Reductions

problem or A — reduction — B — solution/algorithm

A reduction states that B "knows" all solutions of A. So B is at least as complex as A or possibly harder.

Decidable — P (solvable) — NP — NP-complete — NP-Hard

A problem is NP-Hard if it can be transformed (reduced) from all NP problems. Thus, if an NP-Hard problem has an NP or harder solution, then all NP problems can be solved in NP time.

NP-complete: A problem is NP-complete if it can simulate any problem in NP. (these problems are reducible to each other!) A poly-time solution to one implies it for all!

Graph isomorphism: $G \cong G'$

	1	2	3
1	0	1	1
2	1	0	1
3	1	1	0

	x	y	z
x	0	1	1
y	1	0	1
z	1	1	0

adjacency matrices

Are the two graphs actually the same with different vertex labels? (is $G \cong G'$?)
(is their adjacency matrices give the same graph drawing?)

Note: If problem can be solved in polynomial-time on a nondeterministic TM (i.e. it's in NP) then immediately any solution can be verified quickly on a deterministic TM.
This is obvious: The computation trace of the NTM has polynomial-length and a standard computer can easily check if the solution is valid since all the correct transitions are already given.

Reductions

Satisfiability Problem: SAT = $\{\langle\psi\rangle \mid \psi$ is a satisfiable boolean expression$\}$.

$\psi = (\overline{x} \vee y) \wedge (x \vee \overline{z})$ — literals (true or false) — clause

$x = F$
$y = T$
$z = F$
$(T \vee T) \wedge (F \vee T)$
$(T) \wedge (T)$
$\psi = T$

SAT was the first NPC problem.
3SAT: Each clause has 3 literals.

Recall CLIQUE: Does there exist a clique of size k in a given graph.

Theorem 7.32 3SAT is polynomial-time reducible to CLIQUE.

Proof: Let ψ be a SAT expression with k clauses. Organize k triples, each corresponding to a clause in ψ, to form graph G.

$\psi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_k \vee b_k \vee c_k)$.

Ex: if $\psi = (x_1 \vee \overline{x}_1 \vee x_2) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee x_2 \vee x_2)$

We will transform every instance of 3SAT to some instance of CLIQUE

Here is an example reduction:

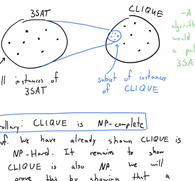


Each literal in ψ has a vertex in G. Now, add all edges to G except between the same triple or between two nodes with contradictory labels, such as x_1 & $\overline{x}_1$.

$\Rightarrow$ Suppose ψ is satisfiable. So one literal must true in each clause. In each triple of G, we select one node corresponding to a "True" literal. The nodes selected form a k-clique.

$\Leftarrow$ Suppose G has a k-clique. No two vertices of the clique appear in the same triple. We make each literal corresponding to the vertex true. Thus, ψ is satisfiable. □

We have shown CLIQUE is NP-Hard.



-A polynomial-time algorithm to CLIQUE would immediately imply a poly-time solution to 3SAT.

Corollary: CLIQUE is NP-complete.

Proof: We have already shown CLIQUE is NP-Hard. It remains to show CLIQUE is also NP. We will prove this by showing that a solution to an instance of CLIQUE can be verified in polynomial-time via a deterministic computer.

Given a graph G, let $S \subseteq G$ be a given solution. It is easy to iterate through the vertices of S and check that all pairs are adjacent and $|S| \geq k$. □



Independent Set: A subgraph of a graph G s.t. each pair of vertices is not adjacent.

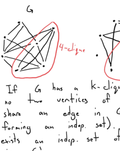


IS: Does there exist an indep. set of size k in a graph G? $k=4$?

Theorem: CLIQUE is polynomial-time reducible to IS.

Proof: Given a graph G, we construct $G' = (V', E')$ s.t. $V' = V$ and $E' = \overline{E}$.

Ex:



$\Rightarrow$ If G has a k-clique, then no two vertices of the set shown an edge in G' (thus forming an indep. set). Hence, there exists an indep. set of size k in G'.

$\Leftarrow$ If G' has an indep. set of size k, then all of its vertices are adj. in G, giving a k-clique. □

We have that IS is NP-Hard.

Corollary: IS is NPC.

Proof: Given a graph $G = (V, E)$, a solution to IS $H \subseteq G$, it can be verified that H is an indep. set in $O(|V|+|E|)^2)$ by iterating through the pairs of vertices in H and checking that they are all not adjacent. □
Since IS is in NP and NP-Hard, it is NP-complete. □

