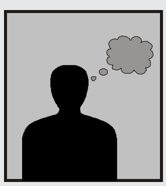
GATE

MADE EASY

Workbook 2026



Detailed Explanations of Try Yourself *Questions*

Computer Science & IT

Theory of Computation

MADE EASY PUBLICATIONS

1 Finite Automata

Detailed Explanation of Try Yourself Questions

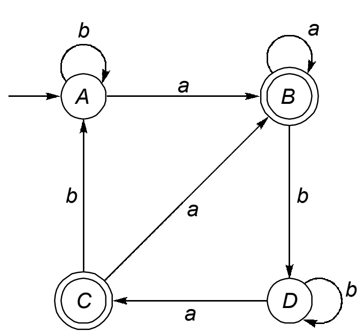
T1 : Solution

(c)

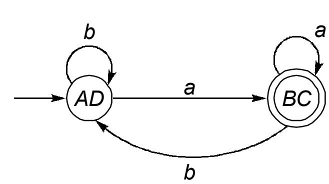
The minimum string that would be accepted by the NFA is “ab”. Therefore it accepts all strings containing “ab” as substring.

T2 : Solution

(b)



is equivalent to



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

T3 : 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*.

T4 : Solution

(d)

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

Example 1:

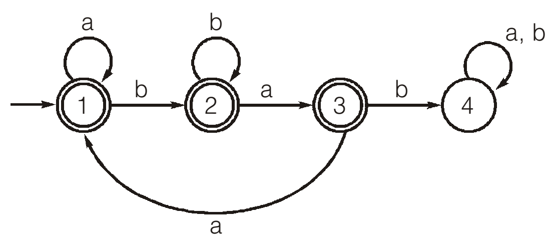
$L^* = (a + b)^*$ is regular, $L = (a + b)$ is regular.

Example 2:

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

T5 : Solution

(b)



T6 : Solution

(c)

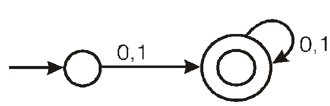
No string can start with 1, as there is no path from M. So all strings should start with other than 1. If string starts with '0' then from A any sequence will be accepted by going to Y state.

T7 : Solution

(b)

Contains a substring '0' or '1'

$\cong$ starts with 0 or 1



$= (0+1)^+$

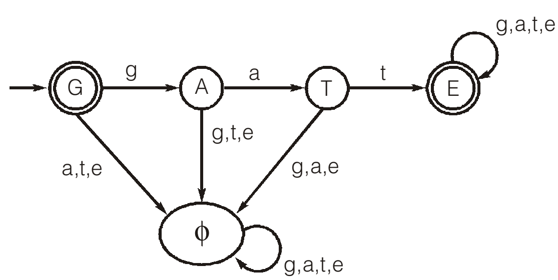
$\therefore$ It needs 2 states

T8 : 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



T9 : 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.

T10 : Solution

(a)

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

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

$\boxed{\text{Last letter of } w_2}$ $\boxed{\text{Last letter of } w_2}$

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

∴ L is regular.

T11 : Solution

(b)

Class of languages recognized by NFA's

≅

Class of languages recognized by DFA's

≅

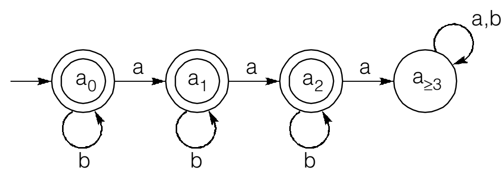
Class of regular languages

⇓

Closed under complement.

T12 : Solution

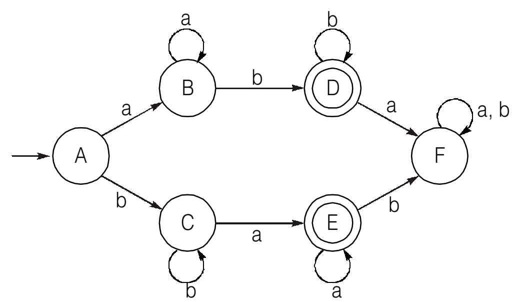
(c)



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

T13 : Solution

(c)



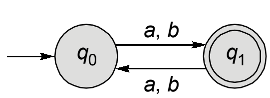
$\therefore$ 6 states are required to accept $a^+b^+ + b^+a^+$

T14 : Solution

(b)

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

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



is equivalent DFA.

$\therefore$ DFA accepts all strings of odd lengths over Σ.

T15 : 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

■■■■

2 Regular Expression

Detailed Explanation *of* Try Yourself Questions

T1 : 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.

$\therefore\ R_1 \supseteq R_2$ is correct relation.

T2 : Solution

(c)

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

(*i*) $(aa)^* = a(aa)^*a + \varepsilon$ is subset

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

(*ii*) $(ba)^* = b(ab)^*a + \varepsilon$ 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

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

$\therefore$ (*i*), (*ii*) and (*iii*) are correct subsets of R.

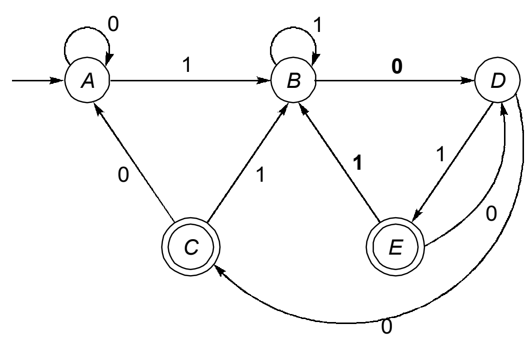
T3 : Solution

(c)

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

T4 : Solution

(c)



$$RE = (0 + 1)^*\ 10\ (0 + 1)$$

Every string either end with 100 or 101.

T5 : Solution

(d)

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

(*i*) If $R_1 = R_2$

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

(*ii*) If $R_1 = R_3$

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

(*iii*) If $R_1 = \phi$,

$$\begin{aligned} R_1 + R_2R_3 &= (R_1 + R_2)(R_1 + R_3) \\ \phi + R_2R_3 &= (\phi + R_2)(\phi + R_3) \\ R_2R_3 &= R_2R_3 \text{ is correct.} \end{aligned}$$

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

T6 : Solution

(c)

$$n = 3$$

Then L contains all strings of length 3

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

$$\therefore \quad |L| = 2^3 = 8 \text{ strings}$$

T7 : Solution

(d)

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

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

$\therefore$ X is regular [in one case]

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

Case 3 : $\{\varepsilon\} \cup \{abc\} \cup \{a^2b^2c^2\} \cup \{a^3b^3c^3\} \cup \ldots.. = \{a^nb^nc^n\}$ $\therefore$ 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.

T8 : Solution

(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

T9 : 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*.

T10 : Solution

(a)

$$\begin{aligned}
1 \times 10 &= 10 \Rightarrow 0^*1.0^*1010 \quad [\because \ \text{dec}(1) = 0^*1, \text{dec}(10) = 0^*1010] \\
10 \times 1 &= 10 \Rightarrow 0^*1010.0^*1 \\
2 \times 5 &= 10 \Rightarrow 0^*10.0^*101 \\
5 \times 2 &= 10 \Rightarrow 0^*101.0^*10
\end{aligned}$$

$$\Rightarrow \quad \text{R.E.} = 0^*10^*1010 + 0^*1010\,0^*1 + 0^*100^*101 + 0^*1010^*10$$

$\therefore$ L is regular language.

T11 : Solution

(d)

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

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

$\therefore$ L_2 is regular but it may be finite or infinite.

T12 : 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".

■■■■

3 Grammar

Detailed Explanation of Try Yourself Questions

T1 : 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.

T2 : Solution

(c)

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

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

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

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

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

T3 : 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.

T4 : Solution

(d)

$$S \to aSa \mid aAa$$

$$A \to bA \mid b$$

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

T5 : Solution

(c)

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

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

'S' generates atleast one more a than A generates

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

T6 : Solution

(c)

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

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

T7 : Solution

(d)

G_1 : $S \rightarrow aS \mid B$

$B \rightarrow b \mid bB$

G_2 : $S \rightarrow aA \mid bB$

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

$B \rightarrow bB \mid \varepsilon$

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

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

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

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

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

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

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

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

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

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

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

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

So correct answer is choice (d).

T8 : Solution

(d)

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

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

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

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

$$L = \{1^n \mid n > 1\}$$

T9 : Solution

(a)

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

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

■■■■

4 Pushdown Automata

Detailed Explanation of Try Yourself Questions

T1 : Solution

(d)

$$\{\ \underbrace{\underset{\text{Push a's}}{\underset{\downarrow}{a^m}} \qquad \underset{\text{PoP a's}}{\underset{\downarrow}{b^n}}}_{\text{match}} \qquad \underbrace{c^k \quad d^l}_{\text{match if m = n}} \ \mid \text{If } (m = n) \text{ then } (n = l)\ \}$$

$$\{\ \underset{\text{Push a's}}{\underset{\downarrow}{a^m}} \qquad \underbrace{\underset{\text{Push b's}}{\underset{\downarrow}{b^n}} \qquad \underset{\text{Pop b's}}{\underset{\downarrow}{c^k}}}_{\text{match}} \qquad \underset{\substack{\text{If n = k then match with}\\ \text{a's by Pop operation}}}{\underset{\downarrow}{d^l}} \ \mid \text{If } (m = k) \text{ the } (m = l)\ \}$$

Both (a) and (b) are CFL.

T2 : Solution

(b)

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

T3 : 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.

T4 : 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.

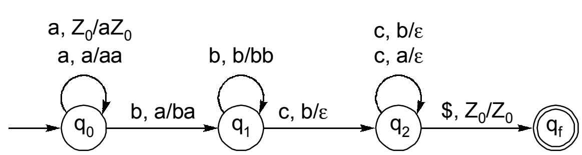
(c) $L = \{a^m b^n c^k \mid (m = n) \text{ or } (n = k)\}$ is CFL but not DCFL.

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

T5 : 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.

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

T6 : Solution

(a)

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

T7 : 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.

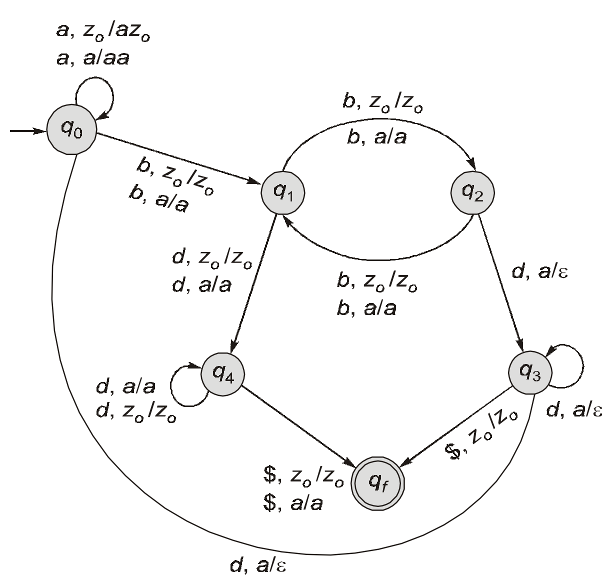
T8 : Solution

(c)

$$L = \{a^m b^n b^k d^\ell \mid \text{if } n = k \text{ then } m = \ell\}$$

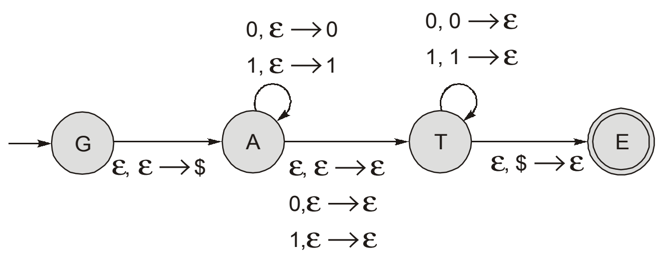
$$= \{a^m b^{2n} d^m\} \cup \{a^m b^{2n+1} d^k\}$$

$$= \text{DCFL} \cup \text{regular} = \text{DCFL}$$



T9 : Solution

(c)



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

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

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

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

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

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

T10 : Solution

(d)

All given languages are DCFL.

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

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

■■■■

5 Turing Machine and Undecidability

Detailed Explanation of Try Yourself Questions

T1 : Solution

(d)

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

T2 : 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.

T3 : 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.

T4 : 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.

T5 : 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)

T6 : 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.

T7 : Solution

(d)

Recursive languages are not closed under Homomorphism and Substitution operations.

T8 : 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

T9 : 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

■■■■