Solution 1:

Correct Answer: B
Explanation: Option A is true since we can first reduce a language A to B and then B to C to get a reduction from language A to C.
Option B is false. For example, any regular language can be reduced to halting problem. The reverse is not true.
Option C and D are true because the given condition would put the language in both the set of Turing recognizable languages and co-Turing recognizable languages and thus making them decidable.

Solution 2:

Correct Answer: D
Explanation:
Option A is not correct: Let $L_1 = L(D)$ and let $L_2$ be the complement of $L_1$. Since regular languages are closed under complementation, we know that $L_2$ is regular and hence have a DFA $D_2$ that accepts it. We can hence design a Turing machine which simply checks whether $L_2$ is empty or not.
Option B is not correct: We can simply simulate $M$ over $w$
Option C is not correct: We can check whether derivation tree of a sufficiently large string contains a cycle or not.
Option D is correct: This is due to the diagonalization argument covered in the lectures.

Solution 3:

Correct Answer: C
Explanation:
Option A is not correct because we can reduce any decidable language into a regular language.
Option B is not correct because we can reduce any regular language into, say, halting problem.
Option C is correct. To decide A, we first reduce it to B and then decide B.

Solution 4:

Correct Answer: B,C
Explanation: Only $L_2$ and $L_3$ can be checked by looking at the description of the input machine. Hence option $B$ and $C$ are correct.

Solution 5:

Correct Answer: C
Explanation: We can run Turing machine simultaneously on all strings and accept when three strings are accepted. Hence C is correct.

Solution 6:

Correct Answer: B
Explanation: To recognize $L_1$ we simulate $M$ on $w$ and accept if $M$ accepts. By diagonalization argument, $L_2$ is not Turing recognizable.

Solution 7:

Correct Answer: A
Explanation: For every machine $M$, the set $L(M)$ is Turing recognizable. Hence the given language is a set of all encoding of Turing machines, which is decidable.
Solution 8:

Correct Answer: D
Explanation: The Set of decidable languages is contained in the set of Turing recognizable languages. Hence the option D is the correct answer.

Solution 9:

Correct Answer: D
Explanation: Let us assume that the language $L_1 \cup L_2$ is Turing recognizable. We give an algorithm to decide the language $L = \{<M, w> | M \text{ accepts } w\}$. Let the machine $M'$ be the one which recognizes $L_1 \cup L_2$. We run $M'$ simultaneously on $<M, w, 0>$ and $<M, w, 1>$ and halt as soon as $M'$ accepts either of them. We accept if $<M, w, 1>$ is accepted by the machine else we reject. Since one of these strings must be in $L_1 \cup L_2$ our algorithm always halts making the language $L$ decidable. This is a contradiction. Similar argument can be used to prove that it is not co-Turing recognizable.

Solution 10:

Correct Answer: A
Explanation: Let $M$ be a halting turing machine which accepts the language $A$. We fix a string $w_0 \in \overline{A}$ and $w_1 \notin \overline{A}$ Our machine for reduction in as follows. For a given string $w$, we simulate $M$ over $w$. If $M$ accepts $w$, we output $w_0$ and if $M$ rejects $w$ we output $w_1$. It is easy to see that the reduction is correct.