1. (a) \( \text{REJECT}_T \) is defined as \( \{ \langle M, w \rangle | M \text{ is a Turing machine, and } M \text{ rejects } w \} \). Prove that \( \text{REJECT}_T \) is Turing recognizable.

**Solution:** \( \text{REJECT}_T \) is recognized by the following Turing machine \( V \):

\[
V = \text{ "On input } \langle M, w \rangle \text{:
} 1. \text{ Run } M \text{ on } w.
2. \text{ If } M \text{ rejects, accept. If } M \text{ accepts, reject."
}\]

(b) Show that \( \text{REJECT}_T \) is undecidable using diagonalization. Your proof should be similar to, but not the same as, the proof that \( A_{TM} \) is undecidable.

**Solution:** We assume that \( \text{REJECT}_T \) is decidable and obtain a contradiction. Suppose that the Turing machine \( H \) decides \( \text{REJECT}_T \). This means that

\[
H(\langle M, w \rangle) = \begin{cases} 
\text{reject} & \text{if } M \text{ accepts } w \\
\text{accept} & \text{if } M \text{ rejects } w \\
\text{reject} & \text{if } M \text{ loops on } w
\end{cases}
\]

Using \( H \), we define another Turing machine \( D \)

\[
D = \text{ "On input } \langle M \rangle \text{ where } M \text{ is a Turing machine
} 1. \text{ Run } H \text{ on } \langle M, \langle M \rangle \rangle.
2. \text{ If } H \text{ accepts, accept. If } H \text{ rejects, reject."
}\]

We have

\[
D(\langle M \rangle) = \begin{cases} 
\text{reject} & \text{if } M \text{ accepts } \langle M \rangle \\
\text{accept} & \text{if } M \text{ rejects } \langle M \rangle \\
\text{reject} & \text{if } M \text{ loops on } \langle M \rangle
\end{cases}
\]

Applying this to \( M = D \), we get

\[
D(\langle D \rangle) = \begin{cases} 
\text{reject} & \text{if } D \text{ accepts } \langle D \rangle \\
\text{accept} & \text{if } D \text{ rejects } \langle D \rangle \\
\text{reject} & \text{if } D \text{ loops on } \langle D \rangle
\end{cases}
\]

No matter what \( D \) does on \( \langle D \rangle \), we get a contradiction, so \( D \) can’t exist, which means that \( H \) can’t exist and \( \text{REJECT}_T \) is undecidable.

(c) Give a second proof that \( \text{REJECT}_T \) is undecidable by reducing \( A_{TM} \) to \( \text{REJECT}_T \).

[This will involve some creativity because the technique we used to reduce \( A_{TM} \) to \( \text{HALT}_T \) will not work here.]

**Solution:** Suppose that the Turing machine \( R \) decides \( \text{REJECT}_T \). Then we define a Turing machine \( S \) that decides \( A_{TM} \) as follows:

\[
S = \text{ "On input } \langle M, w \rangle
\]
1. Construct a TM $M'$ from $M$ by reversing the accept and reject states.
2. Run $R$ on $\langle M', w \rangle$.
3. If $R$ accepts, accept. If $R$ rejects, reject.”

Note that $\langle M, w \rangle \in A_{TM}$ if and only if $M$ accepts $w$ if and only if $M'$ rejects $w$ if and only if $\langle M', w \rangle \in REJECT_{TM}$ if and only if $R$ accepts $\langle M, w \rangle$ if and only if $S$ accepts $\langle M, w \rangle$, so $S$ decides $A_{TM}$. Since $A_{TM}$ is undecidable, no such Turing machine $S$ exists, so $R$ cannot exist and $REJECT_{TM}$ is undecidable.

2. The proof is the same as the one given in Theorem 5.3, except that in step 1 of the instructions for $M_2$, “has the form $0^n1^n$” is replaced by “has the form $0^n1^n2^n$”. Now if $M$ accepts $w$, then $L(M_2)$ is the context-free language $\Sigma^*$ and if $M$ does not accept $w$, then $L(M_2)$ is the non-context-free language $\{0^n1^n2^n | n \geq 0 \}$.

3. Let $NONCONTEXTFREE_{TM} = \{ \langle M \rangle | M$ is a Turing machine, and $L(M)$ is not a context-free language$\}$. Suppose that you want to reduce $A_{TM}$ to $NONCONTEXTFREE_{TM}$ by transforming $\langle M, w \rangle$ to $\langle M_2 \rangle$. (So if $\langle M, w \rangle$ is in $A_{TM}$, then $\langle M_2 \rangle$ is in $NONCONTEXTFREE_{TM}$, and if $\langle M, w \rangle$ is not in $A_{TM}$, then $\langle M_2 \rangle$ is not in $NONCONTEXTFREE_{TM}$.)

(a) Fill in the blanks in the following two statements in a way that states what you have to do to make the reduction work. Make your statements as general as possible. (In both cases you will be writing down something about the behavior of the Turing machine $M_2$.)

- If $M$ accepts $w$, then $L(M_2)$ is not context-free.
- If $M$ does not accept $w$, then $L(M_2)$ is context-free.

(b) Give the definition of the desired Turing machine $M_2$, given $M$ and $w$.

$M_2 = “On input x$

1. If $x$ does not belong to $\{0^n1^n2^n | n \geq 0 \}$, then reject.
2. If $x = 0^n1^n2^n$ for some $n \geq 0$, run $M$ on $w$.
3. If $M$ accepts $w$, accept.
   If $M$ rejects $w$, reject.”

If $M$ accepts $w$, then $L(M_2) = \{0^n1^n2^n | n \geq 0 \}$, which is not context-free, and if $M$ does not accept $w$, then $L(M_2) = \emptyset$, which is context-free.
4. **Problem 5.9**

We show that $T$ is undecidable by reducing $A_{TM}$ to $T$. Suppose that $R$ is a TM that decides $T$. We will give a TM $S$ that decides $A_{TM}$. Since $A_{TM}$ is not decidable, this will show that no such TM $R$ can exist, so $T$ is not decidable.

$S$ will have this form:

$S = \text{On input } \langle M, w \rangle \text{ where } M \text{ is a Turing machine and } w \text{ is an input:}$

1. Produce the TM $M_1$.
2. Run the TM $R$ that decides $T$ on $\langle M_1 \rangle$.
3. If $R$ accepts, then accept. If $R$ rejects, then reject.”

To complete the proof, we have to show how, given $\langle M, w \rangle$, $S$ can produce the description of a TM $M_1$ such that if $M$ accepts $w$, then $\langle M_1 \rangle \in T$, i.e., whenever $M_1$ accepts a string $w$, then it also accepts $w^R$, and if $M$ does not accept $w$, then $\langle M_1 \rangle \notin T$, i.e., there is some string $w$ such that $M_1$ accepts $w$ but does not accept $w^R$. Here is a description of $M_1$.

$M_1 = \text{On input } x,$

1. If $x = 01$, then accept $x$.
2. If $x \neq 01$, then run $M$ on input $w$ and accept if $M$ accepts.”

If $M$ accepts $w$, then $M_1$ accepts every string, so $\langle M_1 \rangle$ belongs to $T$. If $M$ does not accept $w$, then the only string accepted by $M_1$ is 01, so $\langle M_1 \rangle$ does not belong to $T$.

5. **Problem 5.15**

The corresponding language is $MOVELEFT_{TM} = \{ \langle M, w \rangle | M \text{ is a Turing machine and on input } w M \text{ attempts to move its head left at some point in the computation} \}$.

A Turing machine to decide $MOVELEFT_{TM}$ implements the following algorithm. Given $M$ and $w$, simulate $M$ on $w$ until either a) $M$ attempts to move left, b) $M$ halts without ever trying to move left, or c) $M$ moves right beyond the original input and then repeats some state without ever trying to move left. One of these three things has to happen eventually because if $M$ never tries to move left and never halts, it will always move right and eventually move past the original input. Since there are only finitely many states, eventually a state will be repeated after $M$ has gone past its original input. If a) happens, then $\langle M, w \rangle \in MOVELEFT_{TM}$. If b) happens, then $\langle M, w \rangle \notin MOVELEFT_{TM}$. If c) happens, then we also have $\langle M, w \rangle \not\in MOVELEFT_{TM}$ because if $M$ goes from a configuration $up\downarrow$ to a configuration $uvp\downarrow$ without ever moving left, then $M$’s computation will continue as $uvvp\downarrow, uvvvp\downarrow, \ldots$ without ever trying to move left. Since a Turing machine can implement this algorithm, $MOVELEFT_{TM}$ is decidable.
6. Problem 5.26

(a) If \( M \) is a 2DFA and \( x \) is an input for \( M \) of length \( n \), then a tape head position for \( M \) on \( x \) is a number between 0 and \( n + 1 \) that indicates which cell the tape head is on. (0 indicates that the tape head is on the blank to the left of the input and \( n + 1 \) indicates that the tape head is on the blank to the right of the input.) A configuration for \( M \) on \( w \) is a quadruple \((w, p, i, j)\) where \( p \) is the current state of \( M \), \( i \) is the position of the first tape head and \( j \) is the position of the second tape head. Since \( M \) cannot write on its tape, if \( M \) ever repeats a configuration, then \( M \) is in an infinite loop.

A Turing machine \( N \) to decide \( A_{2DFA} \) is given by

\[
N = \text{"On input } (M, x) \text{ where } M \text{ is a 2DFA and } x \text{ is an input string:}
\]

1. Simulate \( M \) on \( x \) until either a) \( M \) accepts, b) \( M \) tries to move left from the initial blank, c) \( M \) tries to move right from the final blank, or d) \( M \) repeats a configuration.

2. If a) happens, then accept. If b), c) or d) happens, then reject.”

If \( M \) is in an infinite loop, then since there are only finitely many configurations, \( M \) must eventually repeat a configuration, so one of a), b), c), d) must eventually happen. Thus, \( N \) decides \( A_{2DFA} \).

(b) To show that \( E_{2DFA} \) is undecidable, we will reduce \( A_{TM} \) to \( E_{2DFA} \).

Given a Turing machine \( M \) and an input \( w \), we can construct a 2DFA \( B \) such that \( L(B) \) is the set of accepting configurations of \( M \) on \( w \).

This \( B \) is similar to the LBA \( B \) constructed in the proof of Theorem 5.9, except that to check if \( C_{i+1} \) follows from \( C_i \) legally, this \( B \) does not zig-zag back and forth using dots as markers, but instead it puts one tape head on \( C_i \) and the other tape head on \( C_{i+1} \) and compares the two without having to write anything on the tape.

Suppose that \( R \) is a TM that decides \( E_{2DFA} \). Then, the following TM \( S \) decides \( A_{TM} \).

\[
S = \text{"On input } (M, w) \text{, where } M \text{ is a TM and } w \text{ is an input string:}
\]

1. Construct the 2DFA \( B \) as described above.
2. Run \( R \) on \( (B) \).
3. If \( R \) rejects, then accept; if \( R \) accepts, then reject.”

Since \( A_{TM} \) is not decidable, there cannot be such a TM \( S \), so there is not such TM \( R \) and \( E_{2DFA} \) is undecidable.

7. Exercise 5.4

No. For example, let \( A = \{0^n1^n | n \geq 0 \} \) and \( B = \{0^n1^m | n, m \geq 0 \} \). We will show that \( A \leq_m B \) even though \( A \) is not regular and \( B \) is regular. A mapping reduction \( f \) from \( A \) to \( B \) is computed by the Turing machine \( F \) given by

\[
F = \text{"On input } w \in \{0,1\}^*,
\]
1. Determine if $w$ belongs to $A$. (Since $A$ is decidable, $F$ can do this with no further information.)

2. If $w$ belongs to $A$, then output 01. If $w$ does not belong to $A$, then output 10.

(In fact, the same argument shows that if $A$ is any decidable language and $B$ is any language other than $\emptyset$ and $\Sigma^*$, then $A \leq_m B$.)

8. **Problem 5.22**

$A$ is Turing-recognizable if and only if $A \leq_m A_{TM}$.

**Proof:** First suppose that $A$ is Turing-recognizable and let $M$ be a Turing machine that recognizes $A$. The function $f$ defined by $f(w) = \langle M, w \rangle$ is a reduction from $A$ to $A_{TM}$ because it is obviously computable and we have

$$w \in A \text{ iff } M \text{ accepts } w \text{ iff } \langle M, w \rangle \in A_{TM} \text{ iff } f(w) \in A_{TM}.$$ 

Now suppose that $A \leq_m A_{TM}$. We know that $A_{TM}$ is Turing-recognizable, so by Theorem 5.28, $A$ is Turing-recognizable.

9. **Problem 5.23**

$A$ is decidable if and only if $A \leq_m 0^*1^*$.

**Proof:** First suppose that $A$ is decidable. Define $f$ by $f(x) = 01$ if $x \in A$ and $f(x) = 10$ if $x \in \overline{A}$. Since $A$ is decidable, $f$ is computable and $x \in A$ if and only if $f(x) \in 0^*1^*$, so $A \leq_m 0^*1^*$.

Conversely, suppose that $A \leq_m 0^*1^*$. Since $0^*1^*$ is decidable, $A$ is decidable by Theorem 5.22.

10. **Problem 5.24**

The set $\overline{A_{TM}}$ is mapping reduced to $J$ by the function $f(y) = 1y$. Thus, $J$ is not Turing-recognizable. The set $A_{TM}$ is mapping reduced to $J$ by the function $g(x) = 0x$. This shows that $\overline{A_{TM}}$ is mapping reducible to $\overline{J}$ and hence that $J$ is not Turing-recognizable.

11. **Problem 5.25**

Consider the set $J$ of Problem 5.24. According to that problem, $J$ is not Turing-recognizable, so $J$ is not decidable. We will show that $J \leq_m \overline{J}$, so $B = J$ is a solution to the problem.

First note that $\overline{J} = \{w|w = 0x \text{ for some } x \in \overline{A_{TM}} \text{ or } w = 1y \text{ for some } y \in A_{TM} \text{ or } w = \varepsilon \text{ or } w \text{ begins with a symbol other than 0 or 1}\}$. Let $z_0$ be some fixed string in $J$, for example, $z_0$ could be $0x_0$ for some particular $x_0$ in $A_{TM}$. Define $f : \Sigma^* \to \Sigma^*$ by

$$f(w) = \begin{cases} 1x & \text{if } w = 0x \\ 0y & \text{if } w = 1y \\ z_0 & \text{if } w = \varepsilon \text{ or } w \text{ starts with a symbol other than 0 or 1.} \end{cases}$$
Then, it is clear that $f$ is computable. To see that $f$ mapping reduces $J$ to $\overline{J}$, suppose first that $w \in J$. We must show that $f(w) \in \overline{J}$. If $w \in J$, there are two possibilities. If $w = 0x$ with $x \in A_{TM}$, then $f(w) = 1x$ with $x \in A_{TM}$, so $f(w) \in \overline{J}$. If $w = 1y$ with $y \in A_{TM}$, then $f(w) = 0y$ with $y \in A_{TM}$, so $f(w) \in \overline{J}$. Thus, if $w \in J$, then $f(w) \in \overline{J}$.

Now suppose that $w \notin J$. We must show that $f(w) \notin \overline{J}$. There are four possibilities to consider. If $w = 0x$ with $x \in A_{TM}$, then $f(w) = 1x$, so $f(w) \notin \overline{J}$. If $w = 1y$ with $y \in A_{TM}$, then $f(w) = 0y$, so $f(w) \notin \overline{J}$. If $w = \varepsilon$ or $w$ starts with a symbol other than 0 or 1, then $f(w) = z_0$, so $f(w) \notin \overline{J}$. Thus, if $w \notin J$, then $f(w) \notin \overline{J}$.

This shows that $f$ is a mapping reduction of $J$ to $\overline{J}$.

12. Is $E_{LBA}$ m-reducible to $A_{LBA}$? Explain your answer.

**Solution:** $E_{LBA}$ is not m-reducible to $A_{LBA}$. The proof is by contradiction. Suppose that $E_{LBA} \leq_m A_{LBA}$. By Theorem 5.8, $A_{LBA}$ is decidable, so by Theorem 5.22, $E_{LBA}$ is decidable. This contradicts Theorem 5.10. Therefore, the m-reduction is not possible.

13. Is $A_{LBA}$ m-reducible to $0^*1^*$? Explain your answer.

**Solution:** $A_{LBA}$ is m-reducible to $0^*1^*$. To prove this, first note that by Theorem 5.9, $A_{LBA}$ is decidable, so by Problem 3.15d, $A_{LBA}$ is decidable, so by Problem 5.23, $A_{LBA}$ is m-reducible to $0^*1^*$.