1. Apply the method from class that decides $E_{DFA}$ to the following DFA and answer the questions below.

(a) List the states you mark in the order they get marked.
$p, q, r, t, s$

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

(c) How does your answer to (b) follow from your answer to (a)?
An accept state (s) is marked.

2. Apply the method from class that decides $E_{CFG}$ to the following CFG and answer the questions below.

\[
S \rightarrow VX|aSb \\
T \rightarrow aUTS \\
U \rightarrow aT|bU \\
V \rightarrow YV|WX
\]
\[ W \rightarrow YSXb \]
\[ X \rightarrow bXa \]
\[ Y \rightarrow \varepsilon|Ya \]

(a) List the terminals and variables you mark in the order they get marked. (List each terminal and variable only the first time you mark it. There is more than one possible order.)

\[ a,b|Y,X|W|V|S \]

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

(c) How does your answer to (b) follow from your answer to (a)?

\[ S \] is marked

3. The language \( EQ_{REX} \) is defined as \{\( \langle R, S \rangle \)|\( R, S \) are regular expressions and \( L(R) = L(S) \)}.

Prove that \( EQ_{REX} \) is decidable.

**Solution:**

A Turing machine \( M \) that decides the language \( EQ_{REX} \) is given by

\( M = \) “On input \( \langle R, S \rangle \) where \( R \) and \( S \) are regular expressions,

1. Combining the methods of Lemma 1.55 and Theorem 1.39, produce DFAs \( B \) and \( C \) with \( L(B) = L(R) \) and \( L(C) = L(S) \).
2. Run the TM \( F \) that decides \( EQ_{DFA} \) on \( \langle B, C \rangle \).
3. If \( F \) accepts, then \( \text{accept} \). If \( F \) rejects, then \( \text{reject} \).”

4. **Exercise 4.3**

\( ALL_{DFA} \) is decided by the following Turing machine \( M \).

\( M = \) “On input \( \langle A \rangle \) where \( A \) is a DFA,

1. Construct a DFA \( B \) such that \( L(B) = \overline{L(A)} \).
2. Run the TM \( T \) that decides \( E_{DFA} \) on \( \langle B \rangle \).
3. If \( T \) accepts, then \( \text{accept} \). If \( T \) rejects, then \( \text{reject} \).”

5. Let \( F = \{ \langle A, B \rangle | A, B \) are DFAs and \( L(A) \subseteq L(B) \} \). Prove that \( F \) is decidable.

**Solution:** A Turing machine \( M \) that decides the language \( F \) is given by

\( M = \) “On input \( \langle A, B \rangle \) where \( A \) and \( B \) are DFAs,

1. Produce a DFA \( C \) with \( L(C) = L(A) \cap \overline{L(B)} \).
2. Run the TM \( T \) that decides \( E_{DFA} \) on \( \langle C \rangle \).
3. If \( T \) accepts, then \( \text{accept} \). If \( T \) rejects, then \( \text{reject} \).”

If \( \langle A, B \rangle \) is in \( F \), then \( L(C) = L(A) \cap \overline{L(B)} = \emptyset \), so \( T \) accepts \( \langle C \rangle \) and \( M \) accepts \( \langle A, B \rangle \). If \( \langle A, B \rangle \) is not in \( F \), then \( L(C) = L(A) \cap \overline{L(B)} \neq \emptyset \), so \( T \) rejects \( \langle C \rangle \) and \( M \) rejects \( \langle A, B \rangle \). Thus, \( M \) decides \( F \).
6. Let $K = \{\langle A \rangle | A$ is a DFA and $L(A)$ contains a string with one more $a$ than $b\}$. Show that $K$ is decidable. (The solution to Problem 4.25 [4.23] is useful here.)

**Solution:** The language of all strings in $\{a, b\}^*$ with one more $a$ than $b$ is context-free. (You were asked to give a context-free grammar for this language on Homework 4.) Let $P$ be a PDA that recognizes this language.

A Turing machine $T$ that decides $K$ is given by:

1. Using the method of Problem 2.18a, construct a PDA $S$ that recognizes $L(A) \cap L(P)$, where $P$ is the PDA mentioned above.
2. Convert $S$ into a CFG $G$.
3. Run the TM $R$ that decides $E_{CFG}$ on $\langle G \rangle$.
4. If $R$ accepts, reject. If $R$ rejects, accept.”

7. Let $L = \{\langle G \rangle | G$ is a CFG and $L(G)$ contains at least one even length string\}. Prove that $L$ is decidable.

**Solution:** For any alphabet, the set of even length strings over the alphabet is a regular language.

A Turing machine $T$ that decides $L$ is given by:

1. Let $E$ be a DFA that recognizes the set of even length strings over the alphabet of $G$.
2. Convert $G$ to a PDA $P$.
3. Using the method of Problem 2.18a, construct a PDA $S$ that recognizes $L(E) \cap L(P)$.
4. Convert $S$ into a CFG $H$.
5. Run the TM $R$ that decides $E_{CFG}$ on $\langle H \rangle$.
6. If $R$ accepts, reject. If $R$ rejects, accept.”

8. The set of all finite sequences of natural numbers is countable.

**Proof:** Unlike the case of $\Sigma^*$ where $\Sigma$ is an alphabet, it is not possible to list all the finite sequences of natural numbers in order by their length, because there are infinitely many sequences of natural numbers with any given length other than 0, so this list would never get to any sequences of length 2, much less all the finite sequences.

One method that does work is to list the finite sequences of natural numbers in order according to the sum of their entries. (Because we do not consider 0 to be a natural number, there are only finitely many finite sequences of natural numbers whose sum is any given number.) For a given sum, the sequences can be listed in lexicographic (ie, dictionary) order.
The first sequence listed would be $\varepsilon$, then $\langle 1 \rangle$. Next would come the two sequences whose sum is 2: $\langle 1, 1 \rangle$ and $\langle 2 \rangle$. The sequences whose sum is 3 come next: $\langle 1, 1, 1 \rangle$, $\langle 1, 2 \rangle$, $\langle 2, 1 \rangle$ and $\langle 3 \rangle$. Next come the sequences with sum 4, and so on.