3.3.3.7 Démonstration

Avec les méthodes données jusqu'à présent, on dispose d'un moyen de « prouver » qu'une formule est vraie dans une situation donnée : il suffit de calculer sa valeur de vérité. On peut parler de démonstration sémantique. Mais la notion d'équivalence logique permet d'envisager une autre façon de prouver qu'une formule est vraie, en se livrant à des manipulations d'ordre formel sur les formules, à condition de respecter quelques principes énoncés ci-après. Dans ce cas, on parle de démonstration syntaxique, ou de preuve.

Règle de substitution *Le résultat de substituer la même formule (atomique ou non) à toutes les occurrences de la même lettre dans une tautologie, est une tautologie.*
Par exemple, si l'on sait que la formule (16a) est une tautologie (c'est le cas), alors la substitution de toutes les occurrences de p par une autre formule quelconque, par exemple $(p \to q)$ (cela donne (16b)), est encore une tautologie.

(16) a. $(p \wedge q) \leftrightarrow (q \wedge p)$
b. $((p \to q) \wedge q) \leftrightarrow (q \wedge (p \to q))$

Ce principe nous permet donc de « découvrir » de nouvelles équivalences, sans passer par le calcul des valeurs de vérités. Mais ce principe n'est pas suffisant pour faire toutes les démonstrations que l'on voudrait faire, car il ne nous permet pas de manipuler des formules contingentes. Le principe suivant pallie ce manque.

Règle de remplacement *Soit α une sous-formule de φ. Si $\alpha \equiv \beta$, alors (1) le remplacement de α par β dans φ donne une formule φ' équivalente à φ, et (2) si φ est une tautologie, alors φ' l'est aussi.*
Ce principe nous dit que l'on ne change pas la valeur de vérité d'une formule en remplaçant une sous-formule par une (sous-)formule équivalente. Par exemple, si l'on sait que (17a), alors on peut conclure que (17b) et (17c) sont équivalentes. Et si on sait que (17d) est une tautologie, alors on peut conclure que (17e) en est une aussi.

(17) a. $(p \to q) \equiv \neg(p \wedge \neg q)$
b. $((p \to q) \wedge p) \to q$
c. $(\neg(p \wedge \neg q) \wedge p) \to q$
d. $((p \to q) \wedge q) \leftrightarrow (q \wedge (p \to q))$
e. $(\neg(p \wedge \neg q) \wedge q) \leftrightarrow (q \wedge (p \to q))$

Ce principe permet de produire une démonstration au sens courant en mathématiques : partant d'une formule qu'on sait vraie (on peut la savoir vraie par calcul, ou la supposer vraie et considérer ses conséquences) (on parle d'**axiomes**), on peut produire par remplacement de nouvelles formules en préservant l'équivalence, ce seront donc de nouvelles formules vraies (on parle de **théorème**).
Notons cependant que ces principes ne fournissent pas d'**algorithme** pour décider "syntaxiquement" si une formule est vraie. Il existe pour cela de nombreuses méthodes dites syntaxiques (ou procédurales), par exemple, les systèmes axiomatiques (Frege), la déduction naturelle, ou le calcul des séquents (Gentzen), etc. La plus connue de ces méthodes est la méthode des tableaux (ou des arbres). Ces méthodes (qui sont aussi à la base des systèmes modernes de déduction automatique) définissent une notion de prouvabilité : on notera $\vdash F$ le fait que F est démontrable par la méthode des tableaux (p. exemple), et $\Gamma \vdash F$ le fait que l'on peut démontrer par la même méthode la formule F en partant des prémisses Γ.