

NP-Completeness

A language L is *NP-complete* if

- L is in NP
- L is *NP-hard*: there is a *polynomial time reduction* from any other language in NP L .

The *Cook-Levin Theorem* states that SAT is NP -complete. This implies $SAT \in P$ iff $P = NP$, i.e., SAT is at least as hard as any problem in NP and $SAT \in NP$.

We will prove this theorem later in the course.

Poly-time Reductions

- We want a definition of reduction so that if (1) L is polynomial time reducible to L' and (2) L' is in P , then L is in P .
- A function $f : \Sigma^* \rightarrow \Sigma^*$ is a *polynomial time computable function* if some polynomial time deterministic TM exists which when any input w is input, the TM halts with just $f(w)$ on its tape.
- A language A is *polynomial time reducible* to language B , $A \leq_P B$ if there is a polynomial time computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that:

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

The function f is a *polynomial time reduction* from A to B .

Poly-time Reductions

Theorem: If $A \leq_P B$ and $B \in P$ then $A \in P$.

Proof: Suppose M is a polytime alg. for deciding B and f is a polytime reduction from A to B . Here is a polytime TM to decide A :

1. Compute $f(w)$
2. Run M on $f(w)$ and output whatever M outputs

Why won't the old definition of reduction work?

NP-completeness

Theorem: If B is NP-complete and $B \in P$ then $P = NP$.

Proof: Suppose that B is NP-complete and $B \in P$. Let L be any language in NP. Since B NP-complete, $L \leq_P B$. Since, $B \in P$ we use the previous theorem to conclude that $L \in P$.

So the assumptions imply $NP \subseteq P$. Since it is clear that $P \subseteq NP$, we conclude that $P = NP$.

Proving a Language L is NP-complete

We first show the following

Lemma: If $A \leq_P B$ and $B \leq_P C$, then $A \leq_P C$.

Proof: Let f be a poly-time reduction from A to B and g be a poly-time reduction from B to C . We claim that $f \circ g$ is a poly-time reduction from A to C .

Proof of claim: (Exercise)

Proving a Language L is NP-complete

Theorem: A language L is NP-complete if

1. L is in NP and
2. there is an NP-complete language B and $B \leq_P L$.

Proof: Let L_A be any language in NP. Since B is NP-complete, $L_A \leq_P B$. By assumption (2), $L_A \leq_P L$. Combining this with assumption (1), we can conclude that L is NP-complete.