$\boxed{\text{11 Oct 2023}}$ Reductions and NP-Completeness

Suppose V is a poly-time algorithm with two inputs, x, y.

Say when $|x| = n$ (length of x)

then $|y| = f(n)$ $f(n) \in poly(n)$. $\exists\, P(n)$, a polynomial func of n s.t. $\forall n \in \mathbb{N}$ $f(n) \le P(n)$.

If $*$ is some assoc, comm. binary operation on range(V), we get a new problem

$$*V(x) = \text{given } x, \text{ compute } \underset{y \in \{0,1\}^{f(n)}}{*} V(x,y).$$

Ex. If $V(x,y)$ takes values in $\{0,1\}$ (0 = false, 1 = true)

then

$$\text{OR-}V(x) = \text{given } x, \text{ compute } \bigvee_{y \in \{0,1\}^{f(n)}} V(x,y)$$

$$= \text{given } x, \text{ does } \exists y \text{ s.t. } V(x,y) = 1?$$

Ex. Bipartite Perfect Matching

x = binary encoding of a bipartite adj' matrix.

y = binary encoding of adj matrix of a matching

$V(x,y)$ = check that entries of y are $\{0,1\}$.

row, column sums of y are exactly 1

$y_{ij} \le x_{ij}$ $\forall i,j$.

Ex. Min-Cost bipartite perfect matching can be represented almost the same way, as

$\text{MIN-}V(x)$

where x = binary encoding of bipartite graph with edge costs

y = same as before

$V =$ check if y is a perf matching contained in x.

yes $\to$ output $\sum$ edge costs in y

no $\to$ output ∞

For the first V above, $\oplus V$ counts if # perfect matchings is even or odd.

(can be solved in poly-time using determinants)

$$+V(x) = \sum_{y \in \{0,1\}^{f(n)}} V(x,y)$$

$$= \#\text{ of perfect matchings in } X$$

This is a complete problem for #P.

Def: NP is the class of problems $OR\text{-}V(x)$, where V is a poly-time algorithm with $\{0,1\}$ output.

coNP is the class problems $AND\text{-}V(x)$.

Stereotypical NP, coNP problems:

- Given a Boolean formula, is it satisfiable? (NP)
- Given a Boolean formula, is it a tautology? (coNP)
- Given a graph G and a parameter $k \in \mathbb{N}$, does G contain a clique of size k? (NP)
- Given G and k, is <u>every</u> clique smaller than k vertices? (coNP)

Reductions: A (poly-time Karp) reduction from problem A to problem B is a function R s.t.

$$\forall x \quad A(x) = B(R(x))$$

Informally, R is poly-time alg that lets you solve A by calling a subroutine to solve B once, and outputting the result of that subroutine call.

If such a reduction exists we write $A \leq_p B$.

A problem H is NP-Hard if $A \leq_p H$ for every $A \in NP$, and H is NP-Complete if it is NP-Hard and it belongs to NP.

Equivalently, NP-completeness of H means it is a maximal element of NP under $\leq_p$.

Thm (Cook-Levin): NP-Complete problems exist. In fact, 3SAT is NP-complete.

3SAT: given (a binary string representing) a Boolean formula in the form $\bigwedge_{i=1}^{m} C_i$ where each clause C_i is a disjunction of ≤ 3 Boolean literals, is there a satisfying truth assignment?

Ex. $(\bar{x}_1 \vee x_2) \wedge (x_3 \vee x_1 \vee \bar{x}_2) \wedge (\bar{x}_3 \vee x_2)$

satisfied by $x_1 = 0, \ x_2 = 1, \ x_3 = 1$.

When someone hands you a decision problem D, and you suspect it's NP-complete, try:

1. find a polytime verifier for D, i.e. find $V(x,y)$ s.t. $D(x) = \text{OR-}V(x)$.

   (usually easy)

2. find a problem H already known to be NP-Hard, and show $H \leq_p D$.

   This requires reducing FROM H TO D.

   In other words the reduction transforms an instance of the known hard problem H to the new problem.

Ex. 4SAT $\leq_p$ 3SAT

Given $\phi = \bigwedge_{i=1}^{m} C_i$ $\qquad C_i = x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4$

$\Downarrow$

$C_i^1 : \; x_1 \vee \bar{x}_2 \vee z_i$ $\qquad C_i^1 \wedge C_i^2$ is logically equiv to C_i.

$C_i^2 : \; \bar{z}_i \vee x_3 \vee \bar{x}_4$