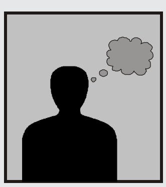
GATE PSUs

State Engg. Exams

MADE EASY WORKBOOK 2025



Detailed Explanations of Try Yourself *Questions*

Computer Science & IT

Theory of Computation

MADE EASY PUBLICATIONS

1 Regular Languages and Finite Automata

Detailed Explanation of Try Yourself Questions

T1 : Solution

(c)

The minimum string that would be accepted by the NFA is “ab”. Therefore it accepts all strings containing “ab” as substring.

T2 : Solution

(c)

R_1 = (a + b)*a(a + b) (a + b)* = (a + b)*aa(a + b)*+ (a + b)*ab(a + b)*

R_2 = a*bbab

All strings generated by R_2 are also derived from R_1 and R_1 can derive many other strings.

∴ $R_1 \supseteq R_2$ is correct relation.

T3 : Solution

(c)

R = (a+b)*a(ab)*+ε

(*i*) (aa)* = a(aa)*a + ε is subset

$$R = \frac{(a+b)^*}{a(aa)^*}\ \frac{a}{a}\ \frac{(ab)^*}{\varepsilon}\ \frac{+\varepsilon}{\varepsilon}$$

(*ii*) (ba)* = b(ab)*a + ε is subset of R

$$R = \frac{(a+b)^*}{b(ab)^*}\ \frac{a}{a}\ \frac{(ab)^*}{\varepsilon}\ \frac{+\varepsilon}{\varepsilon}$$

(*iii*) (aa)*(ba)* = (aa)* + (ba)* + (aa)$^+$(ba)$^+$ is subset of R

$$R = \frac{(aa)^*+}{R}\ \frac{(ba)^*+}{R}\ \frac{(ab)^+\ b(ab)^*}{(a+b)^*}\ \frac{a}{a}$$

∴ (*i*), (*ii*) and (*iii*) are correct subsets of R.

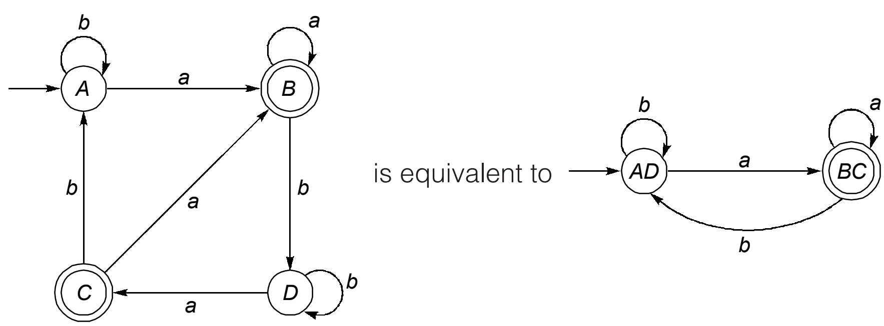
T4 : Solution

(c)

$$A = \{a^*b^*\},\ B = \{bb, ba, bbb\}$$
$$A/B = \{a^*b^*\}/\{bb, ba, bbb\}$$
$$= \{(a^*b^*)/bb\} \cup \{(a^*b^*)/ba\} \cup \{(a^*b^*)/bbb\}$$
$$= \{a^*b^*\} \cup \phi \cup \{a^*b^*\} = \{a^*b^*\}$$

T5 : Solution

(b)



The given DFA accepts the language of all strings where every string ends with a.

T6 : Solution

(a)

DFA1 and DFA2 are equivalent. Both accepts the same language that has all strings contain b.

[RE = $(a + b)^*b(a + b)^*$] = $a^*b(a + b)^*$.

DFA3 accepts the universal language: $(a + b)^*$.

DFA4 accepts $a^*bb^*a^*$.

T7 : Solution

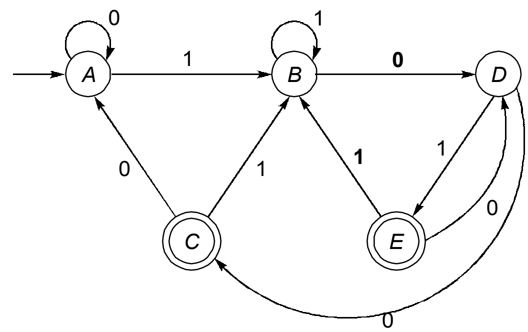
(b)

$$RE = (a + b)^*(a + b + \varepsilon)a = (a + b)^*a$$
$$(\varepsilon + a + b^*)^+ a = (a + b)^* a$$

$\therefore$ Option (b) is equivalent to given RE.

T8 : Solution

(c)



$$\text{RE} = (0+1)^*\,10\,(0+1)$$

Every string either end with 100 or 101.

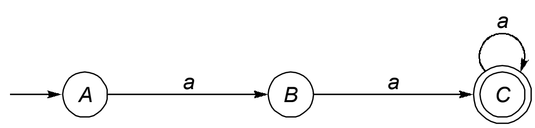
T9 : Solution

(c)

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

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

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



This accepts L.

T10 : Solution

(c)

(a) $L = \{wxwy \mid x, y, w \in (a+b)^+\} \Rightarrow L$ is non-regular language.

(b) $L = \{xwyw \mid x, y, w \in (a+b)^+\} \Rightarrow L$ is non-regular language.

(c) $L = \{wxyw \mid x, y, w \in (a+b)^+\} \Rightarrow L$ is non-regular language.

T11 : Solution

(d)

$$R_1 + R_2R_3 = (R_1 + R_2)(R_1 + R_3)$$

(*i*) If $R_1 = R_2$

$$R_1 + R_2R_3 = (R_1 + R_2)(R_1 + R_3)$$
$$R_2 + R_2R_3 = (R_2 + R_2)(R_2 + R_3)$$
$$R_2 + R_2R_3 = R_2(R_2 + R_3)$$
$$= R_2^2 + R_2R_3 \text{ which is incorrect.}$$

(*ii*) If $R_1 = R_3$

$$R_1 + R_2R_3 = (R_1 + R_2)(R_1 + R_3)$$
$$R_3 + R_2R_3 = (R_3 + R_2)(R_3 + R_3)$$
$$= (R_2 + R_3)R_3$$
$$= R_2R_3 + R_3^2 \text{ is incorrect.}$$

(*iii*) If

$$R_1 = \phi,$$
$$R_1 + R_2R_3 = (R_1 + R_2)(R_1 + R_3)$$
$$\phi + R_2R_3 = (\phi + R_2)(\phi + R_3)$$
$R_2R_3 = R_2\,R_3$ is correct.

$\therefore$ Only (*iii*) is correct.

T12 : Solution

(d)

If L* is regular, L may or may not be a regular.

Example 1:

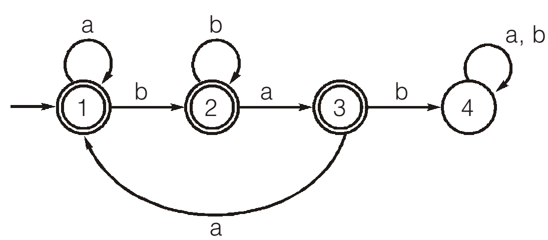
L* = (a + b)* is regular, L = (a + b) is regular.

Example 2:

L* = $\{(a^P)^* \mid P$ is prime$\}$ is regular but L = $\{a^P \mid P$ is prime$\}$ is non-regular.

T13 : Solution

(b)



T14 : Solution

(c)

No string can start with 1, as there is no path from M. So all strings should start with other than 1. If string starts with '0' then from A any sequence will be accepted by going to Y state.

T15 : Solution

(c)

$$n = 3$$

then L contains all strings of length 3

L = {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}

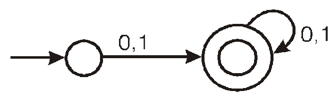
$\therefore$ $|L| = 2^3 = 8$ strings

T16 : Solution

(b)

Contains a substring ‘0’ or ‘1’

≅ starts with 0 or 1



$= (0+1)^+$

∴ It needs 2 states

T17 : Solution

(a)

L is recursive language, finite subset of L is a finite set

∴ L is regular language

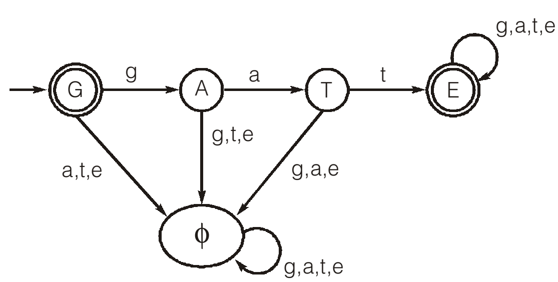
Note : Finite subset of any language is regular.

T18 : Solution

(b)

The given NFA accepts a language where each string starts with ‘gat’ [including Null string]

∴ Number of states required in DFA = 4 + 1 = 5 states



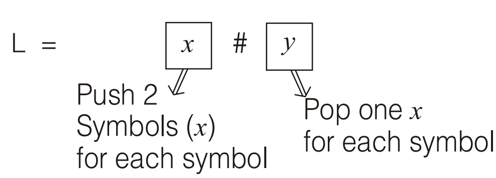
T19 : Solution

(b)

The reversal of any DFA will change direction of arrow, and final state into start state and vice-versa. So number of states after reversal of a DFA remains same.

T20 : Solution

(b)



∴ It is DCFL

It is clear when to push, how much to push, when to pop and how much to pop.

T21 : Solution

(a)

$$L = \{w_1\, w_2\, w_3\, w_2 \mid w_1, w_2, w_3 \in \{0, 1\}^+\}$$

$$= [(0+1)^+0\,(0+1)^+0] + [(0+1)^+1\,(0+1)^+1]$$

Last letter of w_2 Last letter of w_2

[w_1 and w_3 combined with remaining letters of w_2 and makes $(0 + 1)^+$]

∴ L is regular.

T22 : Solution

(b)

Class of languages recognized by NFA's

$\cong$

Class of languages recognized by DFA's

$\cong$

Class of regular languages

$\Downarrow$

Closed under complement.

T23 : Solution

(d)

$$X = L_1 \cup L_2 \cup L_3 \ldots\ldots$$

Case1: $\Sigma^* \cup L_2 \cup L_3 \ldots\ldots = \Sigma^*$

∴ X is regular [in one case]

Case 2 : $\{\varepsilon\} \cup \{ab\} \cup \{a^2b^2\} \cup \ldots\ldots = \{a^nb^n\}$ ∴ X is non-regular but CFL

Case 3 : $\{\varepsilon\} \cup \{abc\} \cup \{a^2b^2c^2\} \cup \{a^3b^3c^3\} \cup \ldots\ldots = \{a^nb^nc^n\}$ ∴ X is not CFL

(a), (b), (c) options can not be correct. We can conclude that X need not be regular, and need not CFL.

T24 : Solution

(c)

$$L_1 = \phi \rightarrow L_1^* = \{\varepsilon\} \text{ is finite}$$

$$L_1 = \{a\} \rightarrow L_1^* = \{a^*\} \text{ is infinite}$$

∴ L_2 need not be infinite

T25 : Solution

(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*.

T26 : Solution

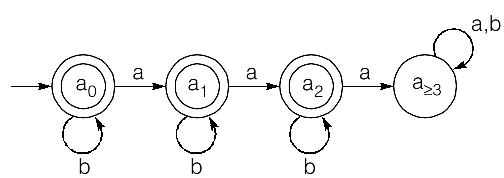
(a)

$$1\times 10 = 10 \Rightarrow 0^*1.0^*1010 \quad [\because \ \text{dec}(1) = 0^*1, \text{dec}(10) = 0^*1010]$$
$$10\times 1 = 10 \Rightarrow 0^*1010.0^*1$$
$$2\times 5 = 10 \Rightarrow 0^*10.0^*101$$
$$5\times 2 = 10 \Rightarrow 0^*101.0^*10$$
$$\Rightarrow \quad \text{R.E.} = 0^*10^*1010 + 0^*1010\,0^*1 + 0^*100^*101 + 0^*1010^*10$$

∴ L is regular language

T27 : Solution

(c)



All strings with atmost two a's are accepted by DFA.

T28 : Solution

(d)

Example 1 $L_1 = \{ba\}, L_2 = \{a^*b^*\}$

$L_1 \cup L_2 = ba + a^*b^*$

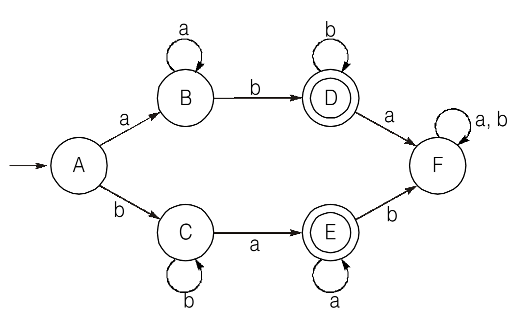
Example 2 $L_1 = \{ba\}, L_2 = \{a\}$

$L_1 \cup L_2 = ba + a$

∴ L_2 is regular but it may be finite or infinite.

T29 : Solution

(c)



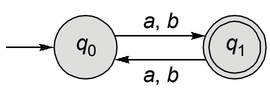
∴ 6 states are required to accept $a^+b^+ + b^+a^+$

T30 : Solution

(b)

Let $\Sigma = \{a, b\}$

Given, $\boxed{\delta(q_i, x) = q_{1-i}, x \in \Sigma, i = 0, 1}$



is equivalent DFA.

∴ DFA accepts all strings of odd lengths over Σ.

T31 : Solution

(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

T32 : Solution

(b)

We wish to find regular expression “for all binary strings containing two consecutive 0’s and two consecutive 1’s”.

Now, choice (a) cannot generate “00011”

Choice (b) is correct

Choice (c) “00” which does not belong to given language.

Choice (d) always ends with 11 or 00 and hence cannot generate “001101”.

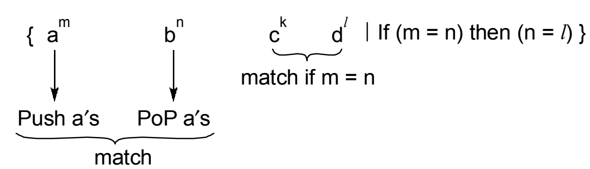
■■■■

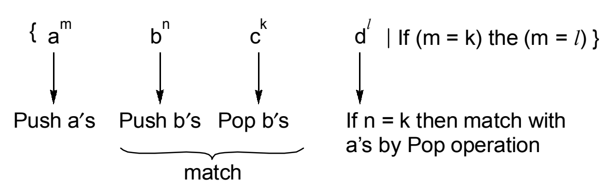
2 Context Free Languages and Pushdown Automata

Detailed Explanation of Try Yourself Questions

T1 : Solution

(d)





Both (a) and (b) are CFL.

T2 : Solution

(b)

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

T3 : Solution

(b)

(a) $S \to aAb$

$A \to aB \mid \varepsilon$

$B \to Ab$

$\Rightarrow \quad L = \{a^m b^n \mid m = n,\ m,\ n \geq 1\}$

(b) $S \to aABb$

$A \to aA \mid \varepsilon$

$B \to bB \mid \varepsilon$

$\Rightarrow \quad L = \{a^+ b^+\}$ is regular.

T4 : Solution

(c)

$S \to aSa \mid bSb \mid A$

$A \to aBb \Rightarrow L(A) = a(a+b)^*b$

$B \to aB \mid bB \mid \varepsilon \quad \Rightarrow L(B) = (a+b)^*$

$$L = \{wa(a+b)^*bw^R \mid w \in (a+b)^*\}$$

$$L = \{waxbw^R \mid x,\ w \in (a+b)^*\}$$

T5 : Solution

(a)

Given push down automata is non-deterministic. The language accepted by NPDA is a(a + b)*b.

q_0 pushes 'a' onto the stack, then q_1 can read any symbol from input.

To reach final state, last symbol must be 'b'. If last input symbol is 'b' then Pop a and match \$ with Z_0 from q_3, then goto q_f.

The language is regular but infinite.

T6 : Solution

(c)

(a) $L = \{a^P \mid P \text{ is prime}\}$ is non CFL.

(b) $L = \{a^m b^n \mid (m < n) \text{ or } (m > n)\}$ is DCFL.

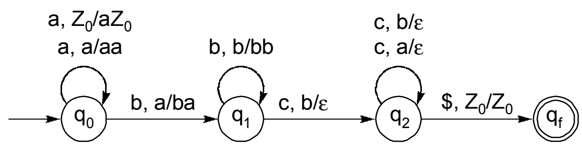
(c) $L = \{a^m b^n c^k \mid (m = n) \text{ or } (n = k)\}$ is CFL but not DCFL.

(d) $L = \{a^m b^n c^k \mid m = n,\ n = k\} = \{a^n b^n c^n\}$ is non CFL.

T7 : Solution

(c)

Given PDA can be redrawn as following:



At q_0, all a's are pushed.
At q_1, all b's are pushed.
At q_2, all c's are matched with b's and a's.
If # c's = # a's + # b's then goes to final state.

$\Rightarrow$ $$L = \{a^m b^n c^k \mid m, n, k \geq 1, k = m + n\}$$

T8 : Solution

(a)

$$L_1 - L_2 = \{a^n b^m c^n\} - \{a^n c^n\}$$
$$= \{a^n b^m c^n \mid m > 0, n \geq 0\} \text{ is DCFL.}$$

T9 : Solution

(c)

DCFLs are closed under
(*i*) Complement
(*ii*) Inverse homomorphism
(*iii*) Prefix operation
(*iv*) Union with regular
(*v*) Intersection with regular, etc.

DCFLs are not closed under
(*i*) Reversal
(*ii*) Concatenation
(*iii*) Intersection
(*iv*) Union, etc.

T10 : Solution

(a)

(a) $$L = \{a^m b^n c^k \mid m = n \text{ or } n = k\}$$

(b) $$L = (a + b)^*$$

Option (a) is inherently ambiguous language, because no equivalent unambiguous grammar exist for the language.
Option (b) is unambiguous language, because many unambiguous grammars exist for the language.

T11 : Solution

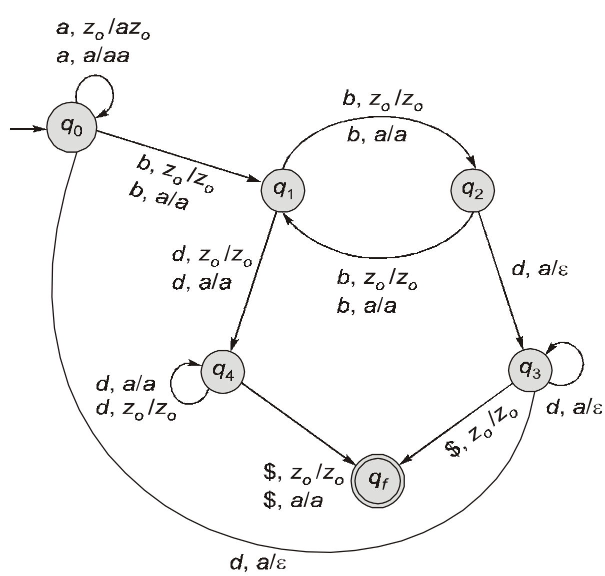
(d)

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

T12 : Solution

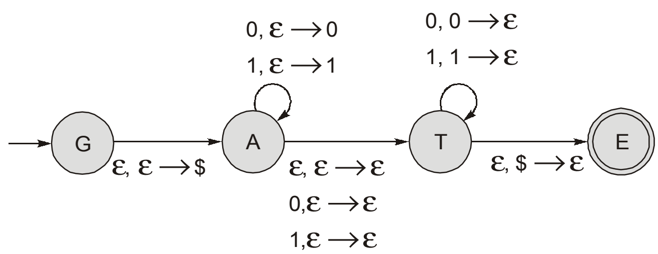
(c)

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



T13 : Solution

(c)



$G \rightarrow A$: Pushes "\$" onto stack initially

$A \rightarrow A$: Pushes 0 for input 0 and Pushes 1 for input 1

$A \rightarrow T$: Moves A to T without reading an input (or)

Read 0 or 1 from input tape and does no operation on the stack

$T \rightarrow T$: Pop 0 for input 0 and Pop 1 for input 1

$T \rightarrow E$: 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 : $S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon$

T14 : Solution

(c)

$$S \rightarrow aA \mid bSS \mid SS$$
$$A \rightarrow aAb \mid bAa \mid AA \mid \varepsilon$$

'A' generates equal number of a's and b's

'S' generates atleast one more a than A generates

$\therefore$ Grammar G generates all strings with atleast one more 'a' than number of b's.

T15 : Solution

(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.

T16 : Solution

(c)

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

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

T17 : Solution

(d)

G_1 : $S \to aS \mid B$

$B \to b \mid bB$

G_2 : $S \to aA \mid bB$

$A \to aA \mid B \mid \varepsilon$

$B \to bB \mid \varepsilon$

G : $B \to b \mid bB \Rightarrow B \to b^+$

Now substitute in $S \to aS \mid B$

We get $S \to aS \mid b^+ \Rightarrow S \to a^* \, b^+$

So, $L(G_1) = \{a^m b^n \mid m \geq 0 \text{ and } n > 0\}$

$G_2 = B \to bB \mid \varepsilon \Rightarrow B \to b^*$

Substitute in $A \to aA \mid B \mid \varepsilon \Rightarrow A \to aA \mid b^* \mid \varepsilon$

$\Rightarrow \quad A \to aA \mid b^*$

$\Rightarrow \quad A \to a^* b^*$

Now substitute A and B in $S \to aA \mid bB$

$\Rightarrow S \to aa^* \, b^* \mid bb^*$

$S \to aa^* \, b^* + bb^*$

So $L(G_2) = \{a^m b^n \mid m > 0 \text{ or } n > 0\}$

So correct answer is choice (d).

■■■■

3 Recursive, Recursively Enumerable Languages and Turing Machines

Detailed Explanation *of* Try Yourself Questions

T1 : Solution

(d)

REL is computable (TM computable), recognizable and enumerable by Turing machine.

T2 : Solution

(d)

$$S \rightarrow A1B$$
$$A \rightarrow AA$$
$$A1 \rightarrow 11B$$
$$B \rightarrow \varepsilon$$

$S \rightarrow A1B \Rightarrow 11B \Rightarrow 11$

$S \rightarrow A1B \Rightarrow AA1B \Rightarrow A11BB \Rightarrow 11B1BB \Rightarrow 111$

$S \rightarrow A1B \rightarrow AA1B \Rightarrow AAA1B \Rightarrow AA11BB \Rightarrow A11B11BB \Rightarrow 11B1B1BB \Rightarrow 1111$

$$L = \{1^n \,|\, n > 1\}$$

T3 : Solution

(d)

Option (a) is not a decidable language (REC).

Since although the number of strings of less than 100 length is finite and can be generated one-by-one and tested in a UTM for a given string the halting problem is undecidable.

Option (b) is not a decidable language since this involves checking whether a given string from the set {00, 11} is a member of the given Turing machine. But Turing machine membership is undecidable.

Option (c) is not a decidable language since this involves checking equivalence of

$L(M_1)$ with $L(M_2) \cup L(M_3)$ and equivalence of Turing machines is undecidable.

So, correct answer is option (d) none of these.

T4 : Solution

(e)

(a) For a given input string, particular state may or may not be reached. Finding the reachability for a particular string is undecidable. [State entry problem is undecidable]

(b) Writing a particular symbol 'x' on tape is undecidable since that particular symbol may be written only from some particular state and whether that state is entered or not is the state entry problem, which is undecidable.

(c) This problem is undecidable since the Turing machine may make a left move only from a particular state and whether that state is entered or not is the state entry problem, which is undecidable.

(d) Language accepted by M is finite or not, is undecidable. [Finiteness problem for Turing machines is undecidable, this problem is non-trivial and Rice's theorem applies].

So the correct answer is (e) none of these.

T5 : Solution

(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.

T6 : Solution

(d)

Language generated by a grammar is recursively enumerable hence it is Turing recognizable and partially decidable (semidecidable) language. (Need not be totally decidable)

T7 : Solution

(d)

(a) L is not recursive, TM accepts a regular language is undecidable.

(b) L is not recursive, TM accepts a regular language is undecidable.

(c) L is not recursive language, state entry problem is undecidable.

(d) L is recursive language since ten steps is finite and in a finite amount of time can be simulated on a UTM and hence decidable.

T8 : Solution

(a)

Each rule $A \to BC$ increases the length of the string by 1, which gives $(n - 1)$ steps and exactly n rules $A \to a$ to convert variables into terminals.

Therefore exactly $2n - 1$ steps are required for CNF CFG.

T9 : Solution

(d)

Recursive languages are not closed under Homomorphism and Substitution operations.

T10 : Solution

(c)

$L_1 \leq L_2$ and $L_2 \leq L_3$

L_3 is decidable $\Rightarrow$ L_2 is decidable

$\Rightarrow$ L_1 is decidable

$\therefore$ L_1 and L_2 are decidable

T11 : Solution

(a)

L_1 and L_2 are decidable

$\therefore$ $L_1 \cap L_2$ is also decidable

Turing decidable languages are recursive languages.

$\therefore$ Recursive languages are not closed under homomorphism

■■■■