Note:
$P \subseteq NP$: $L \in P \Rightarrow$
$\exists$ ptime D deciding L
$\Rightarrow \exists$ verifier V verifying L
[V runs D, ignoring the proof string] $\Rightarrow L \in NP$

Open: $P \stackrel{?}{=} NP$

Prop: If any NP-hard language is in P, then $P = NP$.

Proof: Let L be NP-hard and $L \in P$. For any $B \in NP$ have $B \leq_p L$ (def. of NP-hard)
but $L \in P$, so
$B \in P$ [P is closed under p-reducibility]
$\therefore NP \subseteq P$
$\therefore P = NP$. //

Def: 3-SAT is the decision problem:
Instance: A cnf formula φ with exactly 3 literals in each clause, [2 literals may appear in the same clause]
Q: Is φ satisfiable?

3-SAT $\leq_p$ CNF-SAT
trivially by the identity function.
3-SAT is a restriction of CNF-SAT (restricted instance, same question).

Prop: CNF-SAT $\leq_p$ 3-SAT.

Proof: Given a cnf fmla φ, we construct in ptime a 3-cnf formula φ' (3 literals/clause) such that
φ is satisfiable $\Leftrightarrow$ φ' is satisfiable;

Let
$\varphi = \underbrace{C_1 \wedge \cdots \wedge C_k}_{\text{clauses}}$
Get φ' by replacing each clause C of φ by one or more clauses.

Case 1: C has exactly 3 literals. Leave C unchanged.

Case 2: C has 2 literals, say $C = \ell_1 \vee \ell_2$.
Let x be a fresh variable [occurs nowhere else].
Replace C by 2 clauses
$(\ell_1 \vee \ell_2 \vee x) \wedge (\ell_1 \vee \ell_2 \vee \bar{x})$

Case 3: C has 1 literal:
$C = \ell$
Let x, y be fresh boolean vars.
Replace C by
$(\ell \vee x \vee y) \wedge (\ell \vee x \vee \bar{y}) \wedge (\ell \vee \bar{x} \vee y) \wedge (\ell \vee \bar{x} \vee \bar{y})$
An assignment making C true satisfies all replacement clauses (regardless of how x, y are set).
Conversely, any assignment satisfying all 4 replacement clauses must satisfy C.

Case 4: C has ≥ 4 literals:
$C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4 \vee \cdots$

(*) Replace C by 2 clauses: [x is a fresh var]
$(\ell_1 \vee \ell_2 \vee x) \wedge (\bar{x} \vee \ell_3 \vee \ell_4 \vee \cdots)$
Satisfying both replacement clauses $\Leftrightarrow$ satisfying C (by the same assignment restricted to the vars in C)
Repeat (*) on the 2nd clause until you have all clauses having 3 literals.
End construction of φ'.

Summary: From φ (instance of CNF-SAT) constructed φ' (instance of 3-SAT) in ptime such that φ satisfiable $\Leftrightarrow$ φ' is satisfiable

talked through

k-SAT (k literals per clause)

k-SAT is NP-hard for all $k \geq 3$

BUT: 2-SAT $\in P$

[resolution method]