CLASS TEST

S.No. : 01 PT_CS_ABC_060822

MADE EASY

India's Best Institute for IES, GATE & PSUs

Delhi | Bhopal | Hyderabad | Jaipur | Lucknow | Pune | Bhubaneswar | Kolkata | Patna

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

THEORY OF COMPUTATION

COMPUTER SCIENCE & IT

Date of Test : 28/08/2022

ANSWER KEY ➤

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

1. **(d)**

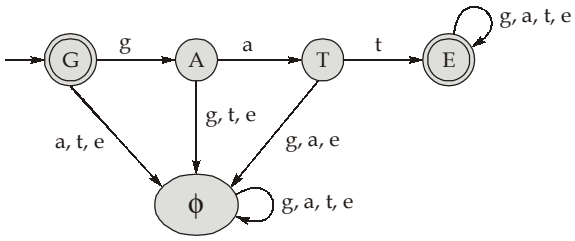
2. **(d)**
You can get only 'b' from both.

3. **(a)**
Complement in CFL not closed but in DCFL it is closed.
L_1 is DCFL L_2 is also DCFL.

4. **(a)**

5. **(c)**
$$((11))^* (11111)^*)^* = (11 + 11111)^*$$
Which is the language corresponding to given grammar.

6. **(b)**
The given NFA accepts a language where each string starts with 'gat' [including Null string]
$\therefore$ Number of states required in DFA = 4 + 1 = 5 states

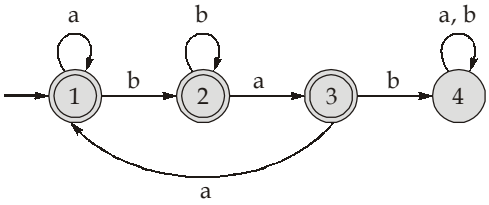


7. **(a)**
Simulate M on all strings of length atmost n for n steps and keep increasing n. We accept if the computation of M accepts some string.

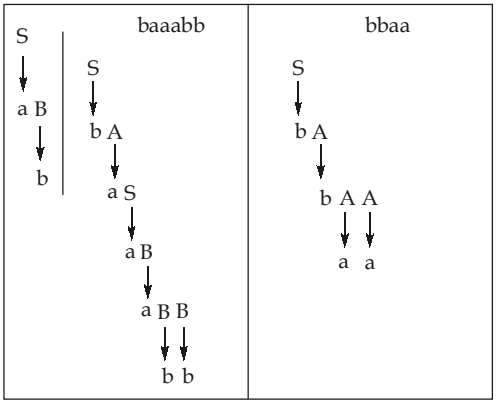
8. **(a)**
Each rule A $\rightarrow$ BC increases the length of the string by 1, which gives $(n - 1)$ steps and exactly n rules A $\rightarrow$ a to convert variables into terminals.
Therefore exactly $2n - 1$ steps are required for CNF CFG.
So option (a) is correct.

9. **(c)**
$$L_1 = \phi \rightarrow L_1^* = \{\varepsilon\} \text{ is finite}$$
$$L_1 = \{a\} \rightarrow L_1^* = \{a^*\} \text{ is infinite}$$
$\therefore$ L_2 need not be infinite

10. **(b)**



11. (c)



12. (b)

13. (c)

Clearly L_1, L_2 are DCFL's and hence CFL's.

$$L_1 \cap L_2 = \{a^i b^j c^k \mid i < j \text{ and } i < k\}$$

is not a CFL, since 2 comparisons must be made before acceptance and this is not possible using a single stack.

So, choice (c) is correct.

14. (b)

B1000B
1R000B
10R00B
100R0B
1000RB
1000LB
100L0B
10L00B
1L000B
L1000B

15. (d)

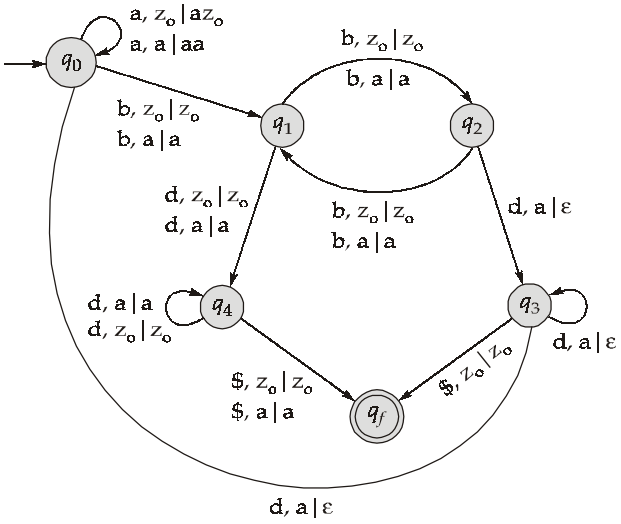
$S \rightarrow aSa \mid aAa$

$A \rightarrow bA \mid b$

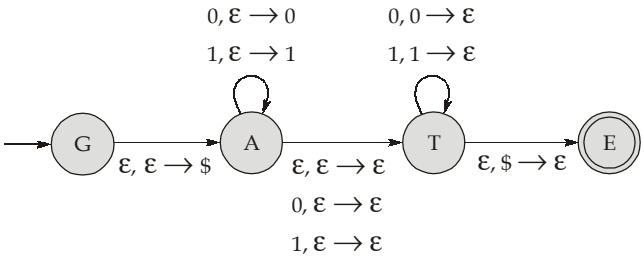
$$L(A) = b^+$$
$$L(S) = a^n(ab^+a)a^n,\ n \geq 0$$
$$= a^{n+1} b^+ a^{n+1}$$
$$= a^m b^+ a^m \mid m > 0$$
$$= \{a^m b^n a^k \mid m = k,\ m, n, k > 0\}$$

16. (c)

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



17. (c)



G → A : Pushes "$" onto stack initially.
A → A : Pushes 0 for input 0 and Pushes 1 for input 1.
A → T : Moves A to T without reading an input (or)
R read 0 or 1 from input tape and does no operation on the stack.
T → T : Pop 0 for input 0 and Pop 1 for input 1.
T → E : Pop "$" from stack and reaches to final state [input string has completed reading]
$\therefore$ $L = \{ \varepsilon, 0, 1, 00, 11, 000, 010, 101, 111, ...\}$
G is : $S \to 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon$
So option (c) is correct.

18. (d)

(a) L is not recursive [and not REL], TM accepts a regular language is undecidable.
(b) L is not recursive [and not REL], TM accepts a regular language is undecidable.
(c) L is not recursive language [But REL], State entry problem is undecidable.
(d) L is recursive language
So option (d) is correct.

19. (c)

$$R = (a + \varepsilon)\,(bb^*a)^*$$

R generates the language that do not contain two or more consecutive a's and do not end with b.

20. (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.

21. (c)

$$\left.\begin{array}{l} S \to AAaSb \mid \varepsilon \\ A \to a \mid \varepsilon \end{array}\right\} \cong S \to aSb \mid aaSb \mid aaaSb \mid \varepsilon$$

$$L(G) = \{a^m b^n \mid n \le m \le 3n\}$$

22. (b)

$$\begin{aligned} L_2 - L_1 &= L_2 \cap \overline{L_1} \\ &= \text{REL} \cap \overline{\text{RECURSIVE}} \\ &= \text{REL} \cap \text{RECURSIVE} \\ &= \text{REL} \end{aligned}$$

$\therefore$ $L_2 - L_1$ is Recursive Enumerable Language (REL).

23. (c)

$\overline{L}$ 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]

$\overline{L}$ is CFL but not DCFL.

24. (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.

25. (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 $L_1 = \phi$,

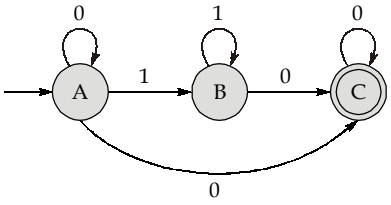
$\Rightarrow$ $L_1.L_2 = \phi.\ \{a^n\,b^n\,c^n\} = \phi$ is regular

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.

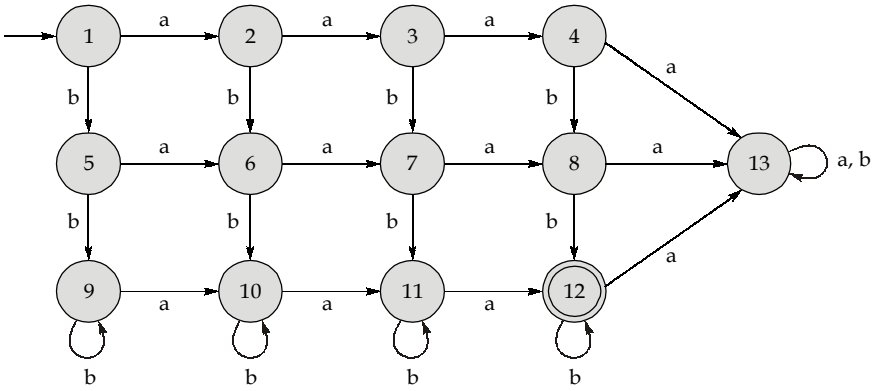
26. (c)



R.E. = 0*(11* 0 + 0) 0*
= 0*((11* + ε) 0) 0*
= 0* 1* 0 0*
= 0* 1* 0* 0

So, option (c) is correct.

27. (b)



Number of states = 13 states

28. (b)

(a) Regular language: 1 [(0 + 1) (0 + 1)]*

(b) Non regular language (finding middle symbol is not possible)

(c) Regular language: [(0 + 1) (0 + 1)]* 1

29. (d)

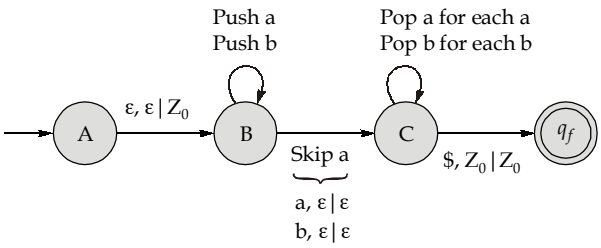
All given languages are DCFL.

(a) $\{w \mid \#_0(w)! = \#_1(w), w \in (0+1)^*\}$ is DCFL.

(b) $\{xwx \mid x \in (0+1),\ w \in (0+1)^*, \#_0(w) = \#_1(w)\}$ is DCFL.

(c) If string starts with 1 then it accepts 0^n1^n as next symbols of the string. If string starts with 11 then it accepts 0^K1^{2K} as next symbols of the string, which is also DCFL.

30. (c)



$$L = \left\{ \underbrace{w}_{\text{push}} \underbrace{x}_{\text{skip}} \underbrace{w^R}_{\text{pop}} \mid w \in (a+b)^*\ x \in (a+b) \right\}$$

∴ Option (c) is correct.

■■■■