Problem 7.1. (30 points) Consider the following four languages:

- $L_1 = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \subseteq \{00, 11\}\}$
- $L_2 = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \{00, 11\}\}$
- $L_3 = \{\langle M, q \rangle \mid \text{TM } M \text{ visits state } q \text{ on some input }\}$
- $L_4 = \{\langle M, q \rangle \mid \text{TM } M \text{ visits state } q \text{ on some input within 1000 steps}\}$

Consider the following four mutually exclusive statements about a language $L$:

- **S1** The language $L$ is recursive
- **S2** The language $L$ is r.e., but not recursive.
- **S3** The language $L$ is co-r.e., but not recursive.
- **S4** The language $L$ is not r.e., nor co-r.e.

You are asked to determine for each language $L_1$ to $L_4$ which one of the statements **S1** to **S4** is true. You need to justify your answers as follows:

- To justify **S1**, give a high-level description of a Turing decider that accepts $L$.
- To justify **S2**, give (i) a high-level description of a Turing recognizer that accepts $L$, and (ii) a mapping reduction from either $A_{TM}$ or $E_{TM}$ to $L$.
- To justify **S3**, give (i) a high-level description of a Turing recognizer that accepts the complement of $L$ and (ii) a mapping reduction from $\overline{A}_{TM}$ or $E_{TM}$ to $L$.
- To justify **S4**, give (i) a mapping reduction from either $A_{TM}$ or $E_{TM}$ to $L$ and (ii) a mapping reduction from $\overline{A}_{TM}$ or $E_{TM}$ to $L$.

Solution:

1. $L_1$ is co-r.e., but not recursive (S2).
   - $L_1$ is co-r.e.
     
     **Proof**: Here is a Turing Machine that accepts the complement of $L_1$, where the complement of $L_1$ is $\overline{L_1} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \not\subseteq \{00, 11\}\}$:

     On input $\langle M \rangle$,
     - Let $f : \mathbb{N} \rightarrow \Sigma^*$ be a function that enumerates all words in $\Sigma^*$.
     - For $j = 0, 1, 2, \ldots$,
       - For $i = 0$ to $j$,
         - Simulate $M$ on $f(i)$ for $j$ steps,
         - if $\langle M \rangle$ accepts and $f(i) \not\in \{00, 10\}$ then ACCEPT.
• \(L_1\) is not recursive.

Proof: \(\overline{A_{TM}} \leq_m L_1\).

Given \(\langle M, w \rangle\) construct \(\langle M' \rangle\) such that \(\langle M, w \rangle \in \overline{A_{TM}}\) iff \(\langle M' \rangle \in L_1\).

That is, \(w \notin L(M)\) iff \(M'\) accepts at most the words \(\{00, 11\}\). \(M'\) behaves as follows:

On input \(x\),
- If \(x = 00\) or \(x = 11\) then ACCEPT.
- Else if \(x = 111\) then,
  - Simulate \(M\) on \(w\). If \(M\) accepts then ACCEPT.
  - Otherwise REJECT.

We need to prove that \(w \notin L(M)\) iff \(M'\) accepts at most the words \(\{00, 11\}\).

- \(\Rightarrow\)
  If \(w \notin L(M)\) the language of \(M'\) is \(\{00, 11\}\) by construction, so \(M'\) accepts exactly the words \(\{00, 11\}\). Thus, \(L(M') = \{00, 11\}\) and \(M' \in L_1\).

- \(\Leftarrow\)
  Proving this is equivalent with proving that if \(w \in L(M)\) then \(M'\) accepts a word that is not in \(\{00, 11\}\). If \(w \in L(M)\) the language of \(M'\) is \(\{00, 11, 111\}\) by construction. So \(M'\) also accepts the word \(111 \notin \{00, 11\}\), and therefore \(M' \notin L_1\).

2. \(L_2\) is not r.e, nor co-r.e. (S4)

• \(L_2\) is not r.e.

Proof: \(\overline{A_{TM}} \leq_m L_2\).

Given \(\langle M, w \rangle\) construct \(\langle M' \rangle\) such that \(\langle M, w \rangle \in \overline{A_{TM}}\) iff \(\langle M' \rangle \in L_2\).

That is, \(w \notin L(M)\) iff \(M'\) accepts exactly the two words \(\{00, 11\}\). \(M'\) behaves as follows:

On input \(x\),
- If \(x = 00\) or \(x = 11\) then ACCEPT.
- Else if \(x = 111\) then,
  - Simulate \(M\) on \(w\). If \(M\) accepts then ACCEPT.
  - Otherwise REJECT.

We need to prove that \(w \notin L(M)\) iff \(M'\) accepts exactly the two words \(\{00, 11\}\).

- \(\Rightarrow\)
  If \(w \notin L(M)\) the language of \(M'\) is \(\{00, 11\}\) by construction, so \(M'\) accepts exactly \(\{00, 11\}\).

- \(\Leftarrow\)
  If \(w \in L(M)\) the language of \(M'\) is \(\{00, 11, 111\}\) by construction. So \(M'\) accepts the three words \(\{00, 11, 111\}\), and not only the two words \(\{00, 11\}\).

• \(L_2\) is not co-r.e.

Proof: \(A_{TM} \leq_m L_2\).

Given \(\langle M, w \rangle\) construct \(\langle M' \rangle\) such that \(\langle M, w \rangle \in A_{TM}\) iff \(\langle M' \rangle \in L_2\).

That is, \(w \in L(M)\) iff \(M'\) accepts exactly the two words \(\{00, 11\}\). \(M'\) behaves as follows:

On input \(x\),
- If \(x = 00\) then ACCEPT.
- Else if \(x = 11\) then
– Simulate \( M \) on \( w \). If \( M \) accepts then \textbf{ACCEPT}.
– Otherwise \textbf{REJECT}.

We need to prove that \( w \in L(M) \) iff \( M' \) accepts exactly the two words \{00, 11\}.

– ’\( \Rightarrow \)’
  If \( w \in L(M) \) the language of \( M' \) is \{00, 11\} by construction, so \( M' \) accepts exactly the two words \{00, 11\}.

– ’\( \Leftarrow \)’
  If \( w \notin L(M) \) the language of \( M' \) is \{0\} by construction, so \( M' \) accepts only the word 00, and not both words 00 and 11.

3. \( L_3 \) is r.e., but not recursive. \( \text{(S2)} \)

  - \( L_3 \) is r.e.
    
    \textit{Proof:} Here is a Turing Machine that accepts \( L_3 \)

    On input \( \langle M, q \rangle \)
    – Let \( f : \mathbb{N} \rightarrow \Sigma^* \) be a function that enumerates all words in \( \Sigma^* \).
    – For all \( j = 0, 1, 2, \ldots \),
      – For all \( i = 0 \) to \( j \),
        – Simulate \( M \) on \( f(i) \) for \( j \) steps,
        – if \( M \) has visited \( q \) then \textbf{ACCEPT}.
      – otherwise continue.

  - \( L_3 \) is not recursive.
    
    \textit{Proof:} \( \overline{E_{TM}} \leq_m L_3 \).

    Given \( \langle M \rangle \) construct \( \langle M', q' \rangle \) such that \( \langle M \rangle \in \overline{E_{TM}} \) iff \( \langle M', q' \rangle \in L_3 \).

    That is, \( L(M) \) is not empty iff \( M' \) visits \( q' \) on some input. We choose \( M' = M \) and \( q' = q_a \), where \( q_a \) is the accepting state of \( M \).

    We need to prove that \( L(M) \) is not empty iff \( M' \) visits \( q' \) on some input.

    – ’\( \Rightarrow \)’ Let \( w \in L(M) \). This word exists as \( L(M) \) is not empty. When running \( w \) on \( M' \) the state \( q' \) will be visited as \( q' \) is the accepting state of \( M' \).
    – ’\( \Leftarrow \)’ The word on which \( M' \) visits \( q' \) is accepted so \( L(M') \) is not empty. As \( M' = M \), it follows that \( L(M) \) is not empty.

4. \( L_4 \) is recursive. \( \text{(S1)} \)

  \textit{Proof:} Here is a Turing Decider that accepts \( L_4 \)

  On input: \( \langle M, q \rangle \)
  – Let \( f : \mathbb{N} \rightarrow \Sigma^* \) be a function that enumerates all words in \( \Sigma^* \) of length at most 1000.
  – For \( i = 0, 1, \ldots \),
    – Simulate \( M \) on \( f(i) \) for 1000 steps,
    – if \( M \) visits \( q \) then \textbf{ACCEPT}.
    – otherwise continue.
  
  - \textbf{Reject}.