Problem 4

(a) Turing machine $T_1$ that decides DISJ$_{DFA}$:

On input $w$ to $T_1$:

1. Check if $w = <A, B>$ for DFAs $A$ and $B$:
   - "yes": go to step 2
   - "no": $T_1$ rejects/halts on $w$

2. Construct a DFA $C$ from DFAs $A$ and $B$ such that
   $L(C) = L(A) \cap L(B)$ (via Cartesian product of state-spaces of $A$ and $B$)

3. $T_1$ simulates the decider for $E_{DFA}$ on input $<C>$
   (see textbook Theorem 4.4)
   - if the decider accepts/halts on $<C>$
     then $T_1$ accepts/halts on $w$
   - else $T_1$ rejects/halts on $w$

Check:
1. $L(T_1) = \text{DISJ}_{DFA}$ and
2. $T_1$ halts on every input.

(b) Turing machine $T_2$ that decides INF$_{CFG}$:

In lecture, we have developed a "pumping lemma-like" condition for the
finiteness condition for regular languages and for context-free languages
for a CFG $G$ in Chomsky Normal Form with $m$ variables,

$L(G)$ is infinite iff $\exists x \in L(G)$ with $|x| \geq 2^m$.

On input $w$ to $T_2$:

1. Check if $w = <G>$ for some CFG $G$:
   - "yes": go to step 2
   - "no": $T_2$ rejects/halts on $w$
2. Convert an CFG $G$ to an equivalent CFG $G'$ in Chomsky Normal Form, say, with $m$ variables.

3. \[ \text{Length} := 0; \]

\[ \text{for} \quad \text{Length} \geq 2^m; \]

\[ \text{Generate all strings of length \text{Length};} \]

\[ \text{for each generated string } x, \text{To simulate the} \]
\[ \text{decider for } A_{CFG} \text{ on input } <G', x> \]
\[ \text{if the decider for } A_{CFG} \text{ accepts/halts on } <G', x> \]
\[ \text{then } T_2 \text{ accepts/halts on } w \]
\[ \text{else continue the generation,} \]

\[ \text{Length} := \text{Length} + 1; \]

\[ \text{and loop,} \]

4. Upon exiting the loop in Step 3, \[ T_2 \text{ accepts/halts on } w. \]

Check:
1. \[ l(T_2) = \text{INF}_{CFG}, \quad \]

2. \[ T_2 \text{ halts on every input} \]
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On input $w$, proceed to step 2.</td>
</tr>
<tr>
<td>2</td>
<td>For each constituent DFA $B_i$, simulate DFA $B_i$. If the simulation of DFA $B_i$ on input $w$ ends in an accepting state, then $T_3$ accepts $w$. Otherwise, reject $w$.</td>
</tr>
<tr>
<td>3</td>
<td>If the decision for $E(A_i, B_i)$ on input $w$ is negative, $T_3$ rejects $w$. Otherwise, continue to step 2.</td>
</tr>
</tbody>
</table>

*Note: $T_3$ is a non-deterministic Turing machine that decides $MIN_{DFA}$. The decision is made based on the acceptance of $w$ by at least one of the constituent DFAs $B_i$. If all DFAs reject $w$, then $T_3$ also rejects $w$. Otherwise, $T_3$ accepts $w$. The process continues until a decision is made.*
Problem 5: Turing machine \( T \) that recognizes \( L \):

On input \( w \):

1. Check if \( w = \langle M \rangle \) for some TM \( M \):
   - "yes": go to step 2
   - "no": \( T \) rejects/halts on \( w \).

2. (Long to recognize if \( M \) accepts at least one input)
   - \( \text{Number of steps} := 1 \)
   - For all strings \( z \) with \( |z| \leq \text{Number of steps} \)
     - \( T \) simulates \( M \) on input \( z \) for \( \text{Number of steps} \) steps
     - if \( M \) halts on \( z \), then \( T \) accepts/halts on \( w \)

   \( \text{Number of steps} := \text{Number of steps} + 1 \)

Choose 1. \( L(T) = L \)

Note that \( T \) does not necessarily halt on every input.
Problem 6. Similar to the machine constructed in the proof of Theorem 4.22.

Denote the TMs recognizing $A$ and $B$ by $T_A$ and $T_B$, respectively.

Construct the following TM $T$:

On input $w$:

1. Number of steps := 1;

2. $T$ simulates $M_A$ on input $w$ for Number of steps steps:
   - if $M_A$ accepts/halts on $w$ during the simulation, then $T$ accepts/halts on $w$.

3. $T$ simulates $M_B$ on input $w$ for Number of steps steps:
   - if $M_B$ accepts/halts on $w$ during the simulation, then $T$ rejects/halts on $w$.

4. Number of steps := Number of steps + 1;
   go to Step 2.

Check 1. The language accepted by $T$, $L(T)$, satisfies that $A \cap B \subseteq L(T)$ and $A \cup B \subseteq L(T)$, and

2. $T$ halts on every input.