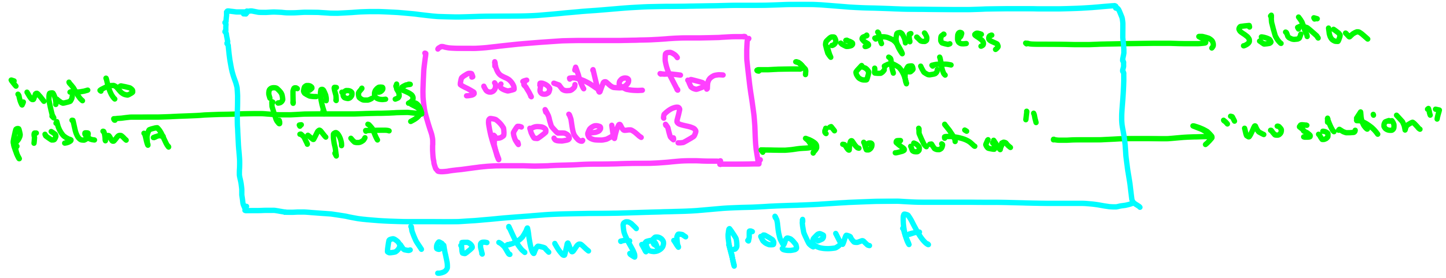
Section 23.6: NP-Completeness

Levin Reductions



Note: defined only when A, B both search problems.

Required format: (1) Given instance I of A, construct instance I' of B. (in polynomial time)

(2) Invoke subroutine on I'.

(3) If subroutine returns "no solution," return "no solution."

(4) " feasible solution to I', convert it in poly-time to a feasible solution to I.

The Hardest Problems in NP

Recall: A problem B is NP-hard if, for every problem A in the complexity class NP, there is a (Cook) reduction from A to B.

Definition: A search problem B is NP-complete if:

1. for every problem A in NP, there is a Levin reduction from A to B.
2. B is a member of the class NP.

Warning: many books define NP-completeness using only decision problems and Karp reductions.

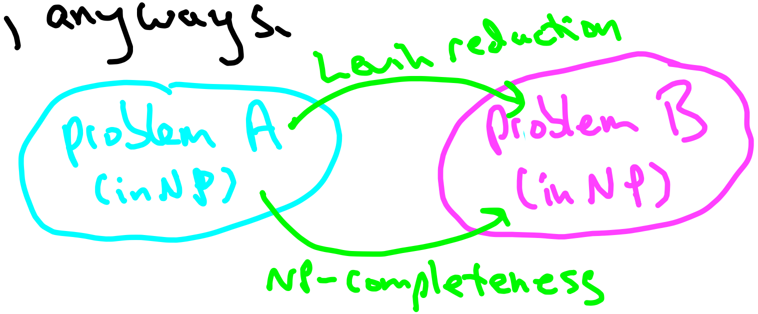
(algorithmic implications the same either way)

Existence of NP-Complete Problems

Cook-Levin Theorem (stronger version): 3-SAT is NP-complete.

Reason: proof only used a Levin reduction, anyways.

How to prove a problem B is NP-complete:



1. Prove that B is an NP problem.
2. Choose an NP-complete problem A.
3. Prove that there is a Levin reduction from A to B.

[see Garey-Johnson book for 300+ examples]