2 Apr 2021

Announcement

CS research night Apr. 8 5:30–7:30 over Discord.

https://discord.com/invite/cP6HKGqD

Learn about research in CS at Cornell, opportunities to get involved.

Def. A decision problem A belongs to NP if $A \le_p 3SAT$.

A is NP-hard if $3SAT \le_p A$.

A is NP-Complete if both NP and NP-Hard:

$A \le_p 3SAT \le_p A \qquad A \equiv_p 3SAT$

NP and efficient verification

Problems like 3SAT and INDEP SET have the property that (we believe) it's hard to find a solution but easy to verify a solution is correct.

__Ex.__ in 3SAT if someone tells you a truth assignment of $x_1, \ldots, x_n$ it's easy to check every clause is satisfied.

In ind set if someone gives you a set of vertices in a graph, it's easy to check there are k of them, all distinct, no edges btw them.

__Def.__ Decision problem A belongs to NP if and only if there exists a poly-time algorithm ("verifier") V with two inputs x, y ("problem" & "solution") such that

$$A(x) = \text{TRUE} \iff \exists y \text{ s.t. } |y| \le \text{poly}(|x|)$$
$$\text{and } V(x,y) = \text{TRUE}$$

__Theorem.__ (Cook-Levin) The two definitions of NP given above are equivalent. A has a poly-time verifier if and only if $A \le_p$ 3SAT.

Observation. Problem A is NP-Hard if and only if $\exists$ an NP-Hard B s.t. $B \leq_p A$.

Proof. Def of NP-Hard is $3SAT \leq_p A$. So if $\exists$ NP-Hard B s.t. $B \leq_p A$ then $3SAT \leq_p B \leq_p A$, by transitivity $3SAT \leq_p A$ $\therefore$ A NP-Hard. Conversely if A is NP-Hard then $A \leq_p A$ so $\exists$ NP-Hard B, namely $B = A$, satisfying $B \leq_p A$.

Practical advice. When attempting to show A is NP-Hard, it helps to pick a known NP-Hard B that has some similarity to A and try reducing from B to A.

Ex. If A says "Does there exist a structure that satisfies a list of constraints?" try $3SAT \leq_p A$.

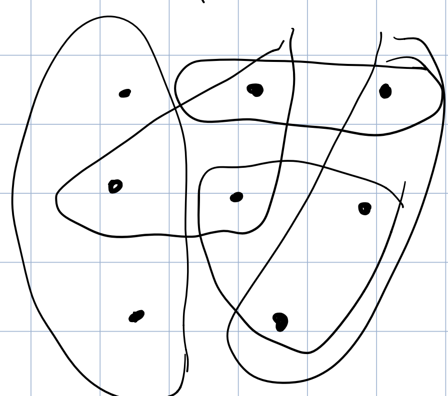
If A says "Does there exit a set of at least k things that satisfies some constraints?" try $\text{IND SET} \leq_p A$.

Similarly "at most k" means try
VTX COVER $\leq_p$ A.

Known NP-Hard problems at this point in 4820:

3SAT INDEP SET
CLIQUE VTX COVER

Some new problems coming from set theory.



Given a set U with n elements and a collection of subsets
$S_1, S_2, \ldots, S_m \subseteq U \ldots$
and a positive integer $k \ldots$

SET COVER: Can you find k of the sets whose union is U?

SET PACKING: Can you find k of the sets that are all disjoint from each other?

Reducing INDEP SET to SET PACKING.

- Given $G=(V,E)$ and $k>0$
  (an instance of INDEP SET)
- Construct $\mathcal{U}=E$.
- For each $v\in V$ create set S_v consisting of all edges that have v as an endpoint.
- Use the same k.

Does this work?

($\Rightarrow$) if G has a k-element indep set, I, then there are k mutually disjoint subsets in our SET PACKING instance.
Yes! $\{S_v \mid v\in I\}$.

($\Leftarrow$) if SET PACKING has k mutually disjoint sets $S_{v_1}, S_{v_2}, \ldots, S_{v_k}$ does G have a k-element indep set?
Yes! $\{v_1, \ldots, v_k\}$.

The same reduction reduces VERTEX COVER to SET COVER.

Don't reduce in the wrong direction!

E.g. when trying to show SET COVER NP-Hard, don't start by reasoning "Let's say I had a set cover problem. How would I transform it into a graph?"

Instead say "Let's say I had a graph representing a VTX COVER problem. How would I transform it into SET COVER?"