Midterm Test Solutions:

1. (15 points) Determine whether the statements below are true or false. If true, provide a proof. If false, explain why the statement is false. Recall that a language \( L \) is non-trivial if \( L \neq \emptyset \) and \( L \neq \Sigma^* \).
   
   (a) (5 points) For any non-trivial semi-decidable language \( A \), there is a computable mapping reduction \( A \leq_m \overline{A} \) where \( \overline{A} \) is the complement.

   **Solution:** False. If \( A \) is semi-decidable and there is a mapping reduction \( A \leq_m \overline{A} \), then \( A \) is decidable. But there are decidable languages that are not semi-decidable (eg. \( A = A_{TM} \)) so not every semi-decidable language has a reduction \( A \leq_m \overline{A} \).

   (b) (5 points) For any non-trivial decidable language \( B \), there is a computable mapping reduction \( B \leq_m \overline{B} \).

   **Solution:** True. Since \( B \) is non-trivial, there is a string \( y \in B \) and a string \( z \notin B \). The function \( f : \Sigma^* \mapsto \Sigma^* \) given by
   
   \[
   f(x) = \begin{cases} 
   z & x \in B \\
   y & x \notin B 
   \end{cases}
   \]

   provides a computable mapping reduction \( B \leq_m \overline{B} \) since \( B \) is decidable, so we can check for membership in \( B \) in finite time.

   (c) (5 points) If \( A \) is an infinite non-trivial decidable language, there is a subset \( B \subseteq A \) that is not decidable.

   **Solution:** True. The set of decidable languages is countable but there is an uncountable number of subsets of \( A \) if \( A \) is infinite, using a diagonalization argument. So there must be a subset of \( A \) that is undecidable even if \( A \) is decidable itself. A explicit undecidable subset of \( A \) be constructed as well using a computable bijection \( f : \Sigma^* \mapsto A \).

Marking notes: For each problem, up to 2 marks for getting the True/False answer right and up to 3 marks for quality of explanation.

2. (10 points) Consider the language

\[
SEP = \{ \langle M_1, M_2 \rangle : L(M_1) = \overline{L(M_2)} \}
\]

consisting pairs of Turing machines \( \langle M_1, M_2 \rangle \) where for every string \( x \in \Sigma^* \), exactly one of \( M_1 \) or \( M_2 \) accepts \( x \). Is \( SEP \) semi-decidable? Prove your answer.

**Solution:** \( SEP \) is not semi-decidable. We prove this using a mapping reduction from a non-semidecidable language.

**Solution 1:** Let \( E_{TM} \) be the set of Turing machine encoding \( \langle M \rangle \) where \( L(M) = \emptyset \). Note that \( E_{TM} \) is not-semidecidable. There is a reduction \( E_{TM} \leq_m SEP \) by using the function \( f(\langle M \rangle) = \langle M, M_a \rangle \) where \( M_a \) is a Turing machine that always accepts its input. Note that \( \langle M \rangle \in E_{TM} \) if and only if \( \langle M, M_a \rangle \in SEP \) so we have constructed a valid reduction \( E_{TM} \leq_m SEP \).
Solution 2: Let $A_{TM}$ be the language of pairs $\langle M, w \rangle$ where $M$ accepts $w$. We show a reduction $A_{TM} \leq_m SEP$ to show that $SEP$ is not semi-decidable. Given a pair $\langle M, w \rangle$, we define $M_1$ be a Turing machine that on any input $x$, firstly runs $M$ on $w$ and then accepts $x$ whenever $M$ accepts $w$ and rejects $x$ if $M$ rejects $w$. Let $M_2$ be a Turing machine that accepts all inputs. The mapping $f(\langle M, w \rangle) = \langle M_1, M_2 \rangle$ is computable and it is a reduction, since $L(M_1) = \Sigma^*$ if $M$ accepts $w$, and $L(M_1) = \emptyset$ otherwise. Clearly, we have $L(M_2) = L(M_1) = \emptyset$ if and only if $M$ does not accept $w$, so we have constructed a valid reduction $A_{TM} \leq_m SEP$.

Marking notes:

1. up to 2 marks for the right answer (yes or no)
2. up to 3 marks for proposing a proof strategy to show your answer
3. up to 5 marks for details of the proof depending on the quality

If you answered that $SEP$ is semidecidable, you could get up to 4 marks for the first two points.

3. (5 points) Recall the definition of the Kolmogorov complexity $K(x)$ of a string and recall that the set of Kolmogorov-incompressible strings is defined as

$$ I = \{ x \in \{0,1\}^* : K(x) \geq |x| \}.$$

(a) (5 points) Show that $I$ is co-semidecidable.

**Solution:** Notice that there are finitely many descriptions $\langle M, w \rangle$ where $|\langle M, w \rangle| < |x|$. Hence, given input $x$, a Turing machine can go through all of these descriptions, testing each one to see if they produce output $x$ if one exists. To avoid infinite loops when testing a particular description, a Turing machine can use the “dovetailing” technique discussed in class. Since we can construct a Turing machine which when given an input $x$, tests if it has a description of length less than $|x|$, the language $I$ is co-semidecidable.

Marking notes:

- up to 2 marks for a valid proof strategy (e.g. constructing a Turing machine $M$)
- up to 2 marks for applying the definition of Kolmogorov complexity
- up to 1 mark for arguing that $M$ correctly recognizes $I$.

(b) (3 points) EXTRA CREDIT (do only if you have finished all other problems!): Let $A$ be a decidable language where $A_n = A \cap \{0,1\}^n$ are its elements of length $n$. Suppose $|A_n| \leq 2^\epsilon n$ for some $0 \leq \epsilon < 1$. Show that $A$ can have only finitely many incompressible strings (i.e. $|A \cap I| < \infty$).

**Solution:** It can be proven using enumerators that under the conditions given, we have $K(x) \leq \log |A_n| + 2 \log n + c$ for some constant $c$ given some string $x \in A_n$. Hence, if $|A_n| \leq 2^\epsilon n$ for some $\epsilon < 1$, we have $K(x) \leq \epsilon n + O(\log n)$, for any $x \in A_n$. If there were infinitely many incompressible strings in $A$, then $(1 - \epsilon)n \leq O(\log n)$ for infinitely many $n$, which is a contradiction.