1. If $A$ is a language, then $\text{PREFIX}(A)$ is the language
\[ \{u|uv \in A \text{ for some string } v\} \]

(a) Prove that if $A$ is decidable, then $\text{PREFIX}(A)$ is Turing recognizable.

(b) Prove that if $A$ is Turing recognizable, then $\text{PREFIX}(A)$ is Turing recognizable.

[Since every decidable language is Turing recognizable, this part implies the first part, but since the proof is harder, I made it a separate part.]

Solution:

a) Let $A$ be a language over an alphabet $\Sigma$ and let $M$ be a Turing machine that decides $A$. Since $\Sigma^*$ is countably infinite, we may list the strings in $\Sigma^*$ as $s_1, s_2, s_3, \ldots$. A Turing machine $N$ that recognizes $\text{PREFIX}(A)$ is given by

\[ N = \text{"On input } w \]

1. For $i = 1, 2, 3, \ldots$

2. Run $M$ on $ws_i$. If $M$ accepts, accept. If $M$ rejects, next $i$.

b) Now assume that $M$ only recognizes $A$. Then, $M$ may go into an infinite loop on some inputs, and we have to modify our definition of $N$ to take account of this fact. Instead of just running $M$ on $ws_i$ and waiting for $M$ to halt, which may not happen, $N$ runs $M$ in parallel on several inputs, for a fixed number of steps.

\[ N = \text{"On input } w \]

1. For $i = 1, 2, 3, \ldots$

2. Run $M$ on $ws_1, ws_2, \ldots, ws_i$ for $i$ steps each. If $M$ accepts any of the strings in $i$ steps accept, else, next $i$.

2. Apply the method from class that decides $E_{\text{DFA}}$ to the following DFA and answer the questions below.
(a) List the states you mark in the order they get marked.
   \[ p, q, s, t, u \]

(b) Does the DFA belong to \( E_{DFA} \)? \( \text{No} \)

(c) How does your answer to (b) follow from your answer to (a)?
   An accept state \( \langle u \rangle \) is marked.

3. The language \( EQ_{NFA} \) is defined as \( \{ \langle A, B \rangle \mid A, B \text{ are NFAs and} \ L(A) = L(B) \} \). Prove that \( EQ_{NFA} \) is decidable.

**Solution:**
A Turing machine \( M \) that decides the language \( EQ_{NFA} \) is given by

\[ M = \text{“On input} \langle A, B \rangle \text{ where} A \text{ and} B \text{ are NFAs,} \]

1. Using the method Theorem 1.39, produce DFAs \( C \) and \( D \) with \( L(C) = L(A) \) and \( L(D) = L(B) \).
2. Run the TM \( F \) that decides \( EQ_{DFA} \) on \( \langle C, D \rangle \).
3. If \( F \) accepts, then \( \text{accept} \). If \( F \) rejects, then \( \text{reject} \).
4. Let $ALL_{REX} = \{(R)|R$ is a regular expression and $L(R) = \Sigma^*\}$. Show that $ALL_{REX}$ is decidable.

**Solution:**

$ALL_{REX}$ is decided by the following Turing machine $M$.

$M =$ “On input $\langle R \rangle$ where $R$ is a regular expression,

1. Combining the methods of Lemma 1.55 and Theorem 1.39, construct a DFA $A$ with $L(A) = L(R)$.
2. Obtain a DFA $B$ from $A$ by reversing accept and reject states.
3. Run the TM $T$ that decides $E_{DFA}$ on $\langle B \rangle$.
3. If $T$ accepts, then accept. If $T$ rejects, then reject.”

Note that you have to transform $R$ into a DFA and not just into an NFA because the complementation construction does not always work for NFAs.