Logique propositionnelle

Dans ce chapitre, on étudie les outils mathématiques permettant de modéliser les expressions booléennes sans entrer dans le détail des sous-expressions non booléennes. On fera bien la distinction entre ce qui relève de la syntaxe des formules, c'est-à-dire comment elles sont écrites, et ce qui relève de leur sémantique, c'est-à-dire les valeurs qu'on leur donne.

0.1 Introduction et rappels

Rappel (*arbres de syntaxe*) :

On peut représenter les expressions booléennes à l'aide d'arbres de syntaxe.

Exemple : Pour l'expression `y mod x == 0 && false`, on a l'arbre suivant :

La logique propositionnelle de modéliser un grand nombre de situations ou de problèmes.

Exemple : Considérons un jeu de Sudoku en 4×4, comme représenté ci-dessous :
Notons $(M_{i,j})_{\substack{i\in[1..4]\\ j\in[1..4]}}$ les cases du tableau. Alors, en notant $\mathcal{P}_{i,j,k}$ la propriété "$M_{i,j} = k$", on a :

$$\bigwedge_{k=1}^{4} \bigwedge_{i=1}^{4} \left(\bigvee_{j=1}^{4} \mathcal{P}_{i,j,k} \right)$$

Cette formule traduit/modélise la règle : "pour tout $k \in [1..4]$, il existe sur chaque ligne i une colonne j telle que la case (i, j) contient k", permettant de décider si une solution est valide.

Remarque : On voit à travers cet exemple que les quantificateurs sur des ensembles finis se tradusent par des conjonctions et des disjonctions finies. La logique avec des quantificateurs, que nous n'étudions pas dans ce cours, s'appelle la logique du 1er ordre.

1 Syntaxe de la logique propositionnelle

Pour toute cete partie, on fixe $\mathcal{Q}$ un ensemble non vide de symboles appelés variables propositionnelles.

1.1 Définition inductive des formules

1.1.1 Avec des règles de construction

Définition (*construction des formules de la logique propositionnelle*) :

L'ensemble des formules de la logique propositionnelle, noté $\mathbb{F}_p(\mathcal{Q})$, est construit par induction à partir des règles de construction suivantes :

- $\mathcal{V}ar|^0_{\mathcal{Q}}$
- $\top|^0_{\{\cdot\}}$ (vrai)
- $\bot|^0_{\{\cdot\}}$ (faux)
- $\neg|^1_{\{\cdot\}}$ (négation)
- $\wedge|^2_{\{\cdot\}}$ (conjonction)
- $\vee|^2_{\{\cdot\}}$ (disjonction)

- $\rightarrow|^2_{\{\cdot\}}$ (implication)
- $\leftrightarrow|^2_{\{\cdot\}}$ (équivalence – mais attention, ce n'est pas une relation d'équivalence)

Remarque : Ce système n'est pas minimal, autrement dit on pourrait obtenir un ensemble de formules équivalent à partir de moins de règles. Par exemple, on peut omettre $A \vee B$, que l'on désignerait alors par $\neg\left((\neg A) \wedge (\neg B)\right)$.

Afin d'alléger les écritures, on note désormais :

$\top$ *pour* $(\top, \_)$
$\bot$ *pour* $(\bot, \_)$
$\mathcal{V}ar(q)$ *pour* $(\mathcal{V}ar, q)$
$\neg A$ *pour* $(\neg, \_, A)$
$A \wedge B$ *pour* $(\wedge, \_, A, B)$
$A \vee B$ *pour* $(\vee, \_, A, B)$
$A \rightarrow B$ *pour* $(\rightarrow, \_, A, B)$
$A \leftrightarrow B$ *pour* $(\leftrightarrow, \_, A, B)$

1.1.2 En français

1.2 Représentation sous forme d'arbres

Une formule de $\mathbb{F}_p(\mathcal{Q})$ peut être réprésentée par un arbre binaire non vide dont les feuilles sont étiquetées par $\mathcal{Q} \cup \{\top, \bot\}$ et dont les nœuds d'arité 1 et 2 sont respectivement étiquetés par $\{\neg\}$ et $\{\vee, \wedge, \rightarrow, \leftrightarrow\}$.

Exemple : Pour $\mathcal{Q} = \{x, y, z\}$, on a par exemple :

Remarque : Dans cette représentation, une sous-formule correspond à un sous-arbre.

Définition (*hauteur et taille d'une formule*) :

La hauteur (resp. taille) d'une formule est la hauteur (resp. taille) de l'arbre qui le représente. Formellement, on définit la hauteur et la taille des formules de $\mathbb{F}_p(\mathcal{Q})$ par induction comme suit :

$$h = \begin{pmatrix} \mathbb{F}_p(\mathcal{Q}) \rightarrow \mathbb{N} \\ \left.\begin{matrix} \top \\ \bot \\ q \in Q \end{matrix}\right\} \mapsto 0 \\ \neg A \mapsto 1 + h(A) \\ A \,\alpha\, B \mapsto 1 + \max(h(A), h(B)) \end{pmatrix} \qquad s = \begin{pmatrix} \mathbb{F}_p(\mathcal{Q}) \rightarrow \mathbb{N} \\ \left.\begin{matrix} \top \\ \bot \\ q \in \mathcal{Q} \end{matrix}\right\} \mapsto 1 \\ \neg A \mapsto 1 + s(A) \\ A \,\alpha\, B \mapsto 1 + s(A) + s(B) \end{pmatrix}$$

où à chaque fois on a $\alpha \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$.

Exercice : Donner de même la définition de quatre fonctions donnant les nombres et les ensembles de feuilles et de variables.

1.3 Conjonction et disjonction

Notations (*conjonction et disjonction de n formules*) :

Soit $(A_i)_{i\in[1..n]} \in \mathbb{F}_p(\mathcal{Q})^n$ avec $n \in \mathbb{N}^*$. On note :
- $\bigwedge_{i=1}^{n} A_i$ pour désigner la formule $((A_1 \wedge A_2) \wedge A_3 \dots) \wedge A_n$
- $\bigvee_{i=1}^{n} A_i$ pour désigner la formule $((A_1 \vee A_2) \vee A_3 \dots) \vee A_n$.

Formellement : ***i.*** $\bigwedge\limits_{i=1}^{1} A_i = A_1$ et $\forall\, n \geq 2$, $\bigwedge\limits_{i=1}^{n} A_i = \left(\bigwedge\limits_{i=1}^{n-1} A_i\right) \wedge A_n$

ii. $\bigvee\limits_{i=1}^{1} A_i = A_1$ et $\forall\, n \geq 2$, $\bigvee\limits_{i=1}^{n} A_i = \left(\bigvee\limits_{i=1}^{n-1} A_i\right) \vee A_n$

Plus généralement, si $I \neq \emptyset$ est un ensemble fini et ordonné et si $(A_i)_{i\in I} \in \mathbb{F}_p(\mathcal{Q})^I$, on s'autorisera à écrire $\bigwedge\limits_{i\in I} A_i$ et $\bigvee\limits_{i\in I} A_i$. Enfin, on conviendra que $\bigwedge\limits_{i\in\emptyset} A_i = \top$ et $\bigvee\limits_{i\in\emptyset} A_i = \bot$.

1.4 Formes normales

Définitions (*littéraux et clauses*) :

Soit $A \in \mathbb{F}_p(\mathcal{Q})$. Alors :
- A est un littéral s'il existe $q \in \mathcal{Q}$ tel que $A = q$ ou $A = \neg q$.
- A est une clause (disjonctive) si c'est une disjonction de littéraux.
- Pour $n \in \mathbb{N}^*$, A est une n-clause si c'est une disjonction d'exactement n littéraux.

Définition (*formes normales*) :

Soit $A \in \mathbb{F}_p(\mathcal{Q})$.

- A est sous forme normale conjonctive (FNC) si c'est une conjonction de clauses disjonctives.
- A est sous forme normale disjonctive (FND) si c'est une disjonction de conjonctions de littéraux, c'est-à-dire une disjonction de clauses conjonctives.

Exemples : $((x \vee y) \vee \neg z) \vee \neg x \vee x$ est une 5-clause.
$(a \vee b) \wedge (a \vee b \vee c)$ est une formule sous FNC.

Remarque : Pour $q \in Q$, q et $\neg q$ sont des clauses (plus précisément, des 1-clauses).

2 Algèbre booléenne

2.1 Définition

Définition (*algèbre de Boole*) :

Notons $\mathbb{B}$ l'ensemble $\{\mathsf{V}, \mathsf{F}\}$. On munit $\mathbb{B}$ des trois opérations suivantes :

$$+ = \begin{pmatrix} \mathbb{B}\times\mathbb{B} \to \mathbb{B} \\ (\mathsf{F},\mathsf{F}) \mapsto \mathsf{F} \\ (\mathsf{F},\mathsf{V}) \mapsto \mathsf{V} \\ (\mathsf{V},\mathsf{F}) \mapsto \mathsf{V} \\ (\mathsf{V},\mathsf{V}) \mapsto \mathsf{V} \end{pmatrix} \qquad \times = \begin{pmatrix} \mathbb{B}\times\mathbb{B} \to \mathbb{B} \\ (\mathsf{F},\mathsf{F}) \mapsto \mathsf{F} \\ (\mathsf{F},\mathsf{V}) \mapsto \mathsf{F} \\ (\mathsf{V},\mathsf{F}) \mapsto \mathsf{F} \\ (\mathsf{V},\mathsf{V}) \mapsto \mathsf{V} \end{pmatrix} \qquad \overline{\cdot} = \begin{pmatrix} \mathbb{B} \to \mathbb{B} \\ \mathsf{V} \mapsto \mathsf{F} \\ \mathsf{F} \mapsto \mathsf{V} \end{pmatrix}$$

$\mathbb{B}$ muni de ces opérations, c'est-à-dire $(\mathbb{B}, +, \times, \overline{\cdot})$, est appelé l'algèbre de Boole.

Remarque : $\overline{\cdot}$ ne correspond pas au passage à l'opposé (inversion pour $+$), ni au passage à l'inverse (inversion pour $\times$).

Propriétés (*lois* $+$, $\times$ *et* $\overline{\cdot}$) :

On a les propriétés suivantes : ***i.*** $+$ et $\times$ sont associatives et commutatives
ii. F est neutre pour $+$ et absorbant pour $\times$
iii. V est neutre pour $\times$ et absorbant pour $+$
iv. $\overline{\cdot}$ est involutive

Preuve :
i. On montre l'associativité en étudiant tous les cas possibles, que l'on présente dans un tableau appelé table de vérité (*cf.* définition plus loin). Soit donc $(a, b, c) \in \mathbb{B}^3$. On a :

Propriétés (*distributivité*) :

i. $+$ est distributive par rapport à $\times$, c'est-à-dire :

$$\forall\,(a,b,c) \in \mathbb{B}^3,\ \begin{cases} (a \times b) + c = (a + c) \times (b + c) \\ c + (a \times b) = (c + a) \times (c + b) \end{cases}$$

ii. $\times$ est distributive par rapport à $+$, c'est-à-dire :

$$\forall\,(a,b,c) \in \mathbb{B}^3,\ \begin{cases} (a + b) \times c = (a \times c) + (b \times c) \\ c \times (a + b) = (c \times a) + (c \times b) \end{cases}$$

Preuve :

2.2 Fonctions booléennes

Définition (*fonction booléenne*) :

Pour $n \in \mathbb{N}^*$, on appelle fonction booléenne d'arité n une fonction de $\mathbb{B}^n$ dans $\mathbb{B}$.

Remarque : $\mathrm{Card}(\mathbb{B}^n) = 2^n$ et $\mathrm{Card}(\mathbb{B}^{\mathbb{B}^n}) = \mathrm{Card}(\mathcal{F}(\mathbb{B}^n, \mathbb{B})) = 2^{2^n}$.

Définition (*table de vérité d'une fonction booléenne*) :

Soit $n \in \mathbb{N}^*$ et $f \in \mathcal{F}(\mathbb{B}^n, \mathbb{B})$.

On appelle table de vérité de f un tableau T ayant 2^n lignes (indicées de 1 à 2^n) et $n + 1$ colonnes (indicées de 1 à $n + 1$), à valeurs dans $\mathbb{B}$ et tel que :

- $\left\{(T_{i,j})_{j\in[1..n]} \,\middle|\, i \in [1..2^n]\right\} = \mathbb{B}^n$
- $\forall\, i \in [1..2^n],\ f((T_{i,j})_{j\in[1..n]}) = T_{i,n+1}$

Autrement dit, les lignes de T couvrent l'ensemble des n-uplets de $\mathbb{B}^n$ et pour chaque ligne, la dernière colonne donne l'image par f du n-uplet constitué des premières cases de cette ligne.

Remarque : Il est recommandé d'énumérer les 2^n n-uplets en comptant en binaire afin de plus facilement les lister exhaustivement et sans doublons.

Remarque : Par ailleurs, on s'autorisera à rassembler les tables de vérité de plusieurs fonctions booléennes de même arité dans un même tableau en factorisant les premières colonnes.

3 Sémantique de la logique propositionelle

Dans cette partie, on fixe à nouveau $\mathcal{Q}$ un ensemble non vide de symboles et on note $\mathbb{B}$ l'algèbre de Boole.

3.1 Interprétation

Définition (*environnement propositionnel*) :

On appelle environnement propositionnel une fonction de $\mathcal{Q}$ dans $\mathbb{B}$. On l'appelle également valeurs de vérité, ou encore assignation des variables.

Définition (*interprétation selon un environnement*) :

Soit $\rho \in \mathbb{B}^{\mathcal{Q}}$ un environnement propositionnel. On définit l'interprétation selon ρ des formules de la logique propositionnelle sur $\mathcal{Q}$ par :

$$[\cdot]^\rho = \begin{pmatrix} \mathbb{F}_p(\mathcal{Q}) \to \mathbb{B} \\ \top \mapsto \mathsf{V} \\ \bot \mapsto \mathsf{F} \\ q \in \mathcal{Q} \mapsto \rho(q) \\ \neg A \mapsto \overline{[A]^\rho} \\ A \vee B \mapsto [A]^\rho + [B]^\rho \\ A \wedge B \mapsto [A]^\rho \times [B]^\rho \\ A \to B \mapsto \overline{[A]^\rho} + [B]^\rho = [\neg A \vee B]^\rho \\ A \leftrightarrow B \mapsto ([A]^\rho \times [B]^\rho) + (\overline{[A]^\rho} \times \overline{[B]^\rho}) \end{pmatrix}$$

Définition (*satisfiabilité, tautologie, antilogie*) :

Soit $A \in \mathbb{F}_p(\mathcal{Q})$.

- Si $\rho \in \mathbb{B}^{\mathcal{Q}}$ est telle que $[A]^\rho = \mathsf{V}$, alors on dit que ρ satisfait A.
- On dit que A est satisfiable s'il existe $\rho \in \mathbb{B}^{\mathcal{Q}}$ tel que ρ satsifait A.
- On dit que A est valide ou bien que A est une tautologie ssi $\forall\, \rho \in \mathbb{B}^{\mathcal{Q}}$, ρ satisfait A.
- On dit que A est insatisfiable ou que c'est une antilogie ssi $\forall\, \rho \in \mathbb{B}^{\mathcal{Q}}$, ρ ne satisfait pas A.

Exemples : $\top$ et $(x \vee \neg x)$ sont des tautologies. $\bot$ et $(x \wedge \neg x)$ sont quant à elles des antilogies.

Remarque : Notons qu'antilogie n'est pas la négation de tautologie.

3.2 Fonction booléenne associée à une formule

On change maintenant de point de vue : en effet, $[A]^\rho$ dépend de de A mais aussi de ρ. Dans la section précédente, on a vu que la dépendance en A pour $\rho \in \mathbb{B}^{\mathcal{Q}}$ fixé était donnée par la fonction $[\cdot]^\rho = A \mapsto [A]^\rho$.

Pour la dépendance en ρ à $A \in \mathbb{F}_p(\mathcal{Q})$ fixé, on a la fonction $[\![\cdot]\!]^A = \rho \mapsto [A]^\rho$ définie ci-après.

Définition (*fonction booléenne associée à une formule*) :

Soit $A \in \mathbb{F}_p(\mathcal{Q})$. On appelle fonction booléenne associée à la formule A la fonction :

$$[\![\cdot]\!]^A = \begin{pmatrix} \mathbb{B}^{\mathcal{Q}} \to \mathbb{B} \\ \rho \mapsto [A]^\rho \end{pmatrix}$$

Remarque : Toute fonction booléenne d'arité $n \in \mathbb{N}^*$ est la fonction booléenne associée d'une formule propositionnelle sur un ensemble de variables propositionnelles de cardinal n (*cf.* section "mise sous forme normale").

Définition (*équivalence logique*) :

On définit la relation binaire $\equiv$ sur $\mathbb{F}_p(\mathcal{Q})$ par :

$$\begin{aligned} \forall\, (A,B) \in \mathbb{F}_p(\mathcal{Q})^2,\ A \equiv B &\iff [\![\cdot]\!]^A = [\![\cdot]\!]^B \\ &\iff \forall\, \rho \in \mathbb{B}^{\mathcal{Q}},\ [\![\rho]\!]^A = [\![\rho]\!]^B \\ &\iff \forall\, \rho \in \mathbb{B}^{\mathcal{Q}},\ [A]^\rho = [B]^\rho \end{aligned}$$

On dit que A et B sont logiquement équivalentes, et on parlera d'équivalence logique, si $A \equiv B$.

Propriété : La relation $\equiv$ est une relation d'équivalence sur $\mathbb{F}_p(\mathcal{Q})$.

Exemple : Pour $(A,B) \in \mathbb{F}_p(\mathcal{Q})^2$, on a toujours $A \vee B \equiv B \vee A$.
En effet, $\forall\, \rho \in \mathbb{B}^{\mathcal{Q}}$, $\begin{aligned}[t][A \vee B]^\rho &= [A]^\rho + [B]^\rho \text{ par définition de l'interprétation} \\ &= [B]^\rho + [A]^\rho \text{ par commutativité de } + \\ &= [B \vee A]^\rho \text{ en réutilisant la définition de l'interprétation.}\end{aligned}$

Exercice : Soit $(A,B) \in \mathbb{F}_p(\mathcal{Q})^2$. Démontrer les équivalences logiques suivantes :
- ***a)*** $A \wedge B \equiv B \wedge A$
- ***b)*** $A \to B \equiv (\neg A) \vee B$
- ***c)*** $A \to B \equiv (\neg B) \to (\neg A)$
- ***d)*** $A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$
- ***e)*** $A \vee \neg A \equiv \top$
- ***f)*** $(A \to B) \wedge A \equiv A \wedge B$.

Correction :
a) $\forall\, \rho \in \mathbb{B}^{\mathcal{Q}}$, $[A \wedge B]^\rho = [A]^\rho \times [B]^\rho = [B]^\rho \times [A]^\rho = [B \wedge A]^\rho$ d'où $A \wedge B \equiv B \wedge A$.

b) $\forall\, \rho \in \mathbb{B}^{\mathcal{Q}}$, $[A \to B]^\rho = \overline{[A]^\rho} + [B]^\rho = [\neg A]^\rho + [B]^\rho = [\neg A \vee B]^\rho$, d'où $A \to B \equiv \neg A \vee B$.

d) $\forall\, \rho \in \mathbb{B}^{\mathcal{Q}}$, $\begin{aligned}[t][A \leftrightarrow B]^\rho &= ([A]^\rho \times [B]^\rho) + (\overline{[A]^\rho} \times \overline{[B]^\rho}) \text{ par définition} \\ &= ([A]^\rho + \overline{[A]^\rho}) \times ([A]^\rho + \overline{[B]^\rho}) \times ([B]^\rho + \overline{[A]^\rho}) \times ([B]^\rho + \overline{[B]^\rho}) \\ &= ([A]^\rho + \overline{[B]^\rho}) \times ([B]^\rho + \overline{[A]^\rho}) \text{ car } \forall\, X \in \mathbb{B}, X + \overline{X} = V \\ &= [B \to A]^\rho \times [A \to B]^\rho \text{ par commutativité de } + \text{ puis par définition} \\ &= [A \to B]^\rho \times [B \to A]^\rho \text{ par commutativité de } \times \\ &= [(A \to B) \wedge (B \to A)]^\rho.\end{aligned}$

D'où $A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$.

3.3 Conséquence logique

Définition (*conséquence logique*) :

Soit $(A, B) \in \mathbb{F}_p(\mathcal{Q})^2$. On dit que B est conséquence logique de A, ce que l'on note $A \vDash B$, ssi tout environnement propositionnel satisfaisant A satisfait aussi B, autrement dit

$$\{\rho \in \mathbb{B}^{\mathcal{Q}} \mid [A]^\rho = \mathsf{V}\} \subseteq \{\rho \in \mathbb{B}^{\mathcal{Q}} \mid [B]^\rho = \mathsf{V}\}$$

Remarque : L'équivalence logique correspond au cas d'égalité de l'inclusion précédente.

Propriété : La relation binaire $\vDash$ est réflexive et transitive.

Exercice : Prouver cette propriété.

Définition (*généralisation pour un ensemble de formules*) :

Soit $X \subseteq \mathbb{F}_p(\mathcal{Q})$ et $B \in \mathbb{F}_p(\mathcal{Q})$. On note $X \vDash B$ ssi tout environnement propositionnel satisfaisant toutes les formules de X satisfait aussi B, autrement dit :

$$\{\rho \in \mathbb{B}^{\mathcal{Q}} \mid \forall\, A \in X, [A]^\rho = \mathsf{V}\} \subseteq \{\rho \in \mathbb{B}^{\mathcal{Q}} \mid [B]^\rho = \mathsf{V}\}$$

Remarque : Il faut distinguer ceci de : "B est conséquence logique de la conjonction des formules de X" (c'est-à-dire $(\bigwedge_{A \in X} A) \vDash B$ $(*)$). En effet, X pouvant être de cardinal infini, la définion précédente caractérise une propriété plus forte que $(*)$, qui ne coïncide avec elle que si l'ensemble X est fini.

Exemples : Soit $(A, B) \in \mathbb{F}_p(\mathcal{Q})^2$. Montrer que :
- ***a)*** $\{(A \to B), A\} \vDash B$
- ***b)*** $\{(A \to B), \neg B\} \vDash \neg A$.

3.4 Reformulations avec des équivalences

Propriétés :

Soit $(A, B) \in \mathbb{F}_p(\mathcal{Q})^2$. Alors :
- ***i.*** A est une tautologie ssi $A \equiv \top$.
- ***ii.*** A est une antilogie ssi $A \equiv \bot$.
- ***iii.*** A est une tautologie ssi $\neg A$ est une antilogie.
- ***iv.*** $A \equiv B$ ssi $A \leftrightarrow B \equiv \top$ (*i.e.* $A \leftrightarrow B$ est une tautologie.)
- ***v.*** $A \vDash B$ ssi $A \to B \equiv \top$ (*i.e.* $A \to B$ est une tautologie).

L'espace des formules logiques quotienté par équivalence, $\mathbb{F}_p(\mathcal{Q})/\equiv$, est en bijection avec $\mathcal{F}(\mathbb{B}^{\mathcal{Q}}, \mathbb{B})$. En effet, une classe d'équivalence selon $\equiv$ est caractérisée par la fonction booléenne à laquelle sont associés tous ses éléments.

Cela justifie que $[\![\cdot]\!]^A$ soit parfois appelée la *représentation* de A.

On peut alors être amené à se demander quel représentant l'on préfèrera pour une classe donnée, et s'il est envisageable de choisir une forme canonique pour représenter une classe de formules équivalentes.

4 Mise sous forme normale

4.1 Mise sous FND à partir d'une table de vérité

4.1.1 Sur un exemple

On cherche à mettre sous FND la formule $A = (a \vee b) \to (c \wedge a)$, dont voici la table de vérité :

On en déduit, en prenant la disjonction des clauses conjonctives formées par les lignes du tableau "satisfaisant" A (c-à-d telles que la dernière case contienne V) :

$$A \equiv (a \wedge b \wedge c) \vee \underbrace{(a \wedge \neg b \wedge c) \vee (\neg a \wedge \neg b \wedge c)}_{\equiv(\neg b \wedge c)} \vee (\neg a \wedge \neg b \wedge \neg c)$$

ce que l'on peut encore simplifier en $(a \wedge b \wedge c) \vee (\neg b \wedge c) \vee (\neg a \wedge \neg b \wedge \neg c)$.

4.1.2 Table de vérité d'une formule

On étend ici la définition de la table de vérité aux formules pour une numérotation des variables fixée, $\mathcal{Q} = \{q_1, q_2, ..., q_n\}$ où $n = \mathrm{Card}(\mathcal{Q})$.

Définition (*table de vérité d'une formule*) :

Une table de vérité d'une formule $A \in \mathbb{F}_p(\mathcal{Q})$ est une table de vérité T de la fonction booléenne associée $[\![\cdot]\!]^A$, de sorte qu'on a :

- $\left\{ (T_{i,j})_{j \in [1..n]} \,\middle|\, i \in [1..2^n] \right\} = \mathbb{B}^{\mathcal{Q}}$
- $\forall\, i \in [1..2^n],\ T_{i,n+1} = [\![\rho^i]\!]^A$

où pour tout $i \in [1..2^n]$, ρ^i est défini par $\forall\, j \in [1..n],\ \rho^i(q_j) = T_{i,j}$.

Remarque : En calculant une FND à partir du tableau T, on calcule donc bien quelque chose qui ne dépend pas exactement de A mais de sa classe...

4.1.3 Calculer une FND à partir d'une table de vérité

On a vu sur l'exemple au-dessus une méthode "intuitive" permettant de mettre une formule sous FND à partir de sa table de vérité. Il s'agit à présent de formaliser ce processus et de montrer qu'il "fonctionne" effectivement.

Soit $A \in \mathbb{F}_p(\mathcal{Q})$ où $\mathcal{Q} = \{q_1, q_2, ..., q_n\}$. Soit T une table de vérité de A suivant cette numérotation de $\mathcal{Q}$ et $(\rho^i)_{i \in [1..2^n]}$ la famille des environnements propositionnels définis comme précédemment à partir des valeurs des cases de T.

Notation :

Pour tout $i \in [1..2^n]$ et $j \in [1..n]$, on note $\ell_{i,j}$ comme étant :
- le littéral q_j si $T_{i,j} = \mathsf{V}$
- le littéral $\neg q_j$ si $T_{i,j} = \mathsf{F}$.

Lemme : $\forall\, i \in [1..2^n]$, $\forall\, j \in [1..n]$, $[\ell_{i,j}]^{\rho^i} = \mathsf{V}$.

Preuve :
Soit $i \in [1..2^n]$ et $j \in [1..n]$.

- Si $T_{i,j} = \mathsf{V}$, alors $\ell_{i,j} = q_j$ donc $[\ell_{i,j}]^{\rho^i} = [q_j]^{\rho^i} = \rho^i(q_j)$ par définition de l'interprétation d'une variable. Or par définition de ρ^i, $\rho^i(q_j) = T_{i,j}$ donc $[\ell_{i,j}]^{\rho^i} = \mathsf{V}$.
- Si $T_{i,j} = \mathsf{F}$, alors $\ell_{i,j} = \neg q_j$ donc $[\ell_{i,j}]^{\rho^i} = \overline{\rho^i(q_j)}$ par définition de l'interprétation d'une négation puis d'une variable. Or par définition de ρ^i, $\rho^i(q_j) = T_{i,j}$ donc $[\ell_{i,j}]^{\rho^i} = \mathsf{V}$.

Notation :

Ensuite, on pose, pour tout $i \in [1..2^n]$, $L^i = \bigwedge_{j=1}^{n} \ell_{i,j}$.

Lemme : On a : ***i.*** $\forall\, i \in [1..2^n]$, $[L^i]^{\rho^i} = \mathsf{V}$
ii. $\forall\, (i,k) \in [1..2^n]^2$, $i \neq k$, $[L^i]^{\rho^k} = \mathsf{F}$.

Preuve :
i. Soit $i \in [1..2^n]$. Par définition de l'interprétation d'une conjonction,

$$[L^i]^{\rho^i} = \prod_{j=1}^{n} [\ell_{i,j}]^{\rho^i} = \prod_{j=1}^{n} \mathsf{V} = \mathsf{V}$$

d'après le lemme précédent appliqué aux couples $((i,j))_{j\in[1..n]}$. D'où $[L^i]^{\rho^i} = \mathsf{V}$.

ii. Soit $k \in [1..2^n]$ tel que $k \neq i$. Puisque les lignes de T restreintes à leur n premières colonnes sont deux à deux distinctes (il y en a 2^n et elles couvrent $\mathbb{B}^{\mathcal{Q}}$ qui est de cardinal 2^n), il existe $j_0 \in [1..n]$ tel que $T_{i,j_0} \neq T_{k,j_0}$.

- Si $T_{i,j_0} = \mathsf{V}$, alors $\ell_{i,j_0} = q_{j_0}$ et $T_{k,j_0} = \mathsf{F}$. Par définition de l'interprétation d'une variable, $[\ell_{i,j_0}]^{\rho^k} = \rho^k(q_{j_0})$, or par définition de ρ^k, $\rho^k(q_{j_0}) = T_{k,j_0} = \mathsf{F}$ donc $[\ell_{i,j_0}]^{\rho^k} = \mathsf{F}$.
- Si $T_{i,j_0} = \mathsf{F}$, alors $\ell_{i,j_0} = \neg q_{j_0}$ et $T_{k,j_0} = \mathsf{V}$. Par définition de l'interprétation de la négation d'une variable, $[\ell_{i,j_0}]^{\rho^k} = \overline{\rho^k(q_{j_0})}$ qui vaut, par définition de ρ^k, $\rho^k(q_{j_0}) = \overline{T_{k,j_0}} = \overline{\mathsf{V}} = \mathsf{F}$.

Dans les deux cas, le terme d'indice j_0 du produit qu'est l'interprétation de L^i par ρ^k vaut F.

Ainsi, F étant absorbant pour $\times$, on en déduit que $[L^i]^{\rho^k} = \mathsf{F}$.

Notation :

Finalement, on pose $D = \bigvee_{\substack{i\in[1..2^n] \\ T_{i,n+1}=\mathsf{V}}} L^i$.

Propriété :

On a alors $D \equiv A$.

Preuve :
Soit $\rho \in \mathbb{B}^{\mathcal{Q}}$. On note $I = \{i \in [1..2^n] \,|\, T_{i,n+1} = \mathsf{V}\}$, ainsi $D = \bigwedge_{i \in I} L^i$.
De plus, par définition de l'interprétation d'une conjonction, $[D]^\rho = \sum_{i \in I} [L^i]^\rho$.
Puisque les lignes de de T restreintes à leurs n premières colonnes couvrent $\mathbb{B}^{\mathcal{Q}}$, il existe $i_0 \in [1..2^n]$ tel que $\rho = \rho^{i_0}$.

· Si $[A]^\rho = \mathsf{V}$, on a $\mathsf{V} = [\![\rho]\!]^A = [\![\rho^{i_0}]\!]^A = T_{i_0,n+1}$ donc $i_0 \in I$.
Ainsi, le terme $[L^{i_0}]^\rho$ apparaît dans la somme qu'est $[D]^\rho$, or par le lemme précédent, $[L^{i_0}]^\rho = [L^{i_0}]^{\rho^{i_0}} = \mathsf{V}$ et V étant absorbant pour $+$, on en déduit que $[D]^\rho = \mathsf{V}$, soit $[D]^\rho = [A]^\rho$.

· Si au contraire $[A]^\rho = \mathsf{F}$, alors $T_{i_0,n+1} = \mathsf{F}$ donc $i_0 \notin I$.
Autrement dit, $\forall\, i \in I$, $i \neq i_0$ ce qui donne d'après le lemme précédent $[L^i]^\rho = [L^i]^{\rho^i} = \mathsf{F}$. Une somme de F valant F, on en déduit que $[D]^\rho = \mathsf{F}$, soit $[D]^\rho = [A]^\rho$.

4.2 Mise sous FNC à partir d'une table de vérité

4.2.1 Sur le même exemple

On veut mettre sous FNC la formule vue au début de la section précédente, $A = (a \vee b) \to (c \wedge a)$. Pour cela, reprenons sa table de vérité :

En prenant cette fois-ci la conjonction des négations respectives des clauses conjonctives formées par les lignes satisfaisant $\neg A$ (*c-à-d* telles que la dernière case contienne F), puis en distribuant la négation sur chaque clause, on obtient :

$$A \equiv (\neg a \vee \neg b \vee c) \wedge (\neg a \vee b \vee c) \wedge (a \vee \neg b \vee \neg c) \wedge (a \vee \neg b \vee c)$$

4.2.2 Calculer une FNC à partir d'une table de vérité

Soit $A \in \mathbb{F}_p(\mathcal{Q})$ où $\mathcal{Q} = \{q_1, q_2, ..., q_n\}$ et T une table de vérité de A suivant cette numérotation de $\mathcal{Q}$; on définit comme prédécemment la famille d'environnements $(\rho^i)_{i \in [1..2^n]}$ à partir de T.

Notations :

Pour tout $i \in [1..2^n]$ et $j \in [1..n]$, on note $r_{i,j}$ comme étant :
- le littéral $\neg q_j$ si $T_{i,j} = \mathsf{V}$
- le littéral q_j si $T_{i,j} = \mathsf{F}$.

Ensuite, on pose, pour tout $i \in [1..2^n]$, $R^i = \bigvee_{j=1}^{n} r_{i,j}$.

Enfin, on pose $C = \bigwedge_{\substack{i \in [1..2^n] \\ T_{i,n+1}}} R^i$.

Propriétés :

On a successivement : ***i.*** $\forall\, i \in [1..2^n]$, $\forall\, j \in [1..n]$, $[r_{i,j}]^{\rho^i} = \mathsf{F}$
ii. $\forall\, i \in [1..2^n]$, $[R^i]^{\rho^i} = \mathsf{F}$ et $\forall\, k \in [1..2^n]$, $k \neq i$, $[R^i]^{\rho^k} = \mathsf{V}$
iii. $C \equiv A$

Preuve :
On procède sur le même principe que pour la mise sous FND.

4.3 Bilan sur la FNC/FND

À retenir :

- Certaines formules sont à la fois sous FNC et FND.
- Il y a existence de la FNC/FND équivalente à une formule (on vient de le montrer !). Par contre, il n'y a pas unicité de la FNC ni de la FND équivalente à une formule.
- Il peut y avoir une explosion de la taille lors de la conversion (*cf.* exemple ci-dessous).

Exemple : $\bigwedge_{i=1}^{n}(a_i^1 \vee a_i^2) = \bigvee_{(j_1,j_2,\ldots,j_n)\in\{1,2\}^n} \bigwedge_{k=1}^{n} a_k^{j_k}$ (*cf.* distributivité généralisée).

Dans cet exemple, la transformation de FNC à FND nous fait passer d'une conjonction de n 2-clauses disjonctives à une disjonction de 2^n n-clauses conjonctives.

Exercice : Voici quelques simplifications utiles ; démontrer chaque équivalence logique.

Exercice : Mettre sous FNC et FND les formules suivantes.
- ***a)*** $U = (x \wedge y) \vee (z \wedge \neg z \wedge q) \vee (\neg x \wedge z)$
- ***b)*** $V = (x \wedge q) \rightarrow ((y \vee z) \wedge q)$
- ***c)*** $W = (x \wedge y) \leftrightarrow (\neg x \wedge z)$.

5 Le problème SAT

Dans cette section, on s'intéresse à la satisfiabilité des formules (sur un ensemble de variables fini, i.e. tel que $\mathrm{Card}(Q) \in \mathbb{N}$*).*

En effet, comme déjà évoqué en introduction, une formule propositionnelle peut modéliser un problème concret dont une solution serait donnée par une environnement satisfaisant la formule.

Exemple : Pour le Sudoku, la valeur de vérité d'une variable $p_{i,j,k}$ peut par exemple indiquer si la case (i, j) contient la valeur k. L'environnement complet décrit une solution, et s'il satisfait la formule, alors c'est une solution valide.

Mais on s'intéresse déjà au problème de décision (plus simple) de savoir si une formule est satisfiable, sans demander par quel environnement : c'est le problème SAT, que l'on formalise dans la section suivante.

Remarque : Puisque la satisfiabilité d'une formule ne dépend que de sa classe, on peut résoudre le problème sur une formule ayant une forme particulière, mais il faut alors faire attention au coût de transformation.

5.1 Définitions

On donne ci-dessous la formalisation du problème SAT ainsi que quelques unes de ses variantes :

SAT || **entrée :** $A \in \mathbb{F}_p(\mathcal{Q})$
sortie : oui si A est satisfiable, non sinon

FND-SAT || **entrée :** $A \in \mathbb{F}_p(\mathcal{Q})$ sous FND
sortie : oui si A est satisfiable, non sinon

FNC-SAT || **entrée :** $A \in \mathbb{F}_p(\mathcal{Q})$ sous FND
sortie : oui si A est satisfiable, non sinon

3-SAT || **entrée :** $A \in \mathbb{F}_p(\mathcal{Q})$ sous FNC avec des clauses de 3 littéraux uniquement
sortie : oui si A est satisfiable, non sinon

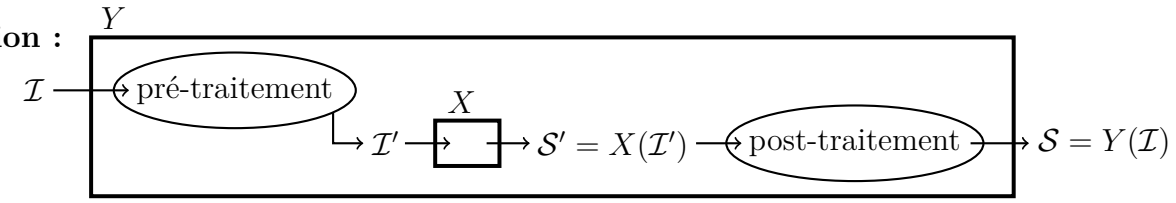
2-SAT || **entrée :** $A \in \mathbb{F}_p(\mathcal{Q})$ sous FNC avec des clauses de 3 littéraux seulement
sortie : oui si A est satisfiable, non sinon

5.2 Réductions

Rappel (*réduction entre problèmes*) :

On dit qu'un problème X est plus dur qu'un autre problème Y, ou que Y se réduit à X, si savoir résoudre X permet de résoudre facilement Y après un éventuel pré-traitement et/ou post-traitement.

Illustration :



Définition (*réduction polynomiale entre problèmes de décisions*) :

Soient X et Y deux problèmes de décision. On note X^+ (resp. X^-) les instances positives (resp. négatives) de X, c'est-à-dire celles pour lesquelles la solution est "Vrai" (resp. "Faux"). On définit de même Y^+ et Y^-.

Alors, on dit que Y se réduit à X en temps polynomial s'il existe une transformation φ calculable en temps polynomial, qui transforme toute instance de Y en une instance de X de sorte que :

$$\forall \mathcal{I},\ \varphi(\mathcal{I}) \in X^+ \iff \mathcal{I} \in Y^+ \text{ et } \varphi(\mathcal{I}) \in X^- \iff \mathcal{I} \in Y^-$$

Illustration :

Compte tenu de ces définitions, on a les réductions suivantes :

- SAT est plus dur que FND-SAT et FNC-SAT, en effet :
  - FNC-SAT se réduit à SAT pour $\varphi = \text{Id}$
  - FND-SAT se réduit à SAT pour $\varphi = \text{Id}$
- FNC-SAT est plus dur que 2-SAT et 3-SAT, en effet :
  - 2-SAT se réduit à FNC-SAT pour $\varphi = \text{Id}$
  - 3-SAT se réduit à FNC-SAT pour $\varphi = \text{Id}$

· Inversement, FNC-SAT se réduit à 3-SAT. En effet, on peut transformer toute clause de taille ≤ 2 d'une FNC en clause de taille 3 suivant le modèle des exemples suivants :

$$a \equiv (a \vee a \vee a) \qquad (a \vee b) \equiv (a \vee a \vee b)$$

De même, on transforme chaque clause de taille > 3 en conjonction de clauses de taille 3 en ajoutant successivement des nouvelles variables comme suit :

$$(a \vee b \vee c \vee d) \text{ devient } (a \vee b \vee z) \wedge (\neg z \vee c \vee d)$$
$$(a \vee b \vee c \vee d \vee e) \text{ devient } (a \vee b \vee z_1) \wedge (\neg z_1 \vee c \vee z_2) \wedge (\neg z_2 \vee d \vee e)$$

On justifie ce processus dans la proposition suivante.

Proposition (*transformations préservant la satisfiabilité*) :

Soit $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee ... \vee \ell_k$ une clause de $\mathbb{F}_p(\mathcal{Q})$ de taille $k \geq 3$.
Soit z une nouvelle variable (*i.e.* $z \notin \mathcal{Q}$). On pose :
- $C_1 = (\ell_1 \vee \ell_2 \vee z)$
- $C_2 = (\neg z \vee \ell_3 \vee ... \vee \ell_k)$.

Alors, on a les deux propriétés suivantes :

i. $\forall\, \rho \in \mathbb{B}^{\mathcal{Q}}, \Big([C]^\rho = \mathsf{V} \Longrightarrow (\exists\, \widetilde{\rho} \in \mathbb{B}^{\mathcal{Q}\cup\{z\}}, \widetilde{\rho}|_{\mathcal{Q}} \text{ et } [C_1 \wedge C_2]^{\widetilde{\rho}} = \mathsf{V})\Big)$

ii. $\forall\, \widetilde{\rho} \in \mathbb{B}^{\mathcal{Q}\cup\{z\}}, ([C_1 \wedge C_2]^{\widetilde{\rho}} = \mathsf{V} \Longrightarrow [C]^{\widetilde{\rho}} = \mathsf{V})$

Preuve :
i. Soit $\rho \in \mathbb{B}^{\mathcal{Q}}$ tel que $[C]^\rho = \mathsf{V}$. Alors, comme $C = \bigvee_{i=1}^k \ell_i$ c'est-à-dire $[C]^\rho = \sum_{i=1}^k [\ell_i]^\rho$, il existe $i_0 \in [1..k]$ tel que $[\ell_i]^\rho = \mathsf{V}$.

· Si $[\ell_1]^\rho = \mathsf{V}$ ou $[\ell_2]^\rho = \mathsf{V}$, on pose :

$$\widetilde{\rho} = \begin{pmatrix} \mathcal{Q} \cup \{z\} & \to & \mathbb{B} \\ q \in \mathcal{Q} & \mapsto & \rho(q) \\ z & \mapsto & \mathsf{F} \end{pmatrix}$$

On a bien $\widetilde{\rho}|_{\mathcal{Q}} = \rho$ par cette définition. De plus :

$$[C_1 \wedge C_2]^{\widetilde{\rho}} = [C_1]^{\widetilde{\rho}} \times [C_2]^{\widetilde{\rho}} = \underbrace{[\ell_1 \vee \ell_2 \vee z]^{\widetilde{\rho}}}_{=\mathsf{V} \text{ par hypothèse}} \times \Big[\bigvee_{i=3}^k \ell_i \vee \neg z\Big]^{\widetilde{\rho}} = \mathsf{V} \times \underbrace{\Big(\sum_{i=3}^k [\ell_i]^{\widetilde{\rho}} + [\neg z]^{\widetilde{\rho}}\Big)}_{=\mathsf{V} \text{ car } [\neg z]^{\tilde{\rho}} = \overline{[z]^{\tilde{\rho}}} = \mathsf{V}} = \mathsf{V}$$

· Si $[\ell_1]^\rho = \mathsf{F}$ et $[\ell_2]^\rho = \mathsf{F}$, alors $i_0 \geq 3$. On pose donc :

$$\widetilde{\rho} = \begin{pmatrix} \mathcal{Q} \cup \{z\} & \to & \mathbb{B} \\ q \in \mathcal{Q} & \mapsto & \rho(q) \\ z & \mapsto & \mathsf{V} \end{pmatrix}$$

On a toujours facilement $\widetilde{\rho}|_{\mathcal{Q}} = \rho$, et on a de plus :

$$[C_1 \wedge C_2]^{\widetilde{\rho}} = [C_1]^{\widetilde{\rho}} \times [C_2]^{\widetilde{\rho}} = \underbrace{[\ell_1 \vee \ell_2 \vee \ell_3]^{\widetilde{\rho}}}_{=\mathsf{V} \text{ par définition de } \tilde{\rho}} \times \Big[\bigvee_{i=3}^k \ell_i \vee \neg z\Big]^{\widetilde{\rho}} = \mathsf{V} \times \underbrace{\Big(\sum_{\substack{i=3 \\ i \neq i_0}}^k [\ell_i]^{\widetilde{\rho}} + [\ell_{i_0}]^{\widetilde{\rho}} + [\neg z]^{\widetilde{\rho}}\Big)}_{=\mathsf{V} \text{ car } [\ell_{i_0}]^{\tilde{\rho}} = \mathsf{V} \text{ puisque } i_0 \in [3..k]} = \mathsf{V}$$

Dans les deux cas, on trouve toujours une fonction qui convient.

Remarque : On a bien des transformations en temps polynomial qui préservent la satisfiabilité. Par contre, elles ne conservent pas nécessairement les classes d'équivalence pour $\equiv$ (c'est entre autres le cas les deux derniers exemples donnés plus haut : considérer l'environnement envoyant toutes les variables sur F).

Remarque : La même stratégie ne permet pas de réduire FNC-SAT à 2-SAT. En fait, FNC-SAT ne se réduit pas à 2-SAT. En effet, on verra dans l'année de Spé que :

- 2-SAT se résout en temps polynomial (*cf.* chapitre "Graphes")
- 3-SAT est NP-complet (*cf.* théorème de Cook en 2$^{\text{de}}$ année)
- FNC-SAT est NP-complet (*idem*)

Donc, à moins que P = NP, ces deux problèmes ne sont pas équivalents.

Remarque : De même, FNC-SAT ne se réduit pas à FND-SAT et ceci pour deux raisons :

- la transformation d'une forme à l'autre se fait en temps exponentiel (vu dans un exemple plus haut) et peut aussi augmenter la taille de l'entrée de manière exponentielle
- on verra dans la suite que FND-SAT se résout en temps polynomial

Encore une fois, à moins que P = NP, ces deux problèmes ne sont pas équivalents.

5.3 Modéliser des FND ou FNC au regard de la satisfiabilité

5.3.1 Modéliser une FND

Rappelons qu'une FND est une disjonction de conjonctions. Compte tenu de ceci, d'un point de vue sémantique, on peut dire que :

- L'ordre des littéraux au sein des conjonctions n'est pas important, et leur multiplicité non plus, puisqu'une conjonction est satisfaite par un environnement propositionnel ssi ce dernier satisfait l'ensemble de ses termes.

  $\rightarrow$ L'objet mathématique adapté pour modéliser les conjonctions de littéraux un ensemble de littéraux ou bien un couple d'ensembles de variables : d'une part, celles apparaissant dans les littéraux positifs, d'autre part, celles apparaissant dans les littéraux négatifs.

- De même, l'ordre et la multiplicité des conjonctions au sein de la disjonction ne sont pas importantes non plus, comme une disjonction est satisfaite par un environnement propositionnel ssi il satisfait l'un de ses termes.

  $\rightarrow$ L'objet mathématique adapté pour modéliser une FND est donc un ensemble de conjonctions.

- Pour ce qui est de la structure de données adaptée pour la modélisation d'une FND, une liste de listes de littéraux convient.

Exemples :
a) Formule initiale : $(a \wedge b \wedge \neg c) \vee (\neg b \wedge \neg c) \vee (a \wedge c)$.
Sous forme d'ensemble d'ensembles de littéraux : $\{\{a, b, \neg c\}, \{\neg b, \neg c\}, \{a, c\}\}$.
Sous forme d'ensemble de couples : $\Big\{\big(\{a, b\}, \{c\}\big), \big(\emptyset, \{b, c\}\big), \big(\{a, c\}, \emptyset\big)\Big\}$.

b) Formule initiale : $(a \wedge b \wedge \neg a) \vee (a \wedge c \wedge \neg a) \vee (\neg b \wedge \neg c \wedge \neg b)$.
Sous forme d'ensemble d'ensembles de littéraux : $\{\{a, b, \neg a\}, \{a, \neg a, c\}, \{\neg b, \neg c\}\}$.
Sous forme d'ensemble de couples : $\Big\{\big(\{a, b\}, \{a\}\big), \big(\{a, c\}, \{a\}\big), \big(\emptyset, \{b, c\}\big)\Big\}$.

5.4 FND-SAT : un problème facile

Voici un algorithme que l'on propose pour la résolution de FND-SAT sur $\mathcal{Q} = \{q_1, q_2, ..., q_N\}$.

Algorithme – FND-SAT

entrée : $\left((\ell_{i,j})_{i\in[1..n_j]}\right)_{j\in[1..n]}$ une famille de familles de littéraux de $\mathcal{Q}$

sortie : Vrai si $A = \bigvee_{j=1}^{n} \bigwedge_{i=1}^{n_j} \ell_{i,j}$ est satisfiable, Faux sinon

Pour j allant de 1 à n
  Créer T un tableau indicé par $[1..N]$, initialisé à -1
  Esaayer :
    Si $\ell_{i,j} = q_k$ alors
      si $T[k] = -1$ alors $T[k] \leftarrow 1$
      sinon si $T[k] = 0$ alors déclencher l'exception "conj. non sat."
    Si $\ell_{i,j} = \neg q_k$ alors
      si $T[k] = -1$ alors $T[k] \leftarrow 0$
    sinon si $T[k] = 1$ alors déclencher l'exception "conj. non sat." Retourner Vrai
  Rattraper "conj. non sat"
Retourner Faux

5.5 Puissance d'encodage de SAT

5.5.1 Modélisation – exemple du solitaire

Dans cette sous-section, on s'intéresse la modélisation d'un exemple de problème concret à l'aide de formules logique. Plus précisément, on veut pouvoir savoir, à partir d'un état initial quelconque du jeu du solitaire, s'il est possible ou non de gagner la partie.

On rappelle d'abord brièvement le principe du jeu : initialement, on a un damier carré de $m \times m$ cases dont chacune contient soit une pierre bleue (b), soit une pierre rouge (r).

Le but du jeu est d'enlever des pierres de manière à avoir, au final, un damier tel que :
- sur chaque colonne, toutes les pierres sont de même couleur
- sur chaque ligne, il y a au moins une pierre.

Formalisons le problème de décision consistant à savoir si une partie donnée peut être gagnée :

JEU

entrée : $m \in \mathbb{N}^*$ la taille du damier
$B \subseteq [1..m]^2$ l'ensemble des cases initialement bleues
$R \subseteq [1..m]^2 \backslash B$ celui des cases initialement rouges

sortie : Vrai s'il existe $B' \subseteq B^{(1)}$ et $R' \subseteq R^{(2)}$ tels que :
- $\forall\, i \in [1..m]$, $\{i\} \times [1..m] \cap (B' \cup R') \neq \emptyset^{(3)}$
- $\forall\, j \in [1..m], [1..m] \times \{j\} \cap B' = \emptyset$ ou $[1..m] \times \{j\} \cap R' = \emptyset^{(4)}$

Réduisons maintenant ce problème à 3-SAT. Soit (m, B, R) une instance de JEU.
On cherche à construire, en temps polynomial, une FNC A telle que :

$$A \in \text{3-SAT}^+ \text{ ssi } (m, B, R) \in \text{JEU}^+$$

c'est-à-dire telle que A est satisfiable ssi il existe $B' \subseteq B$ et $R' \subseteq R$ vérifiant les conditions précédentes.

Pour chaque $(i, j) \in [1..m]^2$, on introduit deux variables :
- $b'_{i,j}$, modélisant $(i, j) \in B'$
- $r'_{i,j}$, modélisant $(i, j) \in R'$.

On modélise ensuite les quatre contraintes précédentes :

$$(1) : A^1_{m,B,R} := \bigwedge_{(i,j)\in[1..m]^2\setminus B} \neg b'_{i,j}$$

$$(2) : A^2_{m,B,R} := \bigwedge_{(i,j)\in[1..m]^2\setminus R} \neg r'_{i,j}$$

$$(3) : A^3_{m,B,R} := \bigwedge_{i=1}^{m} \bigvee_{j=1}^{m} (b'_{i,j} \vee r'_{i,j})$$

$$(4) : A^4_{m,B,R} := \bigwedge_{j=1}^{m} \bigwedge_{(i,i')\in[1..m]^2} (\neg b'_{i,j}) \vee (\neg r'_{i',j}) \equiv \bigwedge_{j=1}^{m} \left(\left(\bigwedge_{i=1}^{m} \neg b'_{i,j} \right) \vee \left(\bigwedge_{i=1}^{m} \neg r'_{i,j} \right) \right)$$

On note $A_{m,B,R}$ la conjonction des $(A^k_{m,B,R})_{k\in[1..4]}$, qui est bien une FNC.
On note par ailleurs $\mathcal{Q} = \{b'_{i,j}, r'_{i,j} \,|\, (i,j) \in [1..m]^2\}$.

Propriété :

> $A_{m,B,R}$ est satisfiable ssi le jeu (m, B, R) est gagnable.

Preuve :
($\Rightarrow$) : Supposons que $A_{m,B,R}$ est satisfiable. Il existe $\rho \in \mathbb{B}^{\mathcal{Q}}$ tel que $[A]^\rho = \mathsf{V}$.
On note alors : · $B' = \{(i,j) \in [1..m]^2 \,|\, \rho(b'_{i,j}) = \mathsf{V}\}$
· $R' = \{(i,j) \in [1..m]^2 \,|\, \rho(r'_{i,j}) = \mathsf{V}\}$

Par définition de l'interprétation d'une conjonction, on déduit de $[A_{m,B,R}]^\rho = \mathsf{V}$ que pour tout $k \in [1..4]$, $[A^k_{m,B,R}]^\rho = \mathsf{V}$.

· En particulier, $[A^1_{m,B,R}]^\rho = \mathsf{V}$ donc :

$$\forall\, (i,j) \in [1..m]^2\setminus B,\ [\neg b'_{i,j}]^\rho = \mathsf{V} \text{ soit } [b'_{i,j}]^\rho = \mathsf{F} \text{ donc } (i,j) \notin B'$$

Ainsi, $[1..m]^2\setminus B = B^{\complement} \subseteq (B')^{\complement}$ et donc $B' \subseteq B$. De même, puisque $[A^2_{m,B,R}]^\rho = \mathsf{V}$, $R' \subseteq R$.

· Soit ensuite $j_0 \in [1..m]$.
Supposons par l'absurde qu'il existe $(i_1, i_2) \in [1..m]^2$ tel que $(i_1, j_0) \in B'$ et $(i_2, j_0) \in R'$. Alors, $[b'_{i_1,j_0}]^\rho = \mathsf{V}$ et $[r'_{i_2,j_0}]^\rho = \mathsf{V}$ c'est-à-dire $[\neg b'_{i_1,j_0}]^\rho = \mathsf{F}$ et $[\neg r'_{i_2,j_0}]^\rho = \mathsf{F}$, donc $[\neg b'_{i_1,j_0} \vee \neg r'_{i_2,j_0}]^= \mathsf{F}$.
Dans ce cas, on a :

$$[A^4_{m,B,R}]^\rho = \left[\bigwedge_{j\in[1..m]} \bigwedge_{(i,i')\in[1..m]^2} (\neg b'_{i,j} \vee \neg r'_{i',j}) \right]^\rho = \mathsf{F}$$

Or, $[A^3_{m,B,R}]^\rho = \mathsf{V}$ puisque $[A_{m,B,R}]^\rho = \mathsf{V}$ donc on a une contradiction.

Ainsi, il n'existe pas deux lignes de couleurs différentes sur la colonne j_0. Ceci valant pour tout $j_0 \in [1..m]$, on en déduit que (B', R') satisfait la condition sur les colonnes.

· Enfin, soit $i_0 \in [1..m]$. Supposons par l'absurde que $\forall\, j \in [1..n]$, $(i_0, j) \notin B' \cup R'$.
Alors, pour tout $j \in [1..m]$, on a : · $[b'_{i_0,j}]^\rho = \rho(b'_{i_0,j}) = \mathsf{F}$ car $(i_0, j) \notin B'$
· $[r'_{i_0,j}]^\rho = \rho(r'_{i_0,j}) = \mathsf{F}$ car $(i_0, j) \notin R'$,
ce qui donne $[b'_{i_0,j} \vee r'_{i_0,j}]^\rho = \mathsf{F}$ et donc $\left[\bigvee_{j=1}^{m} (b'_{i_0,j} \vee r'_{i_0,j})\right]^\rho = \mathsf{F}$. Par conséquent :

$$[A^3_{m,B,R}]^\rho = \left[\bigwedge_{i=1}^{m} \bigvee_{j=1}^{m} (b'_{i,j} \vee r'_{i,j}) \right]^\rho = \mathsf{F}$$

Or c'est impossible puisque comme $[A_{m,B,R}]^\rho = \mathsf{V}$, on a nécessairement $[A^3_{m,B,R}]^\rho = \mathsf{V}$.

Ainsi sur la ligne i_0, il existe $j_0 \in [1..m]$ tel que $(i_0, j_0) \in B'$ ou $(i_0, j_0) \in R'$: il y a bien au

moins une pierre sur cette ligne. Comme c'est vrai quel que soit $i_0 \in [1..m]$, (B', R') vérifie aussi la propriété sur les lignes.

Conclusion : On a bien $(m, B, R) \in \mathsf{JEU}^+$.

($\Leftarrow$) : Réciproquement, montrons que si $(m, B, R) \in \mathsf{JEU}^+$, alors $A_{m,B,R} \in \mathsf{FNC\text{-}SAT}^+$. Supposons que (m, B, R) est gagnable : il existe donc (B', R') une solution de ce jeu. On pose alors :

$$\rho = \begin{pmatrix} \mathcal{Q} \to \mathbb{B} \\ b'_{i,j} \mapsto \begin{cases} \mathsf{V} \text{ si } (i,j) \in B' \\ \mathsf{F} \text{ sinon} \end{cases} \\ r'_{i,j} \mapsto \begin{cases} \mathsf{V} \text{ si } (i,j) \in R' \\ \mathsf{F} \text{ sinon} \end{cases} \end{pmatrix}$$

Il faut montrer que $\forall\, k \in [1..4]$, $[A^k_{m,B,R}]^\rho = \mathsf{V}$.

Exercice : Terminer la preuve précédente.

Exercice : (bonus) Inversement, réduire $\mathsf{3\text{-}SAT}$ à JEU.

À retenir :

La puissance d'encodage de $\mathsf{3\text{-}SAT}$.

5.6 Algorithme de Quine

5.6.1 Substitutions

Définition (*support d'une fonction de $\mathcal{Q}$ dans $\mathbb{F}_p(\mathcal{Q})$*) :

Soit $\sigma \in \mathcal{F}(\mathcal{Q}, \mathbb{F}_p(\mathcal{Q}))$. On appelle support de σ, noté $\mathrm{supp}(\sigma)$, l'ensemble des variables que σ n'envoie pas sur la formule réduite à elle-même :

$$\mathrm{supp}(\sigma) = \{q \in \mathcal{Q} \,|\, \sigma(q) \neq \mathcal{V}ar(q)\}$$

Définition (*substitution*) :

Soit $\sigma \in \mathcal{F}(\mathcal{Q}, \mathbb{F}_p(\mathcal{Q}))$. On dit que σ est une substitution si $\mathrm{Card}(\mathrm{supp}(\sigma)) \in \mathbb{N}$.
Dans ce cas, si $\mathrm{supp}(\sigma) = \{q_1, q_2, ..., q_n\}$, on note :

$$\sigma = [q_1 \mapsto \sigma(q_1), q_2 \mapsto \sigma(q_2), \ldots, q_n \mapsto \sigma(q_n)]$$

Exemple : Pour $\sigma = \begin{pmatrix} \{x,y\} \to \mathbb{F}_p(\{x,y\}) \\ x \mapsto x \\ y \mapsto \neg x \end{pmatrix}$, on a $\mathrm{supp}(\sigma) = \{y\}$ et donc $\sigma = [y \mapsto \neg x]$.

Définition (*application d'une substitution*) :

Soit $\sigma \in \mathcal{F}(\mathcal{Q}, \mathbb{F}_p(\mathcal{Q}))$ une substitution. On appelle application de la substitution σ la fonction suivante :

$$\cdot[\sigma] = \begin{pmatrix} \mathbb{F}_p(\mathcal{Q}) & \to & \mathbb{F}_p(\mathcal{Q}) \\ \top & \mapsto & \top \\ \bot & \mapsto & \bot \\ q & \mapsto & \sigma(q) \\ \neg A & \mapsto & \neg(A[\sigma]) \\ A \,\alpha\, B & \mapsto & A[\sigma] \,\alpha\, B[\sigma] \end{pmatrix}$$

où α désigne toujours un symbole quelconque de $\{\vee, \wedge, \to, \leftrightarrow\}$.

Exercice : Trois personnes a, b et c veulent passer au tableau, selon les règles suivantes :

- Si a passe, alors b ne passe pas.
- Si b passe, alors a et c passent.
- Si c passe, alors a ou b ne passe pas.

1) Modéliser ces affirmations avec des variables de la logique.
2) Peut-on affirmer que a passe forcément ? que b ne passe pas ?

Correction :

Définition (*composée de substitutions*) :

Soit σ_1 et σ_2 deux substitutions. On note $\sigma_1 \cdot \sigma_2$ et on appelle composée de σ_1 et σ_2 la fonction suivante :

$$\sigma_1 \cdot \sigma_2 = \begin{pmatrix} \mathcal{Q} & \to & \mathbb{F}_p(\mathcal{Q}) \\ q & \mapsto & \sigma_2(q)[\sigma_1] \end{pmatrix}$$

Propriété (*associativité de la composée de substitutions*) :

La composée de substitutions est associative, c'est-à-dire :

$$\forall\, (\sigma_1, \sigma_2, \sigma_3) \in \mathcal{F}(\mathcal{Q}, \mathbb{F}_p(\mathcal{Q}))^3 \text{ tel que } \forall\, i \in [1..3],\ \mathrm{Card}(\mathrm{supp}(\sigma_i)) \in \mathbb{N},$$
$$(\sigma_1 \cdot \sigma_2) \cdot \sigma_3 = \sigma_1 \cdot (\sigma_2 \cdot \sigma_3)$$

Propriétés (*caractère interne de $\cdot$, neutre, stabilité du support*) :

i. La composée de deux substitutions est encore une substitution (*i.e.* $\cdot$ est interne).
ii. $\forall\, \sigma_1, \sigma_2$ des substitutions, $\mathrm{supp}(\sigma_1 \cdot \sigma_2) \subseteq \mathrm{supp}(\sigma_1) \cup \mathrm{supp}(\sigma_2)$.
iii. L'application $id = \begin{pmatrix} \mathcal{Q} \to \mathbb{F}_p(\mathcal{Q}) \\ q \mapsto \mathcal{V}ar(q) \end{pmatrix}$ est le neutre pour la loi $\cdot$ (en effet, $\cdot[id] = \mathrm{Id}_{\mathbb{F}_p(\mathcal{Q})}$).

Remarque : Toutes les substitutions ne sont pas symétrisables pour $\cdot$: l'ensemble des substitutions muni de la composition forme donc un monoïde, mais pas un groupe.

5.6.2 Description de l'algorithme

L'algorithme de Quine permet de vérifier la satisfiabilité d'une formule $A \in \mathbb{F}_p(\mathcal{Q})$en passant par des substitutions et des simplifications, selon le principe suivant :

- on substitue à une variable x quelconque la formule $\top$

- on teste (récursivement, c'est-à-dire en continuant à effectuer des substitutions sur d'autres variables) si la formule obtenue après simplification est satisfiable, voire une tautologie
- si c'est le cas, alors tout environnement ρ satisfaisant, prolongé par $x \mapsto \mathsf{V}$, satisfera A
- sinon, on substitue à x la formule $\bot$ pour effectuer les mêmes tests récursifs
- si en faisant cela on aboutit à une formule satisfiable par un environnement ρ, alors le prolongement $x \mapsto \mathsf{F}$ donnera un environnement satisfaisant A
- dans le cas contraire, on en conclut que A ne peut être satisfiable.

Voir le fichier "`algo-quine.ml`" pour une implémentation en OCaml de cet algorithme.

5.6.3 Règles de simplification

Comme l'algorithme de Quine se base en partie sur la simplification de formules en formules équivalentes plus simples, il est important de savoir identifier quand une formule peut être simplifiée.

Voici donc des règles de simplification, valables pour n'importe quelle formule $A \in \mathbb{F}_p(\mathcal{Q})$:

$$\begin{array}{r|l} \neg\top \equiv \bot & A \to \bot \equiv \neg A \\ \neg\bot \equiv \top & A \to \top \equiv \top \\ A \wedge \top \equiv \top \wedge A \equiv A & \bot \to A \equiv \top \\ A \wedge \bot \equiv \bot \wedge A \equiv \bot & \top \to A \equiv A \\ A \vee \top \equiv \top \vee A \equiv \top & A \leftrightarrow \top \equiv \top \leftrightarrow A \equiv A \\ A \vee \bot \equiv \bot \vee A \equiv A & A \leftrightarrow \bot \equiv \bot \leftrightarrow A \equiv \neg A \\ \neg\neg A \equiv A & A \to A \equiv \top \end{array}$$

Propriétés (*autres simplifications*) :

Soit $(A, A', B, B') \in \mathbb{F}_p(\mathcal{Q})^4$.
Si $A \equiv A'$ et $B \equiv B'$, alors on a : ***i.*** $\neg A \equiv \neg A'$
ii. $A \,\alpha\, B \equiv A' \,\alpha\, B'$ pour $\alpha \in \{\wedge, \vee, \to, \leftrightarrow\}$

Exercice : Vérifier la satisfiabilité des formules suivantes :
- ***a)*** $A = \Big((a \to b) \wedge (b \to c)\Big) \to (a \to c)$
- ***b)*** $B = \Big(\Big(s \to (b \vee t)\Big) \wedge \Big(\Big((b \vee a) \to (r \wedge m)\Big) \wedge \neg r\Big)\Big) \to (s \to t)$.

Correction :