TD nº 3

Calcul Propositionnel - Équivalences, algorithmes, français

ÉQUIVALENCES

Soient $\varphi, \theta \in \mathscr{F}_{\text{cp}}$ et $q \in \textsc{Prop}$; la substitution, dans la formule φ, de la variable q par la formule θ, notée $\varphi_{[q\leftarrow\theta]}$, se définit par induction sur la structure d'une formule selon les cas suivants :

$$\begin{aligned}
p_{[q\leftarrow\theta]} &:= \begin{cases} \theta, & \text{si } p = q\,, \\ p, & \text{sinon}\,, \end{cases} \\
(\psi \circ \psi')_{[q\leftarrow\theta]} &:= \psi_{[q\leftarrow\theta]} \circ \psi'_{[q\leftarrow\theta]}\,, \text{ ou } \circ \in \{\wedge, \vee, \Rightarrow\}\,, \\
(\neg\psi)_{[q\leftarrow\theta]} &:= \neg(\psi_{[q\leftarrow\theta]})\,.
\end{aligned}$$

Exercice 3.1 Ecrivez explicitement $\varphi_{[p\leftarrow\theta]}, \varphi_{[q\leftarrow\theta]}$, et $\varphi_{[r\leftarrow\theta]}$, où

1. $\varphi := [(q \vee \neg p) \Rightarrow r] \wedge [r \Rightarrow (\neg p \vee q)]$ et $\theta := \neg\neg q \vee \neg p$,
2. $\varphi := [(q \vee \neg p) \Rightarrow (\neg\neg q \vee \neg p)] \wedge [(\neg\neg q \vee \neg p) \Rightarrow r]$ et $\theta := \neg p \vee q$.

Exercice 3.2 1. Argumentez que la relation $\equiv$ entre formules propositionnelles est réflexive symétrique et transitive (c'est-à-dire, c'est une relation d'équivalence).

2. En utilisant une suite d'équivalences, montrez que la formule

$$\varphi := [(q \vee \neg p) \Rightarrow (\neg\neg q \vee \neg p)] \wedge [(\neg\neg q \vee \neg p) \Rightarrow (\neg p \vee q)]$$

est équivalente à $\top$. Justifiez votre calcul, en faisant appel, à chaque équivalence, à une des équivalences classiques entre formules propositionnelles vues en cours.

Exercice 3.3 Considérez cet enoncé : *si* $\theta \equiv \theta'$, *alors* $\varphi_{[q\leftarrow\theta]} \equiv \varphi_{[q\leftarrow\theta']}$. Prouvez que l'enoncé est vrai, pour tout $\varphi, \theta, \theta' \in \mathscr{F}_{\text{cp}}$ et $q \in \textsc{Prop}$. (Conseil : par induction sur la structure de φ, ...)

FORMES NORMALES

Exercice 3.4 1. Calculer une forme clausale (conjonctive) de chacune des formules suivantes :

(a) $\psi_1 = (p \wedge \neg((q \vee r) \Rightarrow p)) \vee s$;

(b) $\psi_2 = (p_1 \wedge q_1) \vee (p_2 \wedge q_2)$;

(c) $\psi_3 = \neg((p \Leftrightarrow q) \Rightarrow (r \Rightarrow s))$.

Exercice 3.5 Pour chaque $n > 0$, on définit les formules

$$\varphi_n = (p_{1,0} \wedge p_{1,1}) \vee \ldots \vee (p_{n,0} \wedge p_{n,1})\,, \qquad \psi_n = \bigwedge_{f:\{1,\ldots,n\}\to\{0,1\}} p_{1,f(1)} \vee \ldots \vee p_{n,f(n)}\,.$$

Montrez (en utilisant les équivalences et l'induction sur les entiers positifs) que ψ_n est une FNC de φ_n.

Exercice 3.6 Une clause C est complete par rapport à un ensemble de symboles propositionnels $\{p_1, \ldots, p_n\}$ si, pour tout $i = 1, \ldots, n$, soit p_i apparait dans la clause, soit $\neg p_i$ apparait dans C (mais pas tous les deux).

1. Proposez deux formules équivalentes et en FNC, avec ces propriétés : (a) il possèdent un nombre différent de clauses ; (b) les clauses des ces formules sont distinguées. Argumentez donc que la forme normale conjonctive n'est pas une unique, même à associativité, commutativité et idempotence près.
2. Montrez que toute formule dont le symbole propositionnels sont parmi $\{ p_1, \dots, p_n \}$ est équivalente à une formule en FNC, dont toutes les clauses sont complètes (par rapport à $\{ p_1, \dots, p_n \}$). Montrez que cette forme normale est unique à associativité, commutativité et idempotence près.

ALGORITHMES

Exercice 3.7 *(Algorithme de Quine).* Transformez la formule suivante :

$$\varphi = (\,(c \Rightarrow ((b \vee a) \wedge d)) \wedge (b \Leftrightarrow (a \wedge (c \vee d))) \wedge (c \Rightarrow a)) \vee ((b \wedge a) \Rightarrow d).$$

en sa forme normale conjonctive, et ensuite appliquez l'algorithme de Quine pour trouver un modele.

Utilisez ensuite la méthode de Quine pour trouver tous les modèles de φ.

Exercice 3.8 *(Algorithme DPLL).* Transformez la formule suivante :

$$\varphi = \neg[(x \Rightarrow w) \Rightarrow ((y \Rightarrow z) \Rightarrow ((x \vee y) \Rightarrow (w \wedge y \wedge z \wedge \neg x)))]$$

en FNC et appliquez ensuite l'algorithme de DPLL pour trouver un modèle. Répétez ensuite l'exercice avec la formule de l'exercice 3.7.

Exercice 3.9 Montrez que, pour tout $n \geq 1$, il existe un ensemble de clauses $\mathscr{C}_n$, dont les symboles propositionnels sont $\{ p_1, \dots, p_n \}$ tel que l'arbre construit par l'algorithme de Quine est un arbre complet de profondeur n (donc, avec 2^n noeuds) où, par contre, l'arbre exploré par l'algorithme DPLL est une branche de longueur n (donc, seulement $n+1$ noeuds sont explorés par l'algorithme).

FRANÇAIS ET LOGIQUE FORMELLE

Exercice 3.10 Traduire les assertions ci-dessous en associant les variables propositionnelles p, q, r aux énoncés suivants : p : *il pleut* ; q : *Pierre prend son parapluie* ; r : *Pierre est mouillé*.

1. S'il pleut Pierre prend son parapluie.
2. Si Pierre prend son parapluie, Pierre n'est pas mouillé.
3. S'il ne pleut pas, Pierre ne prend pas son parapluie et Pierre n'est pas mouillé.

Montrer que « Pierre n'est pas mouillé » est une consequence logique des trois énoncés précédents.

Exercice 3.11 La finale d'un tournoi de tennis oppose deux joueurs A et B. Après le match, les joueurs s'adressent à la presse :

— A dit : « je ne suis pas le gagnant ».
— B dit : « A ne ment pas ».

Le but de l'exercice est de représenter les informations précédentes par un ensemble de formules du calcul propositionnel. Pour cela on utilisera les symboles propositionnels :

— Ag qui signifie *A est le gagnant du match*,
— Am qui signifie *A ment*
— Bg qui signifie *B est le gagnant du match*
— Bm qui signifie *B ment*

Représenter toutes les informations, à savoir :

— un des joueurs a gagné et l'autre a perdu,
— A dit qu'il n'a pas gagné (si A ne ment pas alors A n'a pas gagné, sinon c'est le contraire),
— B dit que A ne ment pas.