DIRECTIONS: For each of the following languages $A$, determine whether $A$ is semidecidable, and whether $\overline{A}$ is semidecidable. Justify your answers but DO NOT USE RICE’s THEOREM.

1. Let $A_1$ be the set of all Turing machine descriptions $\langle M \rangle$ such that the computation of $M$ on a blank tape writes a nonblank symbol at some point.

Solution:

$A_1$ is decidable (and hence $\overline{A_1}$ is also decidable, and both are semidecidable).

Here is an algorithm for deciding whether a given string $w$ is in $A_1$.

If $w$ is not of the form $\langle M \rangle$ for some TM $M$, then REJECT. Otherwise suppose $w = \langle M \rangle$, where $M$ is a TM. Let $Q$ by the set of states of $M$, and let $s = |Q|$ (so $s$ is the number of states in $M$’s finite state control).

Simulate $M$ for $s$ steps, starting on a blank tape in state $q_0$. If $M$ prints a nonblank symbol during the simulation, then ACCEPT, otherwise REJECT.

The algorithm is correct, because if $M$ does not write a nonblank symbol within $s$ steps, then it never will. This is because after $s$ steps (if not before) it either will halt or it will repeat a state, and so it will be caught in a loop in which it always reads a blank symbol and always writes a blank symbol.

2. Let $A_2$ be the set of all Turing Machine descriptions $\langle M \rangle$ such that $M$, during the computation starting with a blank tape, never attempts to move its head left when its head is on the left-most tape square. (Here we use the textbook’s convention that the tape is one-way infinite to the right.)

Solution:

$A_2$ is not semidecidable, but $\overline{A_2}$ is semidecidable.

To show that $A_2$ is not semidecidable we show

$$\overline{HB} \leq_m A_2$$

Thus we need a computable function $f$ such that for all Turing machines $M$

$$\langle M \rangle \in \overline{HB} \iff f(\langle M \rangle) \in A_2$$

That is, we want $f(\langle M \rangle) = \langle M' \rangle$ where $M'$ is a Turing machine, and $M$ fails to halt on a blank tape iff $M'$ never attempts to move its head left of the left-most square, starting on a blank tape.

We design $M'$ to be similar to $M$, except that $M'$ has an extra tape symbol $\#$. The computation of $M'$ (on a blank tape) begins by writing the symbol $\#$, shifting the head right, and entering the initial state of $M$. Then $M'$ runs $M$ modified so that if at any
step \( M' \) scans the symbol \# in a state \( q \), then \( M' \) shifts its head right and remains in state \( q \) (thus simulating the action of \( M \) if it attempted to shift left from the left-most tape square). Note that so far \( M' \) has not attempted to shift its head left from its left-most square (which contains \#).

If \( M \) halts, then \( M' \) moves its head left repeatedly until it scans \#, and then attempts to shift its head left again.

It is easy to see that \( M' \) has the required property, as expressed above.

To show that \( A_2 \) is semidecidable, just note that a certificate showing that \( M \) attempts to move its head to the left of the left-most tape square, is the computation of \( M \) on a blank tape, up to the time \( M \) makes that attempt.

3. Let \( A_3 \) be the set of all \( \langle M \rangle \) such that \( M \) is a Turing machine and \( L(M) \) contains infinitely many even length palindromes. (Recall that these are strings of the form \( ww^R \), where \( w^R \) is \( w \) written backwards). Here we assume that \( \Sigma = \{0, 1\} \).

**Solution:**

Neither \( A_3 \) nor \( \overline{A_3} \) is semi-decidable, and hence neither is decidable.

To show that \( \overline{A_3} \) is not semidecidable, it suffices to show \( HB \leq_m A_3 \) (because it follows that \( \overline{HB} \leq_m \overline{A_3} \), and \( \overline{HB} \) is not semi-decidable).

To show \( HB \leq_m A_3 \), we show how to computably map \( \langle M \rangle \) to \( \langle M' \rangle \), where \( M \) is any Turing machine and \( M' \) is a TM such that \( M \) halts on a blank tape iff \( L(M') \) contains infinitely many palindromes.

We define \( M' \) as follows: On input \( w \), \( M' \) runs \( M \) on a blank tape. If \( M \) halts, then \( M' \) accepts \( w \).

Thus if \( M \) halts on a blank tape, then \( L(M') = \Sigma^* \), so \( L(M') \) contains infinitely many even palindromes. Conversely, if \( M \) does not halt on a blank tape, then \( L(M') = \emptyset \), so \( L(M') \) does not contain any even palindromes.

To show that \( A_3 \) is not semidecidable, we show that \( \overline{HB} \leq_m A_3 \). To do this, we show how to computably map \( \langle M \rangle \) to \( \langle M' \rangle \), such that \( M \) fails to halt on a blank tape iff \( L(M') \) has infinitely many even palindromes.

We define \( M' \) as follows: On input \( w \), \( M' \) runs \( M \) on a blank tape for \( |w| \) steps. If \( M \) halts, then \( M' \) rejects \( w \). Otherwise \( M' \) accepts \( w \). Thus if \( M \) halts on a blank tape (say in \( t \) steps) then \( L(M') \) does not contain any string \( w \) with \( |w| \geq t \), so \( L(M') \) is finite. Conversely, if \( M \) fails to halt on a blank tape, then \( L(M') = \Sigma^* \), so \( \langle M' \rangle \in A_3 \).

4. \( A_4 \) is the set of all \( \langle G_1, G_2 \rangle \) such that \( G_1 \) and \( G_2 \) are context-free grammars, and \( L(G_1) = L(G_2) \).

**Solution:**

\( \overline{A_4} \) is semi-decidable but not decidable. \( A_4 \) is neither semi-decidable nor decidable.

To show that \( \overline{A_4} \) is semi-decidable, we give a TM \( M_0 \) such that \( L(M_0) = \overline{A_4} \). To do this we use the fact that \( A_{CFG} \) is decidable. That is, it is decidable, given \( \langle G, w \rangle \), whether \( w \in L(G) \).
The machine $M_0$ uses the following algorithm on input $\langle G_1, G_2 \rangle$: (here $s_0, s_1, ...$ is an enumeration of the strings in $\Sigma^*$):

for $i : 0..\infty$
  if $s_i \in \mathcal{L}(G_1)$ and $s_i \notin \mathcal{L}(G_2)$ then ACCEPT
  if $s_i \in \mathcal{L}(G_2)$ and $s_i \notin \mathcal{M}(G_1)$ then ACCEPT
end for

It is easy to see that $\mathcal{L}(M_0) = \overline{A_4}$.

To show that $A_4$ is not decidable, we use the fact that $\text{ALL}_{CFG}$ is not decidable, and we show

$$\text{ALL}_{CFG} \leq_m A_4$$

Given an input $\langle G \rangle$ to $\text{ALL}_{CFG}$ we compute an input $\langle G_1, G_2 \rangle$ to $A_4$ such that $\mathcal{L}(G) = \Sigma^*$ iff $\mathcal{L}(G_1) = \mathcal{L}(G_2)$. To do this, we take $G_1 = G$ and $G_2$ to be any grammar that generates all strings in $\Sigma^*$. 