# Logique propositionnelle

MPSI Lycée Pierre de Fermat

# Introduction

Nous avons vu que les expressions arithmétiques, peuvent s'écrire sous la forme d'arbres. Dans ce chapitre, on étudie les expressions booléennes. On étudiera tout d'abord leur syntaxe, c'est à dire les constructeurs que l'on utilise pour les définir, et de manière plus importante leur sémantique. On parlera de **formules logiques** plutôt que d'expressions.

De nombreux problèmes d'informatique peuvent se réduire à des problèmes logiques tels que "cette formule est-elle toujours vraie ?" ou bien "est-il possible de rendre cette formule vraie ?". A ce titre, on s'attardera sur l'étude de ces problèmes et des algorithmes associés.

# 1 Syntaxe et sémantique

## A Syntaxe

On se donne un ensemble infini $\mathcal{Q}$ de variables. On notera ses éléments $Q, P, Q', Q_1 \dots$.

**Définition 1.** On définit inductivement l'ensemble des formules propositionnelles $\mathcal{F}$, dont les éléments seront notés $\varphi, \psi, \varphi', \dots$, comme suit :

— Toute variable $Q \in \mathcal{Q}$ est une formule : $Q \in \mathcal{F}$

— Il existe deux constantes, notées $\top$ ("top") et $\bot$ ("bottom") : $\top, \bot \in \mathcal{F}$. Elles correspondent respectivement à une formule toujours vraie et une formule toujours fausse.

— Pour $\varphi_1, \varphi_2 \in \mathcal{F}$, on peut construire les formules $\varphi_1 \wedge \varphi_2$ ("$\varphi_1$ ET $\varphi_2$") et $\varphi_1 \vee \varphi_2$ ("$\varphi_1$ OU $\varphi_2$"). Le OU est inclusif.

— Pour $\varphi_1, \varphi_2 \in \mathcal{F}$, on peut construire la formule $\varphi_1 \to \varphi_2$ ("$\varphi_1$ IMPLIQUE $\varphi_2$").

— Pour $\varphi \in \mathcal{F}$, on peut construire la formule $\neg\varphi$ ("NON $\varphi$").

$\wedge, \vee, \neg, \to$ sont appelés **connecteurs logiques** :

- $\neg$ s'appelle "négation" ;
- $\wedge$ s'appelle "conjonction" ;
- $\vee$ s'appelle "disjonction" ;
- $\to$ s'appelle "implication".

**Exemple 1.** Voici quelques formules :

1. $Q \vee \neg Q$
2. $(P \wedge Q) \vee (\neg P \wedge \neg Q)$
3. $(P \wedge \top) \to (\neg P \vee Q)$
4. $(\bot \vee \neg Q) \wedge (P \wedge \neg Q)$

Notons qu'une formule est un objet **syntaxique**, qui n'a pas de sens mathématique en soit. En particulier, bien qu'en mathématiques, le tiers exclus indique qu'une formule de la forme "$X$ ou non $X$" soit toujours vraie, la formule $Q \vee \neg Q \in \mathcal{F}$ n'a pas de valeur de vérité intrinsèque (pour le moment).
De manière générale, une formule n'est pas vraie ou fausse : sa valeur de vérité dépend des valeurs que l'on assigne à ses variables.
De plus, deux formules qui, intuitivement, seraient identiques, comme $X \wedge Y$ et $Y \wedge X$, sont bien deux objets distincts.

## B Algèbre des booléens

Pour donner une sémantique, c'est à dire un sens, aux formules, leur en donner un, il nous faut définir une manière d'évaluer une formule.

**Définition 2.** L'algèbre des booléens, notée $\mathbb{B}$, est l'ensemble $\{0, 1\}$ muni des opérations $+, \times$ et $\bar{(\cdot)}$ définies comme suit :

— $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 1$

— $0 \times 0 = 0$, $0 \times 1 = 0$, $1 \times 0 = 0$, $1 \times 1 = 1$

— $\bar{0} = 1$, $\bar{1} = 0$

0 et 1 correspondent à faux et vrai, et les opérateurs $+, \times$ et $\bar{(\cdot)}$ correspondent au ET, OU et NON logique.

L'algèbre des booléens sert à représenter la sémantique des formules. On souligne à nouveau la différence entre syntaxe et sémantique : les formules $\top \wedge \bot$ et $\bot \wedge \top$ sont distinctes, mais $1 \times 0 = 0 \times 1 = 0$.

**Proposition 1.** 0 est un élément neutre pour $+$ et absorbant pour $\times$. 1 est un élément absorbant pour $\times$ et neutre pour $+$.

Si $f : \mathbb{B}^n \to \mathbb{B}$ est une fonction, on peut la représenter par sa ***table de vérité***. Cette table contient $2^n$ lignes, une pour chaque élément de $\mathbb{B}^n$. Sur chaque ligne, on fait figurer la valeur de chaque entrée de la fonction, ainsi que la valeur de sortie pour cette entrée.

**Exemple 2.** On considère la fonction $f(x, y, z) = \bar{x} + (y \times x) + (\bar{z} \times y)$. Sa table de vérité est :

| $x$ | $y$ | $z$ | $f(x, y, z)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

On remarque que c'est la même table de vérité que $g(x, y, z) = \bar{x} + y$ !

**Exercice 1.**

**Question 1.** Calculer la table de vérité de la fonction $f(x, y, z) = ((x \times y) + (\bar{z} \times x)) \times (\bar{x} + \bar{y} + \bar{z})$

**Question 2.** Si $f(x_1, \ldots, x_n)$ est une fonction $n$-aire, expliquer comment obtenir la table de vérité de $g(x_1, \ldots, x_n) = \overline{f(x_1, \ldots, x_n)}$ à partir de celle de $f$.

**Question 3.** Montrer que $+$ et $\times$ sont associatives, et qu'elles sont distributives l'une sur l'autre, en utilisant des tables de vérité.

## C Sémantique

Nous pouvons maintenant définir la sémantique des formules propositionnelles, ce qui va revenir à donner aux connecteurs logiques et aux constantes le sens intuitif que l'on veut leur donner

**Définition 3.** Soit $\sigma : \mathcal{Q} \to \mathbb{B}$ une fonction, appelée **valuation**. L'interprétation d'une formule $\varphi \in \mathcal{F}$ dans $\sigma$, notée $[\![\varphi]\!]^\sigma$, se définit par induction sur l'ensemble des formules :

— $[\![\top]\!]^\sigma = 1$
— $[\![\bot]\!]^\sigma = 0$
— $[\![\varphi \wedge \psi]\!]^\sigma = [\![\varphi]\!]^\sigma \times [\![\psi]\!]^\sigma$
— $[\![\varphi \vee \psi]\!]^\sigma = [\![\varphi]\!]^\sigma + [\![\psi]\!]^\sigma$
— $[\![\neg\varphi]\!]^\sigma = \overline{[\![\varphi]\!]^\sigma}$
— $[\![\varphi \to \psi]\!]^\sigma = \overline{[\![\varphi]\!]^\sigma} + [\![\psi]\!]^\sigma$

Lorsque $[\![\varphi]\!]^\sigma = 1$, on dit que $\sigma$ satisfait $\varphi$, ou que $\sigma$ est un **modèle** de $\varphi$. On dit que $\varphi$ est **satisfiable** s'il existe une valuation la satisfaisant. On dit que c'est une **tautologie** si toutes les valuations la satisfont.

**Exercice 2.**

**Question 1.** Pour la valuation $\sigma = [P \mapsto 1, Q \mapsto 0, R \mapsto 1]$, calculer l'interprétation des formules suivantes :

1. $Q \vee \neg Q$
2. $(P \wedge Q) \vee (\neg P \wedge \neg Q)$
3. $P \wedge \top$
4. $(\bot \vee \neg Q) \wedge (P \wedge \neg Q)$
5. $(P \wedge Q) \vee (R \wedge \neg Q)$
6. $P \wedge (Q \wedge R)$
7. $(P \wedge Q) \wedge R$

**Question 2.** Pour chacune des formules précédentes, dire si elle est satisfiable, et si c'est une tautologie.

Ainsi, pour $\sigma$ une valuation fixée, on peut regarder son effet sur les formules, c'est à dire considérer la fonction

$$[\![\cdot]\!]^\sigma : \mathcal{F} \to \mathbb{B}$$

Cependant, il est plus intéressant de fixer une formule $\varphi$ et regarder son interprétation dans différentes valuation.

Si l'on numérote les variables de $\varphi$ en les notant $X_1, X_2, \ldots, X_n$, alors on peut voir $\varphi$ comme une fonction de $\mathbb{B}^n$ dans $\mathbb{B}$ :

$$[\![\varphi]\!]^{(\cdot)} : \begin{array}{ccc} \mathbb{B}^n & \longrightarrow & \mathbb{B} \\ (a_1, \ldots, a_n) & \longmapsto & [\![\varphi]\!]^{[X_1 \mapsto a_1, \ldots, X_n \mapsto a_n]} \end{array}$$

**Exemple 3.** La formule $X \wedge (\neg Y \vee Z)$ peut se voir comme la fonction $f : (x, y, z) \mapsto x \times (\bar{y} + z)$

Soulignons à nouveau la différence entre syntaxe et sémantique. Au niveau syntaxique, les formules $X \wedge (Y \wedge \top)$ et $X \wedge Y$ sont distinctes, alors qu'au niveau sémantique, elles donnent lieu à la même fonction de $\mathbb{B}^2$ dans $\mathbb{B}$ : $f(x, y) = x \times y$.

En pratique, on s'autorisera tout de même à écrire $P \wedge Q \wedge R$ sans expliciter le parenthésage : l'interprétation étant la même par associativité de $\times$. Idem pour $P \vee Q \vee R$.

**Définition 4.** On dit que deux formules $\varphi, \psi$ sont logiquement équivalentes, ce que l'on notera $\varphi \equiv \psi$, si $[\![\varphi]\!]^\sigma = [\![\psi]\!]^\sigma$ pour toute valuation $\sigma$.

**Proposition 2.** $\equiv$ est une relation d'équivalence sur $\mathcal{F}$.

**Proposition 3.** Soient $\varphi, \psi, \varphi', \psi' \in \mathcal{F}$ telles que $\varphi \equiv \varphi'$ et $\psi \equiv \psi'$. Alors :

— $\varphi \wedge \psi \equiv \varphi' \wedge \psi'$
— $\varphi \vee \psi \equiv \varphi' \vee \psi'$
— $\varphi \to \psi \equiv \varphi' \to \psi'$
— $\neg\varphi \equiv \neg\varphi'$

**Exercice 3.**

**Question 1.** Pour chaque couple de formules $(\varphi, \psi)$, déterminer si elles sont équivalentes, et si ce n'est pas le cas, donner une valuation permettant de les différencier (i.e. telle que $[\![\varphi]\!]^\sigma \neq [\![\psi]\!]^\sigma$).

— $\varphi = X \wedge Y, \psi = X \vee Y$
— $\varphi = X \vee (Y \wedge Z), \psi = (X \vee Y) \wedge (X \vee Z)$
— $\varphi = (\neg x \vee (Y \wedge Z)) \wedge ((X \wedge Y) \vee (X \wedge Z)), \psi = \bot$
— $\varphi = X \to (\neg X \wedge Y), \psi = \neg X$

**Question 2.** Montrer le principe de contraposée :

$$\forall \varphi, \psi \in \mathcal{F}, \varphi \to \psi \equiv \neg\psi \to \neg\varphi$$

**Proposition 4.** Soient $\varphi, \psi \in \mathcal{F}$. Alors :

— $\varphi \to \psi \equiv \neg\varphi \vee \psi$
— $\neg\neg\varphi \equiv \varphi$
— $\neg(\varphi \wedge \psi) \equiv (\neg\varphi) \vee (\neg\psi)$
— $\neg(\varphi \vee \psi) \equiv (\neg\varphi) \wedge (\neg\psi)$

Les deux dernières identités sont les **lois de De Morgan**.

*Démonstration.* En traçant les tables de vérités. □

On peut également s'intéresser à une relation plus faible que l'équivalence, appelée la conséquence logique :

**Définition 5.** 1. Soient $\varphi, \psi \in \mathcal{F}$. On dit que $\varphi$ est conséquence logique de $\psi$ si toute valuation qui satisfait $\psi$ satisfait aussi $\varphi$. On note $\psi \models \varphi$ :

$$\psi \models \varphi \iff \forall \sigma : \mathcal{Q} \to \mathbb{B}, [\![\psi]\!]^\sigma = 1 \Rightarrow [\![\varphi]\!]^\sigma = 1$$

2. Soit $\Gamma \subseteq \mathcal{F}$ un ensemble de formules, et $\varphi \in \mathcal{F}$. On dit que $\varphi$ est conséquence logique de $\Gamma$ si pour toute valuation $\sigma$ qui satisfait chaque formule de $\Gamma$, $\sigma$ satisfait aussi $\varphi$ :

$$\Gamma \models \varphi \iff \forall \sigma : \mathcal{Q} \to \mathbb{B}, (\forall \psi \in \Gamma, [\![\psi]\!]^\sigma = 1) \Rightarrow [\![\varphi]\!]^\sigma = 1$$

La notation $\Gamma \models \varphi$ se lit "$\varphi$ est conséquence de $\Gamma$." De manière informelle, $\Gamma$ représente un ensemble d'hypothèses, et $\varphi$ une conclusion : $\Gamma \models \varphi$ signifie alors que si l'on suppose vraies les hypothèses $\Gamma$, alors la conclusion $\varphi$ est aussi vraie.

**Exemple 4.** 1. Si $X, Y \in \mathcal{Q}$, alors $X \wedge Y \models X$ car toute valuation qui satisfait $X \wedge Y$ satisfait en particulier $X$. Informellement, cela revient au fait que si l'on sait que $X \wedge Y$ est vrain alors forcément $X$ l'est aussi.

2. Pour $\Gamma = \{X, X \to Y\}$, on a $\Gamma \models Y$. En effet, si $\sigma$ est une valuation telle que $[\![X]\!]^\sigma = 1$ et $[\![X \to Y]\!]^\sigma = 1$, alors $\sigma(X) = 1$ et $\overline{sigma(X)} + \sigma(Y) = 1$, donc $\sigma(Y) = 1$. Informellement, on a dit que si l'on suppose que $X$ est vraie, et que $X$ implique $Y$, alors $Y$ est forcément vraie.

**Exercice 4.** Soient $X, Y, Z \in \mathcal{Q}$. Pour chacune des conséquences logiques suivantes, dire si elles sont vraies ou fausse. Si elles sont vraies, en faire une preuve. Sinon, donner une valuation constituant un contre-exemple adéquat.

— $X \models X \vee Y$
— $X \vee Y \models X$
— $X \to Y \models Y \to X$
— $X \to Y \models \neg Y \to \neg X$
— $X \to Y \to Z \models (X \wedge Y) \to Z$
— $X, X \to Y \models Y$
— $\neg X, X \models Y$
— $X \to Y, Y \to Z \models X \to Z$

Lorsqu'une formule $\varphi$ est conséquence logique de l'ensemble vide, i.e. $\emptyset \models \varphi$, on notera simplement $\models \varphi$. Cela signifie que $\varphi$ est toujours vraie, sans hypothèses nécessaires. Remarque : $\models \varphi$ signifie exactement que $\varphi$ est une tautologie, i.e. que $\varphi \equiv \top$.

**Exercice 5.** Les MPSI unissent leurs forces et organisent le vol des effaceurs fétiches des MP2I. Adrien et Béatrice sont les deux MPSI les plus habiles, mais il faut déterminer qui va aller voler les effaceurs. Après avoir délibéré, les MPSI ont déterminé :

1. Adrien ou Béatrice doit voler l'effaceur ;
2. Si Béatrice vole l'effaceur, elle doit le faire avant minuit ;
3. Si le vol a eu lieu **après** minuit, alors Adrien participera au vol ;
4. Le vol aura lieu avant minuit.

**Question 1.** On considère trois variables propositionnelles $A, B, M$ correspondant respectivement à "Adrien participe", "Béatrice participe" et "le casse a eu lieu après minuit". Donnez 4 formules $\varphi_1, \dots \varphi_4$ correspondant aux 4 informations récoltées par les MPSI.

**Question 2.** On note $\Gamma = \{\varphi_1, \dots, \varphi_4\}$. Quel est le sens naturel de $\Gamma \models A$ et $\Gamma \models B$ ? A t'on l'une (ou les deux) de ces deux conséquences logiques ?

**Question 3.** On rajoute l'information suivante : Si Adrien participe, alors le vol doit avoir lieu après minuit. Conclure.

# 2 Satisfiabilité

De nombreuses situations peuvent se modéliser comme un ensemble de formules logiques correspondant à des contraintes. Les variables propositionnelles correspondent alors aux différentes possibilités, aux évènements qui peuvent survenir, et une valuation est la donnée d'une configuration parmi toutes celles possibles. On peut alors se poser des questions comme "existe-t'il une configuration qui respecte les contraintes ?", autrement dit, existe-t-il une valuation qui satisfait toutes les formules.

## A Le problème SAT

Le **problème de satisfiabilité booléenne**, ou **SAT**, est un problème fondamental en informatique.

**Définition 6.** Le problème **SAT** est de savoir, étant donné une formule propositionnelle $\varphi$, s'il existe $\sigma$ une valuation satisfaisant $\varphi$.

On dit que c'est un problème de **décision**, car il attend une réponse oui/non. En pratique, lorsque l'on résout un problème SAT,on veut aussi savoir **quelle** valuation satisfait la formule.

Les formules propositionnelles peuvent servir à encoder de nombreux problèmes, de telle sorte que résoudre **SAT** équivaut à résoudre le problème initial.

**Exemple 5.** Un club très sélectif affiche à l'entrée certaines règles d'admissions :

1. Tout membre qui a une écharpe doit avoir un chapeau et des lunettes
2. Aucun membre ne peut avoir un chapeau et une écharpe
3. Si un membre n'a pas d'écharpe, il doit avoir des lunettes
4. Si un membre a un chapeau ou une écharpe, alors il a aussi des lunettes

On se demande si ce club admet des membres. Pour cela, on considère des variables propositionnelles $C, E, L$, dont le sens sera "avoir un chapeau", "avoir une écharpe" et "avoir des lunettes". Les règles du club se modélisent alors par les formules suivantes :

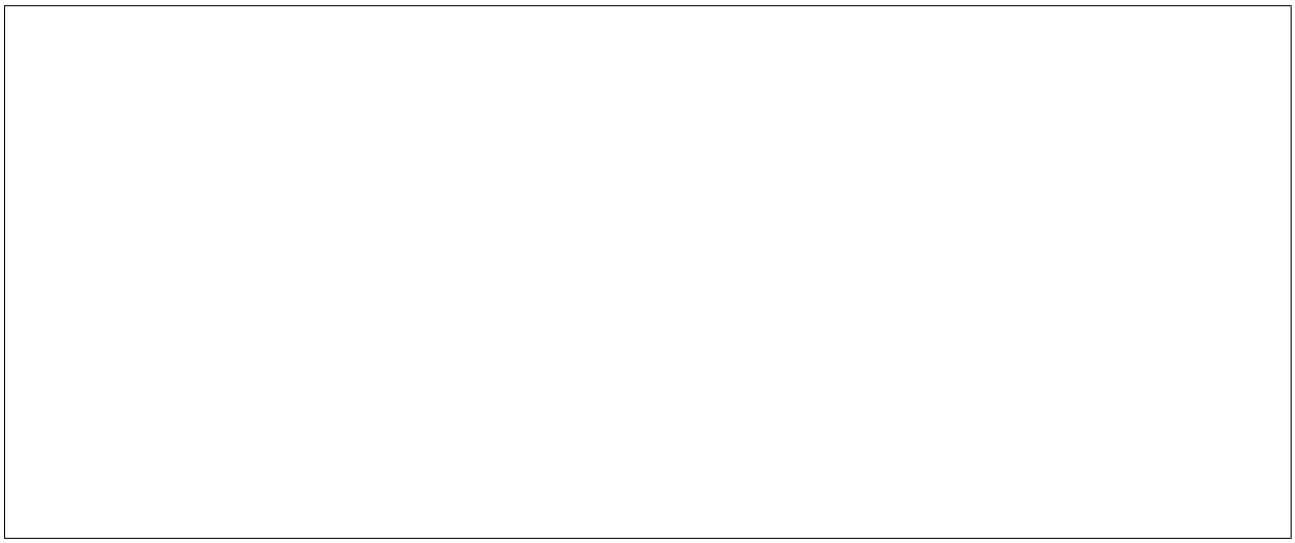
— $\varphi_1 = E \to (C \wedge L)$

— $\varphi_2 = \neg(C \wedge E)$

— $\varphi_3 = \neg E \to L$

— $\varphi_4 = (C \vee E) \to \neg L)$

Alors, le club admet au moins un membre si et seulement si la formule $\Phi = \bigwedge_{i=1}^{4} \varphi_i$ est satisfiable. De plus, une valuation satisfaisant $\Phi$ donne des conditions précises permettant d'intégrer le club. On peut déterminer si $\Phi$ est satisfiable en construisant sa table de vérité, et conclure.



Voyons un deuxième exemple plus complexe :

**Exemple 6.** On considère une grille de Sudoku $4 \times 4$. On encode cette grille en une formule comme suit :

— Pour $(i, j) \in [\![1, 4]\!]$ et $k \in [\![1, 4]\!]$, on considère une variable propositionnelle $M_{i,j,k}$. On veut faire en sorte que cette variable corresponde à l'assertion "La case $(i, j)$ contient le chiffre $k$."

— Pour chaque case $(i, j)$ on considère la formule :

$$A_{i,j} = (\bigvee_{k=1}^{4} M_{i,j,k}) \wedge \neg \left[ \bigvee_{1 \leq k \neq k' \leq 4} (M_{i,j,k} \wedge M_{i,j,k'}) \right]$$

Cette formule exprime qu'une case a au moins un chiffre écrit, mais pas plus : il faut qu'il existe $k$ tel que $M_{i,j,k}$ est vrai, mais pas qu'il existe $k \neq k'$ tels que $M_{i,j,k}$ et $M_{i,j,k'}$ sont vrais en même temps.

— Pour chaque ligne $i \in [\![1, 4]\!]$ et chaque chiffre $k \in [\![1, 4]\!]$, on considère la formule :

$$L_{i,k} = \neg \bigvee_{1 \leq j \neq j' \leq 4} (M_{i,j,k} \wedge M_{i,j',k})$$

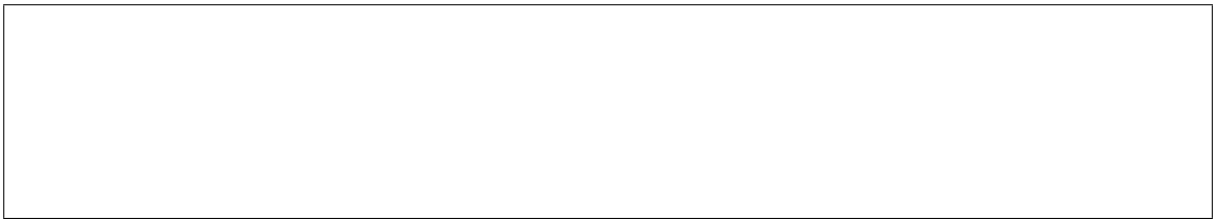
Cette formule exprime que sur la ligne $i$, la couleur $k$ ne figure qu'une seule fois : il ne faut pas qu'il existe $j \neq j'$ tels que $M_{i,j,k}$ et $M_{i,j',k}$ sont vrais en même temps.

— De même, pour chaque colonne $j \in [\![1, 4]\!]$ et chaque chiffre $k \in [\![1, 4]\!]$, on considère la formule :

$$C_{j,k} = \neg \bigvee_{1 \leq i \neq i' \leq 4} (M_{i,j,k} \wedge M_{i',j,k})$$

Cette formule exprime que sur la colonne $j$, le chiffre $k$ ne figure qu'une seule fois.

— Pour $i_0, j_0 \in [\![1, 2]\!]$ et $k \in [\![1, 4]\!]$, on considère une formule $S_{i_0,j_0,k}$ exprimant le fait qu'au sein du grand carré $i_0, j_0$, le chiffre $k$ ne figure qu'une seule fois :



— Enfin, on numérote les cases déjà remplies $(i_1, j_1), (i_2, j_2), \ldots, (i_p, j_p)$ et on note $k_1, k_2, \ldots, k_p$ les chiffres inscrits. On considère la formule :

$$P = \bigwedge_{t=1}^{p} M_{i_t,j_t,k_t}$$

Cette formule exprime que les cases en questions contiennent un chiffre fixé à l'avance.

Enfin, on considère la grande formule composée de la conjonction de toutes les formules précédentes.

$$\Phi = \bigwedge_{i,j} A_{i,j} \wedge \bigwedge_{i,k} L_{i,k} \wedge \bigwedge_{j,k} C_{j,k} \wedge \bigwedge_{i_0,j_0,k} S_{i_0,j_0,k} \wedge P$$

Alors, $\Phi$ est satisfiable si et seulement la grille peut être résolue. De plus, une valuation satisfaisant $\Phi$ donne directement une solution de la grille, et inversement. En effet :

— Si la grille peut être résolue, on note $G = (G[i,j])_{1\leq i,j\leq 4}$ la grille complétée. On considère alors la valuation $\sigma_G$ suivante :

$$\forall i,j,k \in [\![1,4]\!], \sigma(M_{i,j,k}) = \begin{cases} 1 & \text{si } G[i,j] = k \\ 0 & \text{sinon} \end{cases}$$

Remarquons qu'alors, les variables propositionnelles de la forme $M_{i,j,G[i,j]}$ sont mises à 1 par $\sigma_G$, et toutes les autres à 0.

Montrons que $\sigma_G$ satisfait $\Phi$. Il suffit de montrer que $\sigma_G$ satisfait chacune des formules $A_{i,j}, L_{i,k}, C_{j,k}, S_{i_0,j_0,k}$ et $P$. Chacune se vérifie facilement car elles découlent des règles du Sudoku. Par exemple :

— Pour $i,j \in [\![1,4]\!]$, $A_{i,j}$ est satisfaite car $\sigma(M_{i,j,G[i,j]}) = 1$ et $\sigma(M_{i,j,k}) = 0$ pour les autres valeurs de $k$. Donc $[\![\bigvee_{k=1}^{4} M_{i,j,k}]\!]^{\sigma_G} = 1$ et $[\![\bigvee_{1\leq k\neq k'\leq 4}(M_{i,j,k} \wedge M_{i,j,k'})]\!]^{\sigma_G} = 0$, et ainsi $[\![A_{i,j}]\!]^{\sigma_G} = 1$.

— $P$ est satisfaite car si la grille $G$ est une solution de la grille initiale, alors en particulier les cases qui étaient déjà présentes n'ont pas changé. Autrement dit, en reprenant les notations utilisées pour définir $P$, pour $t \in [\![1,p]\!]$, $G[i_t,j_t] = k_t$, et donc $\sigma(M_{i_t,j_t,k_t}) = 1$, et donc $[\![P]\!]^{\sigma_G} = 1$.

— A l'inverse, si $\sigma$ est une valuation satisfaisant $\Phi$, montrons que l'on peut en extraire une solution de la grille. Tout d'abord, remarquons que $\sigma$ satisfait chacune des $A_{i,j}$. Or, cette formule est construite précisément de telle sorte que cela implique qu'il existe un unique $k_{i,j}$ tel que $M_{i,j,k} = 1$. On considère alors la grille $G = (G[i,j])_{1\leq i,j\leq 4}$ avec $G[i,j] = k_{i,j}$. Il reste à vérifier que cette grille est une solution de la grille initiale, en vérifiant que chaque nombre apparaît une seule fois dans chaque ligne, colonne, et carré, et que les cases pré-remplies n'ont pas changé. Pour cela, on repasse par les sous-formules de $\Phi$, qui expriment précisément ces contraintes. Par exemple :

— Vérifions que chaque ligne $i$ contient une seule fois chaque nombre. Pour $k \in [\![1,4]\!]$, la formule $L_{i,k}$ exprime qu'il n'existe aucun couple de colonnes $j \neq j'$ tels que $\sigma(M_{i,j,k}) = 1$ et $\sigma(M_{i,j',k}) = 1$, i.e. tels que $G[i,j] = G[i,j'] = 1$. Autrement dit, le chiffre $k$ n'apparaît pas deux fois sur la ligne.

Le problème **SAT** est un problème fondamental en informatique : on ne connait pas à ce jour d'algorithme polynomial pour le résoudre, et s'il en existe un, alors $\mathbf{P} = \mathbf{NP}$ (ce qui résoudrait l'un des (si ce n'est le) plus célèbres problèmes en informatique théorique). **P** est la classe des problèmes résoluble en temps polynomial, et **NP**, que vous étudierez l'année prochaine, est la classe des problèmes dont on peut vérifier une solution en temps polynomial.

## B Forme normale

Étant donne une formule $\varphi$, il existe une infinité de formules équivalentes. Les formes normales vont donner des formes canoniques pour les formules.

**Définition 7.** Soit $\varphi \in \mathcal{F}$. Alors :

- Si $\varphi$ est de la forme $Q$ ou $\neg Q$, avec $Q \in \mathcal{Q}$, on dit que $\varphi$ est un **littéral**
- Si $\varphi$ est de la forme $L_1 \vee L_2 \vee \cdots \vee L_n$ avec $L_1, \ldots, L_n$ des littéraux, on dit que c'est une **clause disjonctive** de taille $n$
- Si $\varphi$ est de la forme $L_1 \wedge L_2 \wedge \cdots \wedge L_n$ avec $L_1, \ldots, L_n$ des littéraux, on dit que c'est une **clause conjonctive** de taille $n$
- Si $\varphi$ est de la forme $C_1 \wedge \cdots \wedge C_p$ avec $C_1, \ldots, C_p$ des clauses disjonctive, on dit que $\varphi$ est sous **forme normale conjonctive** (FNC)
- Si $\varphi$ est de la forme $C_1 \vee \cdots \vee C_p$ avec $C_1, \ldots, C_p$ des clauses conjonctive, on dit que $\varphi$ est sous **forme normale disjonctive** (FND)

Une formule sous FNC est donc une conjonction de disjonctions de littéraux.

**Exemple 7.** Quelques exemples sur ces notions :

- $X \vee Y \vee Z \vee \neg T$ est une clause disjonctive
- $(X \vee Y \vee \neg T) \wedge (Y \vee \neg Y \vee \neg Z) \wedge (Y \vee Z \vee T)$ est une formule sous FNC.
- $X \wedge (Y \vee (T \wedge Z) \vee (\neg Y \wedge Z))$ n'est pas sous FNC

**Remarque 1.** Soit $\varphi = \bigwedge_{i=1}^{p} \bigvee_{j=1}^{m_i} L_{ij}$ une formule sous forme normale **conjonctive**. Alors une valuation $\sigma$ satisfait $\varphi$ si et seulement pour tout $i \in [\![1, p]\!]$, il existe $j \in [\![1, m_i]\!]$ tel que $\sigma$ satisfait $L_{ij}$.

**Exercice 6.** Proposer une remarque similaire pour les formules sous FND.

**Proposition 5.** Toute formule est équivalente à une formule sous FNC, et à une formule sous FND.

*Démonstration.* Soit $\varphi$ une formule. Commençons par trouver une FND pour $\varphi$. Notons $V = \{X_1, \ldots, X_n\}$ l'ensemble fini des variables apparaissant dans $\varphi$. On trace la table de vérité de $\varphi$ sur $V$. On obtient alors $2^n$ lignes :

| $X_1$ | $\ldots$ | $X_{n-1}$ | $X_n$ | $[\![\varphi]\!]$ |
|---|---|---|---|---|
| 0 | $\ldots$ | 0 | 0 | $s_0$ |
| 0 | $\ldots$ | 0 | 1 | $s_1$ |
| 0 | $\ldots$ | 1 | 0 | $s_2$ |
| 0 | $\ldots$ | 1 | 1 | $s_3$ |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ |
| 1 | $\ldots$ | 1 | 1 | $s_{2^n-1}$ |

avec $s_i = 0$ ou 1 pour $i \in [\![0, 2^n - 1]\!]$.

On note $K \subseteq [\![0, 2^n - 1]\!]$ l'ensemble des $k$ telle que $S_k = 1$. Pour $k \in K$, on note $C_k$ la clause constitué des conjonction des $X_i$ valués à 1 sur la ligne $k$ et des $\neg X_i$ pour $X_i$ valué à 0 sur la ligne $k$. Par exemple, si la ligne $k$ contient $1, 0, 1, 1$ pour $X_1, X_2, X_3, X_4$, alors la clause correspondante sera $X_1 \wedge \neg X_2 \wedge X_3 \wedge X_4$.

Alors $\varphi$ est satisfaite exactement sur les lignes de $K$ :

$$\varphi \equiv \bigvee_{k \in K} C_k$$

Cette méthode permet de construire une FND équivalente pour toute formule.
Pour déterminer une FNC de $\varphi$, on commence par déterminer une FND de $\neg\varphi$ de la forme $\bigvee_{i=1}^{p} C_i$ avec les $C_i$ clauses conjonctives. Alors, $\varphi$ est équivalente logiquement à $\neg \bigvee_{i=1}^{p} C_i$. En appliquant les lois de De Morgan, cette formule est équivalente à $\bigwedge_{i=1}^{p} \neg C_i$. En appliquant à nouveau les lois de De Morgan sur les $C_i$, qui sont des conjonctions, on transforme chaque $\neg C_i$ en une clause disjonctive $D_i$, et donc $\varphi$ est équivalente à $\bigwedge_{i=1}^{p} D_i$. □

**Exemple 8.** Soit $\varphi = (X \to (Y \wedge \neg Z)) \wedge (\neg(Y \wedge Z) \vee X)$.

Sa table de vérité est :

| $X$ | $Y$ | $Z$ | $[\![\varphi]\!]$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Alors les 4 lignes ayant un 1 sur la colonne de sortie permettent de reconstruire une FND de $\varphi$ :

$$\varphi \equiv (\neg X \wedge \neg Y \wedge \neg Z) \vee (\neg X \wedge \neg Y \wedge Z) \vee (\neg X \wedge Y \wedge \neg Z) \vee (X \wedge Y \wedge \neg Z)$$

Les 4 lignes ayant un 1 correspondent aux lignes où $\neg\varphi$ est satisfaite : on en déduit une FNC pour $\varphi$ :

$$\varphi \equiv (X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee Y \vee Z) \wedge (\neg X \vee Y \vee \neg Z) \wedge (\neg X \vee \neg Y \vee \neg Z)$$

**Définition 8.** On définit les problèmes de décision $\mathbf{FNC-SAT}$, $\mathbf{FND-SAT}$ et $\mathbf{3-SAT}$ comme suit :

- $\mathbf{FNC-SAT}$ : étant donné $\varphi$ une variable sous FNC, $\varphi$ est-elle satisfiable ?
- $\mathbf{FND-SAT}$ : étant donné $\varphi$ une variable sous FND, $\varphi$ est-elle satisfiable ?
- $\mathbf{3-SAT}$ : étant donné $\varphi$ une variable sous FNC donc chaque clause contient au plus trois littéraux, $\varphi$ est-elle satisfiable ?

$\mathbf{FNC-SAT}$ et $\mathbf{3-SAT}$ sont aussi durs que $\mathbf{SAT}$ : ils font partie de la classe de problèmes appelée $\mathbf{NP-complets}$, pour lesquels on ne connaît aucun algorithme en temps polynomial. En revanche, $\mathbf{FND-SAT}$ est très simple à résoudre.

**Exercice 7.**

**Question 1.** Montrer que pour $\varphi$ et $\psi$ deux formules quelconques, $\varphi \vee \psi$ est satisfiable si et seulement si $\varphi$ est satisfiable ou $\psi$ est satisfiable.

**Question 2.** Est-ce vrai pour $\varphi \wedge \psi$ ?

**Question 3.** Déterminer un algorithme polynomial permettant de déterminer si une formule sous FND est satisfiable. Cet algorithme prendra en entrée une liste de clauses, chaque étant elle même une liste de littéraux.

Lorsque l'on modélise un problème par une formule, il est courant de chercher à mettre cette formule sous FNC. En effet, on ne perds pas en expressivité, puisque les problèmes $\mathbf{FNC-SAT}$ et $\mathbf{SAT}$ sont aussi durs l'un que l'autre. Cependant, les FNC sont plus structurées, et donc plus maniables.