11.1 Infinite sets. Two sets \(A\) and \(B\) have the same size if there exists a bijective (i.e., one-to-one and onto) function \(f: A \rightarrow B\). A set \(A\) is countable if it is finite or it has the same size as the natural numbers \(\mathbb{N}\); in this case, the bijection \(f: \mathbb{N} \rightarrow A\) is called an enumeration of \(A\). Here are some infinite sets:

1. The set \(\Sigma^*\) of finite words is countable for every finite alphabet \(\Sigma\) (proof by enumeration).
2. The set of infinite words over \(\{0, 1\}\) is uncountable (proof by diagonalization).
3. The set of TMs is countable (proof by enumeration).
4. The set of languages over \(\{0, 1\}\) is uncountable (proof by diagonalization).

It follows that there are languages that are not r.e.

11.2 Two decision problems. We consider the following two languages associated with decision problems about Turing machines:

- **Membership**: \(A_{TM} = \{(M, w) \mid M\text{ is a DTM and } w \in L(M)\}\).
- **Emptiness**: \(E_{TM} = \{(M) \mid M\text{ is a DTM and } L(M) = \emptyset\}\).

The complementary languages/problems are:

- **NonMembership**: \(\overline{A}_{TM} = \{(M, w) \mid M\text{ is a DTM and } w \notin L(M)\}\).
- **NonEmptiness**: \(\overline{E}_{TM} = \{(M) \mid M\text{ is a DTM and } L(M) \neq \emptyset\}\).

11.3 TM membership is r.e.: The universal Turing machine. Here is a high-level description of a Turing machine \(M_{universal}\) which accepts \(A_{TM}\):

- Input: \(\langle M, w \rangle\), where \(M\) is a DTM.
- Simulate \(M\) on input \(w\) until
  1. \(M\) accepts \(w\) (then ACCEPT), or
  2. \(M\) rejects \(w\) (then REJECT).

Note that if \(M\) loops on \(w\), then so does \(M_{universal}\). It follows that \(A_{TM}\) is r.e.

11.4 TM emptiness is co-r.e. We now argue that \(E_{TM}\) is co-r.e. Here is a high-level description of a Turing machine \(M_{TMemptiness}\) which accepts \(\overline{E}_{TM}\):

- Input: \(\langle M \rangle\), where \(M\) is a TM.
- Let \(f: \mathbb{N} \rightarrow \Sigma^*\) be an enumeration of all words in \(\Sigma^*\).
- For \(j = 0, 1, 2, \ldots\) do
  - for \(i = 0\) to \(j\) do
    - if \(M\) accepts \(f(i)\) in \(j\) steps then ACCEPT.

Note that if \(L(M) = \emptyset\), then \(M_{TMemptiness}\) loops. Note also that while \(A_{TM}\) is not recursive, it can be decided if a TM \(M\) accepts an input \(w\) in a given number \(j\) of steps.

11.5 TM Membership is not recursive: Diagonalization. We show that \(A_{TM}\) is not recursive. It follows that also \(\overline{A}_{TM}\) is not recursive, but while \(A_{TM}\) is r.e., \(\overline{A}_{TM}\) is co-r.e. To see that \(A_{TM}\) is not recursive, we assume that there is a Turing decider \(H\) that accepts \(A_{TM}\), and derive a contradiction. (Since the existence of \(H\) will be our only assumption and leads to a contradiction, such an \(H\) cannot exist.) From \(H\) we construct another Turing decider \(D\), with the following high-level description:
Input: \( w \).
Duplicate the input so that \( w\#w \) is on the tape.
If \( w\#w \in L(H) \) then \text{REJECT} else \text{ACCEPT}.

Note that \( D \) uses \( H \) as a subroutine, which is possible because \( H \) never loops. Now consider how \( D \) behaves on input \( w = \langle D \rangle \), i.e., the input to \( D \) is an encoding of \( D \) itself. If \( \langle D \rangle\#\langle D \rangle \in L(H) \) then \( D \) rejects, i.e., \( \langle D \rangle \notin L(D) \); if \( \langle D \rangle\#\langle D \rangle \notin L(H) \), then \( D \) accepts, i.e., \( \langle D \rangle \in L(D) \). But this means that \( H \) does not accept \( A_{TM} \), a contradiction.

11.6 Reductions. We used diagonalization to show that the membership problem for DTM \( s \) is not recursive. Since it is r.e., it cannot be co-r.e. (why?). Hence we have a non-co-r.e. problem (DTM membership), and a non-r.e. problem (DTM non-membership). From these, we can prove other problems non-co-r.e., respectively non-r.e., by a fundamental technique called reduction. A function \( f: \Sigma^* \rightarrow \Sigma^* \) is computable if there exists a Turing decider that accepts all input words \( w \), and when entering \( q_a \), has \( f(w) \) on the tape. For two languages \( A, B \subseteq \Sigma^* \), we say that \( A \) mapping reduces to \( B \), written \( A \leq_m B \), if there exists a computable function \( f: \Sigma^* \rightarrow \Sigma^* \) such that for all \( w \in \Sigma^* \), we have \( w \in A \) iff \( f(w) \in B \). If \( A \leq_m B \), then:

1. if \( B \) recursive, then \( A \) recursive.
2. if \( B \) r.e., then \( A \) r.e.
3. if \( B \) co-r.e., then \( A \) co-r.e.
4. if \( A \) not recursive, then \( B \) not recursive.
5. if \( A \) not r.e., then \( B \) not r.e.
6. if \( A \) not co-r.e., then \( B \) not co-r.e.

11.7 TM emptiness is not recursive: Reduction. We argue that \( E_{TM} \) is not recursive. Since \( E_{TM} \) is co-r.e., we reduce from \( \overline{A_{TM}} \) (rather than \( A_{TM} \)). In order to show that \( \overline{A_{TM}} \leq_m E_{TM} \), given a pair \( \langle M, w \rangle \) of a DTM \( M \) and a word \( w \), we need to construct a TM \( M' \) such that \( w \notin L(M) \) iff \( L(M') = \emptyset \). Here is a high-level description of \( M' \):

Input: \( w' \).
If \( w' \neq w \) then \text{REJECT}.
Simulate \( M \) on the input (which is \( w \)) until

(1) \( M \) accepts (then \text{ACCEPT}), or
(2) \( M \) rejects (then \text{REJECT}).

Note that if \( M \) loops on input \( w \), then so does \( M' \). If \( w \in L(M) \), then \( L(M') = \{w\} \); if \( w \notin L(M) \), then \( L(M') = \emptyset \). It follows that \( E_{TM} \) is not r.e., and therefore not recursive.