2-SAT est NL-complet

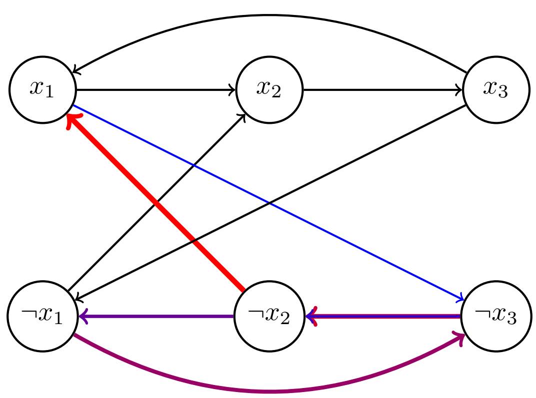
Soit $X = \{x_1, \dots, x_k\}$ un ensemble de variables booléennes. Nous écrivons $\overline{X}$ afin de dénoter $\overline{X} \coloneqq \{\neg x_1, \dots, \neg x_k\}$. Rappelons qu'un *littéral* est une variable ou la négation d'une variable de X, c.-à-d. un élément de $X \cup \overline{X}$. Nous considérons que $\neg(\neg x_i) = x_i$. Nous disons qu'une formule φ est en 2-*FNC* si elle est de la forme:

$$\varphi = \bigwedge_{1 \leq i \leq m} (\ell_i \vee \ell_i') \qquad \text{où chaque } \ell_i, \ell_i' \in X \cup \overline{X}.$$

Nous associons un *graphe d'implication* G_φ à φ. Plus précisément, $G_\varphi = (V, E)$ est le graphe dirigé défini par:

$$\begin{aligned} V &\coloneqq X \cup \overline{X}, \\ E &\coloneqq \{(\neg \ell_i, \ell_i') : i \in [1..m]\} \cup \{(\neg \ell_i', \ell_i) : i \in [1..m]\}. \end{aligned}$$

Un *cycle contradictoire* de G_φ est un chemin de la forme $x_i \to_* \neg x_i \to_* x_i$. Par exemple, le graphe d'implication de la formule $\varphi = (x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_3 \vee x_1)$, illustré ci-dessous, contient le cycle contradictoire $x_1 \to \neg x_3 \to \neg x_2 \to \neg x_1 \to \neg x_3 \to \neg x_2 \to x_1$.



Nous allons démontrer que, contrairement à 3-SAT et 3-UNSAT qui sont respectivement NP-complet et coNP-complet, les problèmes 2-SAT et 2-UNSAT sont NL-complets:

<u>2-SAT</u>

ENTRÉE:	une formule en 2-FNC φ,
DÉTERMINER:	si φ est satisfaisable.

<u>2-UNSAT</u>

ENTRÉE:	une formule en 2-FNC φ,
DÉTERMINER:	si φ n'est pas satisfaisable.

Lemme 1. *Soit φ une formule en 2-FNC satisfaite par une assignation* $\mathrm{val}\colon X \cup \overline{X} \to \{\textit{faux}, \textit{vrai}\}$. *Si $\ell \to_* \ell'$ dans G_φ et* $\mathrm{val}(\ell) = $ *vrai, alors* $\mathrm{val}(\ell') = $ *vrai.*

Démonstration. Montrons que $\ell \to_n \ell'$ et $\mathrm{val}(\ell) = \textit{vrai}$ implique $\mathrm{val}(\ell') = \textit{vrai}$ par induction sur n. Si $\ell \to_0 \ell'$, alors $\ell = \ell'$ et ainsi $\mathrm{val}(\ell') = \mathrm{val}(\ell)$. Soit $n > 0$. Supposons que l'hypothèse d'induction est satisfaite pour $n - 1$. Puisque $\ell \to_n \ell'$, il existe un littéral g tel que $\ell \to_{n-1} g \to \ell'$. Comme $\mathrm{val}(\ell) = \textit{vrai}$, par hypothèse d'induction, nous avons $\mathrm{val}(g) = \textit{vrai}$. De plus, comme $(g, \ell') \in E$, la formule φ contient la clause $\neg g \vee \ell'$. Puisque $\mathrm{val}(g) = \textit{vrai}$ et comme φ est satisfaite, nous avons forcément $\mathrm{val}(\ell') = \textit{vrai}$. □

Lemme 2. *Soit φ une formule en 2-FNC. Tous les littéraux ℓ et ℓ' de G_φ satisfont $\ell \to_* \ell'$ ssi $\neg\ell' \to_* \neg\ell$.*

Démonstration. Montrons $\ell \to_n \ell'$ ssi $\ell' \to_n \ell$ par induction sur n. Pour $n = 0$, $\ell \to_0 \ell'$ ssi $\ell' = \ell$ ssi $\ell' \to_0 \ell$. Soit $n > 0$. Supposons que l'hypothèse d'induction est satisfaite pour $n - 1$. Nous avons:

$$
\begin{aligned}
\ell \to_n \ell' &\iff \exists g \in X \cup \overline{X} : \ell \to_{n-1} g \text{ et } g \to \ell' \\
&\iff \exists g \in X \cup \overline{X} : \neg g \to_{n-1} \neg\ell \text{ et } g \to \ell' && \text{(par hypothèse d'induction)} \\
&\iff \exists g \in X \cup \overline{X} : \neg g \to_{n-1} \neg\ell \text{ et } \neg\ell' \to \neg g && \text{(car } (g, \ell') \in E) \implies (\neg\ell', \neg g) \in E) \\
&\iff \exists g \in X \cup \overline{X} : \neg\ell' \to \neg g \text{ et } \neg g \to_{n-1} \neg\ell \\
&\iff \ell' \to_n \ell.
\end{aligned}
$$

□

Proposition 1. *Une formule φ en 2-FNC est non satisfaisable ssi G_φ contient un cycle contradictoire.*

Démonstration. $\Leftarrow$) Supposons que G_φ contient un cycle contradictoire de la forme $x_i \to_* \neg x_i \to_* x_i$. Afin d'obtenir une contradiction, supposons que φ est satisfaite par une assignation val. Si $\mathrm{val}(x_i) = \textit{vrai}$, alors, par le lemme 1, nous avons $\mathrm{val}(\neg x_i) = \textit{vrai}$, ce qui est impossible. Similairement, si $\mathrm{val}(x_i) = \textit{faux}$, alors $\mathrm{val}(\neg x_i) = \textit{vrai}$ et, par le lemme 1, $\mathrm{val}(x_i) = \textit{vrai}$, ce qui est à nouveau une contradiction.

$\Rightarrow$) Montrons la contraposée. Autrement dit, supposons que G_φ ne contient aucun cycle contradictoire, et montrons que φ est satisfaisable. Nous prétendons que cet algorithme construit une assignation qui satisfait φ:

Algorithme 1 : Construction d'une assignation

Entrées : formule φ en 2-FNC sur variables X telle que G_φ ne contient aucun cycle contradictoire
Résultat : une assignation $\mathrm{val}\colon X \cup \overline{X} \to \{\textit{faux}, \textit{vrai}\}$ qui satisfait φ

1 val := $[\ell \mapsto ? : \ell \in X \cup \overline{X}]$ `// Aucun litéral assigné`
2 **tant que** $\exists \ell \in X \cup \overline{X}$ *t.q.* $\mathrm{val}(\ell) = ?$ *et* $\ell \not\to_* \neg\ell$ `// Construire l'assignation`
3   **pour** $\ell' \in X \cup \overline{X}$ *t.q.* $\ell \to_* \ell'$
4         $\mathrm{val}(\ell') := \textit{vrai}$
5         $\mathrm{val}(\neg\ell') := \textit{faux}$
6 **retourner** val

Il est clair que l'algorithme termine puisqu'un littéral assigné demeure assigné. De plus, l'algorithme assigne une valeur à chaque littéral, car G_φ ne contient aucun cycle contradictoire et ainsi $x \not\to_* \neg x$ ou $\neg x \not\to_* x$ pour toute variable $x \in X$. De plus, les lignes 4 et 5 garantissent qu'une variable et sa négation ne se contredisent pas. Il reste à prouver que l'assignation satisfait φ lorsque l'algorithme se termine.

Montrons d'abord que la boucle **pour** ne modifie pas l'assignation d'un litéral ℓ' déjà assigné. Afin d'obtenir une contradiction, supposons que c'est le cas. Cela signifie qu'il existe $\ell' \in X \cup \overline{X}$ tel que $\ell \to_* \ell'$ et $\ell \to_* \neg\ell'$. Par le lemme 2, nous avons $\neg\ell' \to_* \neg\ell$, et par conséquent $\ell \to_* \neg\ell' \to_* \neg\ell$. Cela contradit le fait que $\ell \not\to_* \neg\ell$.

Montrons maintenant que si un litéral ℓ' est déjà assigné, alors la boucle **tant que** ne modifie pas son assignation. Soit ℓ le littéral choisi au début d'une itération. Supposons sans perte de généralité que $\text{val}(\ell') = vrai$ et qu'on tente d'assigner *faux*. Cela signifie qu'à une itération antérieure nous avons choisi g tel que $g \to_* \ell'$. De plus, comme nous tentons d'assigner *faux*, nous avons $\ell \to_* \neg\ell'$. Par le lemme 2, $\ell' \to_* \neg\ell$, et par conséquent $g \to_* \ell' \to_* \neg\ell$. Rappelons que ℓ est choisi au début de l'itération car il n'est pas assigné. Or, g a été considéré à une itération antérieure, ce qui a déjà assigné *faux* à ℓ en raison de $g \to_* \neg\ell$. Il y a donc contradiction.

Montrons finalement que lorsque l'algorithme se termine, toutes les clauses de φ sont satisfaites. Supposons qu'une clause $C = (g \vee g')$ ne l'est pas. Cela signifie que $\text{val}(g) = \text{val}(g') = faux$. Ces assignations ont été faites à des itérations (possiblement distinctes) de la boucle **tant que** via des littéraux ℓ et ℓ' tels que $\ell \to_* \neg g$ et $\ell' \to_* \neg g'$. Rappelons que $\neg g \to g'$ et $\neg g' \to g$ par définition de C et G_φ. Nous obtenons donc $\ell \to_* g'$ et $\ell' \to_* g$. Ainsi, ℓ ou ℓ' a renversé une assignation de *faux* à *vrai* (pour g' ou g), ce qui contredit nos observations précédentes (c.-à-d. qu'une assignation ne change jamais). □

Proposition 2. 2-UNSAT $\in$ NL.

Démonstration. Par la proposition 1, identifier un cycle contradictoire est équivalent à montrer que la formule φ en entrée n'est pas satisfaisable. Nous pouvons donc deviner une variable x_i; vérifier que $x_i \to_* \neg x_i$; puis vérifier que $\neg x_i \to_* x_i$. Afin d'identifier ces deux chemins simples dans G_φ, nous devinons leur longueur ainsi que leurs sommets (à la volée):

Algorithme 2 : Identification non déterministe d'un cycle contradictoire

Entrées : formule φ en 2-FNC sur variables $X = \{x_1, \ldots, x_k\}$
Résultat : φ est non satisfaisable?

choisir $i \in [1..k]$ de façon non déterministe
choisir $c \in [1..2k]$ de façon non déterministe // $\exists c : x_i \to_c \neg x_i$?
$v \coloneqq x_i$;
tant que $c > 0$
    choisir $v' \in X \cup \overline{X}$ de façon non déterministe
    si $v \not\to v'$ **alors rejeter**
    sinon $v \coloneqq v'$; $c \coloneqq c - 1$
si $v \neq \neg x_i$ **alors rejeter**
choisir $c \in [1..2k]$ **de façon non déterministe** // $\exists c : \neg x_i \to_c x_i$?
tant que $c > 0$
    choisir $v' \in X \cup \overline{X}$ de façon non déterministe
    si $v \not\to v'$ **alors rejeter**
    sinon $v \coloneqq v'$; $c \coloneqq c - 1$
accepter si $v = x_i$, sinon **rejeter**

La procédure stocke un indice i, un compteur c et un sommet v. Leur représentation binaire se stocke sur $\mathcal{O}(\log k)$ bits, et par conséquent en espace logarithmique par rapport à la taille $n \geq k$ de l'entrée. □

Proposition 3. PATH $\leq_m^{\log}$ 2-UNSAT.

Démonstration. Soit $G = (V, E)$ un graphe dirigé et $s, t \in V$. Posons

$$\varphi_{G,s,t} \coloneqq (s \vee s) \wedge \bigwedge_{(x,y) \in E} (\neg x \vee y) \wedge (\neg t \vee \neg t).$$

Observons que $\varphi_{G,s,t}$ est en 2-FNC et que

$$\varphi_{G,s,t} \equiv s \wedge \bigwedge_{(x,y)\in E} (x \to y) \wedge \neg t.$$

Montrons que la fonction f définie par $f(\langle G, s, t\rangle) \coloneqq \varphi_{G,s,t}$ est une réduction. Autrement dit, nous devons montrer que $s \to_* t$ dans G ssi $\varphi_{G,s,t}$ n'est pas satisfaisable.

$\Rightarrow$) Supposons que $s \to_* t$ dans G et que $\varphi_{G,s,t}$ est satisfaite par une assignation val. Nous avons forcément $\mathrm{val}(s) = \textit{vrai}$ et $\mathrm{val}(t) = \textit{faux}$. Par définition de la formule, un raisonnement par induction sur n montre que $\mathrm{val}(y) = \textit{vrai}$ pour tout $s \to_n y$ où $n \in \mathbb{N}$. Comme $s \to_* t$, nous obtenons une contradiction.

$\Leftarrow$) Nous montrons la contraposée. Autrement dit, nous supposons que $s \not\to_* t$ et montrons que $\varphi_{G,s,t}$ est satisfaisable. Posons $S \coloneqq \{v \in V : s \to_* v\}$. Nous prétendons que $\varphi_{G,s,t}$ est satisfaite par l'assignation définie par $\mathrm{val}(x) \coloneqq \textit{vrai} \iff x \in S$. Comme $s \not\to_* t$, nous avons $t \notin S$ et ainsi $\mathrm{val}(t) = \textit{faux}$. De plus, $\mathrm{val}(s) = \textit{vrai}$. Afin d'obtenir une contradiction, supposons qu'il existe un clause $(\neg x \vee y)$ enfreinte. Cela signifie que $\mathrm{val}(x) = \textit{vrai}$ et $\mathrm{val}(y) = \textit{faux}$. Comme $\mathrm{val}(x) = \textit{vrai}$, nous avons $x \in S$ et ainsi $s \to_* x$. De plus, par définition de $\varphi_{G,s,t}$, nous avons $x \to y$. Cela implique $s \to_* y$ et ainsi $y \in S$, ce qui contradit $\mathrm{val}(y) = \textit{faux}$. □

Corollaire 1. 2-SAT *et* 2-UNSAT *sont* NL-*complets.*

Démonstration. La proposition 2 montre que 2-UNSAT $\in$ NL. Comme le problème PATH est NL-difficile, la proposition 3 implique que 2-UNSAT est NL-difficile. Par conséquent, 2-UNSAT est NL-complet. Par le théorème d'Immerman-Szelepcsényi, NL $=$ coNL, ce qui implique que 2-SAT est aussi NL-complet. □