COMPUTER SCIENCE & INFORMATION TECHNOLOGY

THEORY OF COMPUTATION

Volume-1 : Study Material with Classroom Practice Questions
1. **Introduction**

01. **Ans:** (d)
   **Sol:**
   
   (a) \( \{x | x \geq 10 \text{ or } x \leq 5 \} \) is infinite set
   (b) \( \{x | x \geq 10 \text{ or } x \leq 100 \} \) is infinite set
   (c) \( \{x | x \leq 100 \text{ or } x \geq 200 \} \) is infinite set

02. **Ans:** (b)
   **Sol:**
   
   (a) Set of real numbers between 10 and 100 is uncountable
   (b) \( \{x | x \geq 10 \text{ or } x \leq 100 \} \) is finite set. So countable
   (c) Set of real numbers between 0 and 1 is uncountable

03. **Ans:** (d)
   **Sol:**
   
   (a) \(|\varepsilon| = 0\)
   (b) \(|\{\}| = 0\)
   (c) \(|\{\varepsilon\}| = 1\)

04. **Ans:** (b)
   **Sol:**
   
   \( \Sigma = \{0,1\} \)
   00, 01, 10, 11 are 2 length strings

05. **Ans:** (b)
   **Sol:**
   
   \( w = abc \)
   Prefix(w) = \( \{\varepsilon, a, ab, abc\} \)

06. **Ans:** (b)
   **Sol:**
   
   \( w = abc \)
   Suffix(w) = \( \{\varepsilon, c, bc, abc\} \)

07. **Ans:** (d)
   **Sol:**
   
   \( w = abc \)
   Substring(w) = \( \{\varepsilon, a, b, c, ab, bc, abc\} \)

08. **Ans:** (a)
   **Sol:**
   
   Language accepted by finite automata is called as Regular language.

09. **Ans:** (d)
   **Sol:**
   
   Every recursive language is REL but REL need not be recursive language.

10. **Ans:** (b)
    **Sol:**
    
    Every regular grammar is CFG but CFG need not be regular grammar.

2. **Regular Languages**
   (finite automata, Regular expression, regular grammar)

01. **Ans:** (a) & (c)
    **Sol:**
    
    Regular Languages are closed under
   i) string reversal
   ii) intersection with finite sets

02. **Ans:** (c)
    **Sol:**
    
    A minimal DFA that is equivalent to a NFA with n states has atmost \( 2^n \) states.
03. Ans: (a)
Sol: (a) \((1+01)^* (\varepsilon +0)\) generates all strings not containing ‘00’
(b) \((0+10)^* (\varepsilon +1)\) generates invalid string ‘00’
(c) \((1+01)^*\) cannot generate ‘0’
(d) \((\varepsilon +0) (101)^* (\varepsilon +0)\) generates invalid string ‘00’

04. Ans: (a)
Sol:

05. Ans: (d)
Sol: Given grammar generating all strings ending in ‘00’

06. Ans: (a)
Sol:

\[= 4 \text{ states}\]

07. Ans: (a)
Sol:

08. Ans: (b)
Sol: Concatenation of two infinite languages is also infinite. So, infinite languages closed under concatenation.

09. Ans: (c)
Sol: \(\{wxw^R \mid x, w \in (0+1)^+\} = 0(0+1)^+0+1 (0+1)^+ 1\)
\[\therefore \text{It is regular language}\]

10. Ans: (a)
Sol: (I) NFA with many final states can be converted to NFA with only one final state with the help of \(\varepsilon\)-moves.
(II) Regular sets are not closed under infinite union
(III) Regular sets are not closed under infinite intersection
(IV) Regular languages are closed under substring operation
\[\therefore \text{I and IV are correct}\]

11. Ans: (d)
Sol: \(r = (0+1)^* 00(0+1)^*\)
A→0B | 0A | 1A
B→0C\0
C→0C|1C|0|1

12. Ans: (a)
Sol: \(A_n = \{a^k \mid k \text{ is a multiple of } n\}\)
For some \(n\),
\(A_n\) is regular
Let \( n = 5 \),
\[ A_n = A_5 = \{a^k | k \text{ is multiple of } 5\} = \text{regular.} \]

13. Ans: (d)
Sol: \( L = \{a^m b^n | m \geq 1, n \geq 1\} = a^+ b^+ \) is regular.

14. Ans: (c)
Sol: DFA accepts \( L \) and has \( m \) states
It has 2 final states. It implies \( (m-2) \) non-final states.
DFA that accepts complement of \( L \) also has \( m \) states but it has \( (m-2) \) final states and 2 non-final states.

15. Ans: (d)
Sol: (a) \( 0^* (1+0)^* \); It generates invalid string ‘100’
(b) \( 0^* 1010^* \); It cannot generate valid string ‘e’
(c) \( 0^* 1^*01^* \); It cannot generate valid string ‘e’
(d) \( 0^* (10+1)^* \); It generates all strings not containing ‘100’ as substring

16. Ans: (a)
Sol: P1: Membership problem for FA is decidable
P2: Infiniteness problem for CFG is decidable
For P1, CYK algorithm exist
For P2, Dependency graph exist

17. Ans: (b)
Sol: \( L = \) set of all binary strings whose last 2 symbols are same.

18. Ans: (a)
Sol: \( L = a^n b^n \) is not regular
It can be proved using Pumping Lemma
\( L \) does not satisfy Pumping Lemma

19. Ans: (c)
Sol: It requires 29099 remainders to represent the binary numbers of the given language.
So, 29099 states required.

20. Ans: (d)
Sol: The following problems are decidable for regular languages. Equivalence, Finiteness, Emptiness, infiniteness, totality, containment, Emptiness of complement, Emptiness of intersection, Emptiness of complement of intersection.

21. Ans: (a)
Sol: I. \( \{a^n b^{2m} | n \geq 0, m \geq 0\} \Rightarrow \text{Regular} \)
II. \( \{a^n b^{m} | n = 2m\} \Rightarrow \text{not regular} \)
III. \( \{a^n b^{m} | n \neq m\} \Rightarrow \text{not regular} \)
IV. \( \{x \subset y | x, y \in \{a, b\}^*\} \Rightarrow \text{Regular} \)
So, I & IV are correct.
22. Ans: (c)
Sol: Let n = 3
If w = abc,
Substrings of w = {ε, a, b, c, ab, bc, abc}
non empty substrings of w = {a, b, c, ab, bc, abc}
number of substrings of w of length n is ≤ \((\Sigma n)+1\)
number of non empty substrings of w of length n ≤ \((\Sigma n)\).

23. Ans: (c)
Sol:

\[
\begin{array}{c|cc}
\delta & a & b \\
\hline
\rightarrow A & 3 \text{ choices} & 3 \text{ choices} \\
B & 3 & 3 \\
C & 3 & 3 \\
\end{array}
\]

3×3×3×…6 times = 3^6 machines possible with ‘A’ as initial state.
Final states can be any of subset of {A, B, C}
So, 2^3 possible final states combination.
Total 8×3^6 DFAs.
Number of DFAs with atleast 2 final states = 4×3^6.

24. Ans: (a)
Sol:

= 4 states

25. Ans: (b)
Sol:

26. Ans: (b)
Sol:

(i) \(S \rightarrow aAB\)
(ii) \(S \rightarrow aA \mid bB\)
(iii) \(A \rightarrow a \mid b\)
(iv) \(S \rightarrow aA \mid bB\)

(i) & (iii) are equivalent.

27. Ans: (c)
Sol: 
\[L = (a+b)^* b (a+b)^*\]
strings of length ≤ 3:
b, ab, ba, bb, aab, aba, abb, baa, bab, bba, bbb
Number of strings = 11
28. Ans: (b)
Sol: \( r = (0^* + (10)^*) = (0+10)^* \)
\( s = (0^*+10)^* \)
\( \therefore L(r) = L(s) \)

29. Ans: (d)
Sol: The following sets are countable sets.
1) Set of regular sets
2) Set of CFLs
3) Set of Turing Machines
The set of real numbers is uncountable
The set of formal languages is uncountable.

30. Ans: (a)
Sol: 
\[
\begin{align*}
&\text{a} \quad \text{b} \\
&\downarrow \\
&\text{1} \quad \text{2} \quad \text{3} \\
&\text{4} \quad \text{5} \quad \text{6} \\
&\text{a} \quad \text{b} \\
&\text{a} \quad \text{b} \\
&\text{a} \\
&\text{34} \\
&\text{a} \\
\end{align*}
\]
2 Equivalence classes.

31. Ans: (c)
Sol: \( L = ((01)^* 0^*)^* \)
\[
\begin{align*}
&h(a) = 0 \\
&h(b) = 01 \\
&h^{-1}(0) = a \\
&h^{-1}(01) = b \\
&h^{-1}(L) = (b^* a^*)^* = (a+b)^*
\end{align*}
\]

32. Ans: (a)
Sol: \( L_1 = a^*b \\
L_2 = ab^* \)

33. Ans: (d)
Sol: (a) \( L(r^*) \supset L(r^+) \)
(b) \( L((r+s)^*) \supset L(r^*+s^*) \)
(c) \( L((r+s)^*) \supset L((rs)^*) \)
(d) \( L(r^*) = L((r^*)^*) \)

34. Ans: (b)
Sol: Arden’s lemma cannot be applied to NFA with \( \varepsilon \) moves.
Arden’s lemma applied to both DFA and NFA without \( \varepsilon \) moves.

35. Ans: (d)
Sol: Logic circuits, neural sets, toy’s behavior can be modeled with regular sets.

36. Ans: (a)
Sol: \( L = (0+1)^* 00 \)

37. Ans: (c)
Sol:
\[
\begin{array}{c|cc}
& 0 & 1 \\
\hline
q_{10} & q_0 & q_1 \\
q_1 & q_2 & q_3 \\
q_2 & q_4 & q_0 \\
q_3 & q_1 & q_2 \\
q_4 & q_3 & q_4 \\
\end{array}
\]
\( = 5 \) states
38. Ans: (a)  
Sol: 3rd symbol from ending is ‘1’  
\[ \downarrow \]  
DFA has \( 2^3 \) states.

39. Ans: (a)  
Sol:  
\( L = \{a^i b^j | i<100, j = 10000\} \)  
\[ = \{\varepsilon, a, b, \ldots, a^{99} b^{10000}\} \]  
L is finite set.

40. Ans: (a)  
Sol:  
\[ L = (0+1)^* 0001 (0+1)^* \]  
DFA accepts \( L \) with 5 states  
DFA that accepts complement of \( L \) also requires 5 states.  
DFA that accepts complement of \( L \).

41. Ans: (a)  
Sol:  
\[ (00)^* + 0 (00)^* + 00 (000)^* \]  
\( (00)^* = \) set of all even strings  
\( 0(00)^* = \) set of all odd strings  
\( (00)^*+0(00)^* = \) set of all strings  
\[ = 0^* \]  
\[ \therefore (00)^*+0(00)^*+00(000)^* = 0^* \]

42. Ans: (d)  
Sol:  
\[
\begin{array}{c|ccc}
 & 0 & 1 & 2 \\
\hline
 q_0 & q_0 & q_1 & q_2 \\
 q_1 & q_3 & q_4 & q_5 \\
 q_2 & q_0 & q_1 & q_2 \\
 q_3 & q_3 & q_4 & q_5 \\
 q_4 & q_0 & q_1 & q_2 \\
 q_5 & q_3 & q_4 & q_5 \\
\end{array}
\]

Number of states = 3  
\{q_0, q_2, q_4\}, \{q_1, q_3\}, \{q_5\}

43. Ans: (b)  
Sol:  
i) \( a^n \ v n \geq 1 \) is not regular  
ii) \( a^\text{prime} \) is not regular  
iii) \( \{0^i \ 1^j | i<j<1000\} \) is finite. So regular  
iv) Complement of \( L \) where  
\[ L = (0+1)^* 00010101001010010(0+1)^* \]  
is also regular  
\[ \therefore (iii) \& (iv) \text{ are regular sets.} \]

44. Ans: (b)  
Sol:  
i) \( n^\text{th} \) symbol from right end is ‘1’ \( \Rightarrow 2^n \) states  
ii) \( n^\text{th} \) symbol from left end is ‘1’ \( \Rightarrow (n+2) \) states.  
\[ \therefore (i) \text{ has 64 states (ii) has 7 states.} \]

45. Ans: (c)  
Sol:  
\[ L = \{w | w \in (a+b+c)^*, n_a(w) = n_b(w) = n_c(w)\} \]  
L is not regular because symbols have dependency.

46. Ans: (a)  
Sol:  
If \( X = r+Xs \) and \( s \) has no ‘\( \varepsilon \)’ then \( x \) has unique solution otherwise infinite solutions.
3. Context Free Languages (CFG, PDA)

01. Ans: (c)
Sol: CFLs are closed under:
   i) Finite union
   ii) Union
   iii) Concatenation
   iv) Kleene closure
   v) Reversal
CFLs are not closed under:
   i) Intersection
   ii) Complement
   iii) Infinite union

02. Ans: (a)
Sol: CFLs are closed under:
   i) Finite union
   ii) Homomorphism
   iii) Inverse Homomorphism
   iv) Substitution
   v) Reversal
   vi) Init
   vii) Quotient with regular set.

03. Ans: (d)
Sol: CFLs are not closed under:
   i) Intersection
   ii) Intersection with non CFL
   iii) Infinite union

04. Ans: (a)
Sol: Decidable problems for CFLs.
   i) Emptiness
   ii) Finiteness

iii) Non emptiness
iv) Non finiteness (infiniteness)
v) Membership

Following problems are undecidable about CFLs:
   i) Equivalence
   ii) Containment
   iii) Totality

05. Ans: (a)
Sol: i) \{0^n 1^n | n> 99\} is CFL
    ii) \{a^n b^n c^n | n< 990\} is finite, So CFL
    iii) \{a^n b^m c^l | m = l or m = n\} is CFL
    iv) \{ww | w\in (a+b)^* and |w|<1000\} is finite, so CFL
All languages are CFLs

06. Ans: (a)
Sol: \(L_1= \{ww | w\in (0+1)^*\}\) is not CFL
\(\Sigma^* - L_1\) is CFL
\(L_2 = \{a^n b^n c^n | n>1\}\) is not CFL
\(\Sigma^* - L_2\) is CFL.

07. Ans: (b)
Sol: i) \{ww^R | w\in (a+b)^*\} is CFL but not DCFL
   ii) \{wSw^R | w\in (a+b)^*\} is DCFL but not regular
∴ (ii) accepted by DPDA but (i) accepted by PDA.
08. Ans: (b)
Sol: i) \( \{0^n 1^n \mid n>1\} \) is DCFL
ii) \( \{0^n 1^{2n} \mid n>1\} \cup \{0^n 1^n \mid n>10\} \) is CFL but not DCFL
\[ \therefore \text{ (i) accepted by DPDA and (ii) accepted by PDA.} \]

09. Ans: (c)
Sol: \( S \rightarrow SS|a|\varepsilon \)
It is an ambiguous CFG.
Every string generated by the grammar has more than one derivation tree.

10. Ans: (a), (b) and (c)
Sol: \( S \rightarrow a|A \)
\( A \rightarrow a \)
It is an ambiguous CFG and has 2 parse trees for string ‘a’
For string ‘a’, 2 parse trees, 2 LMD’s and 2 RMD’s are there.

11. Ans: (d)
Sol: \( L = \{a^m b^n c^n \mid m, n > 1\} \)
\( L = \{a^n b^n c^n\} \)
unambiguous CFG that generates L:
\( S \rightarrow ABC \)
\( A \rightarrow aA|aa \)
\( B \rightarrow bB|bb \)
\( C \rightarrow cC|cc \)

For given L, there exist unambiguous CFG, So L is called as Inherently unambiguous language.

12. Ans: (d)
Sol: i) \( \{a^p \mid p \text{ is prime}\} \) is not regular
ii) \( \{a^p \mid p \text{ is not prime}\} \) is not regular
iii) \( \{a^n \mid n \geq 1\} \) is not regular
iv) \( \{a^n \mid n \geq 0\} \) is not regular
If language over 1 symbol is not regular then it is also not CFL. So all are not CLFs.

13. Ans: (c) & (d)
Sol: i) \( \{w \mid w \in (a+b)^*\} = (a+b)^* \) is regular
ii) \( \{www \mid w \in (a+b)^*\} \) is not CFL
iii) \( \{www \mid w \in (a+b)^*\} \) is not CFL
iv) \( \{ww^R \mid w \in (a+b)^*\} \) is not CFL
Only (i) is regular and remaining are not regular.
So, only (i) is CFL and remaining are not CFLs.

14. Ans: (c)
Sol: Decidable problems about CFLs:
   i) Emptiness
   ii) Infiniteness
   iii) Membership

15. Ans: (b)
Sol: Finiteness, Infiniteness, Membership are decidable for CFLs.
16. Ans: (c)
Sol: DCFLs are closed under:
   i) Complement
   ii) Inverse homomorphism
   iii) Intersection with regular set

17. Ans: (a)
Sol: DCFLs can be described by LR(k) grammars.

18. Ans: (a)
Sol: L = {N | N = 5^k+1, N \in (0+1)^*} is not regular and also not CFL.

19. Ans: (a)
Sol: L = \{N | N=5^k, N \in (0+1+2+3+4)^*\}
    = 0^*10^*
   L is regular, so CFL.

20. Ans: (d)
Sol: In CNF, if length of string is n then derivation length is always 2^n–1.
   If Derivation length is k then string length is (k+1)/2

21. Ans: (a)
Sol: Top down parsing can use PDA.
   GNF CFG can be converted to PDA. Such PDA derives a string using LMD.

22. Ans: (a)
Sol: If PDA simulated by GNF CFG then the derivation of a string uses LMD.

23. Ans: (b)
Sol: i) L = \{w | w \in (a+b)^*, n_a(w) is divisible by 3 and n_b(w) is divisible by 5\} is regular
   ii) L = \{w \in (a+b)^*, n_a(w) = n_b(w)\} is not regular but CFL
   iii) L = \{w \in (a+b)^*, n_a(w) = n_b(w), n_a(w)+n_b(w) is divisible by 3\} is not regular but CFL
   iv) L = \{w \in (a+b)^*, n_a(w) = n_b(w)\} is not regular but CFLs.
   So, (i) is regular and remaining are CFLs.

24. Ans: (c)
Sol: i) L = (a+b+c)^* is regular
   ii) L = \{w \in (a+b+c)^*, n_a(w) = n_b(w) or n_a(w) = n_c(w)\} is CFL.
   iii) L = \{w \in (a+b+c)^*, n_a(w) = n_b(w) + n_c(w)\} is CFL.
   iv) L = \{w \in (a=b+c)^*, n_a(w) = n_b(w), n_a(w) = 4n_c(w)\} is not CFL.

25. Ans: (a)
Sol: L = \{w \in (a+b+c+d)^*, n_a(w) = n_b(w) = n_c(w) = n_d(w)\}
   L is not CFL but \bar{L} is CFL
   L_1 = \{ww \in (a+b)^*\}
   L_1 is not CFL but \bar{L}_1 is CFL.
### 4. Recursive Enumerable Languages

(REG, TM, REL, CSG, LBA, CSL, Undecidability)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>01. Ans: (d)</td>
<td>Sol: Turing machine is equivalent to the following:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
11. Ans: (a)  
Sol: The class of an assembly programs is equivalent to class of all functions computed by turing machines.

12. Ans: (a)  
Sol: Set of regular languages and set of recursive languages are closed under intersection and complement.

13. Ans: (c)  
Sol:  
- Non-deterministic TM is equivalent to deterministic TM
- Non-deterministic halting TM is equivalent to deterministic halting TM.

14. Ans: (d)  
Sol: Universal TM is equivalent to TM.

15. Ans: (a)  
Sol: \( L = \{ \emptyset \} \)  
\( L \) is REL and \( \overline{L} \) is also REL.  
So, \( L \) is recursive language.

16. Ans: (a)  
Sol: Algorithms \( \cong \) Procedures \( \cong \) TMs

17. Ans: (a)  
Sol: Hyper computer is equivalent to TM.  
TM can accept non-regular.

18. Ans: (c)  
Sol: TM head restricted to input accepts CSL

19. Ans: (b)  
Sol: Type 0 grammar is equivalent to turing machine.

20. Ans: (c)  
Sol: Type 1 grammar is equivalent to linear bounded automata.

21. Ans: (a & d)  
Sol: \( L = \{ \text{wwwww} / w \in (a + b + c)^* \} \)  
\( L \) is CSL but not CFL  
So, \( L \) is also recursive language

22. Ans: (b) & (c)  
Sol: \( L = \{ a^n b^n c^n / n >1 \} \)  
\( L \) is CSL but not CFL  
So, \( L \) is also recursive language

23. Ans: (d)  
Sol: \( L = \{ \text{ww}^k / w \in (a + b)^* \} \)  
\( L \) is CFL but not regular

24. Ans: (d)  
Sol: \( (0 + 1 + \ldots + n + A + B + \ldots + F)^* 1 (0 + 1 + \ldots + 9 + A + B + C + D + E + F)^* \)  
It is regular language

25. Ans: (c)  
Sol: \( L = a^{47^*} \)  
\( L \) is CSL
26. Ans: (d)
Sol: Recursive languages are closed under union, intersection, complement, reversal and concatenation.
Recursive languages are not closed under substitution, homomorphism, quotient and subset.

27. Ans: (d)
Sol:
- Regular sets are closed under finite union, intersection, complement, homomorphism, inverse homomorphism and reversal.
- Containment, equivalence, emptiness, totality problems are decidable for regular sets.

28. Ans: (d)
Sol: The following problems are undecidable for CFL’s
1. Equivalence
2. Totality
3. Containment

29. Ans: (c)
Sol: The following problems are undecidable for CSL’s
1. Finiteness
2. Emptiness
3. Totality ($\Sigma^*$)
4. Equivalence
5. Containment

30. Ans: (d)
Sol: Undecidable problems for recursive sets:
1. Emptiness
2. Infiniteness
3. Regularity
4. Equivalence
5. Containment
Membership problem is decidable for recursive sets

31. Ans: (d)
Sol: Given TM accepts only 2 strings of length one $L = \{0, 1\}$

5. Theory of Complexity

01. Ans: (a)
Sol: $L = \{a^n b^n c^n | n \geq 1\}$ is CSL but P-Problem can be accepted by TM in $O(n^2)$ moves.
It is P-Problem.

02. Ans: (d)
Sol: (a) If $L$ is accepted by DTM in polynomial time then $L$ is P-Problem.
(b) If $L$ is accepted by NTM in polynomial time then $L$ is NP-Problem.
(c) If $L$ is verified by DTM in polynomial time then $L$ is NP-Problem.

03. Ans: (d)
Sol: $L = \{a^n b^n | n \geq 0\}$ is P-Problem
$L = \{a^n b^n c^n | n \geq 1\}$ is P-Problem
$L = \{www | w \in \Sigma^*\}$ is P-Problem
13. Ans: (d)  
Sol: Conversion from NFA to DFA takes \(O(2^n)\) time.

12. Ans: (d)  
Sol: Conversion from NFA to DFA takes \(O(2^n)\) time.

11. Ans: (c)  
Sol: All the problems take exp time.

10. Ans: (d)  
Sol: \(P \subseteq NP \subseteq PSPACE \subseteq EXP\)

9. Ans: (c)  
Sol: If multtape NTM decides a language \(L\) in \(t(n)\) time then single tape NTM requires \(t(n)^2\) time.

8. Ans: (c)  
Sol: If NTM takes \(t(n)\) time to decide any problem then DTM can take \(2^{O(t(n))}\) time to decide the same problem.

7. Ans: (b)  
Sol: P and NP class is closed under homomorphism.

6. Ans: (d)  
Sol: Complement of NP-Problem need not be NP-Problem.

5. Ans: (c)  
Sol: \(L = \{0^n1^n \mid n \geq 1\}\) takes \(O(n^2)\) time and \(O(n)\) space (i) & (ii) are correct.

4. Ans: (a)  
Sol: Regular language can be accepted by DTM in \(O(n)\) time.  
It can be take \(O(1)\) space to accept.

14. Ans: (b)  
Sol: SAT is NP-Problem  
1-SAT and 2-SAT are P-Problems (So NP-Problems)  
3-SAT and n-SAT are NP-Problems.

15. Ans: (b)  
Sol: \(NP = Co-NP\) iff \(L\) and \(\overline{L}\) are in NP.

16. Ans: (b)  
Sol: \(L\) is in NP iff \(L\) is polynomial time verifiable.  
\(L\) is in P iff \(L\) is decidable in polynomial time  
If \(L\) is in P then \(\overline{L}\) is in P  
If \(L\) is in NP then \(\overline{L}\) need not be in NP.

17. Ans: (c)  
Sol: \(L\) is in NPC iff \(L\) is in both NP and NP-Hard.

18. Ans: (a)  
Sol: If \(L \in P\) and \(P = NP\) then NPC = P.  
So, \(L \in NPC\).
19. **Ans: (d)**  
**Sol:**  
(i), (ii) & (iii) are true  
(i) NPH-Problem $\leq L_1$ $\Rightarrow$ $L_1$ is NPH-Problem  
(ii) If NPC $\leq L_1$ and $L_1$ is in NP $\Rightarrow$ $L_1$ is in NPC  
(iii) If $L$ is in NPC and $L \in P$ then $P = NP = NPC$

20. **Ans: (a) & (c)**  
**Sol:**  
(a) Integer Linear Programming is NPC problem  
(b) Primarily is NP-Problem  
(c) 3-CNF is NPC problem

21. **Ans: (b)**  
**Sol:** CYK algorithm is membership algorithm uses dynamic programming. It takes $O(n^3)$ time.