NP Completeness/Polynomial time reductions

05 April 2021 08:51

Def 7.28: $f : \Sigma^* \to \Sigma^*$ is a polynomial time computable function if there is some poly time DTM M that takes input w, writes $f(w)$ on tape and halts.

Def 7.29: language A is polynomial time reducible to language B, denoted $A \leq_p B$ if $\exists$ poly-time computable function f such that, $\forall w \in \Sigma^*$,

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

f is called the polynomial time reduction from A to B

The Goal: A way to decide A efficiently

A to B)

(1) $A \longrightarrow B$ (transform A to B)

(2) Decide B

We will show $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

It is important to have the restriction on the reduction. If not, we could test all the 2^n assignments possible, of a SAT instance ϕ.

$$f(\phi) = \begin{cases} 1 & \text{if satisfiable} \\ 0 & \text{otherwise} \end{cases}$$

We have an easy reduction from SAT to $\{1\}$ if we don't impose restrictions on the power of the reduction function.

Theorem 7.31: $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

Proof: Suppose there is a poly time algorithm for B. We have the following decider for A.

M for B. We have the following decider for A.

Alg for A: On input w

(1) Compute $f(w)$.

(2) Run M on $f(w)$. Accept iff M accepts $f(w)$.

Correctness: Easy

Time: (1) and (2) are both poly time.

<u>Other results</u>

(1) $A \leq_p B$ and $B \in NP \implies A \in NP$

(2) $A \leq_p B$ and $A \notin P \implies B \notin P$

(3) $A \leq_p B$ and $B \leq_p C \implies A \leq_p C$

(4) $A \leq_p B \implies \overline{A} \leq_p \overline{B}$.