Exercise: (Section 5.3.2 from textbook)
Consider the following languages over $\Sigma = \{0, 1\}$

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

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

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

$L_{\text{ac}}$ is the complement of $L_e$.

Claim 1: $L_{\text{ac}}$ is R.E.

Proof: construct N.T.M. $N$ for $L_{\text{ac}}$

(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 U.T.M.)

3) Accept $\mathcal{E}(M)$ if $M$ accepts $w$

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

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

$\iff \mathcal{E}(M) \in L_{\text{ac}}$
Claim 2: \( L_{\text{re}} \) is non-recursively

Proof: by reduction from \( L_{\text{m}} \) to \( L_{\text{re}} \)

Reduction \( R \) is a function computable by a halting T.M. 
with input: instance \( < M, w > \) of \( L_{\text{m}} \)
output: instance \( \varepsilon(M') \) of \( L_{\text{re}} \)
eq \text{set}: \( < M, w > \in L_{\text{m}} \iff \varepsilon(M') \in L_{\text{re}} \)

Description of \( M' \):
- \( M' \) ignores completely its own input string \( X \)
- instead, it replaces its input by the string \( < M, w > \), and runs \( M \) on \( w \) (see (*) below)
- if \( M \) accepts \( w \), then \( M' \) accepts \( X \)
- if \( M \) never halts on \( w \) or rejects \( w \),
  then \( M' \) also never halts on \( w \) or rejects \( X \)

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

hence \( < M, w > \in L_{\text{m}} \iff \varepsilon(M') \in L_{\text{re}} \)

We can construct a halting T.M. \( M_R \) that, given \( < M, w > \) as input, constructs \( \varepsilon(M') \) for an \( M' \) that behaves as above.

\( \text{q.e.d.} \)

(*) \( M' \) has the following form: (let \( w = a_1, \ldots, a_n \))

\[ M \]

To sum up, we have that \( L_{\text{re}} \) is RE but non-recursive.
Hence \( L_{\text{re}} \) must be non-RE.
Exercise 3.2.1

The halting problem, \( L_{H_{1}} \), is the set \( \langle M, w \rangle \) s.t.
\( M \) halts on \( w \) (with or without accepting) in R.E.
but not recursive.

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

\[ \langle M, w \rangle \rightarrow \text{yes} \]
\[ H \]
\[ \text{halts and says no} \]

To show that \( L_{H} \) is not recursive, we assume by contradiction
\( \text{it is R.E.} \), and derive that \( L_{M} \) is recursive.

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

\[ \langle M, w \rangle \rightarrow \text{yes} \rightarrow \text{yes} \]
\[ U \]
\[ \text{triggers} \]
\[ \rightarrow \text{no} \rightarrow \text{no} \]
\[ A_{m} \]

\( A_{m} \) would be an algorithm for \( L_{M} \).

\text{Contradiction}
Let \( L \) be R.E. and \( \bar{L} \) be non-R.E.

Consider \( L' = \{0w \mid w \in L\} \cup \{1w \mid w \notin L\} \).

What do we know about \( L' \) and \( \bar{L}' \)?

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 \( M_{\bar{L}} \) for \( \bar{L} \) as follows:

- on input \( w \), \( M_{\bar{L}} \) changes the input to \( 1w \) and simulates \( M_L \).

- if \( M_L \) accepts \( 1w \), then \( w \notin L \), and \( M_{\bar{L}} \) accepts.
- if \( M_L \) does not terminate or terminates and answers no, then \( w \notin L \), and \( M_{\bar{L}} \) does not terminate or terminates and answers no.

\[ \Rightarrow M_{\bar{L}} \] would accept exactly \( \bar{L} \). Contradiction.

\[ \bar{L}' = \{0w \mid w \in L\} \cup \{1w \mid w \in L\} \cup \{\varepsilon\} \]

Reversing as for \( L' \), we get that \( \bar{L}' \) is non-R.E.
Fl, 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 \( E_n \), 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 \( q \), with \( \delta(q, x) = (q, x, r) \) for all \( x \in \Gamma \).
   - For each \( \delta(q', y) \) that is undefined and \( q' \in F \), add \( \delta(q', y) = (q', y, r) \).

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 \( M \).

\( \text{q.e.d.} \)