Exercise: (Section 5.3.2 from textbook) 12/1/2005 E 5.1

Consider the following languages over $\Sigma = \{0,1\}$ 11/1/2006

$$L_e = \{ \mathcal{E}(M) \mid \mathcal{L}(M) = \emptyset \}$$

$$L_{ne} = \{ \mathcal{E}(M) \mid \mathcal{L}(M) \neq \emptyset \}$$

Hence: L_e ... set of all strings that encode T.M.s that accept the empty language

L_{ne} ... complement of L_e

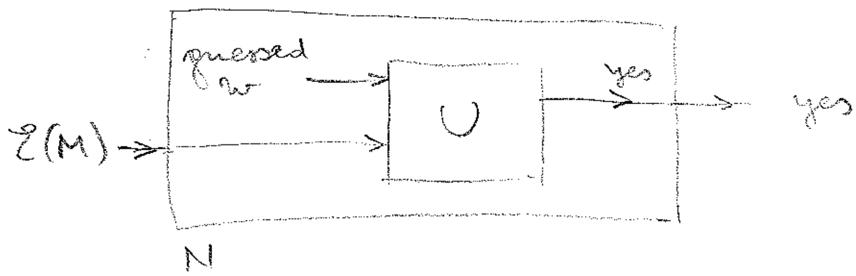
Claim 1: L_{ne} is R.E.

Proof: construct NTM N for L_{ne}

(and then convert N to an ordinary T.M.)

N works as follows: on input $\mathcal{E}(M)$

1) guess a string $w \in \Sigma^*$
2) simulate M on w (like a UTM)
3) accept $\mathcal{E}(M)$ if M accepts w



We have $\mathcal{E}(M) \in \mathcal{L}(N) \iff \exists w \text{ s.t. } \langle M, w\rangle \in \mathcal{L}(U)$

$\iff \exists w \text{ s.t. } w \in \mathcal{L}(M)$

$\iff \mathcal{E}(M) \in L_{ne}$

Claim 2 : L_{ne} is non-recursive 11/11/2006 E5.2

Proof : by reduction from L_u to L_{ne}

Reduction R is a function computable by a halting T.M.

with input : instance $\langle M, w \rangle$ of L_u

output : instance $\mathcal{E}(M')$ of L_{ne}

and s.t. $\langle M, w \rangle \in L_u \iff \mathcal{E}(M') \in L_{ne}$

Description of M':

- M' ignores completely its own input string x
- instead, it replaces its input by the string $\langle M, w \rangle$ and simulates M on w using UTM.
- if M accepts w, then M' accepts x
  if M never halts on w or rejects w, then M' also never halts or rejects x

Note : if $w \in \mathcal{L}(M) \Rightarrow \mathcal{L}(M') = \Sigma^*$

if $w \notin \mathcal{L}(M) \Rightarrow \mathcal{L}(M') = \emptyset$

hence $\langle M, w \rangle \in L_u \iff \mathcal{E}(M') \in L_{ne}$

We can construct a halting T.M. M_R that, given $\langle M, w \rangle$ as input, constructs $\mathcal{E}(M')$ for an M' that behaves as above.

q.e.d.

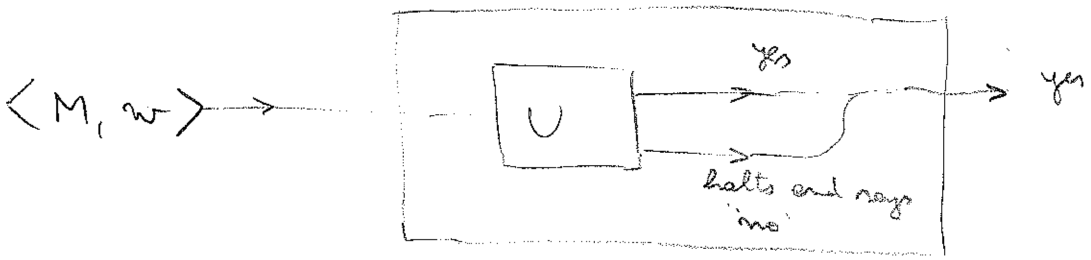
To sum up, we have that L_{ne} is R.E. but non-recursive.

Hence L_e must be non-R.E.

Exercise 9.2.1 11/1/2006 E5.3

The halting problem L_H i.e., the set $\langle M, w\rangle$ s.t. M halts on w (with or without accepting) is R.E. but not recursive.

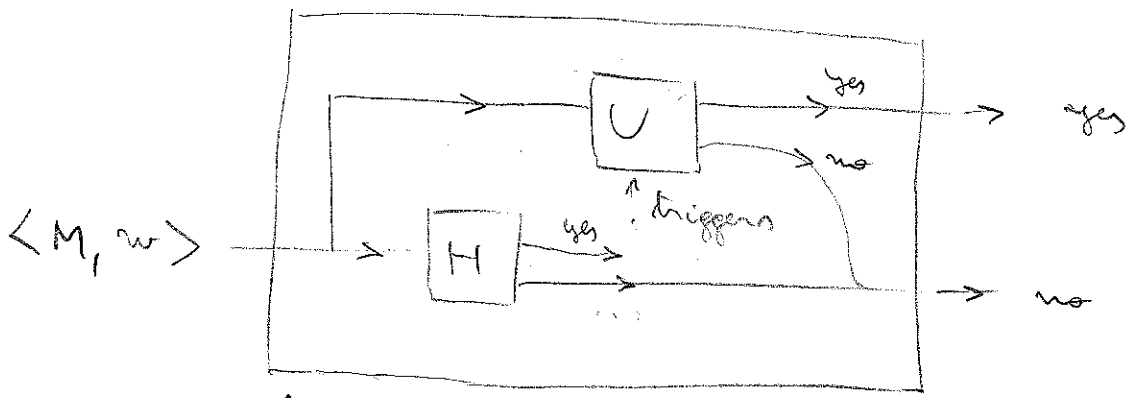
To show R.E., we construct a T.M. H s.t.
$\mathcal{L}(H) := L_H = \{\langle M, w\rangle \mid M \text{ halts on } w\}$



H

To show that L_H is not recursive, we assume by contradiction it is so, and derive that L_u is recursive.

By contradiction, let H be an algorithm for L_H and U a procedure for L_u



A_u

A_u would be an algorithm for L_u.
Contradiction

Exercise 8.2.5

Let L be R.E. and $\overline{L}$ be non-R.E.

Consider $L' = \{0w \mid w \in L\} \cup \{1w \mid w \notin L\}$

What do we know about L' and $\overline{L'}$?

We show that L' is non-R.E.

Suppose by contradiction that we have a procedure $M_{L'}$ for L'.

Then we can construct a procedure $M_{\overline{L}}$ for $\overline{L}$ as follows

- on input w, $M_{\overline{L}}$ changes the input to $1w$ and simulates $M_{L'}$.
- if $M_{L'}$ accepts $1w$, then $w \in \overline{L}$, and $M_{\overline{L}}$ accepts
- if $M_{L'}$ does not terminate or terminates and answers no, then $w \notin \overline{L}$, and $M_{\overline{L}}$ does not terminate or terminates and answers no.

$\Rightarrow$ $M_{\overline{L}}$ would accept exactly $\overline{L}$. Contradiction

$\overline{L'} = \{0w \mid w \notin L\} \cup \{1w \mid w \in L\} \cup \{\varepsilon\}$

Reasoning as for L', we get that $\overline{L'}$ is non-R.E.

Exercise 9.3.7 a)

$\overline{H}$, the complement of the halting problem, i.e., the set of pairs $\langle M, w \rangle$ such that M on input w does not halt, is non-R.E.

Proof: By reduction from $\overline{L_u}$, which is non-R.E.

Idea: we show how to convert any TM M into another TM M_H s.t. M_H halts on w iff M accepts w.

Construction:

1) Ensure that M_H does not halt unless M accepts.
   - add to the states of M a new loop state p, with $\delta(p, x) = (p, x, R)$ for all $x \in \Gamma$
   - for each $\delta(q, y)$ that is undefined and $q \notin F$, add $\delta(q, y) = (p, y, R)$

2) Ensure that, if M accepts, then M_H halts
   - make $\delta(q, x)$ undefined for all $q \in F$ and $x \in \Gamma$

3) The other moves of M_H are as those of M

q.e.d.