CS3313 Lab

10 April 2024

Outline

Review the definition of NP

Recall the SAT problem

Example of reduction to show that a new problem is NP-complete (Vertex Cover)

Reduction exercise (Independent Set)

Verifiability

A verifier for a language L is an algorithm V, where

$$L = \{x \mid V \text{ accepts } \langle x, w \rangle \text{ for some string } w\}$$

- Runtime of V is measured as a function of $|x|$
- V is a polynomial time verifier if it runs in time $poly(|x|)$
- L is polynomially verfiable if it has a polynomial time verifier
- String w is called a witness that $x \in L$

The class NP

Definition

$\mathcal{NP}$ is the class of languages that have polynomial time verifiers.

How to show that a problem A is NP-Complete

1. Show that A is in NP
   a. Provide a poly-time verification algorithm for A
2. Reduce some other NP-hard problem to A

The SAT Problem

Satisfiability Problem

$$SAT = \{\langle\phi\rangle \mid \phi \text{ is a satisfiable Boolean formula}\}$$

Example: $\phi = (\overline{x} \wedge y) \vee (x \wedge \overline{z})$

Can you think of some expressions that are not satisfiable?

Fact: SAT is NP-complete

A common case of SAT: The 3-SAT Problem

Definition 15.5. A propositional formula is in *3-clausal form* if it is in clausal form and has exactly 3 literals per clause. For example, $(x \vee y \vee z) \wedge (\overline{y} \vee z \vee \overline{w})$ is in 3-clausal form. A propositional formula in 3-clausal form is called a *3-clausal propositional formula*.

Definition 15.6. *3-SAT* is the following decision problem.

> **Input.** A 3-clausal propositional formula ϕ.
> **Question.** Is ϕ satisfiable?

Fact: 3-SAT is NP-complete

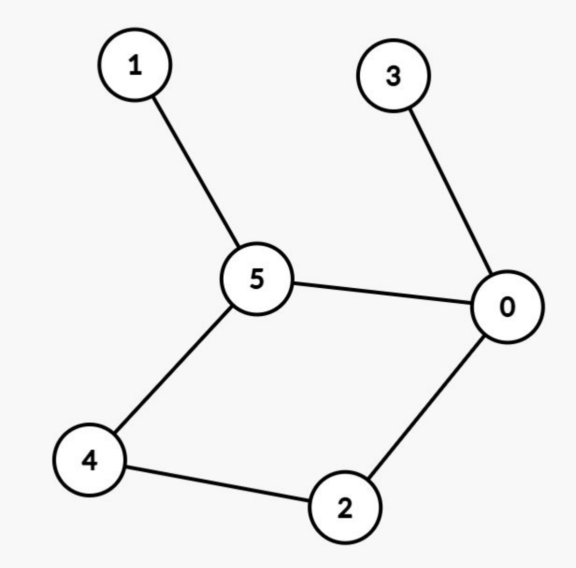
Remark about the Fact: in 7 minutes, you'll want to remember the Fact :-)

Vertex Covers

For a simple graph G=(V,E) a k-vertex cover is a set C⊆V with |C|=k such that every edge in E has an endpoint in C.

Find a vertex cover.

Find the smallest possible vertex cover.



The Vertex Cover Problem

Input. A simple graph G and a positive integer k.
Question. Does there exist a vertex cover C of G where $|C| \leq k$?

Exercise: Prove that the Vertex Cover Problem is in P or is NP-Complete.

Hint: It's not in P.

Solution

Reduce 3-SAT to Vertex Cover

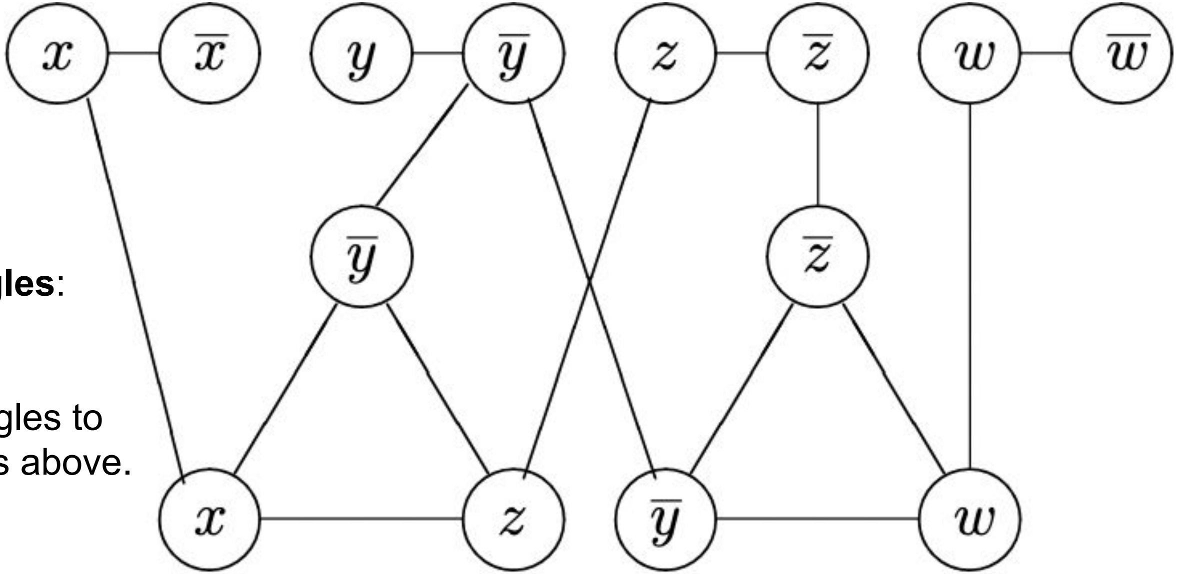
Example: Create graph G as shown; for a 3-clausal formula with v variables and c clauses, the vertex cover problem with k=v+2c solves the 3-SAT problem.

$$\phi = (x \vee \overline{y} \vee z) \wedge (\overline{y} \vee \overline{z} \vee w).$$

First make the **pairs** X and ~X here:

Then make all the clauses into **triangles**:

Finally connect the nodes in the triangles to their corresponding nodes in the pairs above.



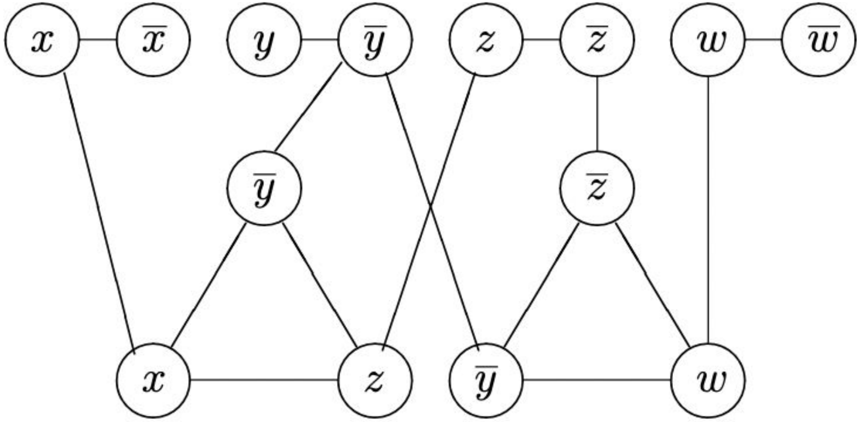
Solution

Continued; we need to show If $x \in A$, $f(x) \in B$
If $x \notin A$, $f(x) \notin B$,

If there is a satisfying assignment of truth values, there is a vertex cover.

- If x is assigned true, add node labeled by x from the **pairs** to the cover. Then any adjacent x in a **triangle** need not be added; so add the other two nodes from that triplet. Same for ~x. (Since the truth assignment satisfies the equation, there will be at least one value in the triangle which is true and thus need not be added to the cover.)

$$\phi = (x \vee \overline{y} \vee z) \wedge (\overline{y} \vee \overline{z} \vee w).$$



Solution

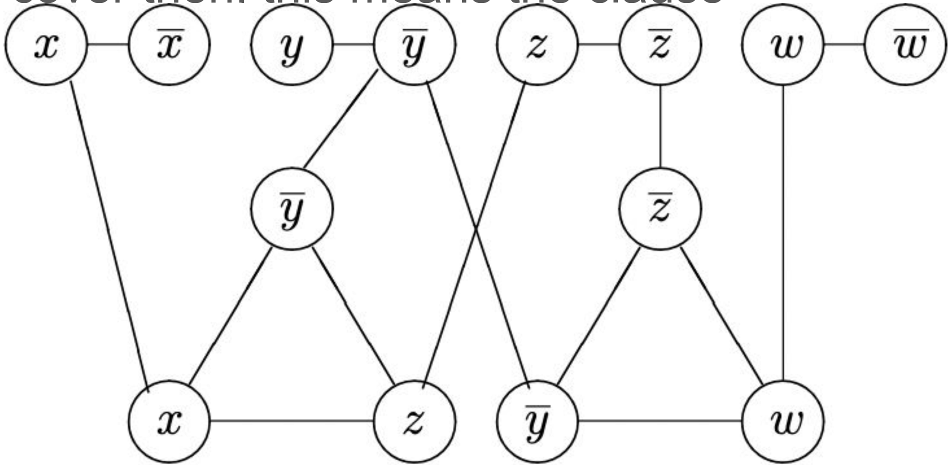
Continued; we need to show If $x \in A$, $f(x) \in B$
If $x \notin A$, $f(x) \notin B$,

For the second part, we can show the contrapositive

If there is a vertex cover, there is a satisfying assignment of truth values:

- For each variable X, assign true if the node labeled X in a **pair** is in the cover and false if the node labeled ~X in a **pair** is in the cover. Now, for each **triangle** there is some node not in the cover; well this one is adjacent to one in a **pair** which is necessarily in the cover then: this means the clause represented by the triangle is true.

$$\phi = (x \vee \overline{y} \vee z) \wedge (\overline{y} \vee \overline{z} \vee w).$$



Definition 15.7. Suppose $G = (V, E)$ is a simple graph. An *independent set* of G is a set $S \subseteq V$ such that no two members of S are connected by an edge. That is, if u and v are different members of S, then$\{u, v\} \notin E$.

Definition 15.8. The *Independent Set Problem* (ISP) is the following decision problem.

> **Input.** A simple graph $G = (V, E)$ and a positive integer k.
> **Question.** Does G have an independent set of size at least k?

Either (1) find a poly-time algorithm for ISP or (2) prove that ISP is NP-complete.