Enumeration Machine

Modification of a TM. It has finite control & two tapes, a read/write work tape and a write only output tape.

Work tape head can move in either direction & can read & write any element of Γ.

Output tape head moves right one cell, when it write a symbol and It can only write symbols in Σ.

The machine starts in its start state with both tapes blank. It moves according to its transition function

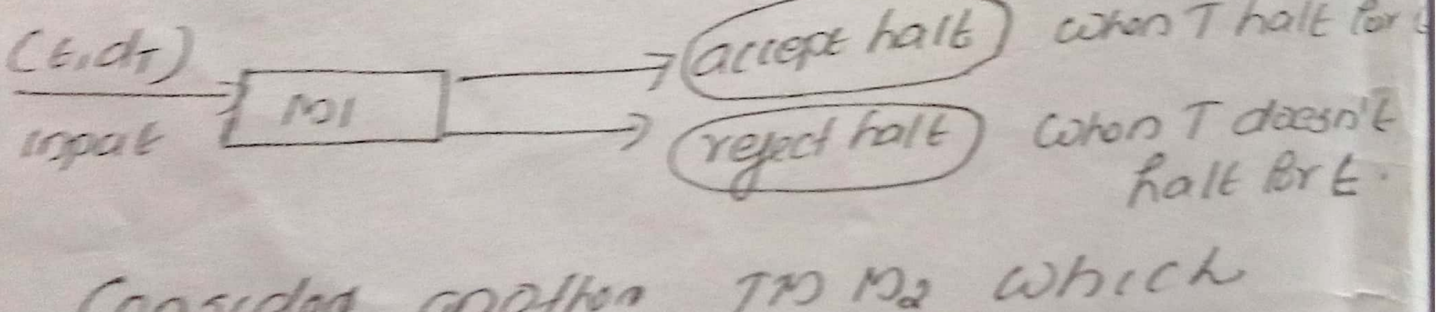
Output tape is automatically erased and the output head moved back to the beginning of the tape and the machine continues from that point. The machine runs forever.

Halting problem

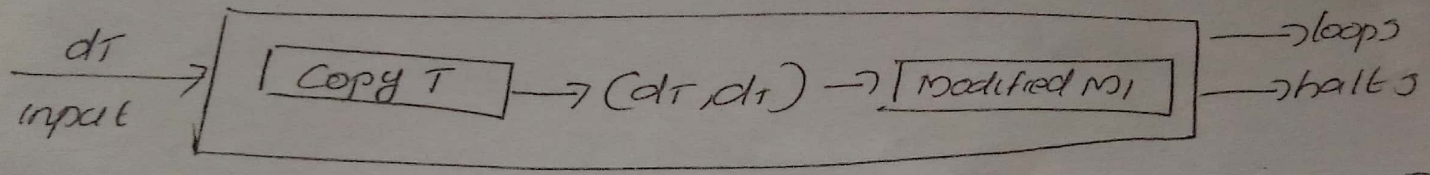
Given any functional matrix, input tape & initial configuration, then is it possible to determine whether the process will ever halt ?. This is called halting pblm.

Halting pblm is <u>unsolvable</u> (undecidable)

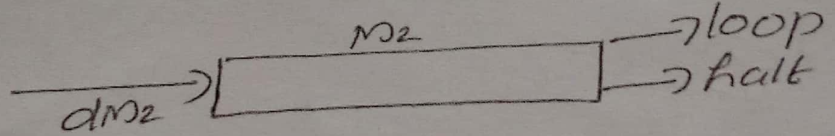
Let there exist a TM M_1 which decides whether or not any computation by a TM T will ever halt when a description d_T of T and tape t of T is given. Then for every input (t, d_T) to M_1 if T halt for input t, M_1 also halt which is called accept halt. Similarly if T doesnot halt for input t, then M_1 will halt which is called reject halt



Consider another TM M_2 which takes an input d_T. It 1st copies d_T and duplicates d_T on its tape and then this duplicated tape information is given as input to machine M_1. But M_1 is a modified machine with the modification that whenever M_1 is supposed to reach an accept halt, M_2 loops forever. Hence behavior of M_2 is as given. It loops if T halts for $T = d_T$ and halt if T doesn't halt for $T = d_T$. The T is any arbitary TM.



As M_2 itself is one TM we will take $M_2 = T$ i.e we will replace T by M_2 from above given machine



Thus machine M_2 halt for d_{M_2} if M_2 doesn't halt for d_{M_2}. This is a contradiction. Hence halting pblm is unsolvable.

Recursive & Recursively Enumerable Languages

A Language L over the alphabet Σ is recursive if there is a TM that accept every word in L & reject every word in L'

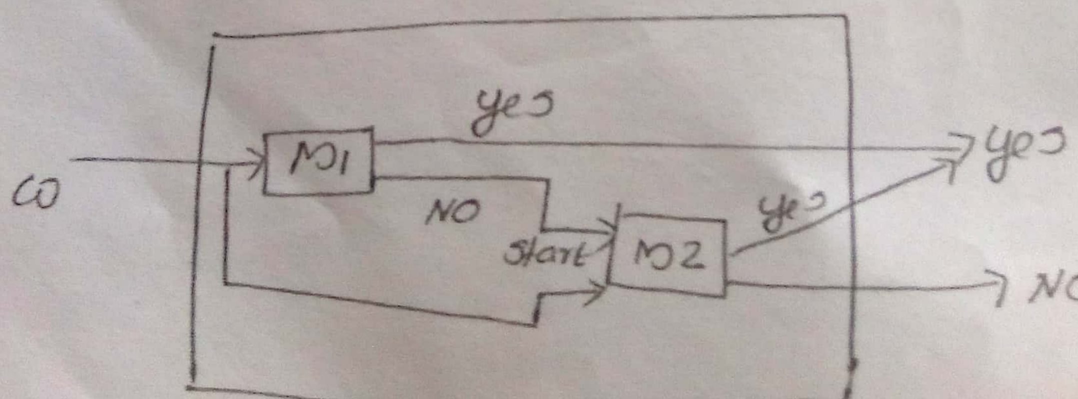
$$Accept(T) = L \quad reject(T) = L' \quad loop(T) =$$

A language L is recursively enumerable if it is accepted by TM & either it rejects or loop forever. For every word in L'

$$Accept(T) = L$$
$$Reject(T) + Loop(T') = L'$$

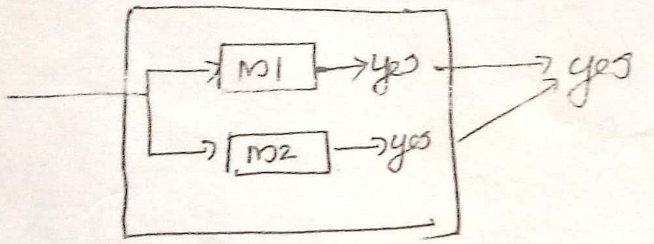
Properties

Union of two recursive language is recursive



Let L_1 and L_2 be recursive languages accepted by M_1 & M_2. We construct 'M' which stimulates 'M_1'. If 'M_1' accepts then M accepts. If M_1 rejects, then M stimulates M_2 and accepts iff M_2 accepts. Since both M_1 & M_2 are algorithms and TM is guaranteed to halt. Clearly M accepts $L_1 \cup L_2$.

Union of two recursively enumerable language is recursively enumerable

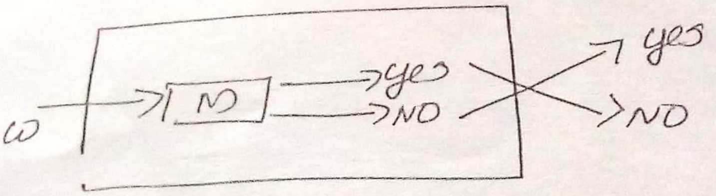


M can be simultaneously stimulate M_1 & M_2 on separate tapes. If either accepts, then 'M' accepts

Complement of a recursive language is recursive

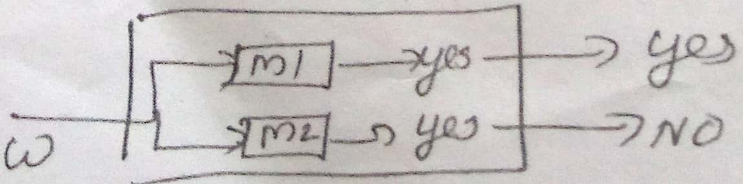
Let L be a recursive language & M a turing machine that halts on all inputs and accept L. Construct M' from

from M so that if M enters final state on input ω, then M' halts without accepting. If M halts without accepting, M' enters a final state. Since one of those two events occurs M' is an algorithm. Clearly L(M') is the complement of L and thus complement of L is a recursive language.



If a language L and its complement L' are both recursively enumerable then L is recursive (hence L')

Let M1 & M2 accepts L & L' resp. Construct M to simulate simultaneously M1 & M2. M accepts ω if M1 accept ω and reject ω if M2 accepts ω. Since ω is in either L or L' we know that exactly one of M1 or M2 will accept. Thus M will always says either yes or no, but will never say both. Since M is an algorithm that accepts L, then L is recursive



Chomsky Hierarchy

4 types of grammer

1] Type 0 grammar (unrestricted grammar)

productions are of the form $\alpha \rightarrow \beta$

$\alpha \in (V \cup T)^{+}$ & $\beta \in (V \cup T)^{*}$

Language generated → recursively enumerable language.

Language recognizer → Turing machine.

eg: $S \rightarrow aAb \mid \epsilon$

$aA \rightarrow bAA$

$bA \rightarrow a$

2] Type 1 grammar (context sensitive)

productions are of the form $\alpha \rightarrow \beta$

where $|\beta| \geq |\alpha|$ α & $\beta \in (V \cup T)^{+}$

Language generated → context sensitive language.

Language recognizer → Linear bounded automata.

eg: $S \rightarrow aAb$

$aA \rightarrow bAA$

$bA \rightarrow aa$

Scanned by CamScanner

3] Type 2 grammar or (Context free grammar)

Productions are of the form $A \to \alpha$
where $\alpha \in (V \cup T)^*$ & $|A| = 1$ & $A \in V$

Language generated → Context free language.

Language recognizer → push down automata

eg: $S \to aB \mid bA \mid \epsilon$
$A \to aA \mid b$
$B \to bB \mid a \mid \epsilon$.

4] Type 3 grammar (regular grammar)

A grammar is said to be type 3 iff the grammar is left linear or right linear.

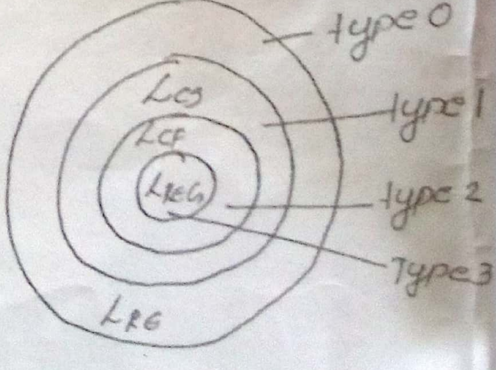
Left linear
$A \to Bw$ or $A \to w$

& Right linear
$A \to wB \mid$ or $A \to w$

Language → regular language.

Language recognizer → finite automata

eg: $S \to aaB \mid bbA \mid \epsilon$
$A \to aA \mid b$
$B \to bB \mid a \mid \epsilon$.



PART E

2017 DEC

Answer any four full questions, each carries 10 marks.

15 State and prove pumping lemma for Context Free Languages. (10)

16 Construct a Turing machine that recognizes the language $L = \{ a^n b^n c^n \mid n>0 \}$ (10)

17 a) What is a Context sensitive grammar(CSG). Design a CSG to accept the language $L = \{ 0^n 1^n 2^n \mid n>0 \}$ (6)

b) Define Linear Bound Automata. (4)

18 a) Write a note on Recursive Enumerable Languages (5)

b) Discuss about Universal Turing Machines. (5)

19 a) Explain Chomsky's Hierarchy of Languages. (6)

b) LetL $= \{x / x \in (a + b + c)^* \text{ and } |x|_a = |x|_b = |x|_c \}$. What class of language does Lbelong? Why? What modification will you suggest in the grammar to accept this language? (4)

20 Discuss the Undecidable Problems About Turing Machines (10)

****

PART E

2018 APRIL

Answer any four full questions, each carries 10 marks

15 a) State pumping Lemma for context free language (5)

b) Define formally Turing machine Model. (5)

16 a) Design Turing machine to accept language $L=\{0^n 1^n \mid n >=1\}$ (6)

b) Describe all instantaneous descriptions (ID) from initial ID $q_0 01$ to Final ID with respect to constructed TM. Assume q_0 as start state. (4)

17 a) Design Turing machine to compute addition of two numbers. Assume unary notation for number representation. (6)

b) Describe all instantaneous descriptions (ID) from initial ID: $q_0 010$ to Final ID: 00 with respect to constructed Turing Machine.(assume q_0 as initial state.) (4)

18 a) Explain the significance of universal Turing machine. (5)

b) Compare and contrast recursive and recursively enumerable languages. (5)

19 a) Prove that union of two recursive languages is recursive. (5)

b) Explain the significance of halting problem. (5)

20 a) Explain general notations for productions of each formal language from Chomsky hierarchy. (5)

b) Prove that complement of a recursive language is recursive. (5)

****

PART E

2018 DeC

Answer any four full questions, each carries 10 marks.

15 a) Consider $L = \{ww \mid w \in \{0, 1\}^*\}$. Prove L is not a CFL. (5)

b) Explain Chomsky hierarchy and corresponding type0, type1, type2 and type 3 formalism. (5)

16 a) Design a Turing machine that determines whether the binary input string is of odd parity or not (5)

b) How does the Universal Turing machine simulate other Turing machines? (5)

17 a) Design a Turing machine that accepts $a^n b^m$ where n>0 and m>n. (5)

b) Explain why Halting problem is unsolvable problem. (5)

18 a) What is the instantaneous description for a Turing machine? Explain with an example. (5)

b) Show that normal single tape Turing machine can perform computations performed by multi-tape Turing machine (informal explanation is sufficient). (5)

19 a) What is a recursive language? Give an example. (5)

b) How does a Turing machine differ from PDA and FSA? (5)

20 a) State pumping lemma for CFL. Mention one application of Pumping lemma (5)

b) What is a non-deterministic Turing machine? (5)

****