1. If \( L_1, L_2 \in \mathcal{NP} \),
   (i) Is \( L_1 \cup L_2 \in \mathcal{NP} \)?

   Yes. Let \( M_1 \) and \( M_2 \) be respectively the NDTM for \( L_1 \) and \( L_2 \) respectively that take \( p_1(n) \) and \( p_2(n) \) steps on an input of length \( n \). An NDTM \( M \) simulates the non-deterministic TM \( M_1 \) for \( L_1 \) on the input string for at most \( p_1(n) \) steps. Simulating an NDTM implies that \( M \) also exercises the non-deterministic choices of \( M_i \). If it is not accepted, then we simulate \( M_2 \). If it is accepted in any of the phases then we accept it. Clearly \( p_1(n) + p_2(n) \) is bounded by some polynomial. The (non-deterministic) transitions of \( M_i \) are built into the transition function of \( M \).

   (ii) Is \( L_1 \cdot L_2 \in \mathcal{NP} \)?

   Yes. Let \( M_2, M_2 \) be NDTM or \( L_1 \) and \( L_2 \) respectively that take \( p_1(n) \) and \( p_2(n) \) steps on an input of length \( n \). Note that \( w \in L_1 \cdot L_2 \) iff there exists \( w_1 \in L_1 \) and \( w_2 \in L_2 \) such that \( w_1 \cdot w_2 = w \). The NDTM \( M \) can guess (alternately try all possible partitions of \( w \) \( w_1, w_2 \) and simulate \( M_1 \) and \( M_2 \) for at most \( p_1(|w_1|) \cdot p_2(|w_2|) \) steps and accept if both \( w_1 \) and \( w_2 \) are accepted for the chosen partition. Clearly the total number of steps is bounded by a polynomial.

2. Define co-NP complete problems under polynomial time reductions. Is the tautology (the formula is true for all truth assignments) problem co-NP complete?

A co-NP complete language \( L \) is in the class co-NP (i.e. \( \bar{L} \) is in \( NP \)) and for all \( L \in co-NP \), \( L \leq_{poly} L' \). From the definition of the many-one reduction function, the same reduction function works for showing the complement of any NPC language is co-NP Complete.

We will show that \( L_{TT} \) (corresponding to Tautology) is co-NP complete by showing that its complement \( L' = \bar{L}_{TT} \) is NPC. \( L' \in NP \) since we can guess an assignment that makes it false. Note that \( L' \) os the set of all boolean formulae that is false for at least one truth assignment.

Moreover \( L_{SAT} \leq_{poly} L' \) using the reduction \( f(F) = \bar{F} \), i.e., \( F \) is satisfiable iff \( \bar{F} \) is false for at least one assignment. The reduction function is clearly polynomial in length of \( F \) as it only adds a negation.

Note that we do not insist that the formula \( F, \bar{F} \) has to be in 3CNF.