

# Problème SAT

Quentin Fortier

April 21, 2022

# Forme normale disjonctive et conjonctive

## Théorème

Toute formule logique  $\varphi$  est équivalente à une formule sous **forme normale disjonctive**, c'est à dire de la forme  $\varphi = c_1 \vee \dots \vee c_n$  où  $c_i$  est de la forme  $x_1 \wedge \dots \wedge x_p$  avec  $x_1, \dots, x_p$  des littéraux (variable ou négation d'une variable).

# Forme normale disjonctive et conjonctive

## Théorème

Toute formule logique  $\varphi$  est équivalente à une formule sous **forme normale disjonctive**, c'est à dire de la forme  $\varphi = c_1 \vee \dots \vee c_n$  où  $c_i$  est de la forme  $x_1 \wedge \dots \wedge x_p$  avec  $x_1, \dots, x_p$  des littéraux (variable ou négation d'une variable).

Preuve :

$$\varphi = \bigvee_{\substack{v \text{ valuation} \\ \text{tq } \llbracket \varphi \rrbracket_v = 1}} \bigwedge_{\substack{x \in V \\ \text{tq } v(x) = 1}} x$$

# Forme normale disjonctive et conjonctive

## Définition

Une **forme normale conjonctive** est une conjonction de disjonctions de littéraux, c'est à dire une formule de la forme  $c_1 \wedge \dots \wedge c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \vee \dots \vee \ell_p$ .

# Forme normale disjonctive et conjonctive

## Définition

Une **forme normale conjonctive** est une conjonction de disjonctions de littéraux, c'est à dire une formule de la forme  $c_1 \wedge \dots \wedge c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \vee \dots \vee \ell_p$ .

## Théorème

Toute formule logique  $\varphi$  est équivalente à une formule sous forme normale conjonctive.

Preuve :

# Forme normale disjonctive et conjonctive

## Définition

Une **forme normale conjonctive** est une conjonction de disjonctions de littéraux, c'est à dire une formule de la forme  $c_1 \wedge \dots \wedge c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \vee \dots \vee \ell_p$ .

## Théorème

Toute formule logique  $\varphi$  est équivalente à une formule sous forme normale conjonctive.

Preuve :  $\neg\varphi$  est équivalente à une forme normale disjonctive, c'est à dire  $\neg\varphi \equiv c_1 \vee \dots \vee c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \wedge \dots \wedge \ell_p$ .

# Forme normale disjonctive et conjonctive

## Définition

Une **forme normale conjonctive** est une conjonction de disjonctions de littéraux, c'est à dire une formule de la forme  $c_1 \wedge \dots \wedge c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \vee \dots \vee \ell_p$ .

## Théorème

Toute formule logique  $\varphi$  est équivalente à une formule sous forme normale conjonctive.

Preuve :  $\neg\varphi$  est équivalente à une forme normale disjonctive, c'est à dire  $\neg\varphi \equiv c_1 \vee \dots \vee c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \wedge \dots \wedge \ell_p$ .  
Alors  $\neg\neg\varphi = \neg(c_1 \vee \dots \vee c_k) \equiv \neg c_1 \wedge \dots \wedge \neg c_k$  (de Morgan).

# Forme normale disjonctive et conjonctive

## Définition

Une **forme normale conjonctive** est une conjonction de disjonctions de littéraux, c'est à dire une formule de la forme  $c_1 \wedge \dots \wedge c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \vee \dots \vee \ell_p$ .

## Théorème

Toute formule logique  $\varphi$  est équivalente à une formule sous forme normale conjonctive.

Preuve :  $\neg\varphi$  est équivalente à une forme normale disjonctive, c'est à dire  $\neg\varphi \equiv c_1 \vee \dots \vee c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \wedge \dots \wedge \ell_p$ .

Alors  $\neg\neg\varphi = \neg(c_1 \vee \dots \vee c_k) \equiv \neg c_1 \wedge \dots \wedge \neg c_k$  (de Morgan).

Et  $\neg c_i = \neg(\ell_1 \wedge \ell_2 \wedge \dots \wedge \ell_p) \equiv \neg\ell_1 \vee \dots \vee \neg\ell_p$  (de Morgan).



# Forme normale disjonctive et conjonctive

## Définition

Une **forme normale conjonctive** est une conjonction de disjonctions de littéraux, c'est à dire une formule de la forme  $c_1 \wedge \dots \wedge c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \vee \dots \vee \ell_p$ .

## Théorème

Toute formule logique  $\varphi$  est équivalente à une formule sous forme normale conjonctive.

Preuve :  $\neg\varphi$  est équivalente à une forme normale disjonctive, c'est à dire  $\neg\varphi \equiv c_1 \vee \dots \vee c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \wedge \dots \wedge \ell_p$ .

Alors  $\neg\neg\varphi = \neg(c_1 \vee \dots \vee c_k) \equiv \neg c_1 \wedge \dots \wedge \neg c_k$  (de Morgan).

Et  $\neg c_i = \neg(\ell_1 \wedge \ell_2 \wedge \dots \wedge \ell_p) \equiv \neg\ell_1 \vee \dots \vee \neg\ell_p$  (de Morgan).

Donc  $\varphi \equiv \neg\neg\varphi$  est bien équivalente à une forme normale conjonctive.

# Forme normale disjonctive et conjonctive

## Définition

Une **forme normale conjonctive** est une conjonction de disjonctions de littéraux, c'est à dire une formule de la forme  $c_1 \wedge \dots \wedge c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \vee \dots \vee \ell_p$ .

## Théorème

Toute formule logique  $\varphi$  est équivalente à une formule sous forme normale conjonctive.

Preuve :  $\neg\varphi$  est équivalente à une forme normale disjonctive, c'est à dire  $\neg\varphi \equiv c_1 \vee \dots \vee c_k$  où chaque  $c_i$  est de la forme  $\ell_1 \wedge \dots \wedge \ell_p$ .

Alors  $\neg\neg\varphi = \neg(c_1 \vee \dots \vee c_k) \equiv \neg c_1 \wedge \dots \wedge \neg c_k$  (de Morgan).

Et  $\neg c_i = \neg(\ell_1 \wedge \ell_2 \wedge \dots \wedge \ell_p) \equiv \neg\ell_1 \vee \dots \vee \neg\ell_p$  (de Morgan).

Donc  $\varphi \equiv \neg\neg\varphi$  est bien équivalente à une forme normale conjonctive.

Autre preuve possible : par induction structurelle sur  $\varphi$ .

**Question 20** Pour chacune des formules suivantes, utiliser l'involutivité de la négation, l'associativité et la distributivité des connecteurs  $\wedge$  et  $\vee$ , ainsi que les lois de De Morgan pour transformer la formule en FNC. Seul le résultat du calcul est demandé :

a)  $(x_1 \vee \neg x_0) \wedge \neg(x_4 \wedge \neg(x_3 \wedge x_2))$

b)  $(x_0 \wedge x_1) \vee (x_2 \wedge x_3) \vee (x_4 \wedge x_5)$

## Problème $k$ -SAT

Le problème  $k$ -SAT consiste à déterminer si une formule  $\varphi$ , sous forme normale conjonctive dont chaque clause comporte  $k$  littéraux, est satisfiable.

## Problème $k$ -SAT

Le problème  $k$ -SAT consiste à déterminer si une formule  $\varphi$ , sous forme normale conjonctive dont chaque clause comporte  $k$  littéraux, est satisfiable.

❶ 1-SAT :

## Problème $k$ -SAT

Le problème  $k$ -SAT consiste à déterminer si une formule  $\varphi$ , sous forme normale conjonctive dont chaque clause comporte  $k$  littéraux, est satisfiable.

- 1 1-SAT : satisfiable ssi  $\varphi$  ne contient pas à la fois une variable et sa négation.

Complexité :  $O(n)$ ,  $n$  étant le nombre de variables dans  $\varphi$ .

- 2 2-SAT :

## Problème $k$ -SAT

Le problème  $k$ -SAT consiste à déterminer si une formule  $\varphi$ , sous forme normale conjonctive dont chaque clause comporte  $k$  littéraux, est satisfiable.

- ❶ 1-SAT : satisfiable ssi  $\varphi$  ne contient pas à la fois une variable et sa négation.

Complexité :  $O(n)$ ,  $n$  étant le nombre de variables dans  $\varphi$ .

- ❷ 2-SAT : se ramène à un problème de graphe dont les sommets sont les littéraux de  $\varphi$ .

Pour toute clause  $\ell_1 \vee \ell_2$ , équivalente à  $\neg\ell_1 \implies \ell_2$ , on ajoute un arc  $(\neg\ell_1, \ell_2)$ .

$\varphi$  est alors satisfiable ssi aucune composante fortement connexe ne contient une variable et sa négation.

# Forme normale disjonctive et conjonctive

## Théorème

Si on peut résoudre 3-SAT en complexité polynomiale (en le nombre de variables), alors on peut aussi résoudre  $k$ -SAT en complexité polynomiale.



# Forme normale disjonctive et conjonctive

## Théorème

Si on peut résoudre 3-SAT en complexité polynomiale (en le nombre de variables), alors on peut aussi résoudre  $k$ -SAT en complexité polynomiale.

Preuve : soit  $\varphi$  une formule  $k$ -SAT et  $c = \ell_1 \vee \dots \vee \ell_k$  une de ses clauses.

# Forme normale disjonctive et conjonctive

## Théorème

Si on peut résoudre 3-SAT en complexité polynomiale (en le nombre de variables), alors on peut aussi résoudre  $k$ -SAT en complexité polynomiale.

Preuve : soit  $\varphi$  une formule  $k$ -SAT et  $c = \ell_1 \vee \dots \vee \ell_k$  une de ses clauses. Alors :

$$c \equiv (\ell_1 \vee \ell_2 \vee x_1) \wedge (\neg x_1 \vee \ell_3 \vee x_2) \wedge (\neg x_2 \vee \ell_4 \vee x_3) \dots \wedge (\neg x_{k-3} \vee \ell_{k-1} \vee \ell_k)$$

où  $x_1, \dots, x_{k-3}$  sont des nouvelles variables.

# Forme normale disjonctive et conjonctive

## Théorème

Si on peut résoudre 3-SAT en complexité polynomiale (en le nombre de variables), alors on peut aussi résoudre  $k$ -SAT en complexité polynomiale.

Preuve : soit  $\varphi$  une formule  $k$ -SAT et  $c = \ell_1 \vee \dots \vee \ell_k$  une de ses clauses. Alors :

$$c \equiv (\ell_1 \vee \ell_2 \vee x_1) \wedge (\neg x_1 \vee \ell_3 \vee x_2) \wedge (\neg x_2 \vee \ell_4 \vee x_3) \dots \wedge (\neg x_{k-3} \vee \ell_{k-1} \vee \ell_k)$$

où  $x_1, \dots, x_{k-3}$  sont des nouvelles variables.

On peut donc transformer  $\varphi$  en une formule 3-SAT, en multipliant au plus par  $k$  le nombre de variables.

## Définition

On appelle **NP** l'ensemble des problèmes possédant un **certificat polynomial**, c'est-à-dire qu'il est possible de vérifier en complexité polynomiale qu'une solution potentielle est correcte.

Exemple : 3-SAT est dans NP puisqu'étant donné une valuation, il est possible de vérifier si une formule est vraie en complexité polynomiale en la taille de la formule (en appelant la fonction `eval`).

## Définition

On appelle **NP** l'ensemble des problèmes possédant un **certificat polynomial**, c'est-à-dire qu'il est possible de vérifier en complexité polynomiale qu'une solution potentielle est correcte.

Exemple : 3-SAT est dans NP puisqu'étant donné une valuation, il est possible de vérifier si une formule est vraie en complexité polynomiale en la taille de la formule (en appelant la fonction `eval`).

## Définition

Si  $P_1$  et  $P_2$  sont des problèmes algorithmiques, une **réduction** de  $P_1$  à  $P_2$  est une transformation polynomiale d'une instance (entrée)  $I_1$  de  $P_1$  en une instance  $I_2$  de  $P_2$ , telle que la résolution de  $I_2$  permette de résoudre  $I_1$ .

Dans ce cas,  $P_2$  est considéré comme étant « plus difficile que  $P_1$  ».

## Définition

On appelle **NP** l'ensemble des problèmes possédant un **certificat polynomial**, c'est-à-dire qu'il est possible de vérifier en complexité polynomiale qu'une solution potentielle est correcte.

Exemple : 3-SAT est dans NP puisqu'étant donné une valuation, il est possible de vérifier si une formule est vraie en complexité polynomiale en la taille de la formule (en appelant la fonction `eval`).

## Définition

Si  $P_1$  et  $P_2$  sont des problèmes algorithmiques, une **réduction** de  $P_1$  à  $P_2$  est une transformation polynomiale d'une instance (entrée)  $I_1$  de  $P_1$  en une instance  $I_2$  de  $P_2$ , telle que la résolution de  $I_2$  permette de résoudre  $I_1$ .

Dans ce cas,  $P_2$  est considéré comme étant « plus difficile que  $P_1$  ».

## Définition

Un problème  $P$  est **NP-difficile** si tout problème de NP se réduit à  $P$ .

Un problème est **NP-complet** s'il est NP-difficile et qu'il est dans NP.

Conjecture à 1 million : il est impossible de résoudre un problème NP-complet en complexité polynomiale.

# Classe de complexité

## Définition

Un problème  $P$  est **NP-difficile** si tout problème de NP se réduit à  $P$ .  
Un problème est **NP-complet** s'il est NP-difficile et qu'il est dans NP.

Conjecture à 1 million : il est impossible de résoudre un problème NP-complet en complexité polynomiale.

## Théorème de Cook

SAT est NP-complet.

On a vu précédemment que SAT se réduit à 3-SAT, donc 3-SAT est aussi NP-complet.



La technique classique pour montrer qu'un problème est NP-complet est de montrer une réduction depuis 3-SAT :

## Problème CLIQUE

Entrée : un graphe  $G$  et un entier  $k$ .

Sortie : **true** si  $G$  possède une clique de taille  $k$  (un sous-graphe complet à  $k$  sommets), **false** sinon.

# Classe de complexité

La technique classique pour montrer qu'un problème est NP-complet est de montrer une réduction depuis 3-SAT :

## Problème CLIQUE

Entrée : un graphe  $G$  et un entier  $k$ .

Sortie : **true** si  $G$  possède une clique de taille  $k$  (un sous-graphe complet à  $k$  sommets), **false** sinon.

## Exercice

Montrer que CLIQUE est NP-complet.

## Exercice

Montrer que CLIQUE est NP-complet.

On montre que 3-SAT se réduit à CLIQUE. Soit  $\varphi = c_1 \wedge c_2 \wedge \dots \wedge c_k$  une formule 3-SAT. On va construire un graphe  $G$  à partir de  $\varphi$ .

## Exercice

Montrer que CLIQUE est NP-complet.

On montre que 3-SAT se réduit à CLIQUE. Soit  $\varphi = c_1 \wedge c_2 \wedge \dots \wedge c_k$  une formule 3-SAT. On va construire un graphe  $G$  à partir de  $\varphi$ . Pour chaque clause  $c$ , on met un sommet pour chaque littéral apparaissant dans  $c$ .

## Exercice

Montrer que CLIQUE est NP-complet.

On montre que 3-SAT se réduit à CLIQUE. Soit  $\varphi = c_1 \wedge c_2 \wedge \dots \wedge c_k$  une formule 3-SAT. On va construire un graphe  $G$  à partir de  $\varphi$ .

Pour chaque clause  $c$ , on met un sommet pour chaque littéral apparaissant dans  $c$ .

On relie deux sommets si les littéraux correspondants sont dans des clauses différentes et qu'ils ne sont pas négation l'un de l'autre.

## Exercice

Montrer que CLIQUE est NP-complet.

On montre que 3-SAT se réduit à CLIQUE. Soit  $\varphi = c_1 \wedge c_2 \wedge \dots \wedge c_k$  une formule 3-SAT. On va construire un graphe  $G$  à partir de  $\varphi$ .

Pour chaque clause  $c$ , on met un sommet pour chaque littéral apparaissant dans  $c$ .

On relie deux sommets si les littéraux correspondants sont dans des clauses différentes et qu'ils ne sont pas négation l'un de l'autre.

Si  $G$  a une clique  $C$  de taille  $k$ , on peut assigner  $T$  à chaque sommet de  $C$ , ce qui satisfait chacune des  $k$  clauses.

# Classe de complexité

Réduction pour la formule  $C_1 \wedge C_2 \wedge C_3$  :

