NP-Completeness

Thm 7.27 [Cook-Levin]: SAT is in P iff P = 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 $w$,

$$w \in A \text{ iff } f(w) \in B$$

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 $A \in P$. 

$$f \text{- poly time}$$

$$B \text{- poly time}$$

$$\Rightarrow A \text{ is also poly time}$$
Thm 7.32: 3SAT is polynomial-time reducible to CLIQUE, where

$$3SAT = \{ <\phi> \mid \phi \text{ is a satisfiable 3-cnf formula} \}.$$
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.

Note: if $B \in P$, then every other problem in NP is polynomial-time, i.e. $P = NP$. 
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 7.35: If B is NP-complete and B ∈ P, then P = 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.

- $\text{SAT} \leq_p \text{CLIQUE}$
- $\text{SAT} \leq_p \text{HAM. PATH}$
- $\text{HAM. PATH} \leq_p \text{LONGEST PATH}$ (use the same graph, the same $s,t$, set $k = \# \text{vertices}$)

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