1. (Problem 5.13) Let $USELESS_{TM} = \{ < M > \mid M \text{ is a TM with one or more useless states} \}$. Show that $A_{TM}$ reduces to $USELESS_{TM}$. Assume for the sake of contradiction that TM $R$ decides $USELESS_{TM}$. Construct TM $S$ that uses $R$ to decide $A_{TM}$. The new TM has a useless state exactly when $M$ doesn’t accept $w$. For this purpose, we use the universal turing machine.

$TM\ S:\$ On input $< M, w >$
$TM\ T:\$ On input $x$:
   a. Replace $x$ on the input by the string $< M, w >$.
   b. Run the universal TM $U$ to simulate $< M, w >$.
      (Note that this TM uses all it’s states)
      If $U$ accepts, enter a special state $q_A$ and accept.
1. Run $R$ on $T$ to determine whether $T$ has any useless states.
2. If $R$ rejects, accept. Otherwise reject.

If $M$ accepts $w$, then $T$ enters all states, but if $M$ doesn’t accept $w$, then $T$ avoids $q_A$. So $T$ has a useless state, $q_A$ iff $M$ doesn’t accept $w$.

2. (Problem 5.15) Let $LM_{TM} = \{ < M, w > \mid M \text{ ever moves left while computing } w \}$. $LM_{TM}$ is decidable. We construct a TM $LEFT_{TM}$ that decides $LM_{TM}$. To construct this TM we claim that any TM that ever makes a left move must do so in at most $w + n_M + 1$ steps, where $n_M$ is the number of states of $M$, and $w$ is the size of the input (we use $w$ to denote both the input and it’s size, but the usage should be clear from context). To see this, assume TM $M$ makes a LEFT move, and consider the shortest computation path $p = q_0, q_1 \cdots, q_s$ of $M$ ending in a LEFT move. First note that since $M$ has only been scanning blanks from state $q_w$, we may remove any cycles in the computation path and still be left with a legal computation path ending in
a left move. Hence, $p$ does not contain cycles, and can have length at most $w + n_M + 1$. Now we can construct $LEFT_{TM}$ as follows:

$TM \ LEFT_{TM} :$ On input $< M, w >$
1. Simulate $M$ on $w$ for $n_M + w + 1$ steps.
2. if $M$ ever makes a left move accept. Otherwise reject.

3. (Problem 5.17) The PCP over a unary alphabet is decidable. We describe a $TM M$ that decides unary PCP. Given a unary PCP instance,

$TM M: On input < P >$
1. Check if $a_i = b_i$ for some $i$. If so, accept.
2. Check if there exist $i, j$ such that $a_i > b_i$ and $a_j < b_j$.
   If so, accept, else reject.

In the first stage, $M$ checks for a single domino which forms a match. In the second stage, $M$ looks for two dominos which form a match. If it finds such a pair, it can construct a match by picking $(b_j - a_j)$ copies of the $i^{th}$ domino, putting them together with $(a_i - b_i)$ copies of the $j^{th}$ domino. This construction has $a_i(b_j - a_j) + a_j(a_i - b_i) = a_i b_j - a_j b_i$ 1’s on top, and $b_i(b_j - a_j) + b_j(a_i - b_i) = a_i b_j - a_j b_i$ 1’s on the bottom. If neither stages of $M$ accept, the problem instance contains dominos with all upper parts having more/less 1’s than the lower parts. In such a case, no match exists and therefore $M$ rejects.

4. (Problem 5.33) First we show that $A_{TM} \leq_M S$. This shows $S$ is not turing recognizable. The function $f$ can be described as follows.

$f: On input < M, w >,$
1. Construct machine $M_1$ that does the following:
   $M_1 :$ On input $x$,
   Run $M$ on $w$. If $M$ accepts $w$, reject.
   Otherwise if $x = < M_1 >$ accept.
2. Output $< M_1 >$. 
If $M$ accepts $w$, then $L(M_1) = \emptyset$. Hence, $\langle M_1 \rangle \in \overline{S}$. Conversely, if $M$ does not accept $w$, then $L(M_1) = \{ \langle M_1 \rangle \}$, and hence $\langle M_1 \rangle \in S$. This shows $S$ is not turing recognizable. We now show that $\overline{S}$ is not turing recognizable by reducing from $A_{TM}$ to $S$.

$g$: On input $\langle M, w \rangle$,
1. Construct machine $M_2$ that does the following:
   $M_2$: On input $x$
   Run $M$ on $w$. If $M$ accepts $w$, check if $x = \langle M_2 \rangle$. If it is, then accept. Otherwise reject.
2. Output $\langle M_2 \rangle$.

In this case, if $M$ accepts $w$, $L(M_2) = \langle M_2 \rangle$, and hence, $\langle M_2 \rangle \in S$. Otherwise $L(M_2) = \emptyset$ and $\langle M_2 \rangle \in \overline{S}$. Hence, $\overline{S}$ is not turing recognizable.

5. (Problem 5.35) Let $X = \{ \langle M, w \rangle \mid M$ is a single-tape TM that never modifies the portion of the tape that contains the input $w \}$. We show that $X$ is undecidable by reducing from $A_{TM}$. Let $R$ be a TM that decides $X$. We use $R$ to construct a TM $S$ that decides $A_{TM}$.

TM $S$: On input $\langle M, w \rangle$

TM $M_X$: On input $\langle M, w \rangle$
   a. Mark the right end of the input with a symbol $\notin \Gamma_M$.
   b. Copy $w$ to the part of the tape after $. Call this part $w'$
   c. Simulate $M$ on $w'$.
   d. If $M$ accepts, write any character on the first cell of the input tape and accept.
   e. Otherwise reject.
1. Input $\langle M_X, w \rangle$ to $R$.
2. If $R$ accepts, accept. Otherwise reject.

Note that $M_X$ ever modifies its input iff $M$ accepts $w$. Hence, we have decided $A_{TM}$, a contradiction.