A Language that is not Recursively Enumerable?

Inorder to prove that a language that is not RE, follow the steps

1) Codes for Turing Machine.
2) Diagonalization Language.
3) L_d is not Recursively Enumerable.

1) Codes for TM?

If w is the binary string, treat $1w$ as a binary integer i.
Then we shall call w the ith string.

ε – first string
0 – second
1 – Third
00 – Fourth
01 – fifth.

To represent a TM $M = \{Q, \{0,1\}, \Gamma, \delta, q_1, B, F\}$ as a binary string, we must first assign integers to the states, tape sym, & directions L & R.

- We shall assume the states are $q_1, q_2 \ldots q_k$ for some k.
  The start state will be q_1 & q_2 will be the only accepting state.
- We shall assume tape symbols $X_1 X_2 \ldots X_m$ for some m.

$X_1 \longrightarrow 0$
$X_2 \longrightarrow 1$
$X_3 \longrightarrow B$

Other tape symbols can be assigned to the remaining integers.

- We shall refer to direction L as D_1 & direction R as D_2
  $L \longrightarrow 0$ $R \longrightarrow 00$

$$\delta(q_i, X_j) = (q_k, X_l, D_m)$$

We shall code as $0^i 1 0^j 1 0^k 1 0^l 1 0^m$.

Scanned by CamScanner

A code for the entire TM M consists of all the codes for the transitions in some order, seperated by pairs of 1's

$$C_1 11 C_2 11 \ldots C_{n-1} 11 C_n$$

where each of the c's is the code for one transistion of M.

Ex:- $M = (\{q_1, q_2, q_3, q_4, q_5, q_6\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \{q_2\})$

where δ consists of the rules.

$\delta(q_1, B) = (q_6, B, R)$

$\delta(q_6, 1) = (q_6, 1, R)$

$\delta(q_6, B) = (q_3, B, L)$

$\delta(q_3, 1) = (q_4, B, L)$

$\delta(q_4, 1) = (q_3, B, L)$

$\delta(q_4, B) = (q_5, B, R)$

$\delta(q_5, B) = (q_2, 1, L)$

Tape Symbols $\begin{cases} 0 - 0 \\ 1 - 00 \\ B = X_3 \ (000) \end{cases}$

$R = D_2 \ (00)$

$L = D_1 \ (0)$

The codes for the seven transitions may be listed in any order, giving us 5040 codes for M.

Diagonalization Language:-

The language L_d, the diagonalization language, is the set of strings w_i such that w_i is not in $L(M_i)$

* L_d consists of all strings w such that the TM M does not accept w as input

We can construct a table for all i & j in which TM M_i accepts input string w_j;

1 means "Yes it does"

0 means "No it doesn't".

$i \downarrow$ \ $j \rightarrow$	1	2	3	4	...
1	0	1	1	1	...
2	1	1	0	0	...
3	0	0	1	1	...
4	0	1	0	1	...
⋮	⋮	⋮	⋮	⋮	⋱

To construct L_d, we complement the diagonal, From the figure diagonal values are 0 1 1 1

Complemented Values 1 0 0 0

The complementented diagonal value "1 0 0 0" will not fallen any where in the table. That means this is not accepted by TM which means it is Not Recursive Enumerable

It works because the complement of the diagonal is itself a characteristic vector describing membership in some language L_d.

The complement of the diagonal cannot be the characteristic vector of any TM.

Universal Language

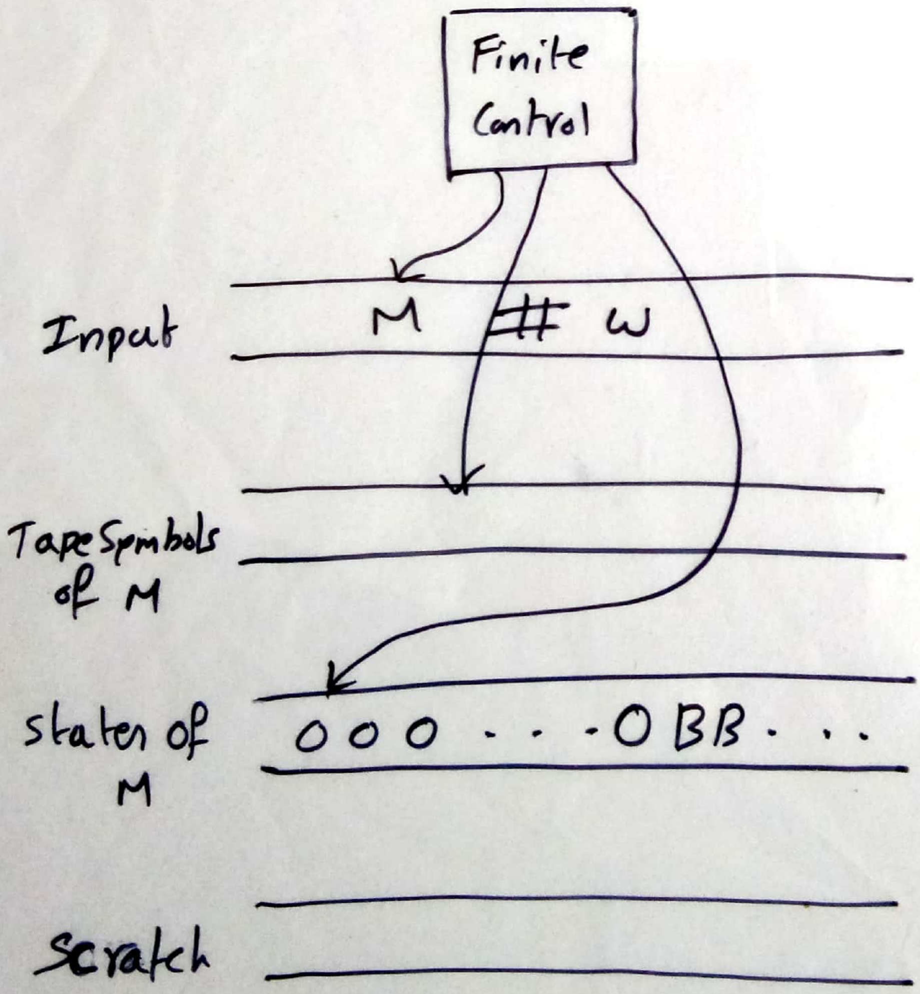
An Undecidable Problem That is RE

Definition :-

L_u is the set of strings representing a TM and an input accepted by that TM. There is a TM U, often called the universal Turing Machine, such that $L_u = L(U)$.

It is easiest to describe U as a multitape TM.

1. First tape holds $\langle M, w \rangle$, $M \rightarrow TM$ & w is the input string
2. Second tape holds tape symbols. of M.
3. Third tape holds the state of M



The operation U can be summarized as follows.

1) Examine the input to make sure that the code for M is legitimate code for some TM. If not, U halts without accepting.
2) Initialize the second tape to contain the input ω, in its encoded form.
   zero (0) $\rightarrow$ 10
   One (1) $\rightarrow$ 100
   Blank (B) $\rightarrow$ 1000
3) Place 0, the start state of M, on the third tape, and move the head of U's second tape to the first simulated cell.
4) To simulate a move of M, U searches on its first tape, for a transition $0^i 1 0^j 1 0^k 1 0^l 1 0^m \Rightarrow (q_i, j) = (q_k, l, m)$ such that 0^i is the state placed in tape 3.
   0^j is the tape symbol placed in tape 2.

The transition can be done by U as follows.
a) Change the contents of tape 3 to 0^k. i.e state change
b) Replace 0^j on tape 2 by 0^l i.e Modify tape symbol.
c) Move the head on tape 2 to the position L (or) R
5) If M has no transitions that matches the simulated state & tape symbol then in (4) no transition will be found. Thus M halts in the simulated configuration. & U must do like wise.
6) If M enters its accepting state, then U accepts.

In this manner, U simulates M on ω. U accepts the coded pair (M, ω) if and only if M accepts ω.