5.7 If $A$ is Turing-recognizable and $A \leq_m \overline{A}$, then (by Theorem 5.28) $\overline{A}$ is Turing-recognizable – i.e., $A$ is co-Turing-recognizable. Therefore, (by Theorem 4.22) $A$ is decidable since $A$ is both Turing-recognizable and co-Turing-recognizable.

5.12 Assume it is decidable to determine if a single-tape Turing machine ever writes a blank symbol over a nonblank symbol during the course of its input string on any computation and assume that $D$ decides this language. We want to use $D$ to decide $A_{\text{TM}}$. Given any $<M,w>$, we want to create a Turing machine that writes a blank over a nonblank if and only if $M$ accepts $w$. We can create a new Turing machine $M_1$ that does the following:

- rejects every string other than $w$
- replaces all transitions that write a blank symbol with transitions that write an $x$, where $x$ is some symbol not in the tape alphabet
- adds a new transition for every transition that reads a blank symbol – the new transition would do exactly the same thing as the original except it would read an $x$ instead of a blank, where $x$ is the same symbol used above
- replaces all transitions to the accept state with three steps: first write any nonblank, then write a blank over the nonblank just written, then go to the accept state

The new TM will only write a blank symbol just before going to the accept state (and this blank symbol will be written over a nonblank symbol). Furthermore, this can only happen if $M$ accepts $w$, since $M_1$ rejects every string other than $w$ and $M_1$ will process $w$ in exactly the same way that $M$ processes $w$ except that it will write $x$’s instead of blanks and in some cases it may read $x$’s instead of blanks. If $M$ accepts $w$, then $M_1$ will write a nonblank, overwrite the nonblank with a blank and then accept $w$. Now we can create our decider for $A_{\text{TM}}$.

$S = \text{“On input } <M,w>, \text{ where $M$ is a TM}$

1. Create $M_1$ as described above
2. Run the decider $D$ on input $<M_1>$
3. If $D$ accepts accept
4. If $D$ rejects reject’

Since $D$ is a decider, $S$ is also a decider. Also, $S$ accepts if $M$ accepts $w$ (this is the only way that $M_1$ will overwrite a nonblank with a blank) and $M$ rejects otherwise. Therefore, $S$ decides $A_{\text{TM}}$. This is a contradiction since $A_{\text{TM}}$ is known to be undecidable. Therefore, $D$ cannot exist so it is undecidable to determine if a TM ever overwrites a nonblank symbol with a blank symbol.
5.20 We can easily show that there are an uncountable number of subsets of \( \{1\}^* \). For any \( n \in \mathbb{N} \), let \( s_n \) be a string of \( n-1 \) 1’s. Then, we can associate any subset \( S \) of \( \{1\}^* \) with an infinite binary string – the \( i^{th} \) symbol of the string would be a 1 if \( s_i \) is in \( S \) and would be 0 otherwise. Clearly, each subset has a unique corresponding infinite binary string. Therefore, then number of subsets of \( \{1\}^* \) is equal to the number of infinite binary strings, which we have shown in class to be an uncountable set. We also showed in class that the number of Turing machines is countable. Therefore, there is some subset of \( \{1\}^* \) that is not Turing recognizable (since there are more subsets than there are Turing machines). Clearly, any non-recognizable subset is also undecidable.

5.22 If \( A \leq_m A_{TM} \), then \( A \) is clearly Turing recognizable since we know \( A_{TM} \) is Turing-recognizable (by Theorem 5.28). It remains to show that if \( A \) is Turing-recognizable then \( A \leq_m A_{TM} \).

Assume \( A \) is Turing-recognizable. Then there exists a Turing machine \( N \) that recognizes \( A \) – i.e., \( A = \{w \mid N \text{ accepts } w\} \). Consider the function \( f(w) = <N,w> \). Clearly, if \( w \) is in \( A \), then \( N \) accepts \( w \) so \( <N,w> \) is in \( A_{TM} \). Also, if \( w \) is not in \( A \), then \( N \) does not accept \( w \) so \( <N,w> \) is not in \( A_{TM} \). Therefore, \( f \) is a mapping reduction from \( A \) to \( A_{TM} \) – i.e., \( A \leq_m A_{TM} \).

5.30 b To use Rice’s Theorem, we need to show that the designated property (in this case the property that \( 1011 \in L(M) \)) is non-trivial – i.e., that (i) the property depends only on the language of \( M \), and (ii) that some but not all Turing machines satisfy the property. The first condition obviously holds in this case. To show that the second condition holds, we observe that we can easily create the Turing machines \( M_1 \) and \( M_2 \) such that \( L(M_1) = \{0,1\}^* \) and \( L(M_2) = \emptyset \). Thus, \( M_1 \) satisfies the property (since \( 1011 \in \{0,1\}^* \)), but \( M_2 \) does not. Therefore, this is a non-trivial property so Rice’s Theorem holds – i.e. it is undecidable to determine if a Turing machine is in the language \( \{<M> \mid M \text{ is a TM and } 1011 \in L(M)\} \).