Exercise: (Section 3.3.2 from textbook)

Consider the following languages over \( \Sigma = \{0, 1\} \):

\[
L_e = \{ \varepsilon(M) \mid \mathcal{L}(M) = \emptyset \}
\]

\[
L_{\text{ne}} = \{ \varepsilon(M) \mid \mathcal{L}(M) \neq \emptyset \}
\]

**Define:** \( L_e \) as the set of all strings that encode T.M.s that accept the empty language.

\( L_{\text{ne}} \) as complement of \( L_e \)

**Claim 1:** \( L_{\text{ne}} \) is R.E.

**Proof:** Construct NTM \( N \) for \( L_{\text{ne}} \)

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

\( N \) works as follows: on input \( \varepsilon(M) \)

1) Guess a string \( w \in \Sigma^* \)

2) Simulate \( M \) on \( w \) (like a UTM)

3) Accept \( \varepsilon(M) \) if \( M \) accepts \( w \)

We have

\[
\varepsilon(M) \in \varepsilon(N) \iff \exists w \text{ s.t. } \langle \varepsilon(M), w \rangle \in \mathcal{L}(U)
\]

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

\[
\iff \varepsilon(M) \in L_{\text{ne}}
\]
Claim 2: \( L_{ne} \) is non-recursive

Proof: by reduction from \( L_{ne} \) to \( L_{ne} \)

Reduction \( R \) is a function computable by a halting T.M.

with input: instance \( \langle \Sigma(M), w \rangle \) of \( L_{ne} \)

output: instance \( \Sigma(M') \) of \( L_{ne} \)

end. s.t.: \( \langle \Sigma(M), w \rangle \in L_{ne} \iff \Sigma(M') \in L_{ne} \)

Description of \( M' \):

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

Note: if \( w \notin \Sigma(M) \Rightarrow L(M') = \Sigma^* \)
  if \( w \notin \Sigma(M) \Rightarrow L(M') = \emptyset \)

hence \( \langle \Sigma(M), w \rangle \in L_{ne} \iff \Sigma(M') \in L_{ne} \)

We can construct a halting T.M. \( M_R \) that, given \( \langle \Sigma(M), w \rangle \) as input, constructs \( \Sigma(M') \) 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: 3.2.1

The halting problem, \( L_{halt} \), i.e., the set \(<E(M), w> \) 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.
\( L(H) = \{ <E(M), w> \mid M \) halts on \( w \}\}

\[
\begin{array}{ccc}
\langle E(M), w \rangle & \rightarrow & y \\
& | & \\
& U & \\
& | & \\
& y & \rightarrow y \\
& | & \\
& halt & \\
& | & \\
& no & \rightarrow no
\end{array}
\]

To show that \( L_H \) is not recursive, we assume by contradiction
it is so, and derive that \( L_{halt} \) is recursive.

By contradiction, let \( H \) be an algorithm for \( L_H \) and
\( U \) a procedure for \( L_{halt} \)

\[
\begin{array}{ccc}
\langle E(M), w \rangle & \rightarrow & y \\
& | & \\
& U & \\
& | & \\
& y & \rightarrow y \\
& | & \\
& \text{triggers} & \\
& | & \\
& no & \rightarrow no
\end{array}
\]

\( A_m \)

\( A_m \) would be an algorithm for \( L_{halt} \)

Contradiction
Let $L$ be R.E. and $\overline{L}$ be non-R.E.
Consider $L' = \{ow | w \in L\} \\
\{ow | w \notin L\}$

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 $ow$ and simulates $M_L$.
  - If $M_L$ accepts $ow$, then $w \in L$, and $M_{\overline{L}}$ accepts.
  - If $M_L$ does not terminate or terminates and answers no, then $w \notin L$, and $M_{\overline{L}}$ does not terminate or terminates and answers no.

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

$\overline{L} = \{ow | w \in L\} \cup \{ow | w \notin L\} \cup \{\varepsilon\}$

Reasoning as for $L'$, we get that $\overline{L}'$ is non-R.E.
F, the complement of the halting problem, i.e., the set of pairs \(<E(M), w>\) such that M on input w does not halt, is non-R.E.

Proof: By reduction from \(E_M\), which is non-R.E.

Idea: we show how to convert any TM \(M\) into another TM \(M^*_k\) such: \(M^*_k\) halts on \(w\) iff \(M\) accepts \(w\).

Construction:
1) Ensure that \(M^*_k\) does not halt unless \(M\) accepts.
   - add to the states of \(M\) a new loop state \(q\), with
     \(\delta(q, \epsilon) = (q, \epsilon, \epsilon)\) for all \(\epsilon \in \Sigma\)
   - for each \(\delta(q, y)\) that is undefined and \(q \in F\),
     add \(\delta(q, y) = (q, y, \epsilon)\)

2) Ensure that, if \(M\) accepts, then \(M^*_k\) halts
   - make \(\delta(q, \epsilon)\) undefined for all \(q \in F\) and \(\epsilon \in \Sigma\)

3) The other moves of \(M^*_k\) are as those of \(M\).

\(\square\)