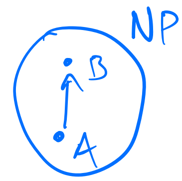
NP-Completeness



Def 7.34: Language B is NP-complete if

(1) $B \in NP$, and

(2) $\forall A \in NP$, $A \leq_p B$.

We can think of NP-complete problems as the "hardest" problems in NP.

Remarks: (1) Suppose B is an NP-complete language.

If $B \in P$, then $\forall A \in NP$, we have $A \in P$. That is $P = NP$.

We know if $A \leq_p B$ and $B \in P \Rightarrow A \in P$.

Since B is NP-complete, $\forall A \in NP$, we have $A \leq_p B$.

Since it is widely believed that $P \neq NP$, it is unlikely that any NP-complete problem is in P.

(2) If B is NP-complete, $C \in NP$ and $B \leq_p C$, then C is NP-complete.

By assumption $C \in NP$. Condition (1) is met.

$\forall A \in NP$, $\underline{A \leq_p B}$ and we have $\underline{B \leq_p C}$.

$\underline{A \leq_p C}$ → Condition (2) is also met.

There are several NP-complete languages known. Many of which we have already come across.

Examples: CLIQUE, 3-SAT, SUBSET-SUM, 3-COLORABLE.

How do we show that a given language is NP-complete?

<u>Cook-Levin Theorem</u>: SAT is NP-complete.