CSC 320 - Lecture 19

#np #membership #np-hard #np-complete #vertex-cover #dominating-set #complexity-theory #optimization #decision

---

NP-Completeness

Definition. A language B is NP-Complete if it satisfies the following two conditions...

1. $B \in NP$
2. $B \in NP$-Hard ("B is at least as difficult as anything in NP")

Where, B is NP-Hard if for every $A \in NP: \ A \leq_p B$

Theorem. If B is NP-Complete and $B \leq_p C$ for $C \in NP$, then C is NP-Complete.

Note. If B is NP-Complete and $B \in P$ then $P = NP$.

Proving for just one NP-complete decision problem / language that it is in P would solve the $P = NP$ question, since then every problem in NP would be solvable by a polynomial time TM.

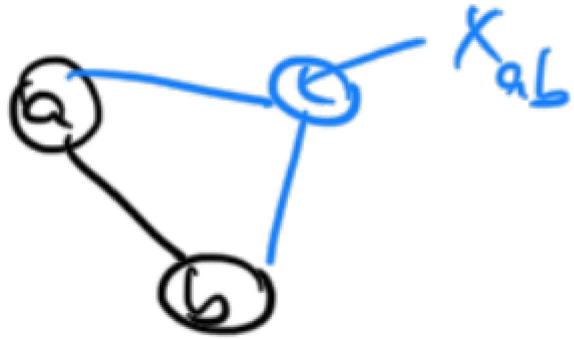
$VC \leq_p DS$ (Revisited)

Reduction from vertex cover instance $\langle G, k \rangle$ to dominating set instance $\langle G', k' \rangle$.

We assume that G has no singletons (vertices not connected to an edge).

Given $G = (V, E)$ we build G' by adding new vertices and edges to G, as follows...

- For each edge $(a, b) \in E$ in G we create a new vertex $X_a b$ and edges $(a, X_a b)$ and $(b, X_a b)$.



More formally: For $G = (V, E)$ we define $G' = (V', E')$...

- $V' = V \cup \{X_{ab} \mid (a, b) \in E \;\&\; X_{ab} \notin V\}$
- $E' = E \cup \{(a, X_{ab}), (b, X_{ab}) \mid (a, b) \in E\}$
- $k' = k$

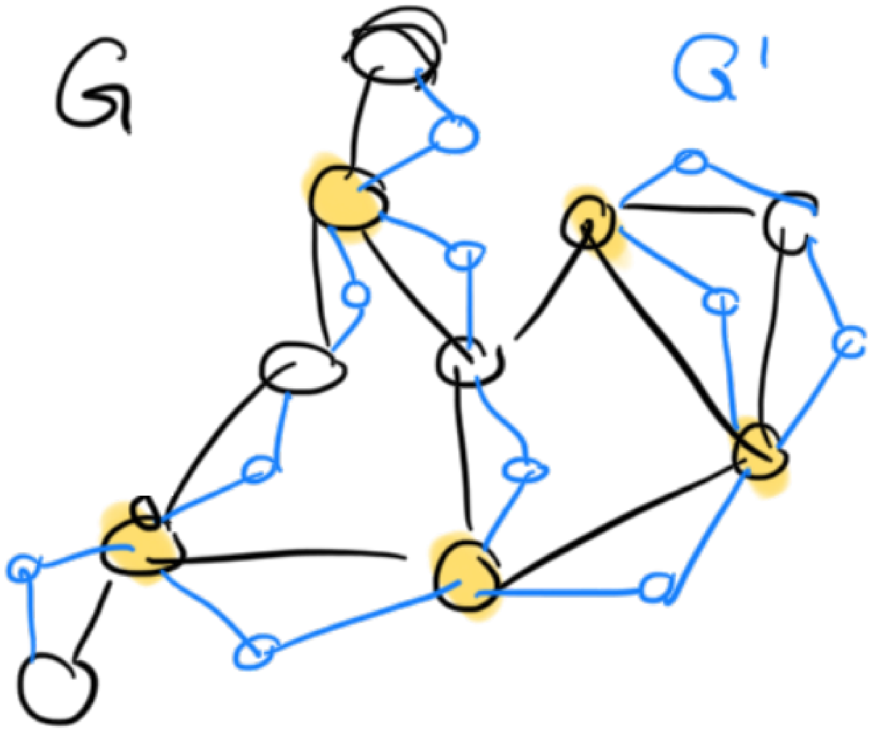
In polynomial time!

Show. $\langle G, k\rangle \in VC$ if and only if $\langle G', k\rangle \in DS$. In other words... G has k-vertex cover if and only if G' has a k dominating set.

$\Rightarrow$: Let $C \subseteq V$ be a vertex cover for G, $|C| = k$.

We show that C is a dominating set for G!

Observe. C is a dominating set for G.



We need to prove that every blue vertex in G' is also eliminated:

- Every X_{ab} is adjacent to black vertices a and b. Since $(a, b) \in E$ and C is a vertex cover, $a \in C$ or $b \in C$ (or both).
  - $\Rightarrow X_{ab}$ is dominated by a or b
  - $\Rightarrow C$ is dominating set for G'

$\Leftarrow$: Let $D \subseteq V'$ be a dominating set for G'.

Show. That we can build a vertex cover C for G with $|C| \leq |D|$.

$\Rightarrow$ We can build C as follows.

$C = (D \cap V) \cup \{a \mid X_{ab} \in D\}$

$\Rightarrow |C| \subseteq |D|$

Note. For each edge in G: $a \in D$ or $b \in D$ or $X_{ab} \in D$.

How Can We Prove That A Decision Problem / Language B is NP-Complete?

1. Show that B is in NP
   A. Give a polynomial time verifier

2. Show that B is NP-Hard
   A. Pick a problem A that is known to be NP-Complete
   B. Prove that $A \leq_p B$

NP-Completeness

- Assume A is NP-Complete, $B \in NP$, and $A \leq_p B$.
  - Is $B \leq_p A$? (Yes)!

Optimization Problems VS Decision Problems

Often, problems come to us as optimization problems. Complexity classes, like P and NP, are defined via languages AKA decision problems.

- Given a decision problem A that is solvable in polynomial time...
  - Is there a polynomial time algorithm for the optimization version of A?
- Given an optimization version A^* of decision problem A, where A^* is solvable in polynomial time...
  - Is there a polynomial-time algorithm that decides A?

Note. Missing class notes from drawing board.

Optimization Versions Of These Decision Problems / Languages?

$$CLIQUE = \{\langle G, k\rangle \mid G \text{ is an undirected graph with } k\text{-clique}\}$$

- **Clique (Decision Version)**
  - Input: $G = (V, E), k$
  - Question: Does there exist a clique for G of size at least k?
- **Maximum Clique (Optimization Version)
  - Input: $G = (V, E)$
  - Output: A maximum-size clique for G

$$IS = \{\langle G, k\rangle \mid G \text{ is an undirected graph with an independent set of size at least } k\}$$

- **Independent Set (Decision Version)**
  - Input: $G = (V, E), k$
  - Question: Does there exist an independent for G of size at least k?
- **Maximum Independent Set (Optimization Version)

- Input: $G = (V, E)$
- Output: A maximum-size independent set for G

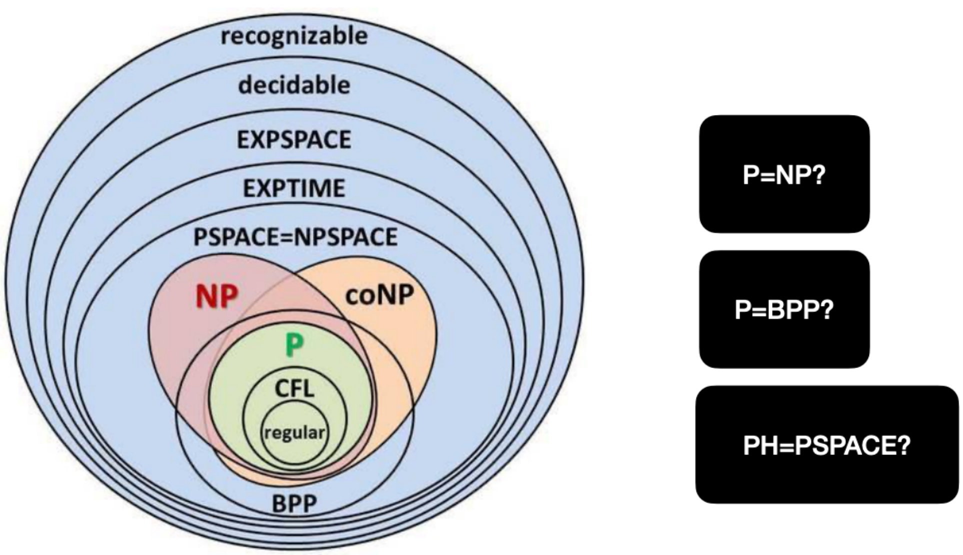
$$VC = \{\langle G, k\rangle \mid G \text{ is an undirected graph with a vertex cover set of size at least } k\}$$

- **Vertex Cover (Decision Version)**
  - Input: $G = (V, E), k$
  - Question: Does there exist a vertex cover for G of size at most k?
- **Minimum Vertex Cover (Optimization Version)
  - Input: $G = (V, E)$
  - Output: A minimum-size vertex cover for G

$$DS = \{\langle G, k\rangle \mid G \text{ is an undirected graph with a dominating set of size at most } k\}$$

- **Dominating Set (Decision Version)**
  - Input: $G = (V, E), k$
  - Question: Does there exist a dominating for G of size at most k?
- **Minimum Dominating Set (Optimization Version)
  - Input: $G = (V, E)$
  - Output: A minimum-size dominating set for G

Complexity Theory



Previous Lecture

Lecture18

Next Lecture

Lecture20