Thm 7.27 [Cook-Levin]: SAT is in P iff P = NP.

SAT is a problem that:
- input a formula \( \varphi \)
- output: YES if \( \varphi \) is satisfiable, i.e. there exists a T/F assignment to the variables s.t. \( \varphi \) is True
- NO otherwise

Alternatively:
\[
\text{SAT} = \{ \varphi \mid \varphi \text{ is satisfiable} \}
\]

Thus, restated (implication \(\Rightarrow\))
if SAT can be solved in polynomial time, then \( P = NP \), i.e. every problem in NP can be solved in poly-time.

notice: SAT \( \in \text{NP} \) because we guess the T/F assignment to the variables and then we verify whether \( \varphi \) is True

PS: most people believe \( P \neq NP \)
Def 7.29: Language $A$ is **polynomial-time reducible** to language $B$, written $A \leq_p B$, if a polynomial-time computable function $f : \Sigma^* \rightarrow \Sigma^*$ exists such that for every $w$, 

$$w \in A \iff f(w) \in B$$

The function $f$ is called **polynomial-time reduction** of $A$ to $B$.

Thm 7.31: If $A \leq_p B$ and $B \in P$, then $A \in P$. 

**Pf:** 

[Diagram showing the reduction process from $A$ to $B$ with $f(w)$ leading to the decision of $B \in P$.]
Thm 7.32: 3SAT is polynomial-time reducible to CLIQUE, where

$$3SAT = \{ \langle \phi \rangle \mid \phi \text{ is a satisfiable 3-cnf formula} \}.$$
Def 7.34: A language $B$ is **NP-complete** if it satisfies both conditions:

- $B$ is in NP, and
- every $A$ in NP is polynomial-time reducible to $B$. 

**Example of a NP-complete problem:**

SAT  
(bec. of the Cook-Levin thm)
Def 7.34: A language $B$ is **NP-complete** if it satisfies both conditions:
- $B$ is in NP, and
- every $A$ in NP is polynomial-time reducible to $B$.

Thm 7.35: If $B$ is NP-complete and $B \in P$, then $P = NP$.

Solving one NP-complete problem in $P$ implies that every problem in NP can be solved in $P$ (polynomial-time).

Thus, NP-complete problems are the "hardest" problems in NP.
Def 7.34: A language $B$ is **NP-complete** if it satisfies both conditions:
- $B$ is in NP, and
- every $A$ in NP is polynomial-time reducible to $B$.

Thm 7.36: If $B$ is NP-complete and $B \leq_p C$ for some $C \in \text{NP}$, then $C$ is NP-complete.
Def 7.34: A language B is **NP-complete** if it satisfies both conditions:

- B is in NP, and
- every A in NP is polynomial-time reducible to B.

Thm 7.37 [Cook-Levin]: SAT is NP-complete.

**Note:** a long list of known NP-complete problems.