Université Paris-Diderot
Licence

Outils Logiques
Année 2017-2018

TD n°3

Systèmes de connecteurs

Rappel sur la notion de complétude d'un ensemble de connecteurs introduite en cours. On peut formaliser cette question comme suit : Un ensemble de connecteurs est complet si **pour tout** nombre naturel n, et **pour toute** fonction f

$$f\colon \underbrace{\{0,1\} \times \ldots \times \{0,1\}}_{n \text{ fois}} \to \{0,1\}$$

il existe une formule propositionnelle p, avec $\mathcal{V}(p) \subseteq \{x_1, \ldots, x_n\}$ qui « réalise » f, c'est-à-dire telle que

$$[\![p]\!][x_1 \mapsto b_1, \ldots, x_n \mapsto b_n] = f(b_1, \ldots, b_n)$$

pour toutes les valeurs booléennes $b_1, \ldots, b_n \in \{0,1\}$.

Exercice 1

1. Montrer que $\{\neg, \wedge\}$ est fonctionnellement complet.
2. Montrer que $\{\neg, \vee\}$ est fonctionnellement complet.
3. Montrer que $\{\rightarrow, \neg\}$ est fonctionnellement complet.
4. Montrer que $\{\rightarrow, \oplus\}$ est fonctionnellement complet.

Négation de la complétude Un ensemble $\mathcal{C}$ de connecteurs n'est pas complet s'il **existe** un nombre naturel n, et il **existe** une fonction f

$$f\colon \underbrace{\{0,1\} \times \ldots \times \{0,1\}}_{n \text{ fois}} \to \{0,1\}$$

telle que **aucune** formule propositionnelle p avec $\mathcal{V}(p) \subseteq \{x_1, \ldots, x_n\}$ ne « réalise » f, c'est-à-dire, **pour toute** formule propositionnelle p, ils **existent** des valeurs booléennes $b_1, \ldots, b_n \in \{0,1\}$ telles que

$$[\![p]\!][x_1 \mapsto b_1, \ldots, x_n \mapsto b_n] \neq f(b_1, \ldots, b_n)$$

Pour prouver qu'une telle formule n'existe pas, on raisonnera par induction sur l'ensemble des formules construites uniquement avec les connecteurs de l'ensemble $\mathcal{C}$.

Exercice 2 Montrer que l'ensemble de connecteurs $\{\wedge, \vee\}$ n'est pas complet.

Exercice 3 Montrer que l'ensemble de connecteurs $\{\rightarrow, \wedge\}$ n'est pas complet.

Exercice 4 Montrer que l'opérateur $\uparrow$ est fonctionnellement complet.

Exercice 5 Montrer que $\{\neg, \leftrightarrow\}$ n'est pas fonctionnellement complet. Pour cela il s'agira de montrer par induction sur l'ensemble des formules construites uniquement grâce à $\neg$ et $\leftrightarrow$, que pour toute formule comportant *au plus deux variables propositionnelles* x et y, le nombre d'affectations qui la satisfont – parmi les quatre affectations dont le support est inclus dans $\{x, y\}$ – est toujours pair.