

NP-Completeness Proofs by Restriction

- Techniques for proving NP-Completeness fall generally into the following three categories :
 - 1) Proof by restriction
 - 2) Proof by local replacement
 - 3) Proof by component design

PROOFS BY RESTRICTION

- As you will recall, to prove that a new decision problem Π is NP-Complete means that you must first show that $\Pi \in NP$, and then you must construct a polynomial transformation from a known NP-Complete problem Π' to Π .
 In a proof by restriction, you want to show that the new problem Π contains the known NP-Complete problem Π' as a special case.
 In other words, the polynomial transformation consists of placing restrictions on the instances of ~~the~~ the new problem Π so that ^{they} look like the instances of Π' . In the rest of this lecture, we will consider some cases of this type of proof.

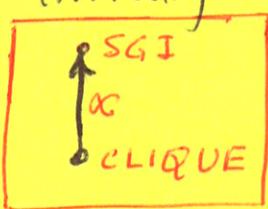


SUBGRAPH ISOMORPHISM (SGI) :

INSTANCE : Two graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$

QUESTION : Does G contain a subgraph isomorphic to H ? That is, does there exist in G a subset $V \subseteq V_1$ of vertices and a subset $E \subseteq E_1$ of edges such that $|V| = |V_2|$ and $|E| = |E_2|$, and there exists a one-to-one function $f: V_2 \rightarrow V$ satisfying $\{u, v\} \in E_2$ iff $\{f(u), f(v)\} \in E$?

PROOF : It is trivially established that SGI $\in NP$. The rest of the proof consists of showing :



Recall from Lecture 30 that we already know that CLIQUE is an NP-Complete problem.

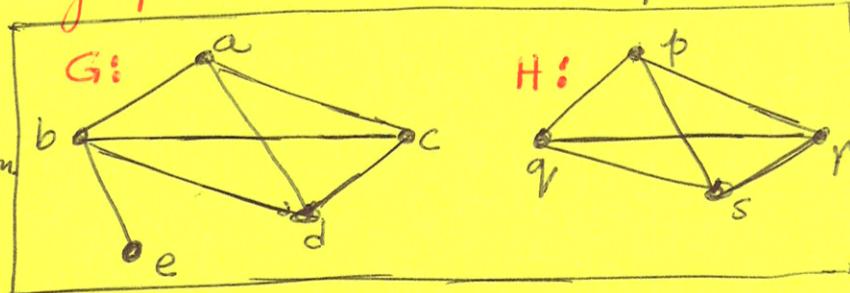
CLIQUE :

INSTANCE : A graph $G = (V, E)$ and a +ve integer J .

QUESTION : Does G contain a clique of size J or greater?

Let's now place the following restriction on the SGI instances : We will consider only those instances in which H is a complete graph. Here is an example :

- It is obvious that any clique of size 4 in G in the SGI instance shown at right would be isomorphic to the H that is also shown.
- So such SGI instances can be solved by using the solution to the CLIQUE problem with $K = |V_2|$.



KNAPSACK :

INSTANCE : A finite set U , a size function $s(u) \in \mathbb{Z}^+$ and a value function $v(u)$ for each $u \in U$, a size constant $B \in \mathbb{Z}^+$ and a value goal $K \in \mathbb{Z}^+$.

QUESTION : Is there a subset $U' \subseteq U$ such that $\sum_{u \in U'} s(u) \leq B$ and $\sum_{u \in U'} v(u) \geq K$?

Imagine a robber with a knapsack of size B who has broken into a jewelry store. The robber wants to fill his knapsack so that the value of the items stolen is maximized.

PROOF: Using the method of proof by reduction, the NP-Completeness of KNAPSACK can be established by : We already know that PARTITION is NP-Complete. We now want to show



that PARTITION is just a special case of KNAPSACK. Recall from Lectures 29 and 30 that an instance of PARTITION is specified by a set U and a size function $s(u) \in \mathbb{Z}^+$. The PARTITION question is whether there exists a subset $U' \subseteq U$ such that $\sum_{u \in U'} s(u) = \sum_{u \in U - U'} s(u)$? If in the KNAPSACK instance we set $s(u) = v(u)$ and $B = K = \frac{1}{2} \sum_{u \in U} s(u)$, the decision question for such a KNAPSACK instance becomes : Does there exist a subset $U' \subseteq U$ such that $\sum_{u \in U'} s(u) \leq B$ and $\sum_{u \in U'} s(u) \geq B$? This amounts to asking whether $\sum_{u \in U'} s(u) = B$, which is nothing but the decision question in PARTITION.

MULTIPROCESSOR SCHEDULING (MS)

INSTANCE : A finite set A of tasks, a length $l(a) \in \mathbb{Z}^+$ for each $a \in A$, a number m of processors, and a deadline $D \in \mathbb{Z}^+$.

QUESTION : Does there exist a partition $A = A_1 \cup A_2 \cup \dots \cup A_m$ of A into m disjoint sets such that

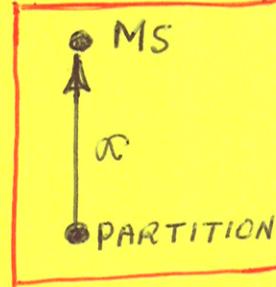
$$\max \left\{ \sum_{a \in A_i} l(a) \right\} \leq D$$

Obviously, the point here is that all tasks in A_1 are assigned to processor 1, all tasks in A_2 to processor 2, and so on, in such a way that we want all tasks to finish running before the deadline.

PROOF : It is easily shown that when $m=2$ and $D = \frac{1}{2} \sum_{a \in A} l(a)$, this restricted version of MS instance we obtain is nothing but an instance of the known NP-Complete problem PARTITION.

With $m=2$ and $D = \frac{1}{2} \sum_{a \in A} l(a)$, the MS instance we get has the following decision question : Does there exist a subset $A' \subseteq A$ such that

$$\max \left\{ \sum_{a \in A'} l(a), \sum_{a \in A - A'} l(a) \right\} \leq D$$



setting $D_1 = \sum_{a \in A'} l(a)$ and $D_2 = \sum_{a \in A - A'} l(a)$, the above decision question boils down to $\max\{D_1, D_2\} \leq D$ under the constraint $D_1 + D_2 = 2D$. There exists only one solution here : $D_1 = D_2 = D$. (You can prove this by trying $D_1 = D+E$ and $D_2 = D-E$.) With $D_1 = D_2 = D$, we are looking for $A' \subseteq A$ such that $\sum_{a \in A'} l(a) = \sum_{a \in A - A'} l(a)$, but that is the same as PARTITION.

BOUNDED DEGREE SPANNING TREE (BDST)

INSTANCE : A graph $G = (V, E)$ and a +ve integer $K \leq |V| - 1$

QUESTION : Is there a spanning tree for G in which no vertex has degree exceeding K ?

[A tree is a connected graph that does not contain cycles. A graph is connected when there exists a path from any one vertex to every other vertex. Note that when we say a graph is connected, it does not mean the graph is fully connected.] The decision question amounts to saying that there exists a subset $E' \subseteq E$ such that $|E'| = |V| - 1$, the graph $G' = (V, E')$ is connected, and no vertex is included in more than K edges in E' .

PROOF : The proof consists of noting that a restricted version of BDST is the same as the HAMILTONIAN PATH problem. The Hamiltonian Path problem is the same as the Hamiltonian Circuit problem except that we drop the requirement that there be an edge between the first and the last vertices. When $K=2$, the spanning tree becomes a Hamil. Path.

The proof of Hamiltonian Path being NP-comp. is only a slight extension of the proof of NP-comp. for Hamilt. Ckt.

