Write a note on multidimensional TM. 5
13 Write a note on universal TM. 5
14 Obtain a Turing machine to accept the language \(L=\{0^n1^n \mid n=1\}\) 8
15 Obtain a Turing machine to accept the language \(L(M)=\{0^n1^{2n} \mid n=1\}\) 8
16 Obtain a Turing machine to accept the language \(L=\{W \mid w\in(0+1)^*\}\) containing the sub string 001 8
17 Obtain a TM to accept the language containing strings of 0’s and ‘s ending with 011 8
18 Give an example of TM that never halts i.e., that goes to infinite loop? How is that represented in instantaneous description? 8
19 Given two positive integers x and y, design a TM that computes x+y 8
20 Design a TM that copies strings of 1’s 8
21 Design a Turing machine that halts at a final state if \(x\geq y\) and at a non-final state if \(x<y\) 8
22 Design a TM that computes the function \(F(x,y) = \begin{cases} x + y & \text{if } x \geq y \\ 0 & \text{if } x<y \end{cases}\) 8
23 Design a TM to implement the macro instruction

\[
\text{If } a \quad \text{Then } qj \\
\text{Else } qk
\]

24 Design a TM that multiplies two +ve integers in unary notation 8
25 Write a note on Turing Thesis. Define algorithm in terms of TM. 8
26 Define equivalence of automata? Demonstrate the equivalence of TM using simulation. 8
27 Obtain a TM to accept a palindrome consisting of a’s and b’s of any length 8
28 Let x and y are two Positive integers. Obtain a Turing machine to perform x+y 8
29 Given a string w design a TM that generates the string ww where we a* 8
30 Define TM with stay on option. Prove that they are equivalent to class of standard TM? 8
31 Prove that class of deterministic TM & class of non-deterministic TM are equivalent. 8
32 Explain what do you mean by countable, uncountable sets and enumeration procedure? 8*
33 Prove that set of all TM, although infinite is countable 8
34 Define linear bounded automata (LBA)? When do we say that a string is accepted by a LBA?

35 Find a LBA that accepts the language \( L = \{ a^n : n \geq 0 \} \)

36 Define TM with semi-infinite tape & prove that they are equivalent to class of standard Turing machine.

37 Define offline TM & prove that they are equivalent to class of standard TM.

38 Construct a TM that stays in the final state qf whenever \( x \geq y \) and non-final state qn whenever \( x < y \) where \( x \) and \( y \) are positive integers represented in unary notation

39 What are the various variations of TM? How to achieve complex tasks using TM

40 Prove that if a Language is accepted by a multitape Turing machine, it is accepted by a single tape Turing machine

41 What are the different techniques for construction of Turing machine

42 What are nondeterministic and multidimensional Turing machine

43 Design Turing machine to compute \( \log_2 n \)

44 Design Turing machine to compute \( n! \)

45 Design Turing machine to compute \( n^2 \)

46 Define Turing Machine , Give Turing Machine to implement , the total recursive function “multiplication”. The Turing machine starts with \( O^m \mid O^n \) on its tape and ends with \( O^{mn} \) surrounded by blanks

47 What is a multi-tape Turing machine? Show how it can be simulated using single tape Turing machine

48 Write short notes on:
   Halting Problem of Turing Machine
   Application of CFG
   Multi Tape Turing Machine
   Post Correspondence Problem

49 Write short notes on:
   Context Sensitive Grammar & Languages
   Chomsky Hierarchy
   Pumping Lemma for Regular Languages
   Post Correspondence Problem

50 Define the following
   Turing machine with stay option
   Turing machine with multiple tracks
   Turing machine with semi-infinite tape
   Off-line Turing machine

1 Define unrestricted grammar.

2 Explain what is Undecidability

3 Define a recursively enumerable language & a recursive language?

4 Define computability and decidability

5 What are Recursive and Recursively Enumerable Languages

6 What is the need for reducing one undecidable problem to other?

7 Define Valid and Invalid Computation of TM’s

8 Discuss the properties of Recursive Enumerable Languages

9 Discuss the Properties of Recursively Enumerable Languages

10 What is the modified version of PCP

11 What are Universal Turing Machines

12 Define Non recursively enumerable Language

13 Define Universal Language

14 Give convincing arguments that any language accepted by an off line Turing machine is also accepted by some standard machine.
<table>
<thead>
<tr>
<th>Question</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15. Prove that Language $L_u$ is Recursively Enumerable</td>
<td>8</td>
</tr>
<tr>
<td>16. Let $S$ be an infinite countable set. Then prove that its power set $2^S$ is not countable.</td>
<td>8*</td>
</tr>
<tr>
<td>17. Discuss on Rice's Theorem and Undecidable Problems</td>
<td>8</td>
</tr>
<tr>
<td>18. Prove that Language $L_u$ is not Recursive</td>
<td>8</td>
</tr>
<tr>
<td>19. Discuss the properties of R.E sets which are not r.e</td>
<td>8</td>
</tr>
<tr>
<td>20. Discuss Rice's theorem for Recursive index sets</td>
<td>8</td>
</tr>
<tr>
<td>21. Discuss the problems about Turing Machine</td>
<td>8</td>
</tr>
<tr>
<td>22. If PCP were decidable, then MPCP would be decidable that is MPCP reduces to PCP</td>
<td>8</td>
</tr>
<tr>
<td>23. Discuss the properties of R.E sets which are R.E</td>
<td>8</td>
</tr>
<tr>
<td>24. What is the Undecidability of PCP</td>
<td>8</td>
</tr>
<tr>
<td>25. Discuss the Application of PCP</td>
<td>8</td>
</tr>
<tr>
<td>26. Prove that PCP is Undecidable</td>
<td>8</td>
</tr>
<tr>
<td>27. What is the Undecidability of Post Correspondence Problem</td>
<td>8</td>
</tr>
<tr>
<td>28. Prove that a language generated by an unrestricted grammar is recursively enumerable.</td>
<td>8</td>
</tr>
<tr>
<td>29. Discuss the Rice’s theorem for recursively enumerable index sets</td>
<td>8</td>
</tr>
<tr>
<td>30. Give the procedure for writing an unrestricted grammar which accepts the language accepted by a given TM.</td>
<td>8*</td>
</tr>
<tr>
<td>31. Prove that for every recursively enumerable language $L$ there exists an unrestricted grammar $G$ such that $L = L(G)$.</td>
<td>8*</td>
</tr>
<tr>
<td>32. Prove that The Complement of a Recursive Language is Recursive</td>
<td>8</td>
</tr>
<tr>
<td>33. Prove that The union of two recursive Language is Recursive</td>
<td>8</td>
</tr>
<tr>
<td>34. Prove that The Union of two recursively enumerable Languages is recursively enumerable</td>
<td>8</td>
</tr>
<tr>
<td>35. Write a note on Chomsky Hierarchy</td>
<td>8</td>
</tr>
<tr>
<td>36. If a Language $L$ and its complement are both recursively enumerable then $L$ and its complement is recursive.</td>
<td>8</td>
</tr>
<tr>
<td>37. Explain state entry problem &amp; blank tape halting problem. How can halting problem be reduced to above problems?</td>
<td>8</td>
</tr>
<tr>
<td>38. Define a context sensitive grammar? Why it is called non-contracting? Define context sensitive language?</td>
<td>10</td>
</tr>
<tr>
<td>39. What is meant by Halting problem of Turing machine? Explain the blank tape halting problem</td>
<td>10</td>
</tr>
<tr>
<td>40. Write a detailed note on The Chomsky hierarchy, Linear bounded automata, Post Correspondence Problem</td>
<td>10</td>
</tr>
<tr>
<td>41. Write a CSG for language $L = { a^n b^n c^n : n \geq 1 }$</td>
<td>10*</td>
</tr>
<tr>
<td>42. For every CSL not including $\lambda$, prove that there exists some linear bounded automaton $M$ such $L = L(M)$. Prove the the converse also</td>
<td>10</td>
</tr>
<tr>
<td>43. Prove that 1) Every CSL $L$ is recursive. 2) There exists a recursive language that is not context sensitive.</td>
<td>10</td>
</tr>
<tr>
<td>44. Prove that it is Undecidable for arbitrary CFG's $G_1$ and $G_2$ whether $L(G_1)\cap L(G_2)$ is empty.</td>
<td>10</td>
</tr>
<tr>
<td>45. Define &amp; Explain TM halting problem? Prove that halting problem is undecidable?</td>
<td>10</td>
</tr>
<tr>
<td>46. Prove that it is undecidable for any arbitrary CFG $G$ whether $L(G)=\Sigma^*$</td>
<td>10</td>
</tr>
<tr>
<td>47. What are the Applications of Greibach’s theorem</td>
<td>10</td>
</tr>
<tr>
<td>48. A Turing machine is one that cannot change a non blank symbol to a blank. Which can be achieved by restriction that $\delta(q_i,a)=(q_j,\varepsilon,L$ or $R)$. Then a must be $\varepsilon$.show that no generality is lost by making such a restriction.</td>
<td>10</td>
</tr>
<tr>
<td>49. Write short notes on: a) Application of Finite Automata</td>
<td>20*</td>
</tr>
<tr>
<td>b) Linear Bounded automata</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>c)</strong> Turing Machine Halting Problem</td>
<td></td>
</tr>
<tr>
<td><strong>d)</strong> Chomsky Hierarchy.</td>
<td></td>
</tr>
<tr>
<td><strong>50</strong></td>
<td>Prove that It is undecidable for arbitrary CFG's G1 and G2 whether ( \text{Complement } L(G1) ) is a CFL and ( L(G1) ) intersection ( L(G2) ) is a CFL</td>
</tr>
<tr>
<td><strong>51</strong></td>
<td>Write short notes on the following:</td>
</tr>
<tr>
<td></td>
<td>a) Chomsky hierarchy</td>
</tr>
<tr>
<td></td>
<td>b) Unrestricted grammar</td>
</tr>
<tr>
<td></td>
<td>c) Post correspondence problem</td>
</tr>
<tr>
<td></td>
<td>d) Linear bounded automata</td>
</tr>
</tbody>
</table>