4.4 Forme clausale - Mise sous forme clausale, simplifications, substitutions

Nous nous concentrons maintenant sur la forme clausale. Comme déjà dit elle sera le point de départ de plusieurs algorithme.

4.4.1 Mise sous forme clausale, en préservant l'équivalence

L'algorithme consiste en l'application des 3 étapes suivantes :

Etape 1 (élimination de l'implication). Appliquer, tant que possible, la substitution suivante :

$$\varphi \Rightarrow \psi \leftarrow \neg\varphi \vee \psi \ .$$

Etape 2 (pousser la négation vers les symboles propositionnels). Appliquer, tant que possible, les substitutions suivantes (en remplaçant le membre gauche par le membre droit) :

$$\neg\neg\varphi \leftarrow \varphi \ , \qquad \neg(\varphi \wedge \psi) \leftarrow \neg\varphi \vee \neg\psi \ , \qquad \neg(\varphi \vee \psi) \leftarrow \neg\varphi \wedge \neg\psi \ .$$

Etape 3 (pousser la disjonction vers les littéraux). Appliquer, tant que possible, les substitutions suivantes (en remplaçant le membre gauche par le membre droit) :

$$\varphi \vee (\psi_1 \wedge \psi_2) \leftarrow (\varphi \vee \psi_1) \wedge (\varphi \vee \psi_2) \ .$$

Exemple 4.35. Considérons $\varphi = (\neg(p \wedge (q \Rightarrow (r \vee s)))) \wedge (p \vee q)$.

— Etape 1 :
Remplacer $q \Rightarrow (r \vee s)$ par $\neg q \vee (r \vee s)$ dans φ :
$\varphi \equiv (\neg(p \wedge (\neg q \vee (r \vee s)))) \wedge (p \vee q)$

— Etape 2 :
$\varphi \equiv (\neg p \vee \neg(\neg q \vee (r \vee s))) \wedge (p \vee q)$
$\varphi \equiv (\neg p \vee (\neg\neg q \wedge \neg(r \vee s))) \wedge (p \vee q)$
$\varphi \equiv (\neg p \vee (q \wedge \neg(r \vee s))) \wedge (p \vee q)$
$\varphi \equiv (\neg p \vee (q \wedge \neg r \wedge \neg s)) \wedge (p \vee q)$

— Etape 3 :
$\varphi \equiv (\neg p \vee q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee \neg s) \wedge (p \vee q)$

Proposition 4.36. *Le calcul précédent termine et donne une formule en forme clausale équivalente à la formule initiale.*

Démonstration. Clairement, l'étape 1 termine et donne une formule ne contenant plus de symboles $\Rightarrow$. Regardons la formule sous sa forme arborescente : à chaque itération de l'étape 2, on fait soit disparaitre 2 négations, soit descendre d'un niveau une négation. Cette étape termine donc et donne une arbre dans lequel les négations sont toutes père d'un symbole propositionnel. La troisième étape conserve cette propriété, de plus, elle consiste à faire remonter, tant que c'est possible, les $\wedge$ au dessus des $\vee$. Donc cette étape termine et donne un arbre dans lequel un $\vee$ est toujours le fils d'un $\wedge$ (sauf dans le cas ou la formule obtenue à la fin de l'étape 2 ne contient pas de $\wedge$, dans ce cas la formule était déjà sous forme clausale). L'algorithme termine donc et la formule obtenue est sous forme clausale.

A chaque étape on a appliqué une substitution d'une sous-formule par une formule équivalente, donc d'après la Proposition 4.28, la formule obtenue à la fin de l'algorithme est équivalente à la formule initiale. □

Remarque 4.37. Il n'y a pas unicité de la forme clausale.

4.4.2 Simplication d'un ensemble de clause

Considérons une formule φ sous forme clausale. Elle est de la forme $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_n$ où chaque C_i est une clause disjonctive. Définissons l'ensemble $\mathcal{C}_\varphi = \{C_1, \ldots, C_n\}$. Satisfaire φ est équivalent à satisfaire chacune des clauses de $\mathcal{C}_\varphi$.

Par exemple, si

$$\varphi = (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_4)$$

(donc φ est en FNC), alors l'ensemble de clauses associé est

$$\mathcal{C}_\varphi = \{\, \neg x_1 \vee x_2 \,, \neg x_2 \vee x_3 \,, \neg x_3 \vee x_4 \,, \neg x_1 \vee \neg x_4 \,\}\,.$$

La plupart des algorithmes que nous allons considérer manipulent des ensembles des clauses.

Algorithme de simplification : La simplification d'un ensemble de clause $\mathcal{C}$ consiste en l'application, tant que possible, de l'ensemble des modifications suivantes :

Tiers exclu : supprimer de $\mathcal{C}$ toutes les clauses C comportant deux littéraux opposés.
Par exemple une clause $p \vee q \vee \neg r \vee \neg q$ doit être supprimée.

Fusion (ou factorisation) : dans chaque clause $C \in \mathcal{C}$ supprimer les répétitions de littéraux.
Par exemple, remplacer la clause $\neg p \vee q \vee \neg r \vee \neg p$ par $\neg p \vee q \vee \neg r$.

Subsumption : si il existe deux clauses C et C' telles que tous les littéraux de C apparaissent dans C', alors supprimer C'.
Par exemple, si $C = p \vee q \vee r$ et $C' = p \vee \neg s \vee t \vee q \vee r$ sont dans $\mathcal{C}$, alors on enlève C' de $\mathcal{C}$.

Notez que l'algorithme termine puisqu'à chaque étape, soit on supprime une clause, soit on diminue la taille d'une clause. L'exercice suivant permet de vérifier que l'application de l'algorithme de simplification ne change pas les modèles d'un ensemble de clauses.

Exercice 4.38. Soit $\mathcal{C}$ un ensemble de clauses et $\mathsf{mod}(\mathcal{C})$ l'ensemble des valuations qui sont modèles de chacune des clauses de $\mathcal{C}$. Montrez que chacune des règles de l'algorithme de simplification (tiers-exclu, fusion, subsumption) préserve l'ensemble des modèles c'est à dire que si $\mathcal{C}'$ est l'ensemble des clauses obtenu en appliquant une fois une des simplifications, alors $\mathsf{mod}(\mathcal{C}) = \mathsf{mod}(\mathcal{C}')$.

4.4.3 Algorithme de substitution

Voyons maintenant l'algorithme de substitution d'une variable propositionnelle par $\bot$ ou $\top$:

$\mathcal{C}_{[p \leftarrow \top]}$ **:** est l'ensemble obtenu de $\mathcal{C}$ en supprimant toutes les clauses contenant le littéral p, et en supprimant $\neg p$ de toutes les clauses contenant $\neg p$.
Notez le cas particulier : la clause $\neg p$ est remplacée par la clause $\bot$. En effet $\neg p \equiv \bot \vee \neg p$, donc en supprimant $\neg p$ de la clause on obtient $\bot$.
Par exemple $\{p \vee q, \neg p \vee r, \neg p\}_{[p \leftarrow \top]} = \{r, \bot\}$.

$\mathcal{C}_{[p \leftarrow \bot]}$ **:** est l'ensemble obtenu de $\mathcal{C}$ en supprimant toutes les clauses contenant $\neg p$, et en supprimant le littéral p de toutes les clauses contenant le littéral p.
Notez le cas particulier : la clause p est remplacée par la clause $\bot$. En effet $p \equiv \bot \vee p$, donc en supprimant p de la clause on obtient $\bot$.
Par exemple $\{p \vee q, \neg p \vee r, p\}_{[p \leftarrow \bot]} = \{q, \bot\}$.

Notation : Si $\ell \in \{\, p, \neg p \,\}$ est un littéral, $\mathcal{C}_{[\ell \leftarrow \top]}$ désigne la substitution
— $\mathcal{C}_{[p \leftarrow \top]}$, si $\ell = p$,
— $\mathcal{C}_{[p \leftarrow \bot]}$, si $\ell = \neg p$.
Il s'agit donc de la substitution rendant le littéral ℓ vrai.
De la même façon, nous désignerons par $\mathcal{C}_{[\ell \leftarrow \bot]}$ la substitution qui force ℓ à être faux.