Fifth Semester B.E. Degree Examination

CBCS - Model Question Paper - 3

AUTOMATA THEORY AND COMPUTABILITY

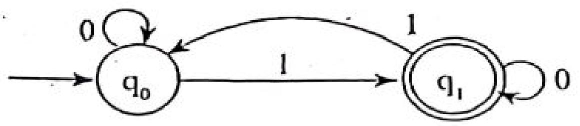
Time: 3 hrs. Max. Marks: 80

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

MODULE - 1

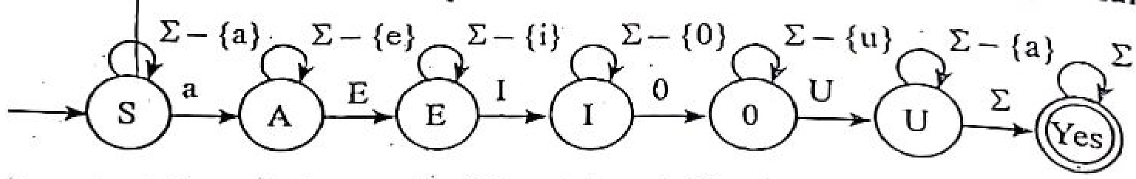
1. a. **Design a DFSM M : Which will check for a odd parity over the binary string 0 and 1.** **(04 Marks)**

Ans. L = {w ∈ {0, 1} * : w has odd parity}



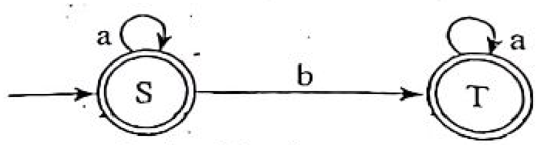
b. **Design a DFSM to accept a string that forms vowels : L = {w ∈ {a – z}*}** **(04 Marks)**

Ans.



c. **Write a simulating code to accept a string of a's and b's where the machine has to reject if it encounter's more than one b.** **(08 Marks)**

Ans. The DFSM for the string a's & b's which accept only one b is as follows.



We could view M as specification for the following program:

Until accept or reject do :

S : S = get-next-symbol
 if S = end-of-file then accept
 Else if s = a then go to S
 Else if s = b then go to T
T : S = get-next-symbol
 if S = end-of-file then accept
 Else if s = a then go to T
 Else if s = b then reject
End

OR

2. a. Define Canonical form for regular languages **(04 Marks)**

Ans. A canonical form for some set of objects C assigns exactly one representation to each class of "equivalent" object in C. Further, each such representation is distinct, so two objects in C share the same representation if they are "Equivalent" in the sense for which we define the form.

The ordered binary decision diagram (OBDD) is a canonical form for Boolean expression that makes if possible for model checkers to verify the correctness of very large concurrent systems and hardware circuits.

b. Explain the Moore Machine with mathematical notion. **(04 Marks)**

Ans. A Moore machine M is a seven - tuple (K, Σ, O, δ, D, S, A) where:

- K is a finite set of states
- Σ is a input alphabet
- O is an output alphabet
- S ∈ K is the start state
- A ⊆ K is the set of accepting states (although for some applications thieole signation is not important)
- δ is the transition function. It is the function from (K) to (O*)

A Moore machine M computes a function f(w) iff, when it reads the input string w, its output sequences is f(w).

c. Construct a minimum state automation equivalent to a DFA whose transition table is shown below : **(08 Marks)**

State	a	b
→q_0	q_1	q_2
q_1	q_4	q_3
q_2	q_4	q_3
(q_3)	q_5	q_6
(q_4)	q_7	q_6
q_5	q_3	q_6
q_6	q_6	q_6
q_7	q_4	q_6

Ans.

$$Q_1^0 = \{q_3, q_4\}, \quad Q_2^0 = \{q_0, q_1, q_2, q_5, q_6, q_7\}$$

$$\pi_0 = \{\{q_3, q_4\}, \{q_0, q_1, q_2, q_5, q_6, q_7\}\}$$

q_3 is 1 – equivalent to q_4. So, $\{q_3, q_4\} \in \pi_1$

q_0 is not 1 – equivalent to q_1, q_2, q_5 but q_0 is 1 – equivalent to q_6.

Hence $\{q_0, q_6\} \in \pi_1$. q_1 is 1 – equivalent to q_2 but not 1 – eqivalent to q_5, q_6 or q_7.

$So_1 \{q_5, q_7\} \in \pi_1$

q_5 is not 1 – equivalent to q_6 but to q_7. $So_1 \{q_5, q_7\} \in \pi_1$

Hence $\pi_1 = \{\{q_3, q_4\}, \{q_0, q_6\}, \{q_1, q_2\}, \{q_5, q_7\}\}$

q_3 is 2 – equivalent to q_4. So $\{q_3, q_4\} \in \pi_2$

q_0 is not 2 – equivalent to q_6. So $\{q_0\}, \{q_6\} \in \pi_2$

q_1 is 2 – equivalent to q_2. So $\{q_1, q_2\} \in \pi_2$

q_5 is 2 – equivalent to q_7. $So_1 \{q_5, q_7\} \in \pi_2$

Hence $\pi_2 = \{\{q_3, q_4\}, \{q_0\}, \{q_6\}, \{q_1, q_2\}, \{q_5, q_7\}\}$

q_3 is 3 – equivalent to q_4; q_4 is 3 – equivalent to q_2 and q_5 is 3 – equivalent to q_7. Hence

$\pi_3 = \{\{q_0\}, \{q_1, q_2\}, \{q_3, q_4\}, \{q_5, q_7\}, \{q_6\}\}$ As $\pi_3 = \pi_2$ the minimum state automation is

$M^1 = (Q^1, \{a, b\}, \delta_1^1 [q_0], \{[q_3, q_4]\})$

State	a	b
$[q_0]$	$[q_1, q_2]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_3, q_4]$	$[q_3, q_4]$
$[q_3, q_4]$	$[q_5, q_7]$	$[q_6]$
$[q_5, q_7]$	$[q_3, q_4]$	$[q_6]$
$[q_6]$	$[q_6]$	$[q_6]$

Module - 2

3. a. Obtain the regular expression for the following language

i) $L = \{a^n b^m \mid m \geq 1, n \geq 1, nm \geq 3\}$ ii) $L = \{a^{2n} b^{2m} \mid n \geq 0, m \geq 0\}$ (08 Marks)

Ans. i) $L = \{a^n b^m \mid m \geq 1, n \geq 1, nm \geq 3\}$

Case 1 : since $nm \geq 3$, if $m = 1$ then $n \geq 3$ i.e., RE is given by aaaa*b

Case 2 : since $mn \geq 3$, if $n = 1$ then $m \geq 3$ i.e., RE is given by abbbb*

Case 3 : since $nm \geq 3$, if $m \geq 2$ and $n \geq 2$ i.e., RE is given by aaa* bbb*

So, final regular expression is

RE = aaaa*b + abbbb* +aaa* bbb*

ii) $L = \{a^{2n} b^{2m} \mid n \geq 0, m \geq 0\}$

For every $n \geq 0$, a^{2n} results in even number of a's and for every $m \geq 0$ b^{2m} results in even number of b's. The regular expression representing even number of a's and b's is given by

RE = (aa)*

RE = (bb)* so final regular expression is

RE = (aa)* (bb)*

b. Let $\Sigma = \{0, 1\}$ $\Gamma = \{0, 1, 2\}$ and $h(0) = 01$, $h(1) = 112$. What is h(010)? If L = {00. 010} What is homomorphism image of L? (04 Marks)

Ans. $h(w) = h(a_1)\, h(a_2) \ldots h(a_n)$

So, h(010) = h(0) h(1) h(0) = 0111201

$L = \{00, 010\} = L(h(00), h(010))$
$= L(h(0)(0), h(0)(h_1) h(0))$
$= L(0101, 0111201)$

Therefore $h(010) = 0111201$
$L(00, 010) = L(0101, 0111201)$

c. What are the various limitations of finite automata? (04 Marks)

Ans. 1) An FA has finite number of states and so it does not have the capacity to remember arbitrary long amount of information.

2) Since it does not have memory, FA can not remember a long string. For example : String is palindrome or not.

3) Finite automata or finite state machine have trouble recognizing various types of languages involving counting, calculating storing the string.

OR

4. a. State and prove that regular grammars Define exactly the regular languages. (08 Marks)

Ans. **Theorem :-** The class of languages that can be defined with regular grammars is exactly the regular languages.

Proof :- We first show that any language that can be defined with a regular grammar can be accepted by some FSM and so is regular. Then we must show that every regular language can be defined with a regular grammar. Both proofs are by construction.

Regular grammar → FSM : The following algorithm constructs an FSM M from a regular grammar $\Sigma = (V, \Sigma, R, S)$ and assures that $L(M) = L(G)$:

Grammar to FSM (G : regular grammar)

1. Create in M a separate state for each non terminal in V.
2. Make the state corresponding to S the start state.
3. If there are any rules in R of the form $x \to w_1$ for some $w \in \Sigma$ the create an additional state labeled #.
4. For each rule of the form $x \to wy$, add a transition from x to y labeled w.
5. For each rule of the form $x \to w$, add a transition from x to # labeled w.
6. For each rule of the form $x \to \varepsilon$, mark state X as accepting.
7. Mark state # as accepting.
8. If M is incomplete, M requires a dead state. Add a new static for every (q, i) pair for which no transition has already been definite create a transition from q to D labeled i. For every i in Σ, create a transition from D to D labeled i.

b. Show that the set $L = \{a^{i^2} \mid i \geq 1\}$ is not regular. (08 Marks)

Ans. Step 1 :- Suppose L is regular. let n be the number of states in the finite automaton accepting L.

Step 2 :- Let $w = a^{n^2}$. Then $|w| = n^2 > n$. By pumping Lemma, we can write $w = xyz$ with $|xy| \leq n$ and $|y| > 0$

Step 3 :- Consider xy^2z $|xy^2z| = |x| + 2|y| + |z| > |x| + |y|$ as $|y| > 0$. This means $n^2 = |xy^2z| = |x| + |y| + |z| < |xy^2z|$. As $|xy| \leq n$,

We have $|y| \leq n$. Therefore

$$|xy^2z| = |x| + 2|y| + |z| \leq n^2 + n$$

i.e, $$n^2 < |xy^2z| \leq n^2 + n < n^2 + n + n + 1.$$

Hence, $|xy^2z|$ strictly lies between n^2 and $(n+1)^2$ but is not equal to any one of them. Thus $|xy^2z|$ is not a prefect square and so xy^2z L, But by pumping Lemma, $xy^2z \in L$, This is a contradiction.

Module - 3

5. a. Obtain a grammar to generate integer number and derive for +1965 from the productions **(08 Marks)**

Ans. G = (V, T, P, S)

V = {D, S, N, I}

T = {+, –, 0, 1, 3, 4, 5, 6, 7, 8, 9}

P = {

I → N | SN (Generate signed / Unsigned number)

N → D | ND | DN (Generate one or more digits)

S → +|–|ε (Generate the sign)

D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (Generate digit)

}

S = I which is start symbol.

The signed number + 1965 can be derived as shown

I ⇒ SN

⇒ +N

⇒ +ND

⇒ +N5

⇒ +ND5

⇒ +N65

⇒ +ND65

⇒ +N965

⇒ +D965

⇒ +1965

b. Define parse tree. Obtain the parse tree for string id + id * is from the grammar **(08 Marks)**

E → E + T | T

T → T * F | F

F → id

Ans. Let G = (V, T, P, S) be a CFG, The tree is derivation tree (parse tree) with following properties.

1. The root has the labels.
2. Every vertex has a label which is in (V u T u ε)
3. Every leaf node has label from T and an interior vertex has a label from v.
4. If a vertex is label A and if $X_1, X_2, X_3, \ldots X_n$ are all children of A from left then $A \to X_1, X_2, X_3, \ldots X_n$ must be a production in P.

E ⇒ E + T

⇒ E + T * F

⇒ E + T * id

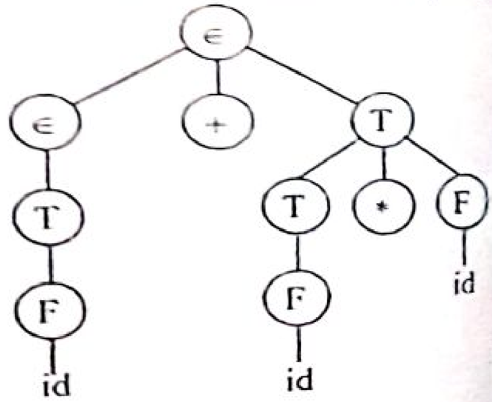
⇒ E + F * id

⇒ E + id * id

⇒ T + id * id

⇒ F + id * id

⇒ id + id * id



OR

6. a. Obtain a PDA to accept the language $L(M) = \{w|w \in (a + b)^* \; n_a(w) = n_b(w)$ by a final state, and show the I_D to reject aabbb. **(08 Marks)**

Ans. $M(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = \{q_0, q_1\}$

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

$\Gamma = \{a, b, z_0\}$

$\delta : \delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, b, z_0) = (q_0, bz_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, b) = (q_0, \varepsilon)$

$\delta(q_0, b, a) = (q_0, \varepsilon)$

$\delta(q_0, \varepsilon, z_0) = (q_1, z_0)$

$q_0 \in Q$ is the start state of the machine

$z_0 \in \Gamma$ is the initial symbol on the stack

$F = \{q_1\}$ is the final state

Initial ID

$(q_0, aabbb, z_0) \leftarrow (q_0, abbb, az_0)$

$\leftarrow (q_0, bbb, aaz_0)$

$\leftarrow (q_0, bb, az_0)$

$\leftarrow (q_0, b, z_0)$

$\leftarrow (q_0, \varepsilon, bz_0)$

(Final configuration)

Since the transition $\delta(q_0, \varepsilon, b)$ is not defined the string aabbb is rejected by PDA.

b. Convert the grammar to chomsky normal form **(08 Marks)**

$G = (\{S, A, B, C, a, c\}, \{A, B, C\}, R, S\})$, where

$R = \{S \rightarrow a\,A\,C\,a$

$A \rightarrow B \mid a$

$B \rightarrow C \mid c$

$C \rightarrow cC \mid \varepsilon\}$

Ans. $S \rightarrow aAca \mid aAa \mid aCa \mid aa$

$A \rightarrow B \mid a$

$B \rightarrow C \mid c$

$C \rightarrow cC \mid c$

Remove unit production

Remove $A \rightarrow B$ Add $A \rightarrow C \mid c$

Remove $B \rightarrow C$ Add $B \rightarrow cC$

Remove $A \rightarrow C$ Add $A \rightarrow cC$

Therefore

$S \rightarrow aACa \mid aAa \mid aCa \mid aa$

$A \rightarrow a \mid c \mid cC$

$B \rightarrow c \mid cC$

$C \rightarrow cC \mid C$

Remove mixed production

$S \rightarrow T_a ACT_a \mid T_a AT_a \mid T_a T_a$

$A \rightarrow a \mid c \mid T_c C$

$B \rightarrow C \mid T_cC$
$C \rightarrow T_cC \mid C$
$T_a \rightarrow a$
$T_c \rightarrow C$
Remove long sequence
$S \rightarrow T_aS_1$ $S \rightarrow T_aS_3$ $S \rightarrow T_aS_4$ $S \rightarrow T_aT_a$
$S_1 \rightarrow AS_2$ $S_3 \rightarrow AT_a$ $S_4 \rightarrow CT_a$
$S_2 \rightarrow CT_a$
Finally
$A \rightarrow a \mid c \mid T_cC$
$B \rightarrow C \mid T_cC$
$C \rightarrow T_cC \mid C$
$T_a \rightarrow a$
$T_c = C$

Module - 4

7. a. Show that $L = \{a^nb^nc^n \mid n \geq 1\}$ is not context free but context sensitivity. **(08 Marks)**

Ans. Step 1 :- Assume L is context free. Let n be the natural number obtained by using the pumping lemma.

Step 2 :- Let $Z = a^nb^nc^n$. then $|Z| = 3n > n$. write Z = uv wxy, where $|vx| \geq 1$, i.e., at least one of V or x is not A.

Step 3 :- $uvwxy = a^nb^nc^n$. As $| \leq |Vx| \leq n$. v or x cannot contain all the three symbol a,b,c. So (i) v or x is of the form a^ib^j(or b^ic^j) for some i, j such taht $i+j \leq n$. or (ii) v or x is a string formed by the repetition of only one symbol among a,b,c.

When v or x is of the form a^ib^j, $v^2 = a^ib^j, a^ib^j$ (or $x^2 = a^ib^j, a^ib^j$). As V^2 is a substring of $uv^2 wx^2y$, we cannot have uv^2wx^2y fo the form $a^mb^mc^m$. $av^2 wx^2y$ L.

When both v and x are formed by the repetition of a single symbol, the string uwy will contain the remaining symbol, say a_i. Also, a_i^n will be substring of uwy as a_i does not occurrences of a_i. So $uv^0wx^0 = uwy$ L

Thus for any choice of v or x, we get a contradiction. There for e, L is not context free, but context sensitive.

b. Prove that context free languages are Nonclosure under Intersection complement and difference. **(08 Marks)**

Ans. Proof :- The context free languages are not closed under intersection. The proof is by counter example Let :

$L_1 = \{a^nb^nc^m : n, m \geq 0)$
$L_2 = \{a^mb^nc^n : n, m \geq 0)$

Both L_1 and L_2 are context free since there exist straight forward context free grammars for them.

$L = L_1 \eta L_2$
$= \{a^n b^n c^n : n \geq 0\}$. If the CFL were closed under intersection L would have to be context free.

The CFL are not closed under complement given any sets L_1 and L_2.

$L_1 \cap L_2 = \neg(\neg L_1 \cup \neg L_2)$

The CFL are closed under union so if they were also closed under complement, they would necessarily be closed under intersection. But it is not. Thus they are not closed under complement either.

The CFL are not closed under difference given any language L,

$\neg L = \Sigma^* - L$

Σ^* is context free, so, if the CFL were closed under difference, the complement of any context - free language would necessarily be context free. But it is not.

Therefore CFL are Non closure under Intersection complement and difference.

OR

8. a. Consider the transition table for the Turning machine. Draw the computation sequence of the input string 00b. (08 Marks)

State	Tape symbol		
	b	0	1
$\rightarrow q_1$	$1Lq_2$	$0Rq_1$	
q_2	bRq_3	$0Lq_2$	$1Lq_2$
q_3		bRq_4	bRq_5
q_4	$0Rq_5$	$0Rq_4$	$1Rq_5$
q_5	$0Lq_2$		

Ans. We describe the computation sequence if term of the contents of the tape and current state. If the string in the tape is $a_1a_2 \ldots a_j a_{j+1} \ldots a_m$ and the TM in state q is to read a_{j+1}, the we write $a_1a_2 \ldots a_j q a_{j+1} \ldots a_m$.

For the input string oob, we get the following sequence

$q_1 00b \vdash 0q_1 0b \vdash 00q_1 b \vdash 0q_2 01 \vdash q_2 001 \vdash q_2 b001 \vdash bq_3 001 \vdash bbq_4 01 \vdash bb0q_4 1 \vdash bb01q_5 b \vdash bb010q_5 \vdash bb01q_2 00 \vdash bbq_2 100 \vdash bbq_2 0100 \dashv bq_2 b0100 \vdash bq_3 0100 \vdash bbbq_2 100 \vdash bbb_1 q_4 00 \vdash bbb10q_4 0 \vdash bbb100q_4 b \vdash bbb1000q_5 b \vdash bbb1000v_2 00 \vdash bbb10q_2 00 \vdash bbb1q_2 0000 \vdash bbbq_2 10000 \vdash bbq_2 b10000 \vdash bbbq_3 10000 \vdash bbbbq_5 0000$

b. Design a TM which can multiply two positive integers. (08 Marks)

Ans. The input (m,n),m, n being given, the positive integers are represented by $0^m|0^n$. M starts with $0^m|0^n$ in its tape. At the end of the computation, 0^{mn} surrounded by b's is obtained as the output. The major steps in the construction are as follows:

1. $0^m|0^n|$ is placed on the tape
2. The leftmost 0 is erased.
3. A block of n 0's is copied onto the right end.
4. Step 2 and 3 are repeated m times and 10^m, 10^{mn} is obtained on the tape.
5. The prefix 10^n1 of $1\,0^n10^{mn}$ is erased, leaving the product mn as the output.

For every 0 in 0^m, 0^n is a doled onto the right end. this requires repetition of step 3 we defined a subroutine called copy for step 3.

State	Tape symbol			
	0	1	2	b
q_1	$q_2 2R$	$q_4 1L$	–	–
q_2	$q_2 0R$	$q_2 1R$	–	$q_3 0L$
q_3	$q_3 0L$	$q_3 1L$	$q_1 2R$	–
q_4	–	$q_5 1R$	$q_4 0L$	–
q_5	–	–	–	–

Module - 5

9. a. Does the pcp with two lists x = (b, bab^3, ba) and y(b^3, ba,a) have a solution? **(04 Marks)**

Ans. We have to determine whether or not there exists a sequence of substrings of x such that string formed by this sequence and the string formed by the sequence of corresponding substrings of y are identical. The required sequence is given by $i_1 = 2$, $i_2 = 1$, $i_3 = 1$, $i_4 = 3$ i.e., (2,1,1,3) and m = 4

The corresponding strings are :

bab^3	b	b	ba	=	ba	b^3	b^3	a
x_2	x_1	x_1	x_3		y_2	y_1	y_1	y_3

Thus pcp has a solution.

b. Prove that pcp with two lists x = (01, 1, 1) y = (0)^2 10, 1^1) **(04 Marks)**

Ans. For each substring $x_i \in x$ and $y_i \in y$ we have $|x_i| < |y_i|$ for all i. Hence the string generated by a sequence of the substring of x is shorter than the string generated by the sequence of corresponding substring of y. Therefore, the pcp has no solution.

c. Let $f(n) = 4n^3 + 5n^2 + 7n + 3$. Prove that $f(n) = 0(n^3)$ **(08 Marks)**

Ans. Let $f(n) = 4n^3 + 5n^2 + 7n + 3$

In order to prove that $f(n) = 0(n^3)$, take c = 5 and $N_0 = 10$, then

$f(n) = 4n^3 + 5n^2\ 7n + 3 \leq 5n^3$ for $n \geq 10$

when n = 10,

$5n^2 + 7n + 3 = 573 < 10^3$ for $n > 10$, $5n^2 + 7n + 3 < n^3$

Then $f(n) = 0(n^3)$

Therefore the function $f(n) = 4n^3 + 5n^2 + 7n + 3$ is $f(n) = 0(n^3)$, the order of the growth of the function is in cubic order.

OR

10. a. Write short notes on:

i. **Growth rate of algorithm**

ii. **Classes of P and NP**

iii. **NP- complete problem**

iv. **Subroutine in TM** **(16 Marks)**

Ans. **i) Growth rate of algorithm:** Algorithms analysis is all about understanding growth rates. That is as the amount of data gets bigger, how much more resource will my algorithm require? Typically, we describe the resource growth rate of a piece of code in terms of a function. The algorithm may have different kind of growth rate, following are few growth rate

1. Constant growth rate
2. Logarithmic growth rate
3. Linear growth rate
4. Log linear
5. Quadratic growth rate
6. Cubic growth rate
7. Exponential growth rate

ii) Classes of P and NP

An algorithm is said to be polynomially bounded if its worst-case complexity is bounded by a polynomial function of the input size. A problem is said to be polynomially bounded if there is a polynomially bounded algorithm for it.

P is the class of all decision problems that are polynomially bounded. The implication is that a decision problem X e P can be solved in polynomial time on a deterministic computation model (such as a deterministic Turing machine).

NP represents the class of decision problems which can be solved in polynomial time by a non-deterministic model of computation. That is, a decision problem X e NP can be solved in polynomial-time on a non-deterministic computation model (such as a non-deterministic Turing machine). A non-deterministic model can make the right guesses on every move and race towards the solution much faster than a deterministic model.

iii) NP-Complete problem

This means that the problem can be solved in Polynomial time using a Non-deterministic Turing machine (like a regular Turing machine but also including a non-deterministic "choice" function). Basically, a solution has to be testable in poly time. If that's the case, and a known NP problem can be solved using the given problem with modified input (an NP problem can be reduced to the given problem) then the problem is NP complete.

The main thing to take away from an NP-complete problem is that it cannot be solved in polynomial time in any known way. NP-Hard/NP-Complete is a way of showing that certain classes of problems are not solvable in realistic time.

iv) Subroutine in TM:

TM program for the subroutine is written. This will have an initial state and a return state. After reaching the return state, there is a temporary halt. For using a subroutine, new states are introduced. When there is a need for calling the subroutine, moves are effected to enter the initial state for the subroutine and return to the main program of TM.