CS 314 - More reductions

Note Title 3/26/2010

Announcements

- HW will be out tonight
  - Graded orally on Tuesday, April 6 (day after Easter break)
  - Could also be in on Monday, since no class, if people have conflicts w/ Tuesday
- From now on, HW due Tues or Wed, since can't do oral grading on Thursdays

Recall:

Dfn: $Y \leq_p X$ (read Y is polynomial time reducable to X) if Y can be solved using a polynomial number of steps plus a polynomial number of calls to an algorithm (or "black box") that solves X.

Thm: Spps $Y \leq_p X$. If X can be solved in polynomial time, then so can Y.

$$\left.\begin{array}{l} p \to q \\ \neg q \to \neg p \end{array}\right\}$$ logically equivalent

While useful for algorithms, the real power here comes from the contrapositive:

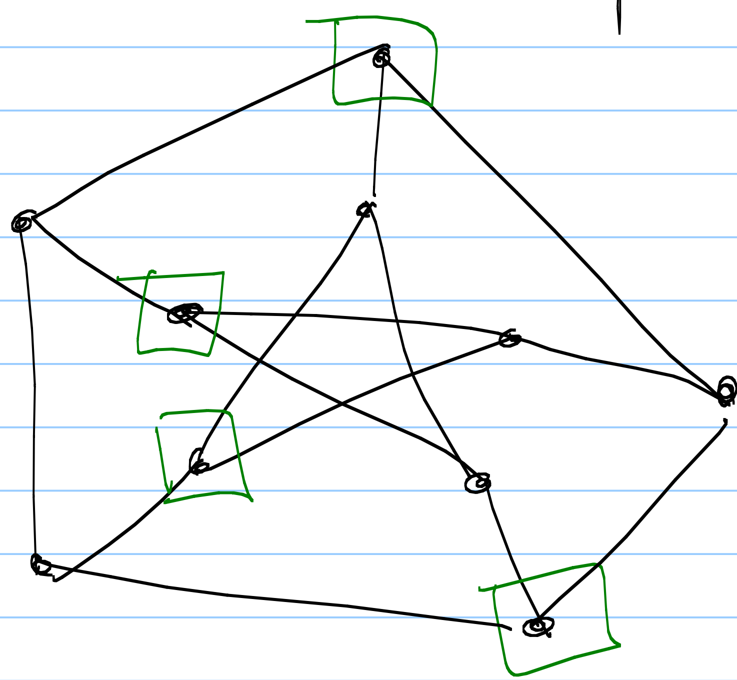
Spps $Y \leq_p X$. If Y cannot be solved in polynomial time, then X can't be solved in polynomial time.

SO:

$\boxed{\text{If we want to show a problem is NP-Hard, reduce a known NP-Hard problem to it!}}$

Independent Set

Q: Given a graph G and number k, is there an independent set of size $\geq k$?



size 4

We gave an algorithm:
in trees, $O(n)$
in general graphs, $O(2^n)$

Independent Set is NP-Complete

Need to show 2 things:

1. In NP
2. Some NP-Hard problem $\leq_p$ Ind. Set

(Don't forget to do 1)

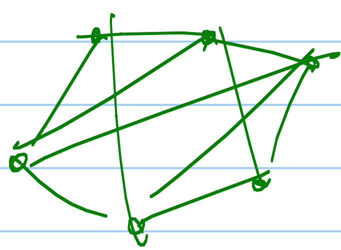
Q: Given G, k, does G have ind. set of size $\geq k$?

1 Ind. Set $\in$ NP:

Need to show a polynomial time check-able "certificate" exists.

If I claim G has an ind. set of size $\geq k$, how can I prove it?

given k vertices, there are $\binom{k}{2} = \frac{k(k-1)}{2}$ possible edges.



Check for all of them in $O(k^2)$ time.

This is polynomial.

3SAT $\leq_p$ Ind. Set :

So given any 3SAT instance, we need to transform it into a graph G in polynomial time so that

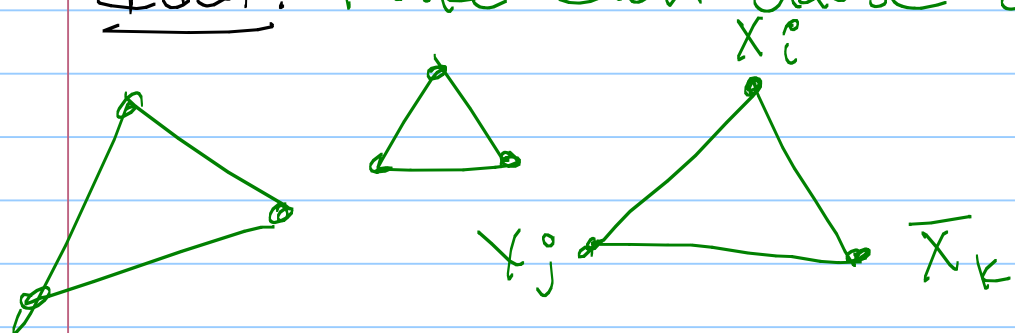
formula is satisfiable $\Longleftrightarrow$ G has ind. set of size $\geq k$

Input to 3SAT: n variables, k clauses

Eg: $(x_i \vee x_j \vee \bar{x}_k) \wedge (\bar{x}_\ell \vee x_j \vee x_k) \wedge \dots$

Want to make a graph which has ind. set of size k $\iff$ all clauses can be satisfied.

Idea: Make each clause a Δ in the graph G.

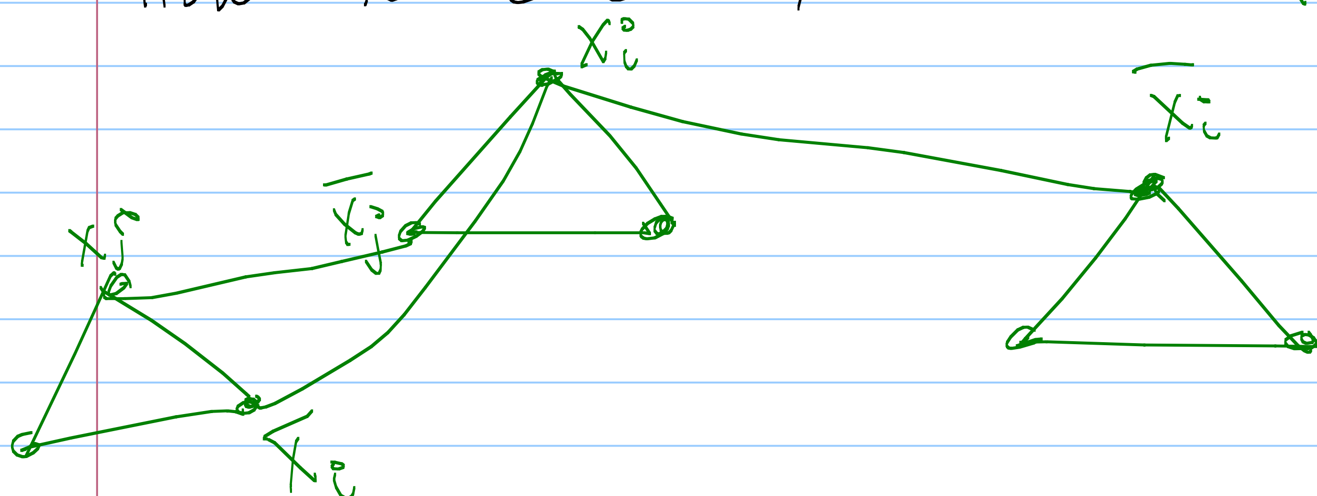


(idea: if x_i is true, then vertex labeled x_i will be in ind. set)

Now think about variables.

If $\underbrace{x_i = \text{true}}_{\text{could be in IS}}$, then $\underbrace{\overline{x_i} \text{ must be false}}_{\text{can't be in IS}}$

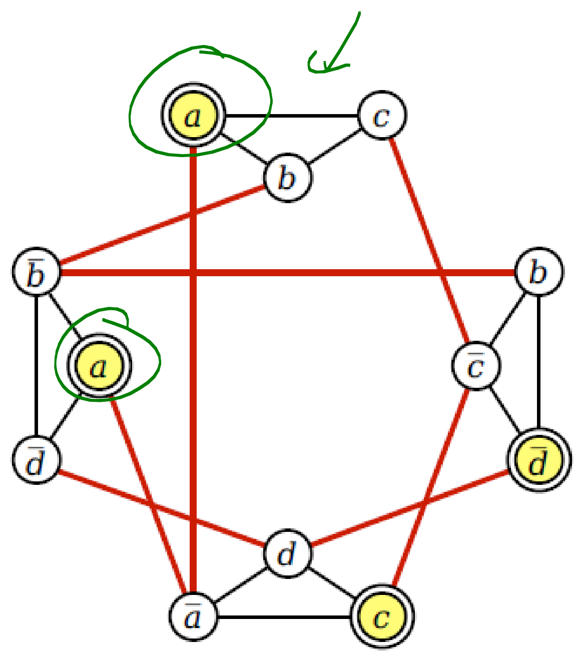
How to ensure?



for every label x_i connect it via an edge to every vertex labeled $\overline{x_i}$

Ex:

$$\overset{T}{(a \vee b \vee c)} \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge \overset{T}{(a \vee \bar{b} \vee \bar{d})}$$



A graph derived from a 3CNF formula, and an independent set of size 4.
Black edges join literals from the same clause; red (heavier) edges join contradictory literals.

→ **Claim:** original formula is satisfiable $\iff$ G has ind. set of size $= k$.

pf:

"$\Rightarrow$": Spps formula is satisfiable.

That means at least one variable in each clause is true.

The corresponding vertices in the graph cannot have edges between them.

So pick the true variable from each clause → that gives 1 vertex per Δ which can't be connected to other true variable.

$\Leftarrow$: Spps G has ind set of size k.

G is k Δ's connected together, so IS must have 1 vertex from each "clause" triangle.

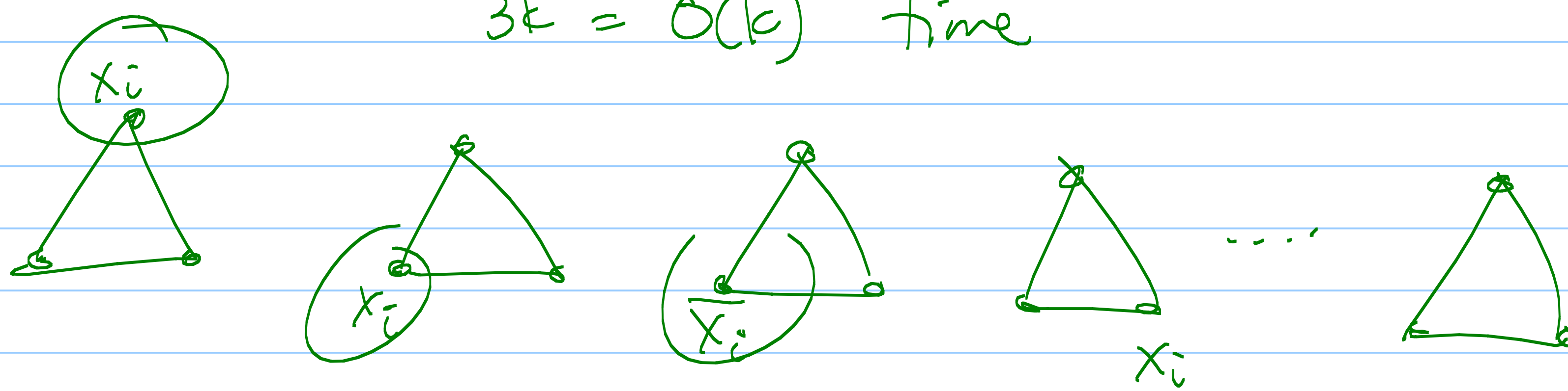
Set those variables in 3SAT formula to be true.

At least 1 variable in each clause must be set to true, and no x_i & $\overline{x_i}$ can both be set to true (b/c we put edges between all such pairs). $\boxtimes$

How long did reduction take?

k clauses became k $\triangle$'s.

$3k = O(k)$ time



$3k$ vertices, each needs to be checked against all other vertices

$O(k^2) \longrightarrow$ polynomial

Recap:

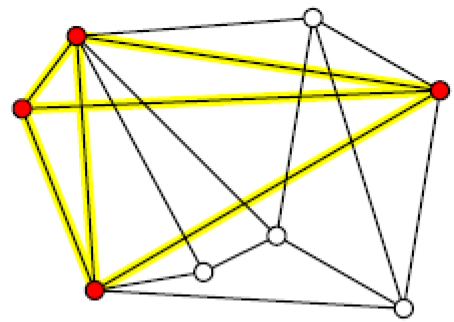
$$
\begin{tikzcd}
\boxed{\text{3CNF formula with } k \text{ clauses}} \arrow[r, "O(n)"] & \boxed{\text{graph with } 3k \text{ nodes}} \arrow[d, Rightarrow, "\textsc{MaxIndSet}"] \\
\boxed{\textsc{True or False}} & \boxed{\text{maximum independent set size}} \arrow[l, "O(1)"']
\end{tikzcd}
$$

$$T_{3SAT}(n) \le \cancel{O(n)} + T_{\textsc{MaxIndSet}}(\cancel{O(n)}) \implies T_{\textsc{MaxIndSet}}(n) \ge T_{3SAT}(\Omega(n)) - \cancel{O(n)}$$

$O(k^2)$ $\quad O(k^2)$ $\quad \Omega(k^2)$ $\quad O(k^2)$

Clique:

Dfn: A clique in a graph G is a subgraph where every pair of vertices is connected by an edge.



A graph with maximum clique size 4.

Q: Given a graph G and value k, is there a clique of size $\geq k$?

K-clique is NP-Complete:

1. in NP
2. NP-Hard problem $\leq_p$ K-clique

① K-clique is in NP:

Given k vertices, check every pair of vertices.

If all have an edge, it is a k-clique.

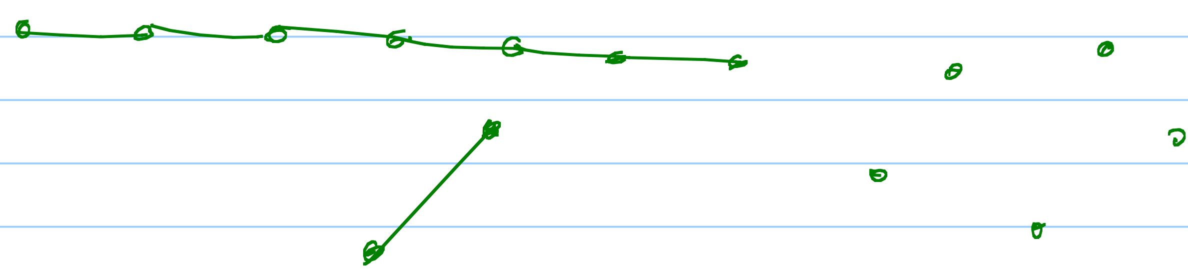
$O(k^2)$ time to verify a "yes" answer

2 K-clique is NP-Hard:

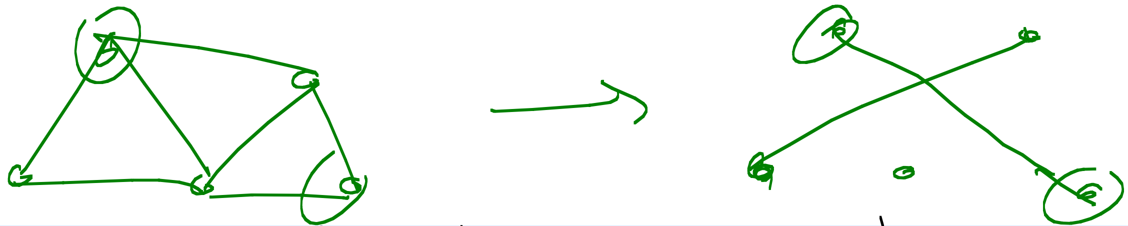
What to reduce to it?

→ Ind. Set $\leq_p$ K-clique

Input: Graph G, # k:
question is: does G have ind set of size $\geq k$?



Idea: Create $\bar{G}$ by adding every edge not in G, & removing every edge in G.

G  $\overline{G}$

G has an ind set of size k

$\iff \overline{G}$ has a clique of size k

pf:

"$\Rightarrow$": Spps S is I.S. of G.

$\Rightarrow$ no 2 vertices of S have an edge

$\Rightarrow$ in $\overline{G}$, every pair of vertices from S will be connected.

So S is a clique in $\overline{G}$.

"$\Leftarrow$": (same idea)

How long did transforming G to $\overline{G}$ take?

$\cancel{O(m)}$ time

$O(n^2)$

Recap: k-clique is NP-Complete:

$$\begin{tikzcd}
 & \boxed{\text{graph } G} \arrow[r, "O(n)"] & \boxed{\text{complement graph } \overline{G}} \arrow[d, Rightarrow, "\textsc{MaxClique}"] \\
 & \boxed{\text{largest independent set}} & \boxed{\text{largest clique}} \arrow[l, "\text{trivial}"']
\end{tikzcd}$$