Lecture 22 NP-hard problems

Wednesday, April 15, 2020 2:57 PM

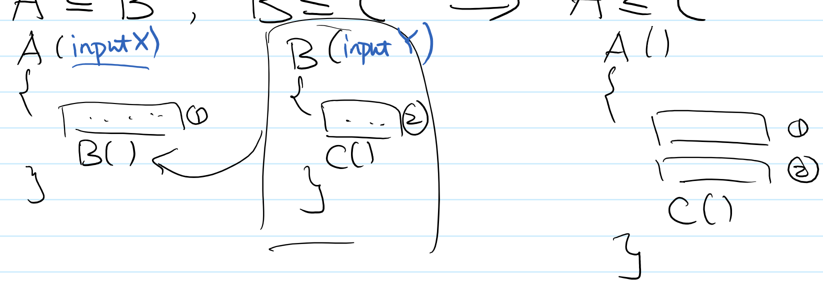
- polynomial time reduction
  - A to B : spend $\boxed{\text{poly time}}$ on input X of A to prepare an input Y for B, then $\boxed{\text{return } B(Y)}$
  - poly time: if there is a poly time reduction from A to B
    if B has a poly time algorithm, then
    A has a poly time algorithm.
    (if $B \in P$, $A \in P$)
  - no post processing : related to definition of NP.

    NP — answer = <u>YES</u> $\exists$ solution s.t. verifier accepts
    answer = <u>NO</u> no solution that verifier accept

- NP-hard problems
  - Problem B is <u>NP-hard</u>, if for any problem $A \in NP$ there is a polynomial time reduction from A to B
  - Problem B is <u>NP-complete</u>, if B is NP-hard, and B is in NP.
  - polynomial time reduction from A to B

    $$\underline{A \le B}$$

  - $A \le B$, $B \le C$ $\Rightarrow$ $A \le C$



  - if NP-hard problem B is in P
    then since every NP problem $A \le B$, $A \in P$
    $\Rightarrow P = NP$

- if both A and B are NP-complete, then if one of them has a poly time algorithm, the other one also has a poly time algorithm.

  B is NP-complete $\Rightarrow$ B is NP hard

  A is NP-complete $\Rightarrow$ A is in NP

  $A \leq B$

  $B \leq A$

- prove a problem B is NP-hard.
  - find an NP-hard problem A
  - do a poly time reduction from A to B

$$A \leq B$$

$$\text{for any } C \in NP \quad C \leq A$$

$$C \leq B$$

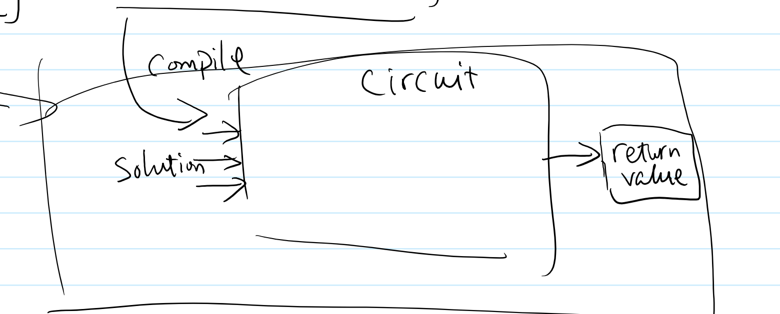
- Cook-Levin Theorem
  - CIRCUIT-SAT problem is NP-hard.
  - For any NP problem A (with $\boxed{\text{input } X}$), there is a poly time

```
Verifier(input X, solution)
{

}
```

```
Verifier (solution)
  ...
      return true
  return false.
```

answer to A for input X = $\begin{cases} \text{YES} & \exists \text{ solution Verifier returns True} \\ \text{NO} & \text{for all solution verifier returns False} \end{cases}$



run CIRCUIT-SAT on this

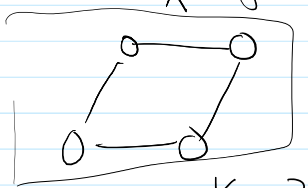
- reduction
  - INDEP SET to CLIQUE

① transform X for INDEP SET $\Longrightarrow$ Y for CLIQUE

- observation: INDEP SET want vertices to have <u>no edges</u>

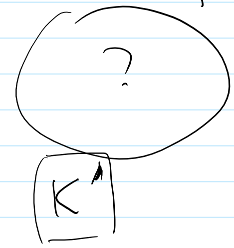
CLIQUE want vertices to be <u>connected</u>.

X for INDEP SET

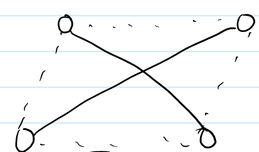


K = 2

Y for CLIQUE



- idea: flip edges



$\boxed{K'} = K = 2$

in first step, cannot use the solution to the INDEP. SET problem

② X is YES $\Longrightarrow$ Y is YES

$\boxed{\text{X has indep. set of size K}}$. Y has clique of size K

<u>solution for X</u> $\Longrightarrow$ solution for Y

Claim: every independent set of the original graph is a clique in the new graph.

③ X is NO $\Longrightarrow$ Y is NO

$\boxed{\text{Y is YES}} \Longrightarrow$ X is YES

<u>solution for Y</u> $\Longrightarrow$ solution for X

Claim: every clique in the new graph is an indep. set in the original graph.

- 3-SAT

$(\underline{x_1 \vee x_2 \vee \overline{x_3}}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee x_4)$

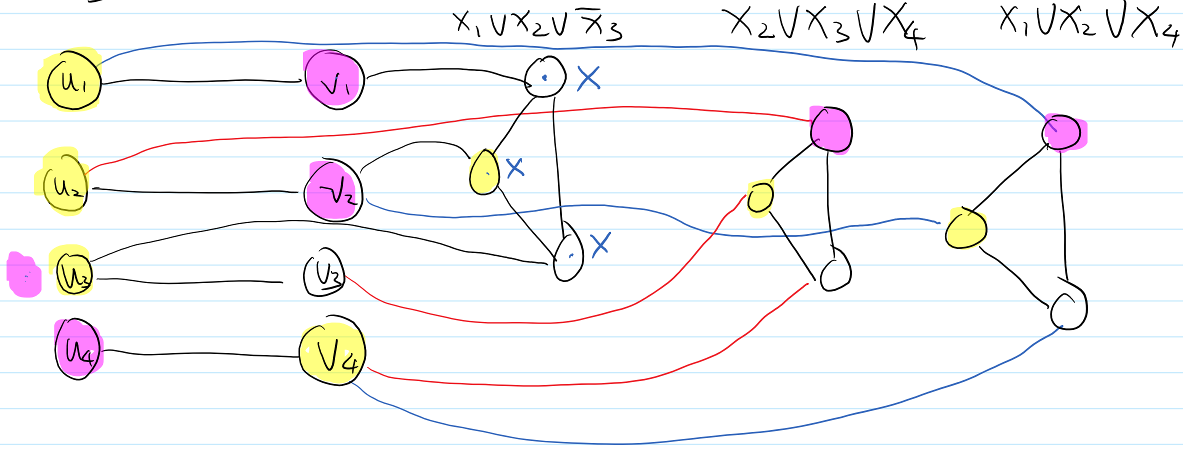
Answer: Yes $x_1 = \text{true}$ $x_2 = \text{true}$

$x_3 = \text{true} \quad x_4 = \text{anything}$

$(x_1 \vee x_2) \wedge (\overline{x_1}) \wedge (\overline{x_2})$

Answer: No

- reduction 3-SAT → INDEP. SET



K: intuition: for a satisfying assignment
can choose 1 vertex from every variable gadget
choose 1 vertex from every clause gadget

$K = n + m$

step 2: 3-SAT YES ⟹ INDEP SET YES