Solution 1:

Correct Answer: B,D
Explanation:
By Rice’s theorem, $L_1$ and $L_3$ are undecidable. $L_2$ is the set of all strings and $L_3$ can be decided by counting the number of states in the description of Turing machine.

Solution 2:

Correct Answer: D
Explanation:
By Rice’s Theorem, $L_1$, $L_2$ and $L_3$ are undecidable. $L_4$ is the set of all strings over some alphabet and hence decidable.

Solution 3:

Correct Answer: C
Explanation:
Every language in P has a deterministic polynomial time algorithm, and since P is a subset of NP, it follows that there are languages in NP which have a deterministic polynomial time algorithm. Therefore option A is wrong.

Every language in P will have a non-deterministic polynomial time algorithm since P is subset of NP. Option B is hence wrong.

Since $P \subseteq NP$, if $NP \subseteq P$ were to be true, it would follow that $P = NP$ contradicting our assumption. Hence Option C is correct.

Option D is wrong since $P \subseteq NP$.

Solution 4:

Correct Answer: B
Explanation:
Since, there can be decidable languages not in NP, A is incorrect.
Languages which are in NP can be decided in polynomial time. Hence B is correct option.

The halting problem is an NP-hard language and undecidable. Hence C is incorrect.

Option D is not correct because there are decidable NP-hard languages which are not NP-complete.

Solution 5:

Correct Answer: A,B,C
Explanation:
Claim: If $SAT \leq_p \overline{SAT}$ then $NP \subseteq \text{co-NP}$

Proof. Let $L \in NP$. Since $SAT \leq_p \overline{SAT}$, as assumed, $L \leq_p \overline{SAT}$. By definition of reduction, this implies $\overline{L} \leq_p SAT$. Hence $\overline{L} \in NP$. Therefore, $L \in \text{co-NP}$.

Claim: If $SAT \leq_p \overline{SAT}$ then $co-NP \subseteq NP$

Proof. Let $L \in co-NP$. So, $\overline{L} \in NP$. Since $SAT \leq_p \overline{SAT}$, as assumed, $\overline{L} \leq_p \overline{SAT}$. By definition of reduction, this implies $L \leq_p SAT$. Hence $L \in NP$.

By above claims, it follows that if $SAT \leq_p \overline{SAT}$ then $NP = \text{co-NP}$. Therefore, option A is correct.

If $NP = \text{co-NP}$, then $\overline{SAT} \leq_p SAT$ (Since every language in NP is polynomial time reducible to SAT). This implies that $SAT \leq_p \overline{SAT}$. Hence option B is correct.

It is known that P is subset of NP. The complement of P is also in P. (We simply reverse the acceptance and rejections) Hence P is also a subset of co-NP. Therefore $P \subseteq NP \cap \text{co-NP}$. Hence option C is correct.
It is unknown whether $NP \cap co-NP = P$. Hence D is incorrect.

Solution 6:

Correct Answer: A,B,D
Explanation: L is NP-complete. This implies that L is both NP-hard and in NP. Since L is an NP-hard language, every language in NP, including SAT is reducible in polynomial time to L. Hence A is correct. Since L is in NP, it is reducible to SAT, which is NP-hard. Hence options A, B and D are correct. C is incorrect because, if it were to be correct, it would imply that every language in NP is NP-hard. This is not known to be true.

Solution 7:

Correct Answer: C
Explanation: We take prime factors of $n$ along with their multiplicity as a certificate. The verifier simply checks whether the given factors are valid (i.e. they are indeed prime factors of $n$) and if anyone of them is smaller than $m$. This puts the language in both NP and co-NP.

Solution 8:

Correct Answer: B,C
Explanation: The class of NP languages is closed under union and intersection. This is so because to decide $L_1 \cup L_2$ we can run the turing machines for $L_1$ and $L_2$ in parallel and accept if any one of them accepts. To decide $L_1 \cap L_2$, we accept if both of them accepts. It follows that both the above mentioned languages are in NP. We can also decide $L_1^*$ by first non-deterministically guessing the length of string checking whether it belongs to $L_1$. We do this repeatedly until we encounter a blank in input tape. Hence option A and D are incorrect. Since the complement of languages mentioned in option B and C are in NP, and it is not necessarily true that $NP = co-NP$, we have option B and C as correct answers.

Solution 9:

Correct Answer: C
Explanation: It is known that P is subset of NP. The complement P is also in P. (We simply reverse the acceptance and rejections) Hence P is also a subset of co-NP. Therefore $P \subseteq NP \cap co-NP$. Every language in NP and co-NP are decidable. However, halting problem is an undecidable language which is Turing recognizable. Hence option C is correct.