CLASS TEST

S.No.:01DLSK_CS_ABCDE_9726

MADE EASY

Leading Institute for IES, GATE & PSUs

Delhi | **Bhopal** | **Hyderabad** | **Jaipur** | **Pune**

Web: www.madeeasy.in | **E-mail:** info@madeeasy.in | **Ph:** 011-45124612

Theory of Computation

COMPUTER SCIENCE & IT

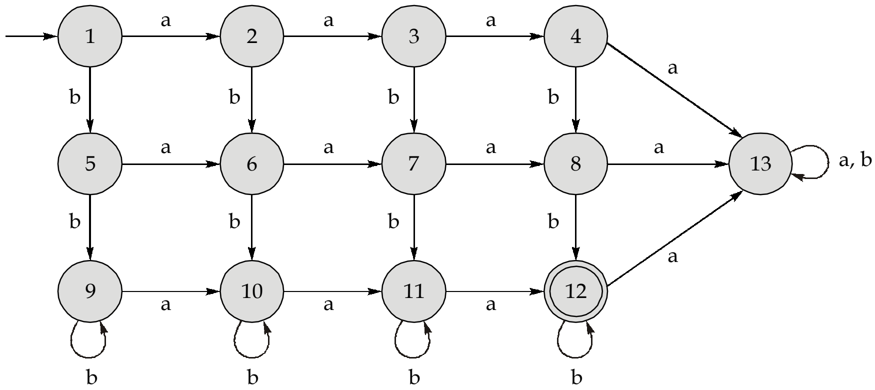
Date of Test : 09/07/2026

ANSWER KEY

1.	(b)	7.	(b)	13.	(a)	19.	(b)	25.	(d)
2.	(c)	8.	(c)	14.	(d)	20.	(d)	26.	(c)
3.	(b)	9.	(b)	15.	(b)	21.	(b)	27.	(a)
4.	(a)	10.	(b)	16.	(c)	22.	(d)	28.	(a)
5.	(b)	11.	(c)	17.	(c)	23.	(a)	29.	(c)
6.	(d)	12.	(c)	18.	(c)	24.	(b)	30.	(b)

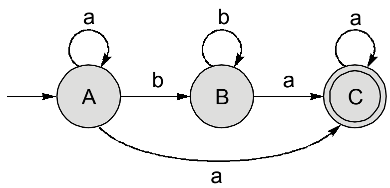
DETAILED EXPLANATIONS

1. (b)



Number of states = 13 states

2. (c)



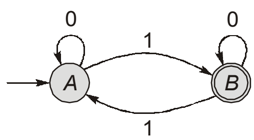
$$\begin{aligned} \text{R.E.} &= a^*(bb^*\,a + a)\,a^* \\ &= a^*((bb^* + \varepsilon)\,a)\,a^* \\ &= a^*\,b^*\,a\,a^* \\ &= a^*\,b^*\,a^*\,a \end{aligned}$$

3. (b)
B is Turing recognizable: Guess the 3 distinct inputs by non-deterministically for each TM and collect those TM's. A is complement of B, so A is not Turing recognizable.
Both A and B are undecidable languages, where A is non-REL and B is REL but not recursive.

4. (a)
When grammar is in CNF i.e., when the production are of the form $S \rightarrow AB$, $S \rightarrow a$.
Then, length of every derivation is $(2n - 1)$.
When grammar is in GNF i.e., when the production are of the form $S \rightarrow VT^*$. Then length of every derivation is n.
Hence, option (a) is correct.

5. (b)

FA for given regular expression



6. (d)
You can get only 'b' from both.

7. (b)

The above language represents L^2.

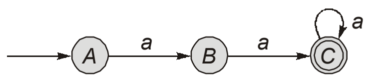
8. (c)

$$L = \{a^{m^n} \mid n \geq 1, m > n\}$$

$\Rightarrow$ $$L = \{a^{m^1} \mid m \geq 2\} \cup \{a^{m^2} \mid m \geq 3\} \cup .....$$

$\Rightarrow$ $$L = \{a^i \mid i \geq 2\} \text{ is a regular language}$$

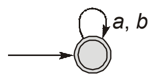
Its DFA will be



Number of states = 3.

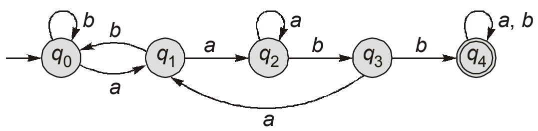
9. (b)

$$R = (a^* (a^* + b^* + ab^* + ba^*)^+)$$
$$= a^* (a + b)^*$$
$$= (a + b)^*$$



$\therefore$ Number of states in minimal DFA is 1.

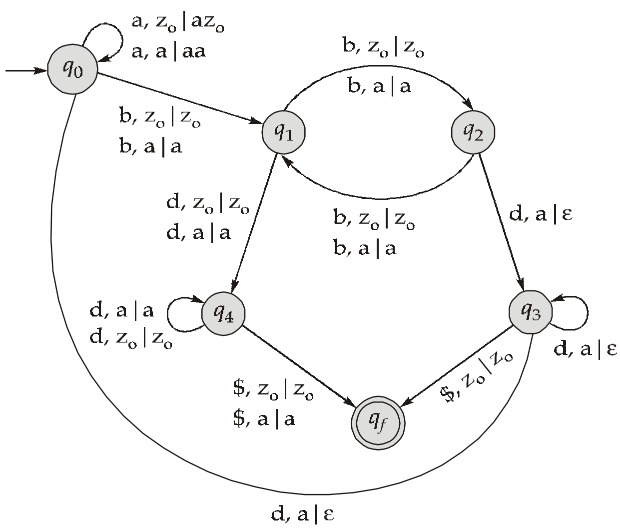
10. (b)



5 states are required in minimal DFA.

11. (c)

$$L = \{a^m b^n b^k d^l \mid \text{if } (n = k \text{ then } m = l\}$$
$$= \{a^m b^{2n} b^m\} \cup \{a^m b^{2n+1} b^k\}$$
$$= \text{DCFL} \cup \text{regular} = \text{DCFL}$$



12. (c)

$$L = \{a^m b^n b^k d^l \mid (n+k = \text{odd}) \text{ only if } m = l\}$$

If, we check the condition carefully, the condition is actually logical implication.

$$L = \{a^m b^n b^k d^l \mid (n+k = \text{odd}) \rightarrow m = l\}$$

Either $n+k$ will be odd or it will be even, if $(n+k)$ is odd, then it's necessary that m should be equal to l, if $(n+k)$ is even then l can be any number.

$$L = \{a^m b^{2n+1} d^l \text{ and } l = m\} \text{ or } \{a^m b^{2n} d^l\}$$

$$= \{a^m b^{2n+1} d^m\} \cup \{a^m b^{2n} d^l\}$$

$\Rightarrow$ DCFL $\cup$ regular = DCFL.

13. (a)

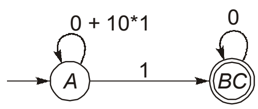
The language $\{0^p 1^{2p}\}$ is context-free language, hence it is recursive also. Since $L(M) \leq_p$ REC, so $L(M)$ also recursive, now given input (i.e. recursive language) to turing machine and finding it is accept or not is non-trival property so it is undecidable by Rice's theorem.

14. (d)

By pumping lemma, we can never say that a language is regular or CFL. It can only be used to prove that a certain language is not regular or not CFL.

Since pumping lemma isn't satisfied for regular, hence we can say it is not regular, but since the lemma is satisfied for context-free, we can't say that the language is CFL.

15. (b)



$= (0 + 10^*1)^*\ 10^*$

16. (c)

$$L_1/L_2 = bba^*baa^* / ab^*$$

$$= bba^*baa^* / a$$

$$= bba^*baa^*$$

17. (c)

$$L_1.L_2 = \text{(Regular)} . \text{(CSL)}$$

L_2 is $a^n b^n c^n$, but L_1 can be any regular language

Case 1: If $L_1 = \phi$,

$\Rightarrow$ $L_1.L_2 = \phi . \{a^n b^n c^n\} = \phi$ is regular

Case 2: If $L_1 = \{\varepsilon\}$

$\Rightarrow$ $L_1.L_2 = \{\varepsilon\} . \{a^n b^n c^n\} = \{a^n b^n c^n\}$ is CSL

$L_1.L_2$ is always CSL but it may or may not be regular.

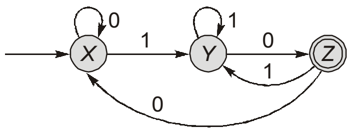
18. (c)

$$((11))^* (11111)^*)^* = (11 + 11111)^*$$

Which is the language corresponding to given grammar.

19. (b)
B1000B
1R000B
10R00B
100R0B
1000RB
1000LB
100L0B
10L00B
1L000B
L1000B

20. (d)



Minimum string accepted by DFA is "10", which is generated by only option (d).

21. (b)

S_1 : Pumping lemma can prove that language is not regular but can't prove that the language is regular. Hence this is false.
S_2 : We can check regular grammar by following productions $V \to T^* \ V + T$ or $V \to V\ T^* + T$.
S_3 : Consider 'L' to be ϕ and 'M' to $\{a^n b^n \mid n \le 0\}$
L.M. = ϕ, which is regular

22. (d)

$S \to AB,$	$S \to \{a^n b^n c^{2m} \mid n > 0,\ m \ge 0\}$
$A \to aAb \mid ab,$	$A \to \{a^n b^n \mid n > 0\}$
$B \to ccB \mid \in,$	$B \to \{c^{2m} \mid m \ge 0\}$

23. (a)
- Finiteness property of a CFG is decidable, which can be decidable with the help of variable dependency graph.
- Push-down automata need not be always deterministic. In fact power of non-deterministic PDA is greater than the deterministic.
- Deterministic CFL are closed under complement, hence recursive too.
- DCFL is not closed under union.

24. (b)
$\in \to$ belongs
$a \to$ does not belongs . So one length string not belongs to the given RE.

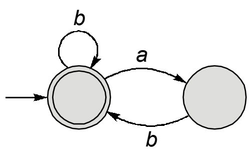
25. (d)
The language accepted by the PDA with finite stack is always regular language. Regular language may be finite or infinite.
$\therefore$ Option (d) is correct.

26. (c)
$\bar{L}$ has every even length string and it contain all odd length strings which are not in the form of $w x w^R$. [It can be implemented by selecting non-deterministic mismatch symbols of w and w^R]
$\bar{L}$ is CFL but not DCFL.

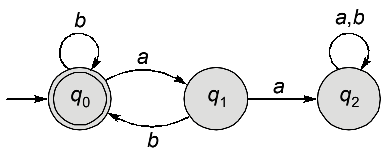
27. (a)
In given PDA at q_1 state 0 is push after that every 0 and 1 ignore (skip) at state q_1. At q_1 for last 1, 0 is popped and reach to q_2 state. At last from q_2 state we go for final state. So the language accepted by given PDA is $0(0+1)^+ 1$ which is regular but infinite.

28. (a)
$$L_3 = L(G_1) \cap L(G_2)$$
$L(G_1)$ and $L(G_2)$ are CFL, infact L_2 is regular language having regular language $0(10)^*$.
$L(G_1)$ is language contain equal number of 0 and 1.
So,
$$\begin{aligned} L_3 &= L(G_1) \cap L(G_2) \\ &= \text{CFL} \cap \text{Reg} \\ &= \text{CFL} \\ L_4 &= L(G_1) \cdot L(G_2) \\ &= \text{CFL} \cdot (\text{Reg})^* \\ &= \text{CFL} \end{aligned}$$
So, both L_3 and L_4 are CFL.

29. (c)



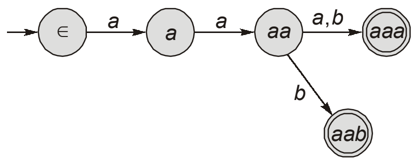
Note is that the NFA has no choice, only dead configuration and hence can be easily converted DFA (put dead configuration trap state).
Minimized DFA for the language R:



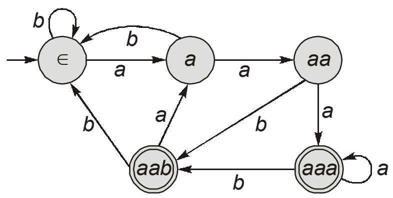
$$\begin{aligned} [q_0] &= (ab + b)^* \\ [q_1] &= (ab + b)^* a \\ [q_2] &= (ab + b)^* aa(a + b)^* \end{aligned}$$
$\therefore$ Three equivalence classes are present.

30. (b)

The R.E. is $(a+b)^*\ (aaa + aab)$ which is nothing but set of string ending with *aaa* or *aab*. For which we can directly design minmal DFA as follow:



The fill missing arrows as follow:



■■■■