CLASS TEST

S.No.:01SKCS_ABCDE_20052024

MADE EASY
India's Best Institute for IES, GATE & PSUs

Delhi | Bhopal | Hyderabad | Jaipur | Pune | Kolkata

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

Theory of Computation

COMPUTER SCIENCE & IT

Date of Test : 23/10/2023

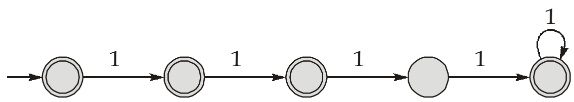
ANSWER KEY

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

DETAILED EXPLANATIONS

1. (d)

$$L = \{1^m \mid m \geq 0 \text{ and } m \neq 3\}$$



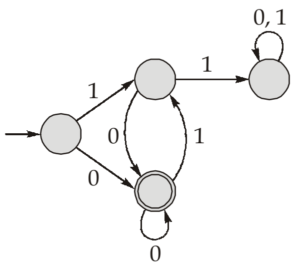
Therefore, total 4 final states and 5 states.

2. (d)

$$L = \{x \in \{0, 1\}^* \mid x \text{ ends with 0 and not contains 2 consecutive 1's}\}$$

$$\text{R.E.} = (0 + 10)^* (0 + 10)$$

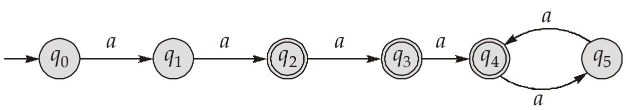
$$= (0 + 10)^+$$



So, option (d) is correct.

3. (d)

$$L = \{a^n : n = 2, 3, 4, 6, 8, 10 \text{ ......}\}$$



DFA

Total 6 states are required.

4. (b)

$$\text{Prefix } (L) = \{\in, b, ba, bab, baba\}$$

$$\text{Suffix } (L) = \{\in, a, ba, aba, baba\}$$

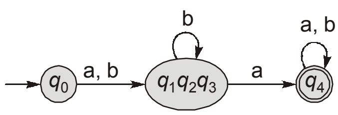
$$A = \{\in, b, ba, bab, baba\} \cap \{\in, a, ba, aba, baba\}$$

$$A = \{\in, ba, baba\}$$

There are 3 strings present in language A.

5. (b)

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



6. (b)

Check the string one by one starting from $\in$, 0, 1, 00, 01,... until we reach the first string that is not generated by the given regular expression. In this case, smallest string not generated by the given regular expression is '0110' whose length is 4.

7. (a)

- S_1 is correct and S_2 is incorrect.
- S_1 can be written as $(000)^n$ where $n \geq 1$. Regular grammar for S_1 is $S \to S000/000$. Hence S_1 is regular.
- S_2 can be written as $(00)^{(x+y)}$ where $x \geq 1$ and $y \geq 1$. S_2 can be further reduced to $(00)^x$ where $x \geq 2$. Regular grammar for S_2 is $S \to S00/0000$. Hence S_2 is also regular language.

8. (b)

INIT(L) is a function which contain all the prefix strings of the language 1.

So, INIT(L) = {∈, a, b, ab, ba, aba, abab}

9. (c)

The minimum pumping length is 3. The pumping length cannot be 2 because the string 11 is in the language and it cannot be pumped. Let s be a string in the language of length at least 3. If s is generated by $0^* 1^+ 0^+ 1^*$, we can write it as xyz, where x is the empty string, y is the first symbol of s and z is the remainder of s. Breaking s up in this way shows that it can be pumped. If s is generated by $10^* 1$, we can write it as xyz, where $x = 1$ and $y = 0$ and z is the remainder of s. This division gives a way to pump s.

10. (c)

$$L = \{a^n b^m \mid n \geq 4;\ m \leq 3\}$$

$$L^c = \{a^n b^m \mid n < 4 \text{ or } m > 3\} \cup \{x\, ba\, x \mid x \in (a+b)^*\}$$

or

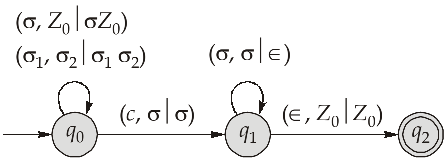
So, $$L^c = (\in + a + aa + aaa)b^* + a^*bbbbb^* + (a+b)^*\, ba(a+b)^*$$

11. (c)

$$L_1 / L_2 = bba^*baa^* \,/\, ab^*$$
$$= bba^*baa^* \,/\, a$$
$$= bba^*baa^*$$

12. (d)

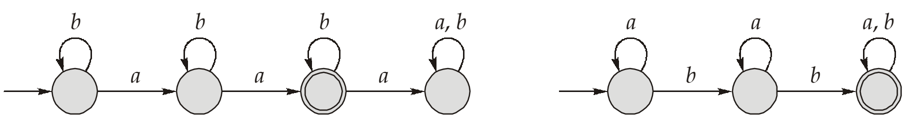
The transition diagram of the PDA is as shown below. In the figure σ, σ_1 and σ_2 represent a or b.



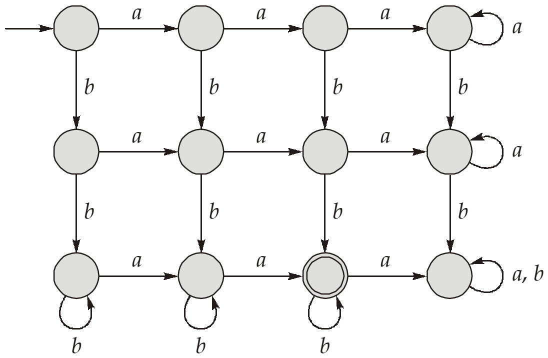
PDA accepting $\{wcw^R \mid w \in (a, b)^* \text{ and } |w| \geq 1\}$.

13. (b)

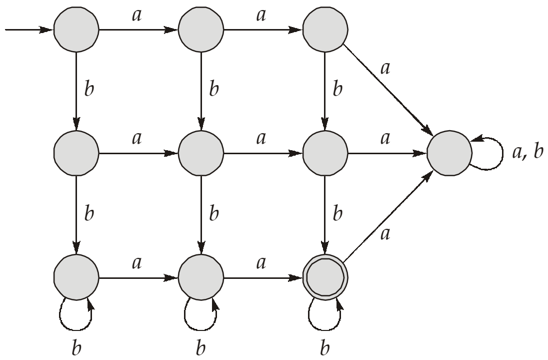
The following are DFAs for the two language $\{w \mid w$ has exactly two a's$\}$ and $w \mid w$ has at least two b's$\}$:



Combining them using the intersection construction gives the DFA:



Certain steps can be minimized



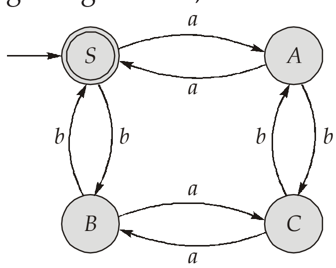
Hence total 10 states required.

14. **(c)**

Given PDA is NPDA, hence two comparison with or is possible. If you observe the automata, at q_2 for every input 'a' there is two transition on epsilon which proves it is NPDA. The upper branch comparing a with b and lower branch comparing a with 'c' and both leads to the final states. L accepts epsilon as well. Hence $L = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i = j \text{ or } i = k\}$.

15. **(a)**

Since grammar is right linear regular grammar, convert it to machine.



Language is (a) $n_a(w)$ and $n_b(w)$ both are even.

16. **(a)**

- Context free languages are not closed under complementation and intersection.
  Hence option (b) and (c) is false.
- DPDA is less powerful than PDA. Hence there is CFL language which can not be accepted by DPDA.
  Hence option (d) is false.

17. **(c)**

S_1 : True

S_2 : True

S_3 : False, checking regularity of TM is undecidable.

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. (b)

Turing-recognizable languages:

- TM halts in an accepting configuration if w is in the language.
- TM may halt in a rejecting configuration or go on indefinitely if w is not in the language.

Turing-decidable languages:

- TM halts in an accepting configuration if w is in the language.
- TM halts in a rejecting configuration if w is not in the language.

20. (c)

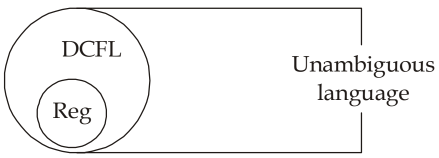
$$L = L_1 \cap L_2$$

$L = \{0^m 1^m \mid m \geq 0\}$ which is CFL but not regular because there is a infinite comparison.

21. (a)

- If there is no context free grammer for L which is unambiguous. Hence L is inherently ambiguous or in other words "A language for which every grammar is ambiguous is called inherently ambiguous language".

  Note: If a language is regular or DCFL it will surely can be written in unambiguous grammar.



- So, surely L_3 can't be inherently ambiguous language as it is regular language.
- L_2 is not inherently ambiguous language as we can write grammar for L_2 which is unambiguous.

$$S \rightarrow aSa \mid bSb \mid \in$$

- L_1 is CFL and it is inherently ambiguous language because L_1 is language with union of two DCFL. So the grammar will always have OR operation which will make them ambiguous. Let's see how we can write grammar.

  $S \rightarrow X \mid Y$
  $X \rightarrow aXc \mid P$
  $P \rightarrow bP \mid \in$
  $Y \rightarrow aYb \mid Q$
  $Q \rightarrow cQ \mid \in$

  If we generate string 'abc' two parse tree possible either through X or Y.
  Hence it is ambiguous and leads to inherently ambiguous language.

22. **(c)**

$S \rightarrow ABA$
$A \rightarrow aA \mid bA \mid \epsilon$
$B \rightarrow bb \mid bbB$

- A generates $(a + b)^*$
- B generates $(bb)^+$

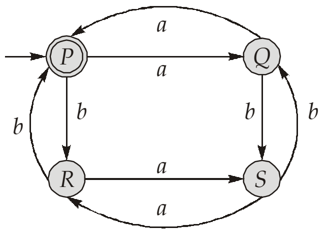
So, $S \rightarrow (a + b)^* (bb)^+ (a + b)^*$
So, correct option is (c).

23. **(c)**

- In option (a) both P and S are final states and cannot generate language L hence not correct.
- In option (b) S is the final state and cannot generate language L hence not correct.
- In option (c) P is the final state as well as initial state and correctly generates the language L.

  $P \rightarrow aQ \mid bR \mid \epsilon$
  $Q \rightarrow bS \mid aP$
  $R \rightarrow aS \mid bP$
  $S \rightarrow aR \mid bQ$

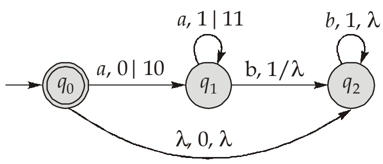
  The machine will be



- Option (d) is wrong because there is no final state.

24. **(b)**

The given PDA, $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{0, 1\}, \delta, q_0, 0, \{q_0\})$ where 6^{th} tuple represents the start stack symbol. So here in this case 0 is start stack symbol



This depicts that on every 'a' in the string '1' is pushed on the stack and stack is popped on every 'b'. Hence we need $a^n b^n$. Now since initial state is also the final state.

So, $L = \{a^n b^n \mid n \geq 0\}$

25. **(d)**

Given, $L = \{a^n b^m \mid (n + m) \text{ is even}\}$

For $(n + m)$ to be even either n and m both are even or n and m both are odd.

Therefore the Regular Expression (R.E.) = $\{(aa)^* (bb)^* + (aa)^*a (bb)^*b\}$

- So, for n and m to be even, grammar is:

  $S_1 \rightarrow aaS_1 \mid A_1$
  $A_1 \rightarrow bbA_1 \mid \epsilon$

- For n and m odd, grammar is:

  $S_2 \rightarrow aaS_2 \mid aA_2$
  $A_2 \rightarrow bbA_2 \mid b$

Now, combine both; then resultant grammar is:

$S \to S_1 | S_2$
$S_1 \to aaS_1 | A_1$
$A_1 \to bbA_1 | \in$
$S_2 \to aaS_2 | aA_2$
$A_2 \to bbA_2 | b$

26. **(d)**
Considering each statement:
CSL are closed under intersection therefore S_1 is true.
Turing decidable language are closed under union and Kleene star operation therefore S_2 is true.
Turing recognizable languages are not closed under complementation therefore S_3 is false.

27. **(d)**

1. L_1 = DCFL and complement of DCFL is DCFL because DCFL is closed under complementation.
2. L_2 = CSL and complement of CSL is CSL because CSL is closed under complementation.
3. $L_1 \cap L_2$ = DCFL $\cap$ CSL
   = DCFL$\uparrow$ $\cap$ CSL(Push DCFL upto CSL in Chomsky hierarchy because operation performed between same language)
   = CSL $\cap$ CSL = CSL
4. $L_1^C \cap L_2^C$ = (DCFL)^C $\cap$ (CSL)^C
   = DCFL $\cap$ CSL
   = CSL $\cap$ CSL = CSL

So all statements are true.

28. **(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 $L = \{a^m b^n | 3m = n; m, n \geq 0\}$.

29. **(c)**
Considering each option:
S_1 **:** Since "hello" is reducible to "world" and "world" is decidable, then "hello" is also decidable.
S_2 **:** Since, language specifies that, $n \leq 10$ and $q < n$ and 2020 is a finite number that means L is regular, hence CFL too.

30. **(c)**
Only S_3 is true:

- 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.

■■■■