1 Reducing from 3-SAT to CNF-SAT

A formula *CNF* (for *Conjunctive Normal Form*) is a set of variables $x_1, \dots, x_n$ and a set of clauses $C_1, \dots, C_k$ where each clause is a disjunction of literals (a literal is either a variable x of its negation $\neg x$). For all $1 \leq j \leq k$, we have $C_j = \bigvee_i v_i^j$ where $v_i^j = x_l$ or $v_i^j = \neg x_l$ for a given l.

A CNF formula is satisfiable if it exists an assignation of the variables $x_1, \dots, x_n$ to true ($\top$) or false ($\perp$) such that each clause is satisfied (i.e. for each clause, at least one of these literals is true). From a logical point of view, a CNF formula is a formula of the form : $\exists x_1, \dots x_n \in \{\top, \perp\}^n : \bigwedge_j \left(\bigvee_i v_i^j\right)$.

Definition 1. *The* CNF-SAT *problem is to decide whether a CNF formula is satisfiable.*

Definition 2. *The* k-SAT *problem is to decide the satisfiability of a CNF formula where each clause has to have at most k literals.*

Exercice 1. Show that the *CNF-SAT* and *k-SAT* problems (for all $k \in \mathbb{N}$)) are $\mathcal{NP}$.

Exercice 2. Show that the formula $(v_1 \vee v_2 \vee v_3 \vee v_4)$ is satisfiable if and only if $(v_1 \vee v_2 \vee l) \wedge (\neg l \vee v_3 \vee v_4)$ is satisfiable (where l is a fresh variable, *i.e.* l does not appear in any of the $v_1, \dots, v_4$).

Exercice 3. Show that 3-SAT is $\mathcal{NP}$-complete. Start from CNF-SAT (which is $\mathcal{NP}$-complete) and show that for each CNF formula φ we can find φ' satisfiable iff φ also is with $|\varphi'|$ polynomial in $|\varphi|$ and where each clause of φ' contains at most 3 literals.

Definition 3. *A DNF formula (for* Disjunctive Normal Form*) is a set of variables $x_1, \dots, x_n$ and a disjunction of clauses where each clause is a conjunction of literals. A DNF formula is thus equivalent to $\exists x_1, \dots, x_n \in \{\perp, \top\}^n : \vee_j(\wedge_i v_i^j)$*

Exercice 4. Is the DNF-SAT (satisfiability of DNF formula) in $\mathcal{NP}$? in co-$\mathcal{NP}$? in $\mathcal{P}$?

2 Reducing from 3-SAT to 3-coloring

Definition 4. *Given a graph $G = (V, E)$, G is k-coloriable if we can color its nodes with k colors such that no two neighboring nodes have the same color. Formally $\exists(c : V \to \{1, \dots, k\})\ \forall(i, j) \in E\ : c(i) \neq c(j)$.*

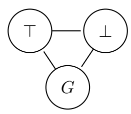
Exercice 5. Show that for each k fixed, the k-coloriability is $\mathcal{NP}$.

Exercice 6. Show that if the $k+1$-coloriability problem is in $\mathcal{P}$ then so is the k-coloriability problem.

Let us show that 3-SAT can be reduced to 3-coloriage. First let us introduce a few gadgets.

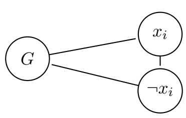
2.1 Color gadget

We will make sure that the colors in our graph represent either true ($\top$), false ($\perp$) or ground G. To encode these colors, our first gadget is to include in the graph the graph drawn below. After that we will often plug nodes to G or $\perp$ to forbid colors in nodes. This gadget is only present once in the graph.



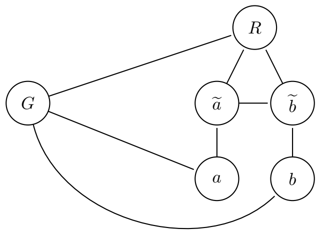
2.2 Literal gadget

For each variable x_i we create two nodes : x_i and $\neg x_i$ that we will connected to each other and to G as represented below :



2.3 OR gadget

Suppose that a and b represent boolean values (we can enforce that by plugging them into G), the OR gadget is the following (the G node is the one of the color gadget) :



Exercice 7. Show that in any valid coloring, the color of R is the color of a or b. And, if we limit the problem to this subgraph then it can be colored with either the color of a or b.

Exercice 8. Find a gadget that performs the OR between three boolean values a, b, c (represented as nodes colored either $\top$ or $\bot$).

2.4 Packing everything

Exercice 9. For each clause, design a gadget to verify it.

Exercice 10. Provide a polynomial reduction from 3-SAT to 3-coloring.

3 Miscellaneous

Exercice 11. In which classes ($\mathcal{P}$? $\mathcal{NP}$? co-$\mathcal{NP}$?) is the 2-coloring problem ?

Exercice 12. Show that SAT can be reduced to 3-SAT (i.e the satisfiability of formula composed of $\vee$, $\wedge$, $\neg$ and variables).

Exercice 13. In which classes is the 2-SAT problem ?

4 Cliques

Definition 5. *The clique problem is to decide whether a given graph contains a k-clique (k nodes all direct neighbors of each others).*

Exercice 14. Show that the clique problem is $\mathcal{NP}$-complete.

Suggestion : find a reduction with 3-SAT. Given $\bigvee_j(\bigwedge_i v_i^j)$ we create a node for each v_i^j (when a literal appears in several clauses we include it several times).