1. Let $coNP$ be the class of languages whose complement is in $NP$. Show that $P \subseteq NP \cap coNP$. Show that if $P = NP$ then $P = coNP$. (Warning: $coNP$ is not the complement of $NP$.)

Solution:
Claim: $P \subseteq NP \cap coNP$.

(1) Let language $L \in P$, then $L \in NP$. Because for every language which is solvable in polynomial time, we can have an algorithm that solves it in nondeterministic polynomial time. Thus, $P \subseteq NP$.

(2) Also, for $L \in P$, we can have $\overline{L} \in P$. Because we can construct a TM that decides $\overline{L}$ in polynomial time, i.e., just reverse the output of TM deciding $L$. This means that $P$ is closed under complement. Based on the previous proof, we also have $\overline{L} \in NP$. According to the definition of $coNP$, $L \in coNP$. Thus, $P \subseteq coNP$. Based on (1) and (2), we have $P \subseteq NP \cap coNP$.

If $P = NP$, we need only show that this would imply $coNP \subseteq P$, since the reverse argument is given above in general. Let $L \in coNP$, then $\overline{L} \in NP = P$. Since we know $P$ is closed under intersection, we then have $L \in P$ and so $P = coNP$.

2. 7.46 Say that two Boolean formulas are equivalent if they have the same set of variables and are true on the same set of assignments to those variables (i.e., describe the same Boolean function). A Boolean formula is minimal if no shorter Boolean formula is equivalent to it. Let MIN-FORMULA be the collection of minimal Boolean formulas. Show that if $P=NP$, then MIN-FORMULA $\in P$.

Solution:
First notice that the language of all pairs of Boolean formulas $(\phi_1, \phi_2)$ such that $\phi_1$ and $\phi_2$ are not equivalent is an NP language. We can guess an assignment and check that the formulas differ on this assignment in polynomial time. Assuming $P=NP$, this language and its complement, the equivalence problem for formulas, are in $P$.

A formula is not minimal if there exists a smaller formula that is equivalent to it. Thus, given the above, MIN-FORMULA $\in NP$ because we can guess a smaller formula and check equivalence. Again, assuming $P=NP$, we have MIN-FORMULA $\in P$ and hence MIN-FORMULA $\in P$.

3. Show that NP is closed under union and concatenation.

Solution:
NP is closed under union.
Let $L_1, L_2$ be two NP languages, and $M_1, M_2$ be their polynomial time nondeterministic decider. We construct a NTM $N_0$ that decides $L_1 \cup L_2$ in polynomial time: $N_0 = “On$ input string $w:$

1. Run $M_1$ on $w$. If it accepts, ACCEPT.
2. Run $M_2$ on $w$. If it accepts, ACCEPT. Otherwise, REJECT.”

In both stages 1 and 2, $N_0$ uses its nondeterminism when the machines being run make nondeterministic steps. $N_0$ accepts $w$ iff either $M_1$ and $M_2$ accept $w$. Therefore, $N_0$ decides the union of $L_1$ and $L_2$. Since both stages take polynomial time, the algorithm runs in polynomial time.
NP is closed under concatenation.

For any two NP languages \(L_1\) and \(L_2\), let \(M_1, M_2\) be the NTMs that decide them in polynomial time. We construct a NTM \(N_1\) that decides \(L_1L_2\) in polynomial time: 

\[N_1 = \text{"On input string } w:\]

1. For each way to cut \(w\) into two substrings \(w = w_1w_2\):
   
   (a) Run \(M_2\) on \(w_1\).
   
   (b) Run \(M_2\) on \(w_2\). If both accept, ACCEPT; otherwise continue with the next choice of \(w_1\) and \(w_2\).

2. If \(w\) is not accepted after trying all the possible cuts, REJECT.”

In both the inner loop stages, \(N_1\) use its nondeterminism when the machines being run make nondeterministic steps. \(N_1\) accepts \(w\) iff \(w\) can be expressed as \(w_1w_2\) such that \(N_1\) accepts \(w_1\) and \(M_2\) accepts \(w_2\). Therefore, \(N_1\) decides the concatenation of \(L_1, L_2\). The inner loop stages run in polynomial time and are repeated for at most \(O(n)\) time, so the algorithm runs in polynomial time.

4. Call graphs \(G\) and \(H\) isomorphic if the nodes of \(G\) may be reordered so that it is identical to \(H\). Let \(ISO = \{\langle G, H \rangle \mid G\ \text{and}\ H\ \text{are isomorphic graphs}\}\). Show that \(ISO \in \text{NP}\).

Solution:

A nondeterministic polynomial time algorithm for \(ISO\) operates as follows: “On input \(\langle G, H \rangle\) where \(G\) and \(H\) are undirected graphs:

1. Let \(m\) be the number of nodes of \(G\) and \(H\). If they don’t have the same number of nodes, REJECT.

2. Nondeterministically select a permutation \(\pi\) of \(m\) elements.

3. For each pair of nodes \(x\) and \(y\) of \(G\) check that \((x, y)\) is an edge of \(G\) iff \((\pi(x), \pi(y))\) is an edge of \(H\). If all agree, ACCEPT. If any differ, REJECT.”

Stage 2 can be implemented in polynomial time nondeterministically. Stages 1 and 3 takes polynomial time. Hence \(ISO \in \text{NP}\).