CLASS TEST

S.No. : 01 SK1_CS_W+Y_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

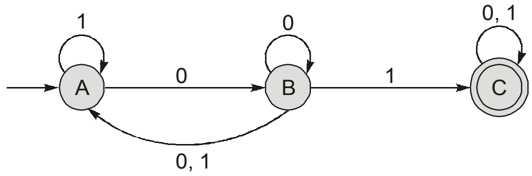
ANSWER KEY ➤ Theory of Computation

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

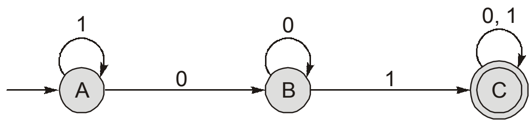
DETAILED EXPLANATIONS

1. (c)

To reach a final state (c) it must cover minimum "01" as a substring.

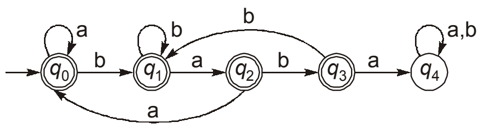


1*00*1(0 + 1)* will cover all strings containing '01'. The cycle between A and B will not affect. Equivalent modified NFA will be same as given NFA.



2. (c)

Complement of above DFA is generate after changing the final state to non-final state and non-final state to final state, which is given below



This DFA is representing the strings which do not contain “baba” as substring.

3. (c)

Languages accepted by push down automata are closed under complementation.
Turing decidable languages are closed under union and Kleen star operation.
Recursive enumerable languages are not closed under complementation.

4. (b)

(a) Regular expression = 1*(1 + 0)* can generate string 101.
(b) Regular expression = 0^*100^+ generate all string which does not contain 101 as substring.
(c) Regular expression = 10*10 contain 1010 as string which can contain 101 as substring.
(d) Regular expression = 1*(01+0)* contain 101 as string which can contain 101 as substring.

5. (a)

L_1 is DCFL $\Rightarrow$ L_2 is also DCFL
$\therefore$ L_1 and L_2 are DCFL but not regular.

6. (b)

Statement S1 and S3 are True.
S2 is false because all $\in$-production can be removed from grammar only when the language do not contain $\in$-string but if language contain $\in$-string then removal of the null productions is not possible.

7. (c)

(a) $S \to AB$
$A \to aA \mid \varepsilon$
$B \to bB \mid \varepsilon$
$\Rightarrow L = \{a^*b^*\}$ is regular.

(b) $S \rightarrow AaB$

$A \rightarrow aA \mid B \mid \varepsilon$

$B \rightarrow bB \mid \varepsilon$

$\Rightarrow L = \{a^*b^*ab^*\}$ is regular.

(c) $S \rightarrow aA \mid \varepsilon$

$A \rightarrow Sb \mid Ab$

$\Rightarrow L = \{a^m b^n \mid m <= n\}$ is non-regular.

8. (b)

$$L_2 = \{a^n b^n\},\ L_1 = \{a^*b^*\}$$
$$L = (a^*b^*) \cap ((a+b)^* - \{a^n b^n\}) = \{a^m b^n \mid m! = n\}$$

9. (a)

$L \leq_P L'$. Since L' is semidecidable then L is semidecidable is one way theorem (semidecidability goes backward).

10. (a)

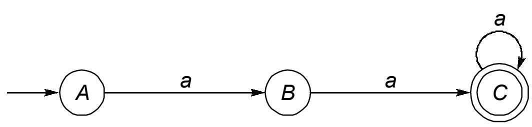
If turing machine has no writing capability on tape then turn around capability of head is not useful. So it accepts only regular language.

11. (c)

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

$$\Rightarrow \quad L = \left\{a^{m^1} \mid m \geq 2\right\} \cup \left\{a^{m^2} \mid m \geq 3\right\} \cup \ldots$$

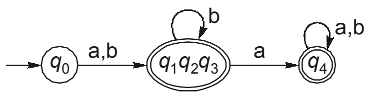
$$\Rightarrow \quad L = \left\{a^i \mid i \geq 2\right\} \text{ is a regular language.}$$



Number of states = 3.

12. (b)

The minimized DFA after combining the q_1, q_2 and q_3 are given below.



13. (c)

$L_1 = \{a^i b^j c^k \mid i = j,\ j < k\}$ is not CFL

$L_2 = \{a^i b^j c^k \mid (i \leq j) \text{ or } (j \leq i),\ j = k\}$

$L_3 = \{a^m b^n c^n d^m \mid m \neq n\}$ is not CFL

$L_4 = \{a^i b^j c^k \mid \text{ if } (i = j) \text{ then k is even}\}$ is CFL

$\therefore$ L_2 and L_4 are CFLs.

$\because$ 2 comparisons occuring
only, comparison
cannot be done using PDA
only, comparison

14. (d)

Strings of one length = 0 $\rightarrow$ Not possible

Strings of two length = 0 $\rightarrow$ Not possible

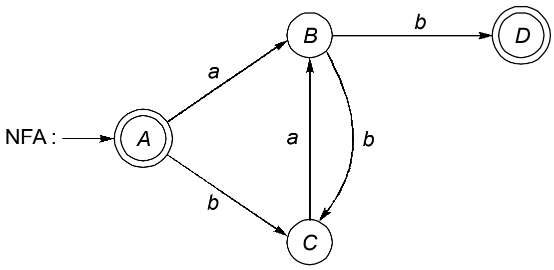
Strings of three length = 0

Strings of four length = abba ...(1 string)

Strings of five length = abbba, abbaa, aabba ...(3 strings)

Total 4 strings possible.

15. (b)



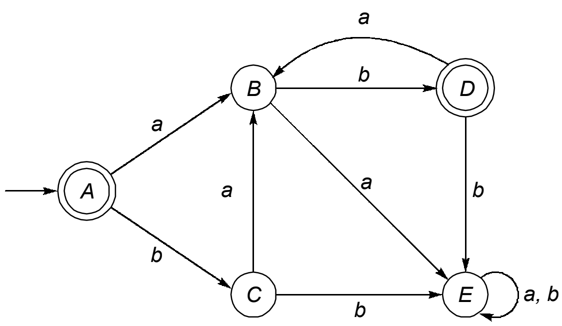
δ	a	b
→A	{B}	{C}
B	{ }	{C,D}
C	{B}	{ }
D	{ }	{ }

Convert NFA into DFA as following

DFA: δ'	a	b
→{A}	{B}	{C}
{B}	{ }	{C,D}
{C}	{B}	{ }
{C,D}	{B}	{ }
{ }	{ }	{ }

After renaming the above states equivalent DFA is:

DFA: δ'	a	b
→A	B	C
B	E	D
C	B	E
D	B	E
F	F	E



∴ Option (b) is correct.

16. (d)

$$q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{0} q_0$$

String 0001100 accepted by FA.

17. (a)

$$Y = A \cup L_1 \cup L_2 \cup \ldots L_n \cup B$$
$$Y = \Sigma^* \cup L_1 \cup L_2 \cup \ldots L_n \cup \phi$$
$$Y = \Sigma^* = (a+b)^*$$

∴ Y is regular and infinite language.

18. (c)

Statement S1 is true.

Example: $\{a^* b^n c^n d^* \mid n \le 2\}$, here comparison between 'b' and 'c' is finite but the language is infinite. So it is regular language.

The statement S2 is true.

Given regular expression is infinite set (because of *) of finite strings. A regular expression can generate any infinite length string (since string length always be finite but language can be infinite).

19. (c)

S1: Pumping lemma is always used in negative sense to prove language is not regular.

S2: This is a relationship between DCFL and LR(k).

DCFL $\leftrightarrow$ LR(k)

but for DCFL and LL(k)

LL(k) $\rightarrow$ DCFL

S2 also correct.

20. (a)

$$L = (bc)^*a + b$$
$$f(a) = \varepsilon,\ f(b) = \phi.(a+b)^* = \phi,\ f(c) = \phi.a^* = \phi$$
$$f(L) = (f(b).f(c))^*\ f(a) + f(b) = (\phi.\ \phi)^*\ \varepsilon + \phi = \phi^*.\varepsilon = \varepsilon.\varepsilon = \varepsilon$$

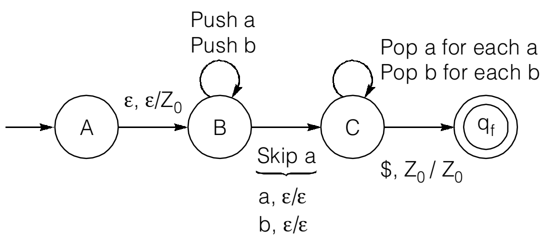
21. (b)

The grammar generates the following language.

$$L(G) = \{a^n b^n c^m \mid n, m \geq 0\} \cup \{a^n b^m c^m \mid n,m \geq 0\}$$

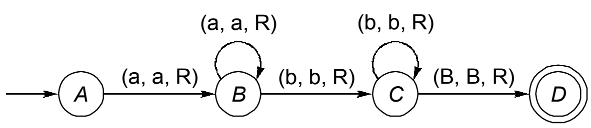
Which is a standard example of inherently ambiguous language i.e. no grammar is possible which is unambiguous. Any string in this language is of the form of $a^n b^n c^n$ and will have 2 derivation trees. Whatever be the grammar, one of these derivation trees will start with S $\rightarrow$ XC and another will start with S $\rightarrow$ AY.

22. (c)



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

23. (b)



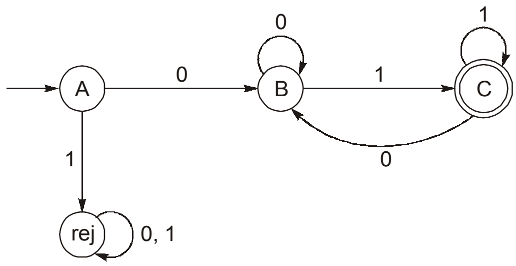
TM accepts $L = \{a^+ b^+\} = \{a^m b^n \mid m, n \geq 1\}$

24. (a)

The intersection of L_1 and L_2 is given by $L_1 \cap L_2 = \{0^n 1^n 2^n \mid n > 0\}$ which is well known CSL.

25. (b)

$$L = \{0w1 \mid w \in (0+1)^*\} = 0(0+1)^*1$$



4 states in minimized DFA. Hence four equivalence classes for L.

26. (b)

Assume that there are two palindrome strings one of them is even palindrome and one is odd palindrome i.e. abba and ababa respectively. The grammar will not generate both the strings. The even palindrome contain equal number of 'a' and 'b' here. So option 'a' and 'c' are not correct.

The grammar will generate all strings over 'a' and 'b' that are not palindrome.

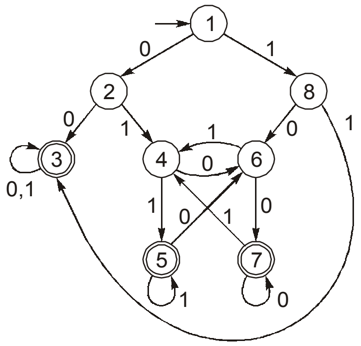
Therefore (b) is true since it will generate all strings which are not palindrome.

27. (c)

This language contains all the strings that either begin or end with 00 or with 11. So the set of strings that will be accepted by DFA are 11010, 001, 1011, 10100,...

The final state will be the state in which we land when the beginning of strings either 00 or 11 and states in which we land when the end of DFA is either 00 or 11.

DFA of the language is given below.



28. (b)

$$RE = \underbrace{(aa+abb)^+}_{aa}\ \underbrace{(a+b+ba)^+}_{a \text{ or } b}\ \underbrace{(a+b)^+}_{a \text{ or } b}$$

$$\left.\begin{array}{l} aaaa \\ aaab \\ aaba \\ aabb \end{array}\right\} \text{are minimal strings for RE}$$

$\therefore$ 4 strings are possible.

29. (d)

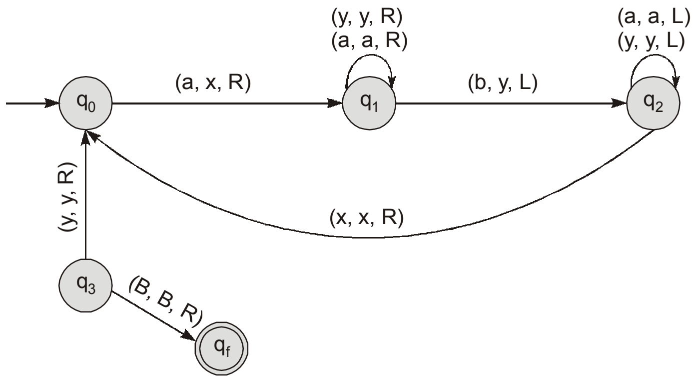
$L_1 = a^*b^* \Rightarrow L_1^* = (a^*\,b^*)^* = (a + b)^*$

$L_2 = \{ab\}$

$L_1^* \cap L_2 = (a + b)^* \cap \{ab\} = \{ab\}$

$L_3 = \text{Prefix}\,(L_1^* \cap L_2) = \{\in, a, ab\}$

30. (c)



■■■■