Problem 1

Consider the problem of determining whether a Turing machine $M$ on an input $w$ ever attempts to move its head left when its head is on the left-most tape cell. Formulate this problem as a language and show that it is undecidable.

**Proof by Contradiction.**

$L_{TM} = \{ <M, w> | M$ on $w$ tries moving its head left from the leftmost tape cell, at some point in its computation $\}$. Assume to the contrary that TM $R$ decides $L_{TM}$. Construct TM $S$ that uses $R$ to decide $A_{TM}$.

$S = \text{ "On input } <M, w>:\$

1. Convert $M$ to $M'$, where $M'$ first moves its input over one square to the right and writes new symbol $\$ on the leftmost tape cell. Then $M'$ simulates $M$ on the input. If $M'$ ever sees $\$ then $M'$ moves its head one square right and remains in the same state. If $M$ accepts, $M'$ moves its head all the way to the left and them moves left off the leftmost tape cell.

2. Run $R$, the decider for $L_{TM}$, on $<M', w>$. 

3. If $R$ accepts then accept. If it rejects, reject”

TM $S$ decides $A_{TM}$ because $M'$ only moves left from the leftmost tape cell when $M$ accepts $w$. $S$ never attempts that move during the course of the simulation because we put the $\$ to ”intercept” such moves if made by $M$.

Problem 2

Consider the problem of determining whether a Turing machine $M$ on an input $w$ ever attempts to move its head left at any point during its computation. Formulate this problem as a language and show that it is decidable.
Let \( LM_{TM} = \{ < M, w > | M \text{ ever moves left while computing on } w \} \). \( LM_{TM} \) is decidable. Let \( M_{LEFT} \) be the TM which on input \( < M, w > \) determines the number of states \( n_M \) of \( M \) and then simulates \( M \) on \( w \) for \( |w| + n_M + 1 \) steps. If \( M_{LEFT} \) discovers that \( M \) moves left during that simulation, \( M_{LEFT} \) accepts \( < M, w > \). Otherwise \( M_{LEFT} \) rejects \( < M, w > \).

The reason that \( M_{LEFT} \) can reject without simulating \( M \) further is as follows. Suppose \( M \) does make a left move on input \( w \). Let \( p = q_0, q_1, \ldots, q_s \) be the shortest computation path of \( M \) on \( w \) ending in a left move. Because \( M \) has been scanning only blanks (it’s been moving right) since state \( q|w| \), we may remove any cycles that appear after this state and be left with a legal computation path of the machine ending in a left move. Hence \( p \) has no cycles and must have length at most \( |w| + n_M + 1 \). Hence \( M_{LEFT} \) would have accepted \( < M, w > \), as desired.

**Problem 3**

Suppose \( A \) and \( B \) are languages and \( A \leq_m B \).

Prove the if \( A \) is undecidable, then \( B \) is undecidable.

**Proof by Contradiction.**

Assume, for the sake of contradiction, that \( B \) is decidable. Then there exists a Turing machine \( D_B \) that decides \( B \). By the definition of a decider, if \( b \in B \), then \( D_B \) accepts the input \( b \). If \( b \notin B \), then \( D_B \) rejects on input \( b \).

Since \( A \leq_m B \), there exists a computable function \( f \) such that if \( a \in A \), then \( f(a) \in B \), and if \( a \notin A \), then \( f(b) \notin B \). By the definition of computable, \( f \) completes in a finite amount of time on all inputs.

Consider the machine \( D_A \) on input \( a \):

1. Let \( b = f(a) \)
2. Run \( D_B \) on input \( b \)
   
   (a) If \( D_B \) accepted \( b \), then ACCEPT
   (b) If \( D_B \) rejected \( b \), then REJECT

Observe that \( D_A \) halts on all inputs. Step 1 runs the function \( f \). Since \( f \) is computable, this is a finite step. Step 2 runs \( D_B \). Since \( D_B \) is a decider, it halts in a finite amount of time.

We claim that \( D_A \) is a decider for \( A \).

**Case analysis:** For any element \( x \), there are two cases: \( x \) is in \( A \) or it is not in \( A \). Suppose that we run \( D_A \) on input \( x \):

**Case 1:** \( x \in A \): By the definition of \( f \) above, \( f(a) \in B \) because \( a \in A \). So \( b \in B \). Therefore, in Step 2, \( D_B \) will accept \( b \), so \( M_A \) accepts.
Case 2: $x \notin A$: By the definition of $f$ above, $f(a) \notin B$ because $a \notin A$. So $b \notin B$. Therefore, in Step 2, $D_B$ will reject $b$, so $M_A$ rejects.

For any element $x$, $D_A$ accepts if $x$ is in $A$ and $D_A$ rejects if $x$ is not in $A$. Therefore, $D_A$ is a decider for $A$, so $A$ is decidable. However, this is a contradiction, as $A$ is known to be undecidable. Therefore, our initial assumption must be incorrect. $B$ is indeed undecidable. □