Chapitre 2

Logique des propositions

2.1 Introduction

Ce chapitre est la première étape qui va nous donner les moyens de représenter de façon formelle et calculable les propriétés sémantiques de la langue naturelle. Comme il est d'usage en la matière, on commence par une version simple de la logique moderne, que l'on appelle logique des propositions, et qui constitue le socle sur lequel se construit ce qu'on appelle aujourd'hui la logique classique (ou logique des prédicats) qui sera présentée au chapitre 3.

2.1.1 Logique, raisonnement

La logique peut être définie comme l'étude des raisonnements valides. Plus précisément, les logiciens se sont intéressés, depuis longtemps, aux propriétés **formelles** des raisonnements valides.

La démarche consiste à considérer des raisonnements valides, comme ceux de (34), et à tenter de mettre en évidence les propriétés sous-jascentes de ces raisonnements, en faisant abstraction des éléments non pertinents. Par exemple, dans le premier cas, le choix du mot *vertu* n'est pas déterminant : si on remplace *vertu* par *sagesse* dans toutes ses occurrences, on conserve la validité du raisonnement.

Nous voulons souligner ici l'analogie entre la démarche du logicien et celle du linguiste : les raisonnements valides constituent les données de départ du logicien, et ce que l'on cherche à faire, c'est élaborer un système qui reproduit ces jugements de validité, produits de la « compétence rationnelle » des sujets[1].

(34)	a.	Prémisses	Toute vertu est accompagnée de discrétion
			Il y a des zèles sans discrétion
		Conclusion	Il y a des zèles qui ne sont pas vertu

[1] Cette analogie rapide néglige des débats importants et délicats sur la question.

b. S'il pleut, la route est mouillée
La route n'est pas mouillée
―――――――――――――――
Il ne pleut pas

c. Tous les hommes sont mortels
Socrates est un homme
―――――――――――――――
Socrates est mortel

La logique des propositions ne va s'intéresser qu'à certains raisonnements parmi ceux que nous avons illustrés ici. Dans (34a), pour trouver les arguments formels qui rendent le syllogisme concluant, il faut distinguer, comme le fait la tradition philosophique antique et médiévale de la syllogistique, le sujet et le prédicat (voir par exemple [Arnauld et Nicole, 1662]). La logique propositionnelle ne va rien avoir à dire là-dessus. Dans (34b), on peut observer que les éléments essentiels qui rendent le syllogisme valide sont les connecteurs *si* et la négation. On a quelque chose comme « Si P Q ; non Q ; donc non P ». C'est là-dessus que la logique propositionnelle va travailler. Enfin, dans (34c), montré ici par anticipation, il faut regarder à l'intérieur, mais d'une façon plus détaillée que dans la syllogistique (qui ne sait rien dire de la seconde proposition, singulière). Ce sera le rôle de la logique des prédicats. La logique des propositions se concentre donc sur deux classes d'objets, les **propositions**, et les **connecteurs**.

2.1.2 Les objets de base

Propositions

Elles reçoivent une définition *externe*[2] : est appelée proposition toute expression qui peut être dite vraie ou fausse, ce qui exclut, entre autres, les questions (35a), les impératifs (35b), les exclamatifs (35c), et plus généralement tous les énoncés dits *non assertifs*, comme certains performatifs (35d), certains énoncés à fonction phatique (35e), ou — on peut en discuter — toute la classe des énoncés modalisés (35f). Sont de même exclues les expressions correspondant à des constituants non phrastiques.

(35) a. Est-ce que Paul aime la marche à pied ?
b. Fermez la porte !
c. Qu'elle est gentille !
d. Je te promets de venir
e. Tu m'entends !
f. La bataille aura lieu demain

Il faut noter qu'en logique propositionnelle, les propositions restent **inanalysées**, **atomiques** — sauf quand elles peuvent se décomposer en d'autres propositions et des connecteurs.

[2]Par contraste avec la définition héritée de la tradition philosophique, que nous qualifierons d'*interne*, qui définit une proposition comme un prédicat appliqué à un sujet.

Étant donnés les objectifs linguistiques que nous avons ici, on peut s'interroger sur le rapport entre phrase (au sens linguistique) et proposition (au sens logique). Il est clair que ce rapport n'est pas direct : (a) deux phrases peuvent exprimer la même proposition (par exemple dans deux langues différentes, ou encore lorsque l'on utilise des variantes lexicales (*dame/femme*), mais aussi parce que la proposition correspond au contenu littéral, qui peut être le même pour deux phrases dont le contenu exprimé est différent). (b) Une phrase peut contenir plusieurs propositions (soit par la présence d'un connecteur (36a), soit par des effets sémantiques (présupposition en (36b)) ou pragmatiques (36d), soit par ambiguïté (36e)).

(36) a. Si Pierre chante, tout le monde se plaint
b. Le Roi de France est chauve
c. Avant son mariage, il était moins bien nourri
d. Même Jean est venu
e. Toutes les françaises admirent un acteur

Connecteurs

Les connecteurs sont des opérateurs qui permettent, en reliant deux propositions, de former une nouvelle proposition : avec cette définition, le prototype du connecteur est le mot *et*, qui (au moins dans certains cas) fonctionne en effet de cette manière-là : la phrase (37c) est une proposition formée au moyen du connecteur *et* et de deux (autres) propositions.

(37) a. Il pleut à Paris
b. Il neige à Ouagadougou
c. Il pleut à Paris et il neige à Ouagadougou

Il est important de noter que si le mot *et* fonctionne dans certains cas comme un connecteur, ce n'est pas nécessairement le cas dans tous ses emplois, qui pourtant seraient qualifiés de connecteur au point de vue linguistique. Ici, c'est une définition logique que nous adoptons, et l'emploi de mots français n'est qu'une commodité.

Les connecteurs qui nous intéressent ici sont caractérisés par leur sensibilité exclusive à la vérité ou la fausseté de leurs opérandes, c'est ce que l'on appelle le caractère **véri-fonctionnel** des connecteurs. Il y a en langue des expressions qui semblent jouer un rôle de connecteur (par exemple les conjonctions de subordination), mais qui ne sont pas véri-fonctionnelles.

(38) a. Jean s'est cogné et il pleure
b. Jean pleure parce qu'il s'est cogné
c. Jean pleure
d. Jean s'est cogné

Supposons que (a) soit vraie. Prenons à la place de 'Jean pleure' une autre proposition vraie, par exemple 'il pleut', alors "Jean s'est cogné et il pleut" est vraie aussi. De même, si on suppose (c) fausse, la valeur de vérité de (38a) n'est pas modifiée par la substitution de 'Jean pleure' par une autre proposition fausse. Autrement dit, par rapport à l'interprétation de *et*, ce qui compte est simplement la valeur de vérité des arguments.

Si on fait la même substitution dans (b), on obtient une phrase fausse. Pourquoi ? Parce que le connecteur *parce que* ne dépend pas seulement des valeurs de vérité de ses opérandes.

En nous restreignant aux connecteurs véri-fonctionnels, on bénéficie d'une conséquence directe de la compositionalité : pour calculer la vérité d'une proposition, il suffit de considérer la vérité des propositions qui la composent.

Un langage (formel) se définit au moyen (1) d'un alphabet ou vocabulaire (c'est-à-dire d'un ensemble de symboles), (2) d'une syntaxe, qui détermine la façon d'organiser les symboles pour former des expressions (bien formées), et (3) d'une sémantique, qui fixe la signification des symboles élémentaires et une méthode de calcul pour la composition des significations.

2.2 Syntaxe

2.2.1 Formules bien formées

Soit L_p le langage de la logique des propositions. Le vocabulaire de L_p est constitué (i) d'un ensemble de *symboles de proposition* P, Q, R..., (ii) du connecteur unaire $\neg$, (iii) des connecteurs binaires $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, et (iv) des parenthèses (et).

Les **formules bien formées** du langage L_p sont données par :

(i). Tous les symboles de propositions sont des formules de L_p.
(ii). Si φ est une formule de L_p, alors $\neg\varphi$ est une formule de L_p.
(iii). Si φ et ψ sont des formules de L_p, alors $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, et $(\varphi \leftrightarrow \psi)$ sont des formules de L_p.
(iv). Rien d'autre n'est une formule (Seules sont des formules les expressions qui peuvent être générées par les règles 1, 2 et 3 en un nombre fini d'étapes).

La signification précise des connecteurs est donnée plus loin, mais on peut déjà indiquer leur nom : $\wedge$ est appelé **et** (logique) ou conjonction ; $\vee$ est appelé **ou** (logique), ou disjonction, ou **ou inclusif** ; $\rightarrow$ est appelé **implication** ou conditionnel (matériel) ; et $\leftrightarrow$ est appelé **équivalence** (matérielle).

Noter que dans cette version du langage, les parenthèses sont associées exclusivement aux connecteurs **binaires** ; l'expression $\neg(P)$, bien que facile à interpréter, n'est pas bien formée. Par ailleurs, on s'autorisera, lorsqu'aucune ambiguïté n'en découle, à ne pas noter les parenthèses les plus externes.

2.2.2 Arbres de construction

La définition récursive précédente permet de donner, pour tout formule bien formée, un **arbre de construction** (ou de décomposition). Par exemple, la formule $((\neg(P \vee Q) \rightarrow \neg\neg\neg Q) \leftrightarrow R)$ peut-être décomposée de la manière représentée à la figure 2.1.

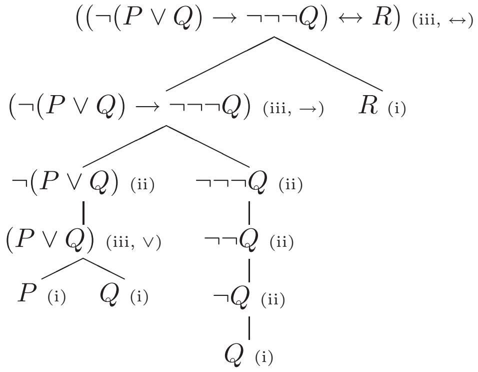

FIG. 2.1 – Exemple d'arbre de construction

L'arbre de construction peut être vu comme la trace de la vérification qu'une suite de symboles est une formule bien formée. Ainsi, la formule précédente est bien formée si et seulement si elle a été construite avec le connecteur $\leftrightarrow$, et les deux (sous-)formules $((\neg(P \vee Q) \rightarrow \neg\neg\neg Q) \leftrightarrow R)$ d'une part et R d'autre part. Le connecteur $\leftrightarrow$ est appelé **signe principal** de la formule. La formulation des règles de syntaxe garantit que toute formule a un et un seul signe principal. On voit que pour garantir que la formule initiale est bien formée, il faut maintenant vérifier que les deux premiers fils dans l'arbre correspondent à des formules bien formées. Il faut donc réitérer le processus, jusqu'à aboutir à des formules comme R, qui sont bien formées en vertu de la règle (i), puisque R est un symbole de proposition.

Il est important de noter que toute formule bien formée a un et un seul arbre de construction (c'est une conséquence du fait que le langage de la logique est syntaxiquement non ambigu) ; et que de surcroît apparaissent dans cet arbre exactement toutes ses **sous-formules** (une sous-formule d'une formule φ est une suite (contiguë) de symboles de φ qui est elle-même une formule bien formée).

2.3 Sémantique

Donner une sémantique à un langage formel consiste à définir une fonction (au sens mathématique) qui est capable d'associer à toute formule bien formée un « sens » : en l'occurrence, le sens d'une formule sera simplement une valeur de vérité (vrai ou faux).

Pour définir cette fonction, on procède en deux étapes : d'abord, on donne un sens aux éléments atomiques du langage, c'est-à-dire d'une part les variables propositionnelles et d'autre part les connecteurs (qu'on appelle quelquefois constantes logiques) (§ 2.3.1) ; ensuite, on donne une méthode de calcul pour calculer le sens de toute formule complexe à partir du sens des constituants plus simples (§ 2.3.2).

Avant de définir tous ces aspects en détail, il faut faire encore une remarque générale : les règles que nous donnerons ci-après ne permettent évidemment pas de décider, en général, si une proposition quelconque (par exemple (39a)) est vraie ou fausse. En revanche, ces règles permettront de décider si (39c) est vraie, dès lors que l'on sait si (39a) et (39b) le sont.

(39) a. Il pleut à Paris
b. Il neige à Ouagadougou
c. Il pleut à Paris et il neige à Ouagadougou

Autrement dit, pour décider si une formule est vraie (par exemple $(P \wedge Q)$), il est nécessaire de fixer les valeurs des propositions élémentaires qui interviennent dans la formule (ici, P et Q) (c'est pour cette raison qu'on parle de variables propositionnelles). Ainsi, si nous nous plaçons dans la situation où (39a) est vraie, et (39b) fausse, alors nous pouvons faire le calcul de la valeur de vérité de (39c).

2.3.1 Sémantique des connecteurs

Au point de vue sémantique, les connecteurs peuvent être vus comme des fonctions : étant données deux propositions, c'est-à-dire deux valeurs de vérité, ils donnent une valeur de vérité. Comme il n'y a que 2 valeurs de vérité possible, il est facile de résumer le comportement d'un connecteur sous la forme d'une table de vérité. Voici les tables de vérité des connecteurs définis dans L_p. (1 représente la valeur 'vrai', et 0 la valeur 'faux'.)

(40)

φ	$\neg\varphi$
0	1
1	0

φ	ψ	$\varphi \wedge \psi$
0	0	0
0	1	0
1	0	0
1	1	1

φ	ψ	$\varphi \vee \psi$
0	0	0
0	1	1
1	0	1
1	1	1

φ	ψ	$\varphi \rightarrow \psi$
0	0	1
0	1	1
1	0	0
1	1	1

φ	ψ	$\varphi \leftrightarrow \psi$
0	0	1
0	1	0
1	0	0
1	1	1

Ces définitions appellent quelques remarques, simplement évoquées ici. D'une part, la question de la correspondance entre les connecteurs logiques et leurs correspondants directs en langue est, c'est bien connu, délicate. Ensuite, on peut noter que la liste ci-dessus n'est pas exhaustive : il y a en fait 16 ($= 2^4$) connecteurs binaires définissables. Mais on sait aussi que ces connecteurs sont « interdéfinissables » par exemple il suffit d'avoir la négation et la conjonction pour pouvoir exprimer tous les autres connecteurs.

2.3.2 Calcul

Tables composites

La méthode de calcul de la valeur de vérité d'une formule repose sur la décomposition de cette formule : on crée un tableau avec une colonne par sous-formule de la formule initiale. Il y a donc en particulier dans ce tableau des colonnes pour chacun des symboles de proposition qui apparaît dans la formule. Chaque ligne de ce tableau va correspondre à une situation possible, où par situation on entend une combinaison particulière de valeurs de vérité pour les propositions élémentaires. Voici sous (41) un exemple simple, avec 6 sous-formules (y compris la formule elle-même). Il y a deux symboles de propositions (p et q), et le tableau envisage toutes les combinaisons possibles ($2^2 = 4$).

(41) $\neg(\neg p \wedge \neg q)$

p	q	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$\neg(\neg p \wedge \neg q)$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

Pour passer d'une colonne à l'autre (remarquer que l'ordre des colonnes correspond à un parcours de bas en haut de l'arbre de décomposition), il faut se reporter à la table de vérité du connecteur (ou signe) principal. Par exemple, pour calculer la colonne $\neg p \wedge \neg q$, il faut voir la formule comme $\varphi \wedge \psi$, avec $\varphi = \neg p$ et $\psi = \neg q$ (colonnes précédentes). Il suffit alors de se reporter à la table de vérité de $\wedge$ pour pouvoir faire le calcul.

Il existe une autre notation (équivalente) pour représenter le calcul que nous venons de décrire : elle consiste à représenter un tableau dont les colonnes correspondent aux symboles de la formule elle-même (parenthèses exclues) : sous chaque connecteur on représente la valeur de vérité de la sous-formule dont il est le signe principal. Voici un exemple, avec une formule comportant 3 symboles de proposition, il y a donc 2^3 lignes dans le tableau (noter, entre autres, la répétition des valeurs de p sous les deux occurrences de ce symbole).

(42)

(	(	p	$\wedge$	(	q	$\rightarrow$	r	)	)	$\vee$	(	r	$\rightarrow$	p	)	)
		0	0		0	1	0			1		0	1	0		
		0	0		0	1	1			0		1	0	0		
		0	0		1	0	0			1		0	1	0		
		0	0		1	1	1			0		1	0	0		
		1	1		0	1	0			1		0	1	1		
		1	1		0	1	1			1		1	1	1		
		1	0		1	0	0			1		0	1	1		
		1	1		1	1	1			1		1	1	1		

Noter que dans un tel calcul, on envisage la totalité des configurations possibles, on est donc à la fois capable de décider si une formule est vraie dans une situation donnée, et aussi capable de décrire l'ensemble des situations qui rendent vraie (« satisfont ») la formule. Intuitivement, c'est plutôt le premier point de vue qui nous intéresse : par exemple, si on s'intéresse à la valeur de vérité de la proposition (43a), dans la situation (43b), on peut utiliser la formule (43c), dont la légende est (43d), pour faire le calcul. La table de vérité composite comprend la ligne donnée sous (44), qui permet de conclure que la phrase (43a) est vraie dans la situation considérée. Mais bien entendu, pour produire cette ligne particulière, il faut être capable de produire l'intégralité du tableau.

(43) a. Il n'est pas vrai que Pierre viendra si Marie ou Jean vient
b. Situation : Pierre ne vient pas, Marie vient, Jean vient.
c. $\neg((Q \vee R) \to P)$
d. P = « Pierre vient » ; Q = « Marie vient » ; R = « Jean vient »

(44)

P	Q	R	$Q \vee R$	$(Q \vee R) \to P$	$\neg((Q \vee R) \to P)$
0	1	1	1	0	1

Équivalence logique

La détermination de la valeur de vérité de la formule dans tous les cas de figure permet de s'intéresser à des propriétés générales de certaines formules, et de les comparer entre elles. Par exemple, le tableau donné sous (41), reproduit ici, comprend une dernière colonne qui ressemble point pour point à celle de $p \vee q$.

(45)

p	q	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$\neg(\neg p \wedge \neg q)$	$p \vee q$
0	0	1	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	0	1	1

Le fait que les deux colonnes soient identiques signifie que les deux formules ($\neg(\neg p \wedge \neg q)$ et $p \vee q$), **dans toutes les situations**, ont la même valeur de vérité. On dira que ces formules sont **(logiquement) équivalentes**. Cette notion d'équivalence joue un rôle dans la notion de démonstration (cf. plus loin)[3].

Tautologies, contradictions

Indépendamment du problème de la comparaison de deux formules, on peut aussi rencontrer des formules qui présentent des propriétés remarquables. Ainsi, dans la table de vérité suivante, on trouve une formule qui est vraie dans toutes les situations. De telles formules, qui vont elles aussi être utilisées pour les démonstrations, sont appelées des **tautologies**. De même, il existe des formules fausses dans toute situation, qu'on appelle **contradictions**. On doit remarquer que la valeur de vérité de telles formules est par conséquent indépendante des valeurs de vérité des symboles de proposition qui y apparaissent.

[3] On peut faire des choses plus subtiles en comparant les colonnes de deux formules dans une table composite : par exemple, si entre deux colonnes on trouve le même rapport de valeurs que dans le tableau de l'implication matérielle ($\to$), on pourra dire que la première implique logiquement la seconde.

(46)

p	q	$(p \to q)$	$\neg q$	$(p \wedge \neg q)$	$(p \to q) \wedge (p \wedge \neg q)$	$(q \vee \neg q)$
0	0	1	1	0	0	1
0	1	1	0	0	0	1
1	0	0	1	1	0	1
1	1	1	0	0	0	1

La plupart des formules, dont la valeur de vérité dépend de la situation (c'est-à-dire des valeurs de vérité des formules atomiques), et ne sont par conséquent ni des tautologies ni des contradictions, sont dites **contingentes**. On utilise la (méta-)notation $\models \varphi$ pour indiquer que φ est une tautologie.

Équivalence matérielle

Il est facile de vérifier (et même de démontrer) que si φ et ψ sont logiquement équivalentes, alors la formule $(\varphi \leftrightarrow \psi)$ est toujours vraie (c'est-à-dire est une tautologie).

(47)

p	$\neg p$	$\neg\neg p$	$p \leftrightarrow \neg\neg p$
1	0	1	1
0	1	0	1

p	q	$p \wedge q$	$q \wedge p$	$(p \wedge q) \leftrightarrow (q \wedge p)$
1	1	1	1	1
1	0	0	0	1
0	1	0	0	1
0	0	0	0	1

En fait, c'est un théorème de la logique des propositions :

(48) *φ et ψ sont logiquement équivalentes ssi $(\varphi \leftrightarrow \psi)$ est une tautologie.*

Ce théorème permet donc de faire un lien entre deux notions que nous avons introduites dans ce chapitre, celle d'équivalence matérielle et celle d'équivalence logique. Mais il est important de bien faire la différence entre ces deux notions. La notion d'équivalence logique concerne deux formules qui, **dans toutes les situations** se comportent de la même façon. Il s'agit d'une relation entre formules, que l'on pourra noter au moyen du symbole $\equiv$: par exemple, $p \equiv \neg\neg p$. Mais cette notation ne fait pas partie du langage que nous avons défini plus haut, L_p. Cette notation relève du métalangage. L'équivalence matérielle, quant à elle, est un opérateur, comme l'opérateur $+$ en arithmétique : étant donnés deux opérandes, il fournit un résultat : $(\varphi \leftrightarrow \psi)$ ne dit rien des formules φ et ψ en général, mais vaut vrai ou faux selon la valeur de ses opérandes dans une situation donnée.

Cette distinction apparaît dans la formulation du théorème que nous avons choisie : c'est seulement lorsque $(\varphi \leftrightarrow \psi)$ est <u>toujours vraie</u> que l'on peut conclure que $\varphi \equiv \psi$.

Valuation

On peut formaliser complètement (« algorithmiquement ») le calcul de la valeur d'une formule, une fois connues les valeurs des propositions qui la composent. En fait, on se donne une sorte de *modèle*, sous la forme d'une fonction mathématique de l'ensemble des symboles de proposition dans l'ensemble $\{0, 1\}$. On appelle une telle fonction, notée V, une *valuation*. Alors, l'interprétation d'une formule quelconque φ (notée $[\![\varphi]\!]_V$ pour rappeler qu'elle dépend de V) peut être donnée par les règles suivantes :

1. Si φ est un symbole de proposition, alors $[\![\varphi]\!]_V = V(\varphi)$;

Pour toutes formules φ et ψ :

2. $[\![\neg\varphi]\!]_V = 1$ ssi $[\![\varphi]\!]_V = 0$;
3. $[\![(\varphi \wedge \psi)]\!]_V = 1$ ssi $[\![\varphi]\!]_V = 1$ et $[\![\psi]\!]_V = 1$;
4. $[\![(\varphi \vee \psi)]\!]_V = 0$ ssi $[\![\varphi]\!]_V = 0$ et $[\![\psi]\!]_V = 0$;
5. $[\![(\varphi \rightarrow \psi)]\!]_V = 0$ ssi $[\![\varphi]\!]_V = 1$ et $[\![\psi]\!]_V = 0$;
6. $[\![(\varphi \leftrightarrow \psi)]\!]_V = 1$ ssi $[\![\varphi]\!]_V = [\![\psi]\!]_V$;

Ces règles sont "calquées" sur la syntaxe, et nous donnent des règles d'interprétation qui en fait décomposent (récursivement) la formule jusqu'à trouver des sous-formules élémentaires dont la valeur de vérité ne dépend plus que du modèle (c'est-à-dire de V).

On peut facilement prouver que l'algorithme précédent, étant donnée une formule propositionnelle quelconque, permet de déterminer quand la formule est vraie ou fausse, au bout d'un nombre fini d'étapes (il suffit de remarquer que la table de vérité d'une formule comportant n atomes comprend au plus 2^n lignes). La logique des propositions est **décidable**.

Conséquence logique

Mais la logique, comme modèle du raisonnement, a pour objectif de permettre de démontrer des syllogismes. Alors, il ne s'agit pas seulement de décider si une formule est vraie ou fausse dans une situation, mais plutôt si la vérité d'une proposition (la conclusion) découle nécessairement de la vérité d'autres propositions (les prémisses). On introduit la notion de **conséquence logique** : une formule F est une conséquence logique d'un ensemble de formules Γ si toute valuation qui donne vrai à toutes les formules de Γ donne vrai à F. On note cela $\Gamma \models F$. Un exemple, avec $\Gamma = \{(p \wedge q), (q \rightarrow \neg r)\}, F = \neg r$:

On a :	$(p \wedge q)$	$\models$	q
De plus :	$\{q, (q \rightarrow \neg r)\}$	$\models$	$\neg r$
Donc :	$\{(p \wedge q), (q \rightarrow \neg r)\}$	$\models$	$\neg r$

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 (49a) 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 (49b)), est encore une tautologie.

(49) 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 (50a), alors on peut conclure que (50b) et (50c) sont équivalentes. Et si on sait que (50d) est une tautologie, alors on peut conclure que (50e) en est une aussi.

(50) 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 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.

2.4 Résultats et repères bibliographiques

Nous donnons succintement ici quelques résultats très connus et importants concernant la logique des propositions, sans les démontrer car cela n'entre pas dans le propos de ce cours.

Décidabilité On a déjà indiqué plus haut que le calcul de la valeur de vérité d'une formule quelconque est toujours réalisable en un temps fini : la logique des propositions est **décidable**.

Déduction Le théorème de déduction permet de relier la notion de conséquence logique avec l'implication matérielle :

$$\varphi \models \psi \text{ si et seulement si } \models (\varphi \rightarrow \psi)$$

Adéquation et complétude La notion *sémantique* de validité ($\models$) et la notion *procédurale* de prouvabilité ($\vdash$) sont reliées, pour la logique des propositions, par les deux théorèmes fameux :

Théorème d'**adéquation** (*soundness*)	Si $\Gamma \vdash F$ alors $\Gamma \models F$
Théorème de **complétude** (*completeness*)	Si $\Gamma \models F$ alors $\Gamma \vdash F$

Repères bibliographiques : voir fin du chapitre 3.