CLASS TEST

S.No. : 01 **PT_CS_A+C_310819**

Theory of Computation

MADE EASY

India's Best Institute for IES, GATE & PSUs

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

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

CLASS TEST 2019-2020

COMPUTER SCIENCE & IT

Date of Test : 31/08/2019

ANSWER KEY ➤ Theory of Computation

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

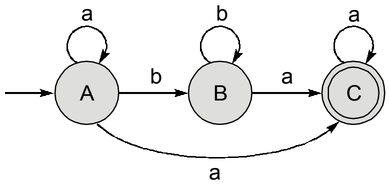
DETAILED EXPLANATIONS

1. (a)

L_2 is even palindrome on $\{a, b\}^*$
L_3 is odd palindrome on $\{a, b\}^*$
L_1 is any palindrome on $\{a, b\}^*$

Clearly, $L_2 \subset L_1$ and $L_3 \subset L_1$ and $L_1 = L_2 \cup L_3$

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. (c)

Traversing the states of the turing machine, it can be seen that for every 'a' as the input, it is accepting 3 b's. For every 'a' machine writes 'X' on the tape, then take right moves till it reaches 'b'. For every 3 b's it writes symbol Y.

Hence accepting the language $L = \{a^m b^n \mid 3m = n;\ m, n \geq 0\}$.

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. (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.

6. (c)

(a) $$L_1 = \{a^n b^n \mid m > n \text{ and } n > m\} = \phi$$
which is regular.

(b) $$\begin{aligned} L_1 \cup L_2 &= \{a^m b^n \mid m > n \text{ or } m < n\} \\ &= \{a^m b^n \mid m \neq n\} \text{ which is CFL.} \end{aligned}$$

(c) $$\begin{aligned} L_1 \cup L_2 &= \{a^m b^n \mid m < n \text{ or } m > n\} \\ L_1 \cup L_2 &= \{a^m b^n \mid m \neq n\} \\ (L_1 \cup L_2)^C &= \{a^m b^n \mid m = n\} \cup (a + b)^* \,ba\, (a + b)^* \end{aligned}$$

(d) $L_1 \cup L_2 = \{a^m\, b^n \mid m < n \text{ or } m > n\}$

$= \{a^m\, b^n \mid m \neq n\}$ which is DCFL and hence unambiguous language

7. (b)

The above language represents L^2.

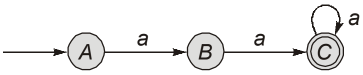
8. (c)

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

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

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

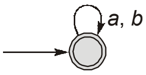
Its DFA will be



Number of states = 3.

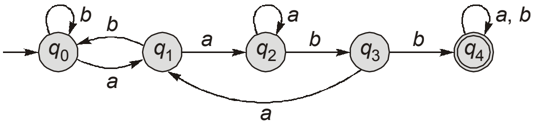
9. (b)

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



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

10. (b)



5 states are required in minimal DFA.

11. (a)

- L_1 is regular , since we can create DFA for given language.
- L_2 is CFL, since their is a comparison between number of a and number of b in strings i.e., difference is less than equal to 10.
- L_3 is regular, since by making $w = \epsilon$ and $c = (a{+}b)^*$ we get language $(a{+}b)^*$ which contain every string belongs to $wcww^r$.

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)

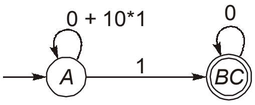
The language $\{0^p 1^{2p}\}$ is context-free language, hence it is recursive also. Since $L(M) \leq_p$ REC, so $L(M)$ also recursive, now given input (i.e. recursive language) to turing machine and finding it is accept or not is non-trival property so it is undecidable by Rice's theorem.

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)

$$\begin{aligned} L_1/L_2 &= bba^*baa^* \,/\, ab^* \\ &= bba^*baa^* \,/\, a \\ &= bba^*baa^* \end{aligned}$$

17. (d)

The given grammar G:

$$\begin{aligned} &E \rightarrow 0XE2 \\ &E \rightarrow 0X2 \\ &X0 \rightarrow 0X \\ &X2 \rightarrow 12 \\ &X1 \rightarrow 11 \end{aligned}$$

$L(G) = \{012, 001122, \dots\}$

$E \rightarrow 0X2 \Rightarrow 012$

$E \rightarrow 0XE2 \Rightarrow 0X0X22 \Rightarrow 0X0122 \Rightarrow 00X122 \Rightarrow 001122$

$\therefore\ L = \{0^n 1^n 2^n \mid n > 0\}$

18. (d)

$$(R_1) = (a^*ba^*ba^*ba^*)^*$$

It represents language that contain strings in which number of b's is multiple of 3 with any number of a.

$$(R_2) = (a^*ba^*ba^*)^*$$

It represents language that contain strings in which number of 'b' are in multiple of 2 with any number of a.

So, $(R_1) \cap (R_2) = (a^*ba^*ba^*ba^*ba^*ba^*ba^*)^*$

Represent string that contain number of b's in multiple of 6 with any number of a's.

19. (c)

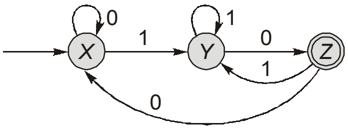
Considering the grammar :

$S \rightarrow SSS \mid a \mid ab$

The minimum string that can be obtain from grammar 'S' is string 'a' and then string 'ab'. On applying the production, $S \rightarrow SSS$, which represents $L \cup L^3 \cup L^5$ ....

Hence, the regular expression will be, $(a + ab)\,((a + ab)\,(a + ab))^*$.

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 $V \rightarrow T^* V + T$ or $V \rightarrow V T^* + T$.

S_3 : Consider 'L' to be ϕ and 'M' to $\{a^n b^n \mid n \leq 0\}$

L.M. = ϕ, which is regular

22. (d)

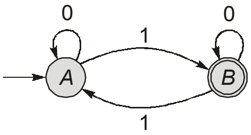
$S \rightarrow AB$, $\quad S \rightarrow \{a^n b^n c^{2m} \mid n > 0, m \geq 0\}$

$A \rightarrow aAb \mid ab$, $\quad A \rightarrow \{a^n b^n \mid n > 0\}$

$B \rightarrow ccB \mid \in$, $\quad B \rightarrow \{c^{2m} \mid m \geq 0\}$

23. (b)

FA for given regular expression



24. (b)

$\in \rightarrow$ belongs

$a \rightarrow$ does not belongs . So one length string not belongs to the given RE.

25. (b)

(a) $\{(ab)^* (cb^n)^* \mid n \geq 1\}$ is regular. R.E. = $(a\, b)^* (cbb^*)^*$

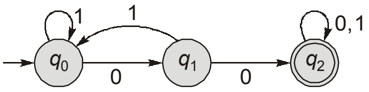
(b) $\{a^n b^m b^n \mid m, n \geq 0\}$ is not regular, since one comparison present between 'a' and 'b'.

(c) $\{(a^n b)^* (cb)^* \mid n \geq 1\}$ is regular. R.E. = $(aa^* b)^* (cbb^*)^*$

(d) $\{(ab)^* (cb)^*\}$ is regular, since there is no bound on any variable.

26. (a)

- Complement of $L(D)$ is:



Which represent all strings in consecutive 0's, so given DFA is no consecutive 0's.

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 $0(10)^*$.

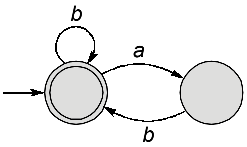
$L(G_1)$ is language contain equal number of 0 and 1.

So,

$$\begin{aligned} L_3 &= L(G_1) \cap L(G_2) \\ &= \text{CFL} \cap \text{Reg} \\ &= \text{CFL} \\ L_4 &= L(G_1) \cdot L(G_2) \\ &= \text{CFL} \cdot (\text{Reg})^* \\ &= \text{CFL} \end{aligned}$$

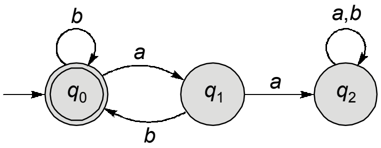
So, both L_3 and L_4 are CFL.

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:

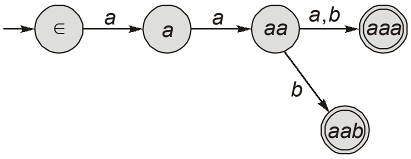


$$\begin{aligned} [q_0] &= (ab + b)^* \\ [q_1] &= (ab + b)^* a \\ [q_2] &= (ab + b)^* \, aa(a + b)^* \end{aligned}$$

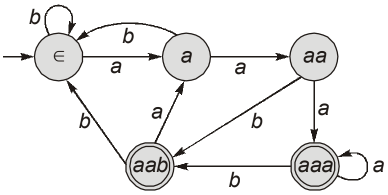
∴ Three equivalence classes are present.

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:



■■■■