1. Let Q,Q' be problems such that $Q \le Q'$ and $Q' \le Q$. We say that Q and Q' are polynomial-time equivalent. Prove/Disprove: Any two NP-Complete problems are polynomial-time equivalent.

Since Q and Q' are NP-Complete, they are in NP. Moreover, since they are NP-hard, we have $Q \le Q'$ and $Q' \le Q$ by definition of NP-hard-ness. So Q and Q' are polynomial-time equivaalent.

2. DOUBLE-SAT: Decide whether a Boolean formula has at least two satisfying assignments. Prove that DOUBLE-SAT is NP-Complete.

Different parts of an NP-completeness proof

1. DOUBLE-SAT is in NP: For any φ in Accept(DOUBLE-SAT), two different truth assignments of the variables of φ, for which φ evaluates to true construct a succinct certificate. Such a certificate can be verified in time polynomial in n plus the size of φ.

2. A reduction from a known NP-complete problem: Here, we show SAT $\leqslant$ DOUBLE-SAT. Let φ be an instance of SAT. Take a variable y not present in φ, and define $\varphi' = \varphi \wedge (y \vee y')$. Take φ' as the converted instance for DOUBLE-SAT. This reduction should satisfy two properties.

   (a) Doable in polynomial time [Clear in this example]
   (b) Correctness: φ is satisfiable if and only if φ' is double satisfiable.
      If φ is satisfiable, you can take y = T or y = F.
      If φ is not satisfiable, φ' cannot be satisfied (let alone double satisfied) irrespective of the truth assignment of y.

3. A CNF formula is called not-all-equal satisfiable if for some truth assignment of the variables, each clause has at least one true literal and at least one false literal.

NAESAT: Decide whether a Boolean formula in CNF is not-all-equal satisfiable.

Prove that NAESAT is NP-Complete.

NAESAT is in NP: A truth assignment with the stated property is a succint certificate for an instance in Accept(NAESAT).

NP-hard-ness: Use the reduction CNFSAT $\leqslant$ NAESAT as follows. Let φ be an instance for CNFSAT. Convert it to an instance φ' of NAESAT as follows. First, choose a variable y not in φ. Let $l_1 \vee l_2 \vee \cdots \vee l_k$ be a clause in φ. Convert this to the clause $l_1 \vee l_2 \vee \cdots \vee l_k \vee y$ for φ'.

Clearly, this construction can be done in polynomial time.

Correctness: We need to prove that φ is satisfiable if and only if φ' is not-all-equal satisfiable.

[$\Longrightarrow$] Let $t_1, t_2, \cdots, t_n$ be a satisfying truth assignment for φ. Take the truth assignment $t_1, t_2, \cdots, t_n, F$ for φ'.

[$\Longleftarrow$] Let $t_1, t_2, \cdots, t_n, \theta$ be a truth assignment that not-all-equal satisfies φ'. If $\theta = F$, then $t_1, t_2, \cdots, t_n$ satisfies φ. If $\theta = T$, then $t'_1, t'_2, \cdots, t'_n$ is a satisfying truth assignment for φ.

4. Let P be a problem. The complement problem Q satisfies Accept(Q) = Reject(P), and Reject(Q) = Accept(P).

Define coNP = {Q | The complement of Q is in NP}.

Examples of problems in coNP:

(a) Complement of SAT
(b) PRIMALITY
(c) TAUTOLOGY
(d) CONTRADICTION

Q is in coNP if and only if every instance I in Accept(Q) has a succinct disqualification.

(a) Complement of SAT: A truth assignment for which the formula evaluates to true.

(b) Primality: A non-trivial divisor of the given integer.

(c) A formula is called a tautology if it evaluates to true for all truth assignments of its variables.

A truth assignment for which the formula evaluates to false is a succinct disqualification.

(d) A formula is called a contradiction if it evaluates to false for all truth assignments of its variables.

A truth assignment for which the formula evaluates to true is a succinct disqualification.

5. Prove that if any NP-Complete problem is in coNP, then NP = coNP.

Let P be an NP-complete problem which is in coNP.

[NP $\subseteq$ coNP] Take any problem Q in NP. Since P is NP-complete, there is a polynomial-time reduction Q $\leqslant$ P that maps I to J. I is in Reject(Q) if and only if J is in Reject(P). If that is the case, P being in coNP, J has a succinct disqualification D. But then, I also has the same disqualification D. This can be verified by first applying the reduction to get J from I, and then verifying that J is a disqualification for P. Since J can be obtained from I in poynomial time, the size of J is polynomial in the size of I. A polynomial of a polynomial is again a polynomial. So D is succinct for Q, and can be verified (as a disqualification) in time polynomial in the size of I. Therefore Q is in coNP.

[coNP $\subseteq$ NP] Take complements of the problems, and use the other inclusion NP $\subseteq$ coNP already proved.

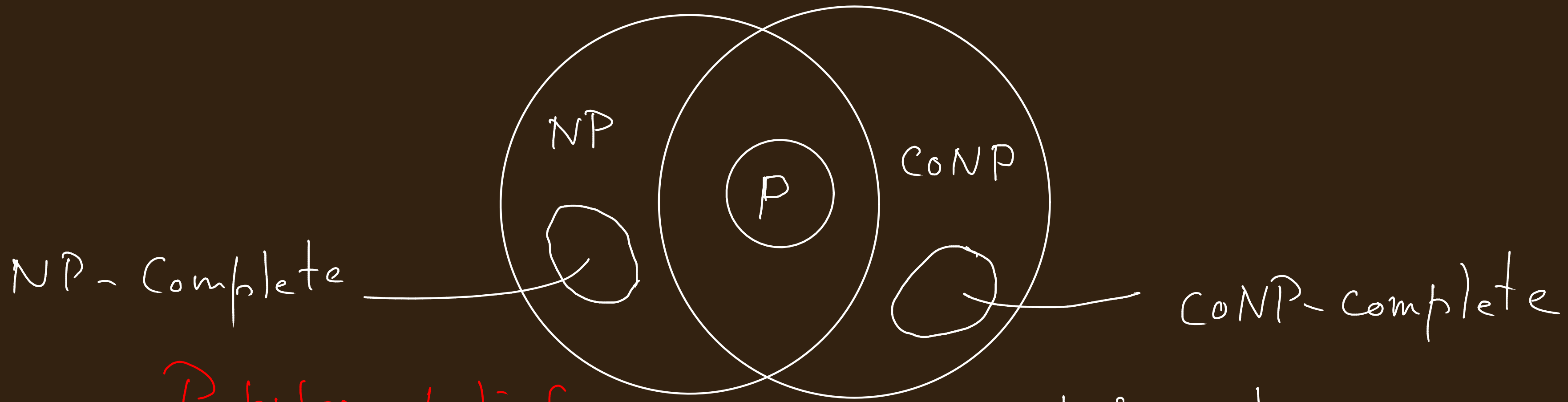
6. Prove that $P \subseteq NP \cap coNP$.

A problem in P can be solved in polynomial time irrespective of the availability of a certificate or disqualification (do not look at it). Stated diffferently, the NULL string can be taken as both a succinct certificate and a succinct disqualification for the problem. Therefore

$$P \subseteq NP$$

and

$$P \subseteq coNP.$$



<u>Popular belief</u>: All the containments are proper.

7. Define coNP-Complete problems.

A problem P is called coNP-complete if

(1) P is in coNP, and
(2) Every problem Q in coNP reduces in polynomial time to P.

Note: A reduction P $\leqslant$ Q is also a reduction $\overline{\text{P}} \leqslant \overline{\text{Q}}$. Therefore coNP-complete problems are precisely the complements of the NP-complete problems.

8. Prove that TAUTOLOGY is coNP-Complete.

We have seen that TAUTOLOGY is in coNP.

The complement of SAT is coNP-complete (because SAT is NP-complete). We use a reduction

$$\overline{\text{SAT}} \leqslant \text{TAUTOLOGY}$$

as

$$\varphi(x_1, x_2, \cdots, x_n) \mapsto \varphi'(x_1, x_2, \cdots, x_n),$$

where φ' is the Boolean complement of φ. Evidently, φ' is a tautology if and only if φ is not satisfiable.