NP-Hard: A problem L is NP-Hard iff satisfiability reduces to L i.e., any nondeterministic polynomial time problem is satisfiable and reducable then the problem is said to be NP-Hard.

NP-Complete: A problem L is NP-Complete iff L is NP-Hard and L belongs to NP (nondeterministic polynomial).

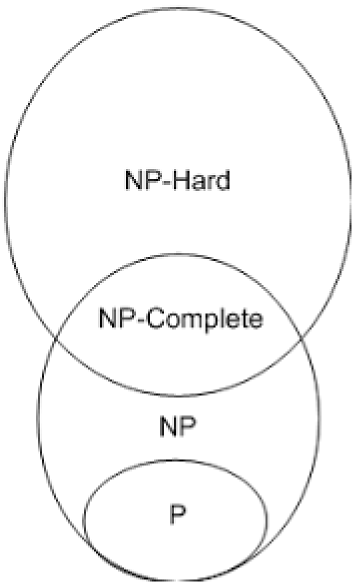
A problem that is NP-Complete has the property that it can be solved in polynomial time iff all other NP-Complete problems can also be solved in polynomial time. (NP=P)

Reducability:

A problem Q1 can be reduced to Q2 if any instance of Q1 can be easily rephrased as an instance of Q2. If the solution to the problem Q2 provides a solution to the problem Q1, then these are said to be reducable problems.

Let L1 and L2 are the two problems. L1 is reduced to L2 iff there is a way to solve L1 by a deterministic polynomial time algorithm using a deterministic algorithm that solves L2 in polynomial time and is denoted by L1α L2.

If we have a polynomial time algorithm for L2 then we can solve L1 in polynomial time. Two problems L1 and L2 are said to be polynomially equivalent iff L1α L2 and L2 α L1.



Cook's Theorem shows that the Boolean Satisfiability Problem (SAT) is NP-complete. To formally connect this to "P = NP," here is the logical reasoning:

1. **Key Insight from Cook's Theorem**:
   - Cook's Theorem states that SAT is NP-complete. That means SAT is one of the hardest problems in NP.
   - If SAT can be solved in polynomial time ($SAT \in P$), then every NP problem can also be solved in polynomial time, making P = NP.
2. **Proof Breakdown**:
   - Suppose $SAT \in P$:
     - By the definition of NP-completeness, every problem in NP can be reduced to SAT in polynomial time.
     - If SAT can be solved in polynomial time, the solution to every NP problem can be derived in polynomial time (via reduction to SAT + solving SAT).
     - Therefore, P=NP.
   - Conversely, suppose P=NP:
     - By definition, every NP-complete problem, including SAT, would belong to P, as all problems in NP would have polynomial-time solutions.
     - Thus, $SAT \in P$.
3. **Conclusion**:
   - SAT is in P if and only if P = NP.

This is why Cook's Theorem is often tied to the P vs. NP question—it identifies SAT as the "representative" problem of NP-completeness. Solving SAT efficiently would imply efficient solutions for all problems in NP.