Lecture 19

CSE 431
Intro to Theory
of Computation

NP-complete Problems:

$\text{CIRCUIT-SAT} \le_m^P \text{3SAT} \le_m^P \text{CNFSAT} \le_m^P \text{INDEPENDENT-SET} \le_m^P \text{CLIQUE}$

$\text{3SAT} \le_m^P \text{EXACT-3SAT} \le_m^P \text{3COLOR}$

$\text{INDEPENDENT-SET} \le_m^P \text{VERTEX-COVER}$

Thm C is NP-complete iff (1) $C \in NP$ (2) $B \le_m^P C$ for some NP-complete B

Proof Given that $C \in NP$ only need to show C is NP-hard
By (2) $\forall A \in NP$, $A \le_m^P B$ but then $A \le_m^P B$ and $B \le_m^P C \Rightarrow A \le_m^P C$
so C is NP-hard too ∎

How to structure a proof that C is NP-complete

(1) $C \in NP$: For input x
(a) Give the form of certificate for x and argue poly length
(b) Give algorithm to verify certificate and argue poly time

and then

(2) C is NP-hard: Choose known NP-complete problem A and write

Claim $A \le^P_m C$

want f s.t. $x \mapsto f(x)$
$x \in A \iff f(x) \in C$

(a) Define function f

(b) Argue f computable in polytime

(c) Correctness:
Argue $x \in A \iff f(x) \in C$.
Usually best argument is of following form since both $A, C \in NP$ and have polytime verifiers

(i) $x \in A \Rightarrow f(x) \in C$:
Since $x \in A$ there is a certificate y for $x \in A$

input $x \mapsto f(x)$
certificate $y \qquad y'$

Show how to use y to build certificate y' for $f(x) \in C$

(ii) $f(x) \in C \Rightarrow x \in A$
do the reverse. Given certificate y'' for $f(x) \in C$ show how to get get certificate y''' for $x \in A$

note only need y' for special form $f(x)$ not general inputs to C

Correctness example:

CIRCUIT-SAT $\langle C \rangle \longmapsto$ 3SAT $\langle \varphi \rangle$

certificate $\Rightarrow$ y assignment s.t. $C(y)=1$ $\longmapsto$ $y' = (y, z)$ z = gate values on input y

$\Leftarrow$ just take input values in y'' $\longleftarrow$ Given y'' making φ true

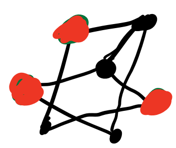
---

Easy reduction: $3SAT \le_m^P CNFSAT$: f: identity (if input not of right form map to garbage)

INDEPENDENT-SET = $\{\langle G, k\rangle :$ G is a graph with an independent set of size $\ge k\}$

where

<u>Def^n</u> For a graph $G=(V,E)$, $U \subseteq V$ is <u>independent</u> iff no pair of vertices in U joined by an edge



independent

<u>Thm</u> INDEPENDENT-SET is NP-complete

<u>Proof</u> 1) $\in$ <u>NP</u>: <u>Certificate</u> for G: set of vertices U forming an independent set of size k (length $\le$ length of encoding of the graph) so poly size

<u>verify</u> Check that
- no edges between elt^s of U
- $|U| \ge k$.

both easily polytime

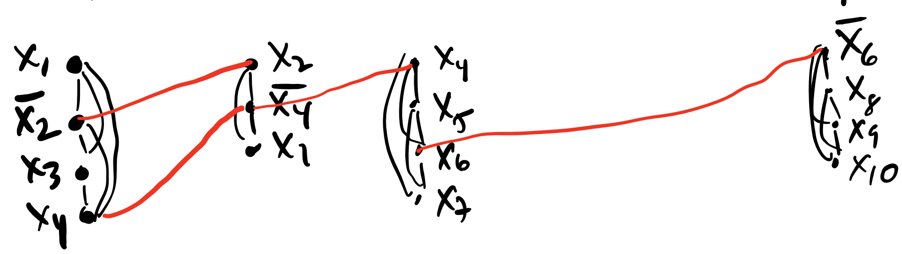
(2) <u>NP-hard</u>: <u>Claim</u> $CNFSAT \le_m^P$ INDEP-SET

Let φ be a CNF formula with clauses $C_1, \dots, C_m$ on n vars.

(a) Define G with one vertex per literal occurrence in φ, organized in columns for each clause

eg $C_1 = (x_1 \vee \overline{x_2} \vee x_3 \vee x_4)$. $C_2 = (\quad)$

G: C_1 C_2 C_3 C_4 ... C_m



Put an edge between every pair of vertices in same column

$\Rightarrow$ independent set has size $\leq m$

Put an edge between every pair of nodes labelled by contradictory literals $x_i, \overline{x_i}$

G has both kinds of edges

map: $\langle \varphi \rangle \xmapsto{f} \langle G, m \rangle$

(b) f is clearly polytime. G has # of vertices $\leq$ size of $\langle \varphi \rangle$ and # of edges at most that squared.

easy to compute

(c) Correctness:

(i) Suppose φ is satisfiable with assignment y making it 1

y must make every clause $C_1 \ldots C_m$ true

i.e. make at least one literal in each clause true

For set U choose one of these true literals per clause/column (doesn't matter which)

That won't contain any black edge because U has at most one literal per column

It won't contain any red edge since y can't make both $x_i, \bar{x}_i$ true.

$\therefore U$ is independent & size m as required.

(ii) Suppose G has an independent set U' of size m

U' must have one node per column

For truth ass't y' set all the literals labelling these nodes to true

(This will be consistent because U' can't contain nodes labelled both $x_i, \bar{x}_i$ because of red edges)

This might leave some vars unassigned so far by y'; assign these remaining vars arbitrarily in y'

Clearly this y' will satisfy φ by construction ∎

We define two more natural problems on graphs

Defn. A set $U \subseteq V$ is a clique in graph $G=(V,E)$ iff every pair of vertices in U is joined by an edge

- A set $W \subseteq V$ is a vertex-cover of G iff every edge of G touches W

CLIQUE $= \{\langle G,k\rangle : G$ has a clique of size $\geq k\}$

VERTEXCOVER $= \{\langle G,k\rangle : G$ has a vertex-cover of size $\leq k\}$

Thm CLIQUE & VERTEX-COVER are both NP-complete

Proof Clearly CLIQUE, VERTEX-COVER $\in$ NP

certificate = set U of size $\geq k$ that is a clique

certificate = set W of size $\leq k$ that is a vertex cover

Claim: INDEPENDENT-SET $\leq_m^P$ CLIQUE

$$\langle G,k\rangle \xmapsto{f} \langle G',k'\rangle$$

where G' = "complement of G"

(poly time) easy to compute

e is an edge in G'

$\iff$ e is not an edge in G

$k'=k$

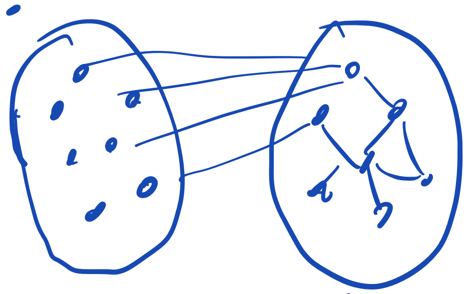
Correctness is immediate: certificate is same set U

Claim INDEPENDENT-SET $\le_m^P$ VERTEX-COVER

Lemma U is an independent set in $G=(V,E)$ $\iff$ $V-U$ is a vertex cover

Proof by picture:

no edges inside



U $\qquad\qquad$ $V-U$

$\blacksquare$

IND-SET $\qquad\qquad$ VERTEX-COVER

$$\langle G,k\rangle \xmapsto{f} \langle G',k'\rangle$$

easy to compute

$$G' = G$$

$$k' = |V| - k$$

Given certificate U for G, certificate for G' is $W = V-U$

Given certificate W for G', certificate for G is $U = V-W$ $\blacksquare$

---

kCOLOR $= \{\langle G,k\rangle : G$ has a proper vertex coloring with at most k colors$\}$

Defn A proper vertex coloring of k colors assigns one of k colors to vertices of G s.t. end points of an edge have different colors

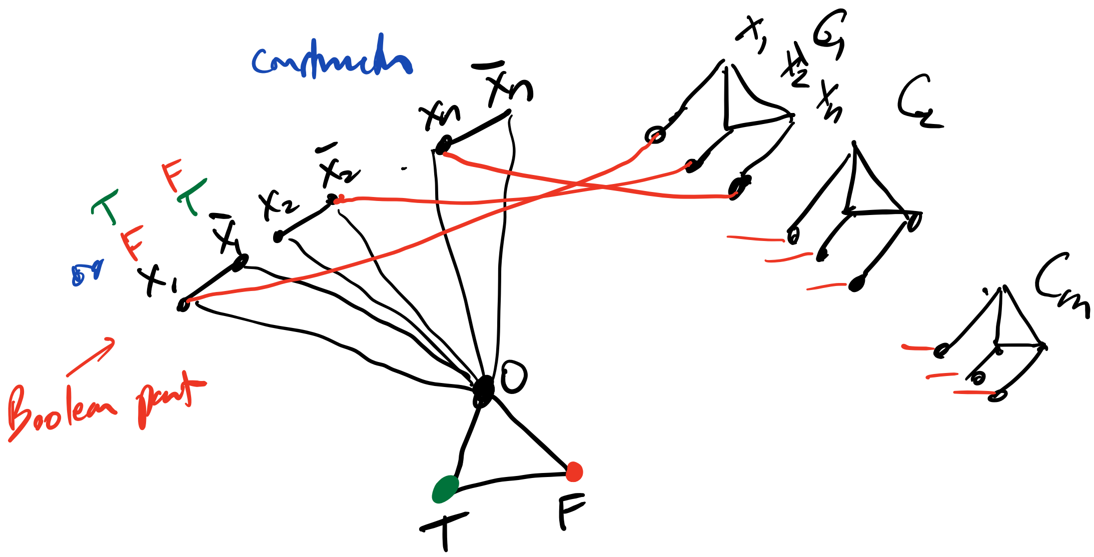
3COLOR = {<G> : <G,3> ∈ kCOLOR}

Thm 3COLOR is NP-complete
(from 421 recall that 2COLOR ∈ P)

Proof ∈ NP : certificate is assignment of colors to edges
check only 3 colors used
every edge has diff colored end points

NP-hard Claim: EXACT3SAT $\leq_m$ 3COLOR

construct



Finish next time