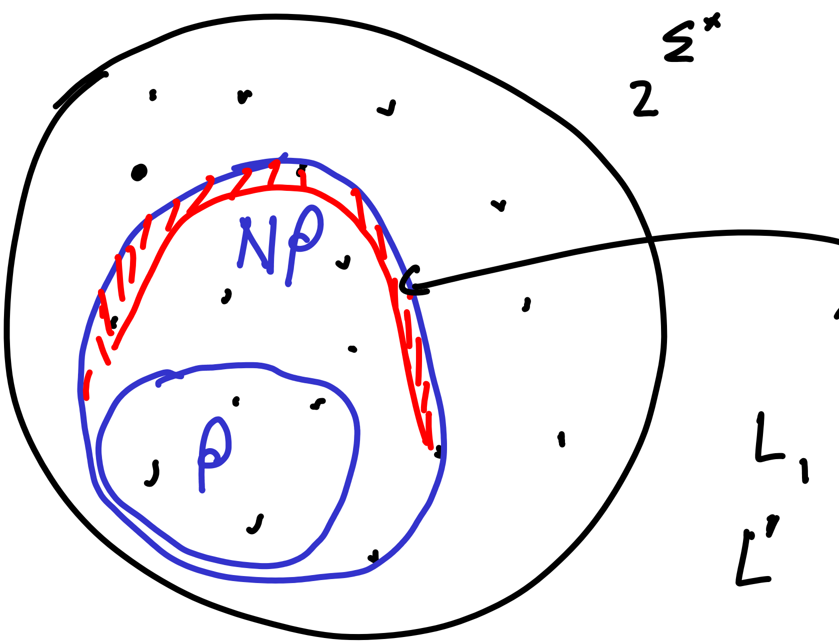
COL 702 Lecture 27 Nov 10

Alphabet Σ Set of strings Σ^*

Language is a subset of Σ^*

Set of languages 2^{Σ^*}

Decision problem : given a string $s \in \Sigma^*$ and a language L, $s \overset{?}{\in} L$



hardest problems in NP

$L_1 \in NP$ $\forall L_1 \in NP$,

L' is a hard problem $\boxed{\text{in NP}}$

$L_1 \leq_{poly} L'$

then L' is an $\boxed{\text{NP-complete problem.}}$

If $L', L'' \in NPC \Rightarrow L' \leq_{poly} L''$ and $L'' \leq_{poly} L'$

NPC defines an equivalence class

Cook. Levin theorem

The Satisfiability Problem of Boolean CNF is NP complete.
(Conjunctive normal form)

Boolean CNF expression is over n boolean variables which are x_1 x_2 ... x_n and they appear as literals x_i or $\bar{x}_i$ (complement)

A clause is a disjunction of some literals $(x_2 \vee \bar{x}_3 \vee x_5 \vee \bar{x}_6)$ etc.

A CNF is True if at least one literal is set to True in every clause

$$(x_2) \wedge (x_3 \vee \bar{x}_4 \vee \bar{x}_2) \wedge (\bar{x}_3 \vee \bar{x}_4)$$

$(\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee x_5)$

is satisfiable

$$(x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (x_1 \vee \bar{x}_2)$$

is not satisfiable

K-CNF is a CNF where every clause has exactly K literals

For K = 3 it is a 3-CNF

$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_2 \vee x_4 \vee \bar{x}_1) \wedge (\quad) \ldots$

<u>Claim</u> : 3 CNF Satisfiability Problem (3 SAT) is NP complete.

(i) It is in NP

Since SAT is in NP 3SAT is also in NP since it is a special case

(ii) All NP problems are polynomial time reducible to 3SAT

If we can show that $L^{CNF} \leq_{poly} L^{3SAT}$

then it follows that L^{3SAT} is NP hard

(If all languages $L \in NP$ are polytime reducible to L' then L' is NP hard)

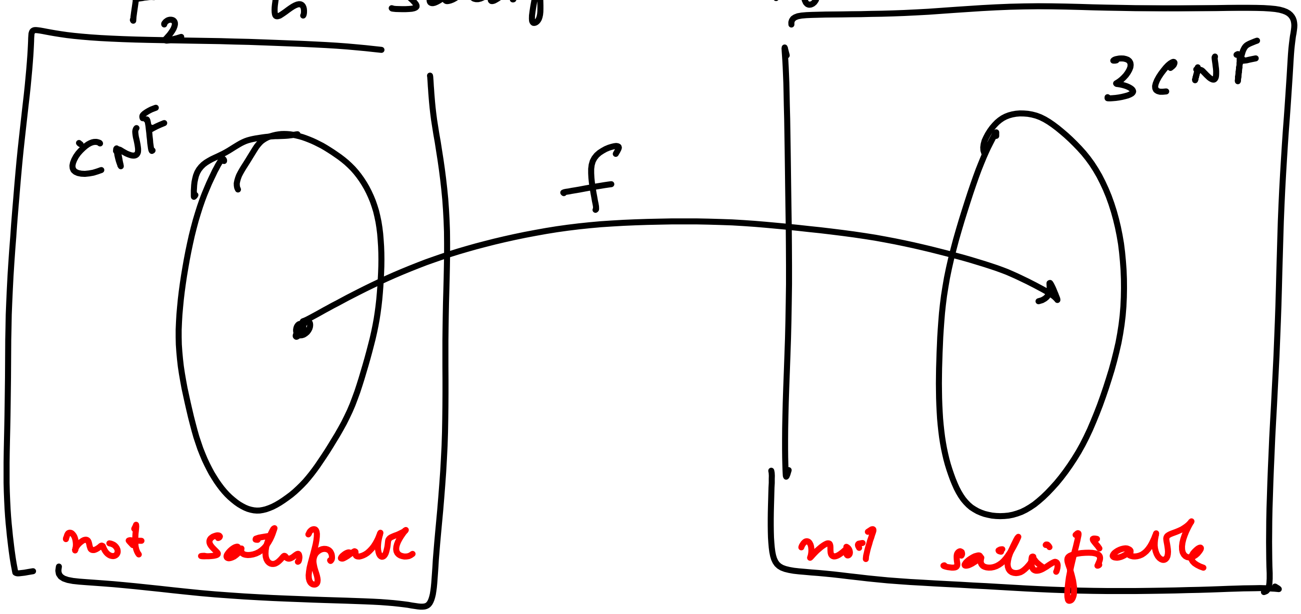
$(x_1 \quad) \wedge (\bar{x}_1 \vee x_3) \wedge (\qquad) \wedge ($

$(\bar{x}_1 \vee x_3 \vee y) \wedge (\bar{x}_1 \vee x_3 \vee \bar{y})$

$(x_1 \vee y_1 \vee y_2) \wedge (x_1 \vee \bar{y}_1 \vee y_2) \wedge (x_1 \vee y_1 \vee \bar{y}_2)$

$\wedge\ x_1 \vee \bar{y}_1 \vee \bar{y}_2)$

Given an arbitrary CNF F_1, we must map it to a 3CNF F_2 s.t. F_2 is satisfiable iff F_1 is satisfiable



Function f is computable in polynomial-time

Clauses in CNF $\longrightarrow$ 1 literal, 2 literals, 3 literals, 4 or more

4 literals in a clause

$$(x_1 \vee x_2 \vee x_3 \vee x_4)$$

$$(x_1 \vee x_2 \vee y_1) \wedge (\bar{y}_1 \vee x_3 \vee x_4)$$

$$(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5)$$

$$(x_1 \vee x_2 \vee y_1) \wedge (\bar{y}_1 \vee x_3 \vee y_2) \wedge (\bar{y}_2 \vee x_4 \vee x_5)$$

If original formula is satisfiable, say x_3 is True we can make the other clauses true by using $y_1 = T$ and $y_2 = F$

If the new formula is satisfiable, we cannot do it just by setting y_1 and y_2 appropriately. At least one of $x_1, x_2 \ldots x_5$ must be true.

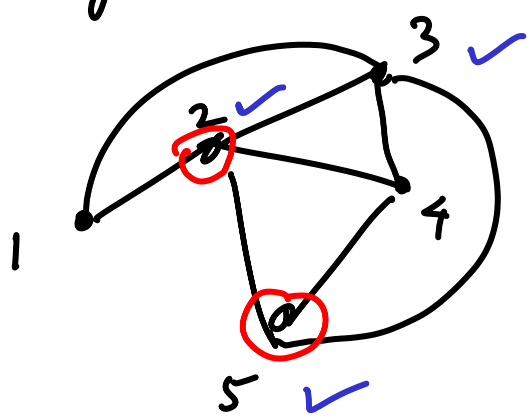
<u>Important</u>: What is the size of the new formula? We have to ensure polynomial time computability. For every clause of size less than 3, we are introducing at most 4 clauses (and 2 new variables). For k-clause $k > 3$, we are creating $k-1$ clauses and $k-2$ variables So it is still polynomial

For the recursive construction* this may not be true, i.e. we could be introducing many more variables almost 2^k * $k \to k-1 \to k-2$ etc.

Vertex Cover Problem

$G = (V, E)$ and an integer K

A subset of vertices $W \subset V$ is a "cover" if at least one endpoint of every edge is incident on W



$(2, 5)$

Is there a subset of size K that is a cover?

(i) Is L^{VC} in NP? ✓

(ii) Is L^{VC} NP hard?

We will reduce $L^{3SAT} \leq_{poly} L^{VC}$

Given an arbitrary 3CNF formula we will map it to an instance of the L^{VC}

$$F_1 \xrightarrow{f} \left[G = (V, E), K \right]$$

s.t. the graph will have a cover of size K iff F is satisfiable

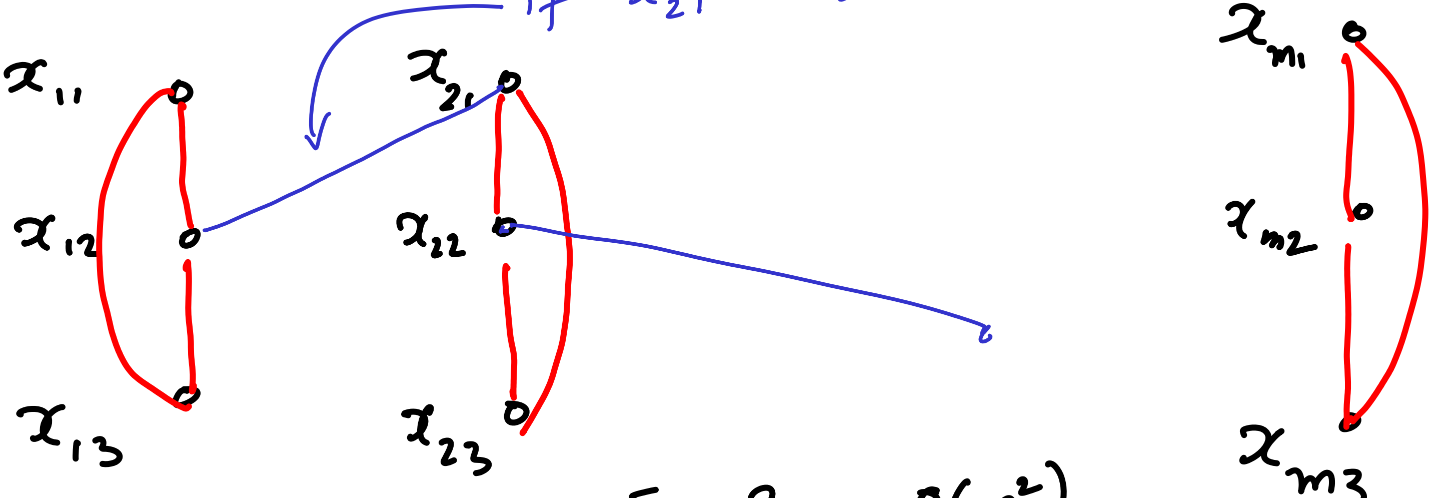
$$(x_{11} \vee x_{12} \vee x_{13}) \wedge (x_{21} \vee x_{22} \vee x_{23}) \cdots$$

$$(x_{m1} \vee x_{m2} \vee x_{m3})$$

m clauses
n variables

$$x_{ij} \in \{ x_1, \bar{x}_1, x_2, \bar{x}_2 \cdots x_n, \bar{x}_n \}$$

if $x_{21} = \bar{x}_{12}$

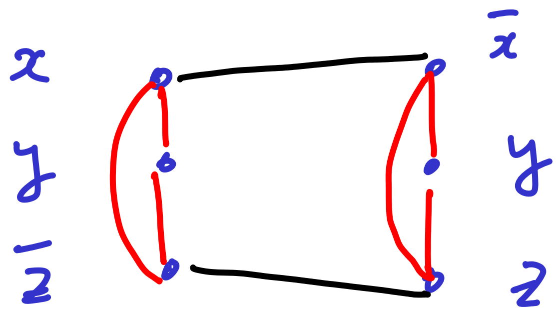


$E = 3m + O(m^2)$

V: $3m$ vertices

$k = 2m$

Example: $(x \vee y \vee \bar{z}) \wedge (\bar{x} \vee y \vee z)$



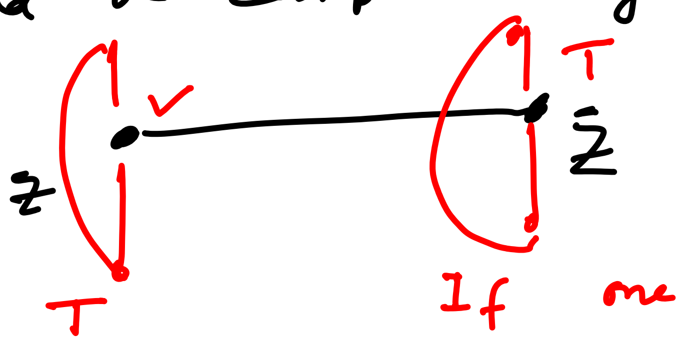
Claim: The graph has a vertex cover of size $2m$ iff the formula is satisfiable

$\Rightarrow$ Suppose the formula is satisfiable. Then there must be at least one literal set to True. Say y_i is the literal that is true in clause i (If there is more than, choose one arbitrarily). Choose the vertices corresponding to the other 2 literals and call this set W. We will show that W is a cover

$\Leftarrow$ If the graph has a cover of size $2m$ then the formula is satisfiable

For a satisfiable assignment, we set the literal in each clause as true that is not in the cover

(1) We must ensure that the assignment is consistent, i.e. the two endpoints of a non-edge $(z, \bar{z})$ should be complementary values



At least one must be chosen

If one of them is chosen then the

(2) If some variables have not been assigned any truth value then assign truth values arbitrarily

(polynomial time) Approximation Algorithms

We want to guarantee that the solution is within a "small factor" of the optimal soln.

Vertex Cover (optimization version) Given a graph $G = (V, E)$ what is the size of the smallest cover?

Say the optimal cover is $O^* \subseteq V$

Then if our algorithm produces W, we want to claim $\frac{|W|}{|O^*|} \leq \alpha$

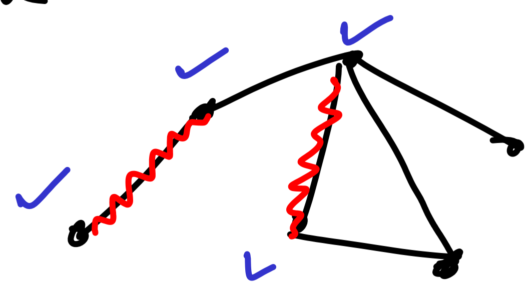
If $\alpha = 1$ it is optimal

$\alpha = 2$ α-factor approx

$\alpha \geq 1$

Approximate Vertex Cover

1. Find a <u>maximal</u> matching of the graph



2. If we have picked edges $e_1, e_2, \ldots e_k$ then we define a cover consisting of the endpoints of all the edges, i.e. $2k$ vertices

<u>Observation</u>: Any V.C. must have at least k vertices to cover $e_1 \ldots e_k$

Approximation ratios can't be better than "2" unless P = NP

(hardness of approximating)

Independent Set : getting below

$\frac{n}{\log^2 n}$ is "hard"