## Introduction

**Definition** [3SAT]**.** We say a formula $\varphi$ is in *conjunctive normal form*, or $\varphi$ is a cnf-formula, if it can be written as a conjunction ($\wedge$) of *clauses*, and each clause is a disjunction ($\vee$) of literals (either a variable $x_i$ or its negation $\neg x_i$).

$$\varphi = \bigwedge_{i=1}^{n} \left( \bigvee_{j=1}^{k_i} f_{ij} \right) = (f_{11} \vee \cdots \vee f_{1k_1}) \wedge \cdots \wedge (f_{n1} \vee \cdots \vee f_{nk_n}).$$

A *3cnf-formula* is a cnf-formula such that each *clause* has exactly *three literals*. For example:

$(x_1 \vee x_3 \vee \neg x_2) \wedge (x_1 \vee \neg x_1 \vee x_4) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_7)$ is a 3cnf-formula.
$(x_1 \vee x_3 \vee \neg x_2) \wedge (x_1)$ and $\neg(x_4 \wedge x_1 \vee x_3) \wedge (x_1 \vee \neg x_1 \wedge x_4)$ are not 3cnf-formulas.

Finally, we can define:

$$\mathsf{3SAT} = \{\langle \varphi \rangle : \varphi \text{ is a satisfiable 3cnf-formula}\} \subseteq \mathsf{SAT}.$$

The following theorem is striking, since it shows that increasing the *vertical complexity* of a formula does not necessarily allow it to be more expressive.

**Theorem.** 3SAT is NP-complete.

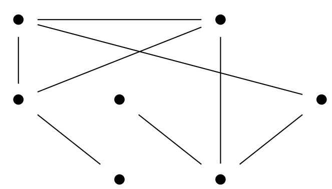
If this theorem feels "obvious", then it may be more surprising that 2SAT is not even NP-complete! This indicates that there is a *fundamental difference in the difficulty* of solving cnf-formulas with clauses of size at most 2, and cnf-formulas with clauses of size at least 3.

## Part 1.

In this tutorial, we will study the problem VERTEX-COVER, and eventually prove that it is NP-complete. VERTEX-COVER is phrased as follows: Given a graph $G$ and $k \in \mathbb{N}$, does there exist a set of $k$ vertices $v_1, \ldots, v_k$ in $G$, such that every edge in $G$ is incident with at least one of the $v_i$'s? If so, we say that $\{v_1, \ldots, v_k\}$ is a $k$-(vertex) cover of $G$.

$$\mathsf{VERTEX\text{-}COVER} = \{\langle G, k \rangle : G \text{ has a } k\text{-cover}\}$$

**Exercise 1.** Try to find a 3-cover of the following graph:



**Exercise 2.** Reason that VERTEX-COVER $\in$ NP using one of the following two methods:

1. Show that there is a polynomial time verifier $V(\langle G, k \rangle, c)$ for VERTEX-COVER. *Hint: the input $c$ codes a "solution" to the problem; the format of $c$ depends on the definition of $V$ and should be chosen by you. What counts as evidence that $G$ has a $k$-cover?*
2. Show that there is a polynomial time NTM $T(\langle G, k \rangle)$ that decides VERTEX-COVER. *Hint: at some point in its computation, $T$ will have to nondeterministically guess a property of $G$, and later it should* ***deterministically*** *check that the guess was a correct one.*

## Part 2

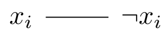
We now show that VERTEX-COVER is NP-hard, hence proving that it is NP-complete. As a sanity check, recall that it suffices to show

$$\mathsf{3SAT} \leq_{\mathrm{P}} \mathsf{VERTEX\text{-}COVER}.$$

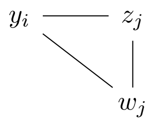
In other words, given any 3cnf-formula $\varphi(x_1, \ldots, x_n) = \varphi_1(\vec{x}) \wedge \cdots \wedge \varphi_\ell(\vec{x})$, we will construct in polynomial time (of $|\varphi|$) a graph $\Gamma\varphi$ and an integer $k$, such that $\Gamma\varphi$ has a $k$-clique iff $\varphi$ is satisfiable.

Let $\Gamma\varphi$ be the graph constructed by the following steps:

1. For each variable $x_i$ in $\varphi$, $\Gamma\varphi$ will have the subgraph

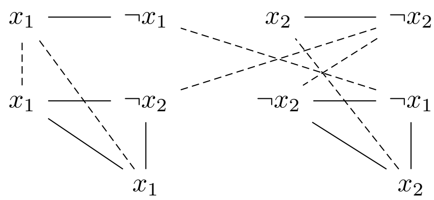



2. For each clause $\varphi_j = (y_j \vee z_j \vee w_j)$, where, for example, $y_j$ is a literal of the form $x_i$ or $\neg x_i$ for some $i$, $\Gamma\varphi$ will have the subgraph




3. Finally, connect each literal $y_j, z_j, w_j$ to the corresponding variable in part 1, respecting negation.

For example, the formula $\varphi(x_1, x_2) = (x_1 \vee \neg x_2 \vee x_1) \wedge (\neg x_2 \vee \neg x_1 \vee x_2)$ will have the graph $\Gamma\varphi$ given by




**Exercise 3.** Practice constructing the following graph:

$$\Gamma((x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_1 \vee x_1) \wedge (x_3 \vee x_2 \vee x_1))$$

**Exercise 4.** Suppose $\varphi$ is satisfiable, meaning we have an assignment $f : \{x_1, \ldots, x_n\} \to \{0, 1\}$. We construct a cover $\mathcal{C}$ of $\Gamma\varphi$ as follows:

1. $\mathcal{C}$ will contain exactly one of $x_i$ and $\neg x_i$ corresponding to whether $f(x_i)$ equals 1 or 0, respectively.
2. Since each clause $\varphi_j = (y_j \vee z_j \vee w_j)$ is satisfied, then at least one of the literals of $\varphi_j$ equals 1. Let $\mathcal{C}$ contain the vertices corresponding to the other two literals.

Using the solution $f(x_1) = 1, f(x_2) = 0, f(x_3) = 0$ for $\varphi$ in exercise 3, find the cover $\mathcal{C}$.

**Exercise 5.** Prove in general that $\mathcal{C}$ is a $k$-cover for $\Gamma\varphi$, where $k = n + 2\ell$, $n$ is the number of variables, and $\ell$ is the number of clauses.

**Exercise 6.** Now work backwards: if $\Gamma\varphi$ has a $k$-cover $\mathcal{C}$, where $k = n + 2\ell$, prove that $\varphi$ is satisfiable. *Hint: first, reason that at least one vertex from each pair $x_i, \neg x_i$ is in $\mathcal{C}$, and similarly that $\mathcal{C}$ must contain vertices corresponding to at least two literals from each clause. Finally, conclude that this choice of vertices actually represents a solution to $\varphi$.*