Theory of Computation, CSCI 438 spring 2023
Class nondeterministic polynomial time, NP, pg. 292-298
NP-Completeness, pg. 299-311
April 27

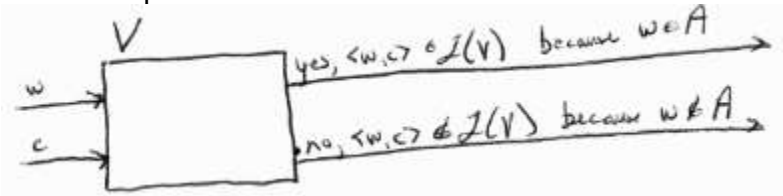
NP

Idea of P is that all "reasonable deterministic computation models are polynomial equivalent". NP problems appear to be harder, but possibly P=NP.

Exponential problems typically solve a problem by brute force, searching every possible branch, while polynomial time algorithms are using some insight into the problem.

Definition 7.18 (page 293) A verifier for a language A is an algorithm V, where
A = {w | V accepts <w,c> for some string c}.
We measure the time of a verifier only in terms of the length of w, so a polynomial time verifier runs in polynomial time in the length of w. A language A is polynomially verifiable if it has a polynomial time verifier.

Verifiers (V in the above) use additional information, c, to verify that string w is a member of A. The additional information, c, is called a certificate, or proof of membership in A.



Def. 7.19 (page 294) NP is the class of languages that have polynomial time verifiers.

Theorem 7.20 (page 294) A language is in NP iff it is decided by some nondeterministic polynomial time Turing machine.

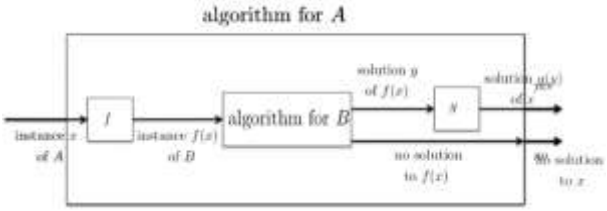
NP-Completeness

Definition 7.29 (page 300)
Language A is *polynomial time mapping reducible*, or simply *polynomially time reducible*, to language B, written $A \leq_p B$, if a polynomial time computable function $f:\Sigma^* \rightarrow \Sigma^*$ exists, where for every w,

$w \in A \Leftrightarrow f(w) \in B$.

The function f is called the polynomial time reduction of A to B.

$A \leq_p B$:



Theorem 7.31 (page 301)
If $A \leq_p B$ and $B \in P$, then $A \in P$.

Def. 7.34 (page 304)
A language B is NP-complete if

1. $B \in NP$
2. $\forall$ language $A \in NP$, A is polynomial time reducible to B ($A \leq_p B$)

NP-complete problems are often considered the hardest problems in NP, and they may be, but more specifically they are the most general/flexible problems in NP, such that all other NP problems can be reduced to them.

Theorem 7.35 (page 304)
If B is NP-complete and $B \in P$, then P=NP.

Theorem 7.36 (page 304)
If B is NP-complete and $B \leq_p C$ for C in NP, then C is NP-complete.