CMPT307: NP-Completeness Proofs

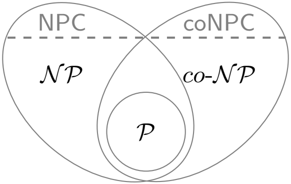
Week 13-2

Xian Qiu

Simon Fraser University

xianq@sfu.ca

Theorem. $\mathcal{P} \in \mathcal{NP} \cap \text{co-}\mathcal{NP}$



Open questions

- $\mathcal{P} = \mathcal{NP}$? consensus opinion: no
  one of the seven millennium prize problems of CMI (worth 1,000,000$)
- $\mathcal{P} = \mathcal{NP} \cap co\text{-}\mathcal{NP}$?

Formula Satisfiability

given a formula F in conjunctive normal form (CNF)

- literals: $x_1, \ldots, x_n \in \{\text{true}, \text{false}\}$
- clauses: $C_1, \ldots, C_m$ of disjunction of literals

$$C_i = x_1 \vee \bar{x}_3 \vee \ldots \vee x_7$$

- formula: $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$

The satisfiability problem (Sat)

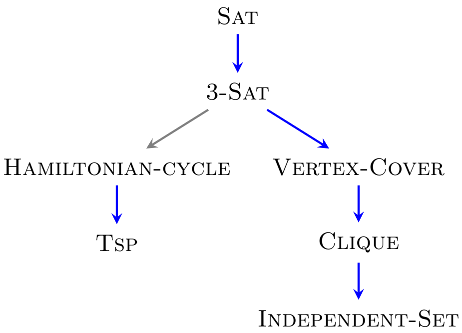
does there exist $x \in \{\text{true}, \text{false}\}^n$ such that $F(x) = \text{true}$?

also denote by Sat the corresponding language

3-Sat: Sat by restricting each clause with exactly 3 literals

NP-complete Problems

Theorem (S.A. Cook, 1971). SAT is NP-complete.



3-SAT $\in \mathcal{NP}$

- a certificate is a vector $x \in \{\text{true,false}\}^n$ s.t. $F(x) = \text{true}$
- x is poly-size and $F(x)$ can be computed in poly-time

SAT $\leq_P$ 3-SAT

1. start with an arbitrary SAT instance I
2. construct a 3-SAT instance I'
3. show that $I = \text{yes} \Leftrightarrow I' = \text{yes}$

let F be an arbitrary CNF formula

$$F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$$

idea: transform C_i with $|C_i| \neq 3$ into clauses of 3 literals each

3-SAT is NP-complete

consider any clause $C_i = (x_{i_1} \vee x_{i_2} \vee \ldots \vee x_{i_k})$ of F

case 1: $k = 3$, done

case 2: $k = 2$, introduce a variable y_{i_1} and replace C_i by

$$C_i^1 = (x_{i_1} \vee x_{i_2} \vee y_{i_1}), \quad C_i^2 = (x_{i_1} \vee x_{i_2} \vee \bar{y}_{i_1})$$

- ▷ $C_i =$ true $\Leftrightarrow C_i^1 \wedge C_i^2 =$ true

case 4: $k = 1$, introduce two variables y_{i_1}, y_{i_2} and replace C_i by

$$C_i^1 = (x_{i_1} \vee y_{i_1} \vee y_{i_2}), \quad C_i^2 = (x_{i_1} \vee y_{i_1} \vee \bar{y}_{i_2})$$
$$C_i^3 = (x_{i_1} \vee \bar{y}_{i_1} \vee y_{i_2}), \quad C_i^4 = (x_{i_1} \vee \bar{y}_{i_1} \vee \bar{y}_{i_2})$$

- ▷ $C_i =$ true $\Leftrightarrow C_i^1 \wedge C_i^2 \wedge C_i^3 \wedge C_i^4 =$ true

3-Sat is NP-complete

case 4: $k \geq 4$, first introduce a variable y_{i_1}

$$C_i = (\underbrace{x_{i_1} \vee x_{i_2}}_{y_{i_1}} \vee \ldots \vee x_{i_k})$$

and replace C_i by C_i^1 and C_i', where C_i' has $k-1$ literals

$$C_i^1 = (x_{i_1} \vee x_{i_2} \vee y_{i_1}), \quad C_i' = (\bar{y}_{i_1} \vee x_{i_3} \vee \ldots \vee x_{i_k})$$

- $C_i = \text{true} \Leftrightarrow C_i^1 \wedge C_i' = \text{true}$
- keep doing this for C_i' until it has 3 literals (introducing $k-3$ variables and $k-2$ clauses)

3-SAT is NP-complete

$$F = C_1 \wedge C_2 \wedge \ldots C_m \qquad \textsc{Sat}$$

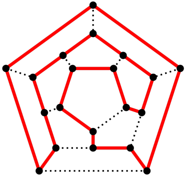
$$F' = (C_1^1 \wedge C_1^2 \wedge \ldots \wedge C_1^{i_1}) \wedge (C_2^1 \wedge C_2^2 \wedge \ldots \wedge C_2^{i_2}) \wedge \ldots \wedge (C_m^1 \wedge C_m^2 \wedge \ldots \wedge C_m^{i_m}) \qquad \text{3-}\textsc{Sat}$$

$$F = \mathsf{true} \Leftrightarrow F' = \mathsf{true}$$

the reduction is polynomial time

- each C_i has at most n literals, thus introducing $O(n)$ literals and clauses
- m clauses in total, thus the reduction can be done in $O(mn)$

HAMILTONIAN-CYCLE (HC): given an undirected graph $G = (V, E)$, does it contain an HC?



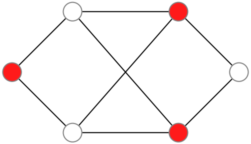
polynomial reduction

- create a complete graph $G' = (V, E')$ and assign distances $d(e) = 1$, $\forall e \in E$, $d(e) = 2$ otherwise
- G has an HC $\Leftrightarrow$ G' has tour of length n

Vertex Cover

Vertex-Cover

- ▷ given: undirected $G = (V, E)$ and integer k
- ▷ question: does G contain a vertex cover of size at most k?



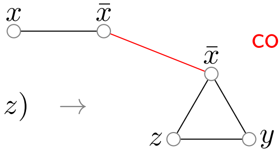
3-Sat $\leq_P$ Vertex-Cover

- ▷ $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m \longmapsto$ graph G
- ▷ how to define vertices and edges?

Vertex-Cover is NP-complete

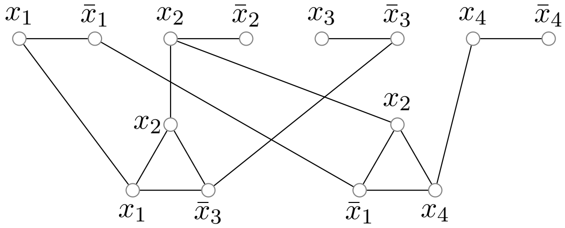
variable gadegt: $x \quad \rightarrow$

clause gadegt: $C = (\bar{x} \vee y \vee z) \quad \rightarrow$



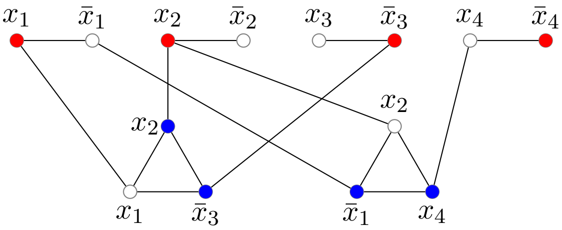
connect "x-x"

example: $F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_4)$



Claim. $F =$ true $\Leftrightarrow G$ has a vertex cover C of size $n + 2m$.

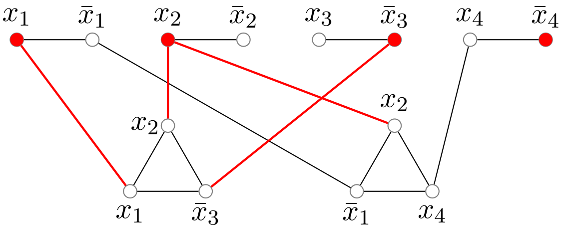
Necessity



$F =$ true $\Rightarrow x_i$ or $\bar{x}_i$ must be true

- $C_1 = \{$true vertices of variable gadgets$\}$
- $C_2 = \{$remaining vertices of clause gadgets$\}$
- $C = C_1 \cup C_2$ is a vertex cover with $|C| = n + 2m$

Sufficiency



$|C| = n + 2m \Rightarrow 2$ nodes per clause, 1 node per variable gadget

- set covered nodes in variable gadgets to be true (red nodes)
- the assignment is feasible as no conflict
- each clause gadget has 3 "x-x edges": at least one edge is covered by a red node

polynomial time reduction?