Chapitre 3

Calcul propositionnel

La *logique propositionnelle* permet essentiellement de discuter des connecteurs grammaticaux comme la négation, la conjonction et la disjonction, en composant des propositions à partir de propositions données. Ces connecteurs sont parfois appelés *aristotéliciens* , car ils ont été mis en évidence par Aristote.

Le *calcul propositionnel* permet essentiellement de parler de *fonctions booléennes*, c'est-à-dire de fonctions de $\{0,1\}^n \to \{0,1\}$. En effet, les variables, c'est-à-dire *les propositions*, ne peuvent prendre que deux valeurs, *vrai* ou *faux*.

Le calcul propositionnel tient une grande place en informatique : ne serait-ce parce que nos ordinateurs actuels sont digitaux, et travaillent en binaire. Ce qui fait que nos processeurs sont essentiellement constitués de portes binaires du type de celles que l'on va étudier dans ce chapitre.

D'un point de vue expressivité logique, le calcul propositionnel reste très limité : par exemple, on ne peut pas écrire en calcul propositionnel l'existence d'un objet ayant une propriété donnée. Le calcul des prédicats, plus général, que nous étudierons dans le chapitre 5Calcul des prédicatschapter.5, permet lui d'exprimer des propriétés d'objets et des relations entre objets, et plus généralement de formaliser le raisonnement mathématique.

Puisque le calcul propositionnel forme toutefois la base commune de nombreux systèmes logiques, et nous allons nous y attarder dans ce chapitre.

3.1 Syntaxe

Pour définir formellement et proprement ce langage, nous devons distinguer la syntaxe de la sémantique : la syntaxe décrit comment on écrit les formules. La sémantique décrit leur sens.

Fixons un ensemble fini ou dénombrable $\mathcal{P} = \{p_0, p_1, \cdots\}$ de symboles que l'on appelle *variables propositionnelles*.

Définition 3.1 (Formules propositionnelles) *L'ensemble des formules propositionnelles $\mathcal{F}$ sur $\mathcal{P}$ est le langage sur l'alphabet $\mathcal{P} \cup \{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, (,)\}$ défini inductivement par les règles suivantes :*

(B) *il contient $\mathcal{P}$: toute variable propositionnelle est une formule propositionnelle ;*
(I) *si $F \in \mathcal{F}$ alors $\neg F \in \mathcal{F}$;*
(I) *si $F, G \in \mathcal{F}$ alors $(F \wedge G) \in \mathcal{F}$, $(F \vee G) \in \mathcal{F}$, $(F \Rightarrow G) \in \mathcal{F}$, et $(F \Leftrightarrow G) \in \mathcal{F}$.*

Il s'agit d'une définition inductive qui est légitime par les considérations du chapitre précédent. Il s'agit d'une définition inductive non ambigüe : on peut reformuler ce fait par la proposition suivante, parfois appelé *théorème de lecture unique*.

Remarque 3.1 *La non-ambiguïté vient essentiellement des parenthèses explicites. On utilise ici l'astuce utilisée dans le chapitre précédent qui considérait $Arith'$ plutôt que $Arith$ pour permettre d'écrire des expressions sans aucune ambigüité de lecture.*

Proposition 3.1 (Décomposition / Lecture unique) *Soit F une formule propositionnelle. Alors F est d'une, et exactement d'une, des formes suivantes*

1. *une variable propositionnelle $p \in \mathcal{P}$;*
2. *$\neg G$, où G est une formule propositionnelle ;*
3. *$(G \wedge H)$ où G et H sont des formules propositionnelles ;*
4. *$(G \vee H)$ où G et H sont des formules propositionnelles ;*
5. *$(G \Rightarrow H)$ où G et H sont des formules propositionnelles ;*
6. *$(G \Leftrightarrow H)$ où G et H sont des formules propositionnelles.*

De plus dans les cas 2., 3., 4., 5. et 6., il y a unicité de la formule G et de la formule H avec ces propriétés.

Le fait qu'une formule se décompose toujours dans un des 6 cas plus hauts est facile à établir inductivement. L'unicité de la décomposition découle de l'exercice suivant :

Exercice 3.1. *Montrer que la définition inductive précédente est non-ambigüe, c'est-à-dire que G et H sont uniquement définis dans chacun des cas plus haut.*

On pourra procéder de la façon suivante.

— *Montrer que dans toute formule F le nombre de parenthèses ouvrantes est égal au nombre de parenthèses fermantes.*
— *Montrer que dans tout mot M préfixe de F, on a $o(M) \geq f(M)$, où $o(M)$ est le nombre de parenthèses ouvrantes, et $f(M)$ le nombre de parenthèses fermantes.*
— *Montrer que pour toute formule F dont le premier symbole est une parenthèse ouvrante, et pour tout mot M préfixe propre de F, on a $o(M) > f(M)$.*
— *Montrer que tout mot M préfixe propre de F n'est pas une formule.*
— *En déduire le résultat.*

On appelle *sous-formule* de F une formule qui apparaît dans la décomposition récursive de F.

p	$\neg p$	q	$p \vee q$	$p \wedge q$	$p \Rightarrow q$	$p \Leftrightarrow q$
0	1	0	0	0	1	1
1	0	0	1	0	0	0
0	1	1	1	0	1	0
1	0	1	1	1	1	1

FIGURE 3.1 – Tableau de vérité.

3.2 Sémantique

Nous allons maintenant définir la *sémantique* d'une formule propositionnelle, c'est-à-dire le sens qu'on lui donne.

La *valeur de vérité* d'une formule se définit comme l'interprétation de cette formule, une fois que l'on s'est fixé la valeur de vérité des variables propositionnelles : le principe est d'interpréter les symboles $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$ par la négation logique, le ou logique (appelé aussi disjonction), le et logique (appelé aussi conjonction), l'implication et la double implication (aussi appelée équivalence) logique.

Formellement,

Définition 3.2 (Valuation) *Une* valuation *est une distribution de valeurs de vérité aux variables propositionnelles, c'est-à-dire une fonction de $\mathcal{P}$ vers $\{0, 1\}$.*

Dans tout ce qui suit, 0 représente faux, et 1 représente vrai.

On représente souvent les conditions de la définition suivante sous la forme d'un *tableau de vérité* : voir la figure 3.1.

Proposition 3.2 *Soit v une valuation.*

Par le théorème 2.5Fonction définie inductivementtheorem.2.5, il existe une unique fonction $\overline{v}$ définie sur tout $\mathcal{F}$ qui vérifie les conditions suivantes

(B) *$\overline{v}$ étend v : pour toute variable propositionnelle $p \in \mathcal{P}$, $\overline{v}(p) = v(p)$;*

(I) *la négation s'interprète par la négation logique :*
si F est de la forme $\neg G$, alors $\overline{v}(F) = 1$ ssi $\overline{v}(G) = 0$;

(I) *$\wedge$ s'interprète comme le et logique :*
si F est de la forme $G \wedge H$, alors $\overline{v}(F) = 1$ ssi $\overline{v}(G) = 1$ et $\overline{v}(H) = 1$;

(I) *$\vee$ s'interprète comme le ou logique :*
si F est de la forme $G \vee H$, alors $\overline{v}(F) = 1$ ssi $\overline{v}(G) = 1$ ou $\overline{v}(H) = 1$;

(I) *$\Rightarrow$ s'interprète comme l'implication logique :*
si F est de la forme $G \Rightarrow H$, alors $\overline{v}(F) = 1$ ssi $\overline{v}(H) = 1$ ou $\overline{v}(G) = 0$;

(I) *$\Leftrightarrow$ s'interprète comme l'équivalence logique :*
si F est de la forme $G \Leftrightarrow H$, alors $\overline{v}(F) = 1$ ssi $\overline{v}(G) = \overline{v}(H)$.

On écrit $v \models F$ pour $\overline{v}(F) = 1$, et on dit que v est *un modèle* de F, ou que v satisfait F. On note $v \not\models F$ dans le cas contraire. La valeur de $\overline{v}(F)$ pour la valuation v est appelée *la valeur de vérité de F sur v*.

3.3 Tautologies, formules équivalentes

On souhaite classer les formules selon leur interprétation. Une classe particulière de formules est celles qui sont toujours vraies que l'on appelle les *tautologies*.

Définition 3.3 (Tautologie) *Une* tautologie *est une formule F qui est satisfaite par toute valuation. On note dans ce cas $\models F$.*

Définition 3.4 (Equivalence) *Deux formules F et G sont dites* équivalentes *si pour toute valuation v, $\overline{v}(F) = \overline{v}(G)$. On écrit dans ce cas $F \equiv G$.*

Exemple 3.1 *La formule $p \vee \neg p$ est une tautologie. Les formules p et $\neg\neg p$ sont équivalentes.*

Remarque 3.2 *Il est important de bien comprendre que $\equiv$ est un symbole que l'on utilise pour écrire une relation entre deux formules, mais que $F \equiv G$ n'est pas une formule propositionnelle.*

Exercice 3.2. *Montrer que $\equiv$ est une relation d'équivalence sur les formules.*

3.4 Quelques faits élémentaires

Exercice 3.3. *Montrer que pour toutes formules F et G, les formules suivantes sont des tautologies :*

$$(F \Rightarrow F),$$
$$(F \Rightarrow (G \Rightarrow F)),$$
$$(F \Rightarrow (G \Rightarrow H)) \Rightarrow ((F \Rightarrow G) \Rightarrow (F \Rightarrow H)).$$

Exercice 3.4. *[Idempotence] Montrer que pour toute formule F on a les équivalences :*

$$(F \vee F) \equiv F,$$
$$(F \wedge F) \equiv F.$$

Exercice 3.5. *[Associativité] Montrer que pour toutes formules F, G, H on a les équivalences :*

$$(F \wedge (G \wedge H)) \equiv ((F \wedge G) \wedge H),$$
$$(F \vee (G \vee H)) \equiv ((F \vee G) \vee H).$$

En raison de l'associativité, on note souvent $F_1 \vee F_2 \vee \cdots \vee F_k$ pour $(((F_1 \vee F_2) \vee F_3) \cdots \vee F_k)$, et $F_1 \wedge F_2 \wedge \cdots \wedge F_k$ pour $(((F_1 \wedge F_2) \wedge F_3) \cdots \wedge F_k)$.

Remarque 3.3 *Exactement comme on le fait avec les expressions arithmétiques : on écrit* $1+2+3$ *pour* $((1+2)+3)$ *comme pour* $(1+(2+3))$. *Voir toutes les discussions sur* $Arith$ *et* $Arith'$ *dans le chapitre précédent.*

Exercice 3.6. *[Commutativité] Montrer que pour toutes formules* F *et* G *on a les équivalences :*

$$(F \wedge G) \equiv (G \wedge F),$$

$$(F \vee G) \equiv (G \vee F).$$

Exercice 3.7. *[Distributivité] Montrer que pour toutes formules* F, G, H *on a les équivalences :*

$$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H)),$$

$$(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H)).$$

Exercice 3.8. *[Lois de Morgan] Montrer que pour toutes formules* F *et* G *on a les équivalences :*

$$\neg(F \wedge G) \equiv (\neg F \vee \neg G),$$

$$\neg(F \vee G) \equiv (\neg F \wedge \neg G).$$

Exercice 3.9. *[Absorption] Montrer que pour toutes formules* F *et* G *on a les équivalences :*

$$(F \wedge (F \vee G)) \equiv F,$$

$$(F \vee (F \wedge G)) \equiv F.$$

3.5 Remplacements d'une formule par une autre équivalente

Nous connaissons maintenant quelques équivalences entre formules, mais nous allons maintenant nous convaincre qu'on peut utiliser ces équivalences de façon compositionnelle : si l'on remplace dans une formule une sous-formule par une formule équivalente, alors on obtient une formule équivalente.

3.5.1 Une remarque simple

Observons tout d'abord que la valeur de vérité d'une formule ne dépend que des variables propositionnelles présentes dans la formule : lorsque F est une formule, on notera $F(p_1, \cdots, p_n)$ pour dire que la formule F s'écrit avec les variables propositionnelles $p_1, \cdots, p_n$ seulement.

Proposition 3.3 *Soit $F(p_1, \cdots, p_n)$ une formule. Soit v une valuation. La valeur de vérité de F sur v ne dépend que de la valeur de v sur $\{p_1, p_2, \cdots, p_n\}$.*

Démonstration : La propriété s'établit facilement par induction structurelle. □

3.5.2 Substitutions

Il nous faut définir ce que signifie remplacer p par G dans une formule F, noté $F(G/p)$.

Cela donne la définition un peut pédante qui suit, mais nous devons en passer par là :

Définition 3.5 (Substitution de p par G dans F) *La formule $F(G/p)$ est définie par induction sur la formule F :*

- (B) *Si F est la variable propositionnelle p, alors $F(G/p)$ est la formule G ;*
- (B) *Si F est une variable propositionnelle q, avec $q \neq p$, alors $F(G/p)$ est la formule F ;*
- (I) *Si F est de la forme $\neg H$, alors $F(G/p)$ est la formule $\neg H(G/p)$;*
- (I) *Si F est de la forme $(F_1 \vee F_2)$, alors $F(G/p)$ est la formule $(F_1(G/p) \vee F_2(G/p))$;*
- (I) *Si F est de la forme $(F_1 \wedge F_2)$, alors $F(G/p)$ est la formule $(F_1(G/p) \wedge F_2(G/p))$;*
- (I) *Si F est de la forme $(F_1 \Rightarrow F_2)$, alors $F(G/p)$ est la formule $(F_1(G/p) \Rightarrow F_2(G/p))$;*
- (I) *Si F est de la forme $(F_1 \Leftrightarrow F_2)$, alors $F(G/p)$ est la formule $(F_1(G/p) \Leftrightarrow F_2(G/p))$.*

3.5.3 Compositionnalité de l'équivalence

On obtient le résultat promis : si l'on remplace dans une formule une sous-formule par une formule équivalente, on obtient une formule équivalente.

Proposition 3.4 *Soient F, F', G et G' des formules. Soit p une variable propositionnelle.*

- *Si F est une tautologie, alors $F(G/p)$ aussi.*
- *Si $F \equiv F'$, alors $F(G/p) \equiv F'(G/p)$.*
- *Si $G \equiv G'$ alors $F(G/p) \equiv F(G'/p)$.*

Exercice 3.10. *Prouver le résultat par induction structurelle.*

3.6 Système complet de connecteurs

Proposition 3.5 *Toute formule propositionnelle est équivalente à une formule qui est construite uniquement avec les connecteurs $\neg$ et $\wedge$.*

Démonstration : Cela résulte d'une preuve par induction sur la formule. C'est vrai pour les formules qui correspondent à des variables propositionnelles. Supposons la propriété vraie pour les formules G et H, c'est-à-dire supposons que G (respectivement H) est équivalente à une formule G' (respectivement H') construite uniquement avec les connecteurs $\neg$ et $\wedge$.

Si F est de la forme $\neg G$, alors F est équivalente à $\neg G'$ et l'hypothèse d'induction est préservée.

Si F est de la forme $(G \wedge H)$, alors F est équivalente à $(G' \wedge H')$ et la propriété d'induction est préservée.

Si F est de la forme $(G \vee H)$, en utilisant la deuxième loi de Morgan et le fait que $K \equiv \neg\neg K$ pour éliminer les doubles négations, on obtient que $F \equiv \neg(\neg G' \wedge \neg H')$ qui est bien construite en utilisant uniquement les connecteurs $\neg$ et $\wedge$.

Si F est de la forme $(G \Rightarrow H)$, alors F est équivalente à $(\neg G' \vee H')$ qui est équivalente à une formule construite uniquement avec les connecteurs $\neg$ et $\wedge$ par les cas précédents.

Si F est de la forme $(G \Leftrightarrow H)$, alors F est équivalente à $(G' \Rightarrow H') \wedge (H' \Rightarrow G')$ qui est équivalente à une formule construite uniquement avec les connecteurs $\neg$ et $\wedge$ par les cas précédents. □

Un ensemble de connecteurs qui a la propriété plus haut pour $\{\neg, \wedge\}$ est appelé un *système complet de connecteurs*.

Exercice 3.11. *Montrer que $\{\neg, \vee\}$ est aussi un système complet de connecteurs.*

Exercice 3.12. *Donner un connecteur logique binaire tel qu'à lui seul il constitue un système complet de connecteurs.*

3.7 Complétude fonctionnelle

Supposons $\mathcal{P} = \{p_1, p_2, \cdots, p_n\}$ fini. Soit V l'ensemble des valuations sur $\mathcal{P}$. Puisqu'une valuation est une fonction de $\{1, 2, \ldots, n\}$ dans $\{0, 1\}$, V contient 2^n éléments.

Chaque formule F sur $\mathcal{P}$ peut être vue comme une fonction de V dans $\{0, 1\}$, que l'on appelle *valeur de vérité de F* : cette fonction est la fonction qui à une valuation v associe la valeur de vérité de la formule sur cette valuation.

Il y a 2^{2^n} fonctions de V dans $\{0, 1\}$. La question qui se pose est de savoir si toutes les fonctions peuvent s'écrire comme des formules. La réponse est positive :

Théorème 3.1 (Complétude fonctionnelle) *Supposons $\mathcal{P} = \{p_1, p_2, \cdots, p_n\}$ fini. Soit V l'ensemble des valuations sur $\mathcal{P}$. Toute fonction f de V dans $\{0, 1\}$ est la valeur de vérité d'une formule F sur $\mathcal{P}$.*

Démonstration : La preuve se fait par récurrence sur le nombre de variables propositionnelles n.

Pour $n = 1$, il y a quatre fonctions de $\{0,1\}^1$ dans $\{0,1\}$, qui se représentent par les formules p, $\neg p$, $p \vee \neg p$, $p \wedge \neg p$.

Supposons la propriété vraie pour $n-1$ variables propositionnelles. Considérons $\mathcal{P} = \{p_1, \cdots, p_n\}$ et soit f une fonction de $\{0,1\}^n$ dans $\{0,1\}$. Chaque valuation v' sur $\{p_1, p_2, \cdots, p_{n-1}\}$ peut se voir comme la restriction d'une valuation sur $\{p_1, \cdots, p_n\}$. Soit f_0 (respectivement f_1) la restriction de f à la valuation v telle que $v(p_n) = 0$ (resp. $v(p_n) = 1$). Les fonctions f_0 et f_1 sont des fonctions définies des valuations sur $\{p_1, \cdots, p_{n-1}\}$ dans $\{0,1\}$ et se représentent par des formules $G(p_1, \cdots, p_{n-1})$ et $H(p_1, \cdots, p_{n-1})$ respectivement par hypothèse de récurrence. La fonction f peut alors se représenter par la formule

$$(\neg p_n \wedge G(p_1, \cdots, p_{n-1})) \vee (p_n \wedge H(p_1, \cdots, p_{n-1}))$$

ce qui prouve l'hypothèse de récurrence au rang n. □

Remarque 3.4 *Notre lecteur assidu aura remarqué que la proposition 3.5 peut aussi se voir comme la conséquence de cette preuve.*

3.8 Formes normales

3.8.1 Formes normales conjonctives et disjonctives

On cherche souvent à ramener les formules sous une forme équivalente la plus simple possible.

Définition 3.6 *Un* littéral *est une variable propositionnelle ou sa négation, i.e. de la forme p, ou $\neg p$, pour $p \in \mathcal{P}$.*

Définition 3.7 *Une* forme normale disjonctive *est une disjonction $F_1 \vee F_2 \cdots \vee F_k$ de k formules, $k \geq 1$ où chaque formule F_i, $1 \leq i \leq k$ est une conjonction $G_1 \wedge G_2 \cdots \wedge G_\ell$ de ℓ littéraux (ℓ pouvant dépendre de i).*

Exemple 3.2 *Les formules suivantes sont des formes normales disjonctives*

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

$$((p \wedge q \wedge \neg r) \vee (q \wedge \neg p))$$

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

Définition 3.8 *Une* forme normale conjonctive *est une conjonction $F_1 \wedge F_2 \cdots \wedge F_k$ de k formules, $k \geq 1$ où chaque formule F_i, $1 \leq i \leq k$ est une disjonction $G_1 \vee G_2 \cdots \vee G_\ell$ de ℓ littéraux (ℓ pouvant dépendre de i).*

Exemple 3.3 *Les formules suivantes sont des formes normales conjonctives*

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

$$(\neg p \vee q) \wedge \neg r$$

$$(\neg p \vee q)$$

Théorème 3.2 *Toute formule sur un nombre fini de variables propositionnelles est équivalente à une formule en forme normale conjonctive.*

Théorème 3.3 *Toute formule sur un nombre fini de variables propositionnelles est équivalente à une formule en forme normale disjonctive.*

Démonstration : Ces deux théorèmes se prouvent par récurrence sur le nombre n de variables propositionnelles.

Dans le cas où $n = 1$, on a déjà considéré dans la preuve précédente des formules qui couvrent tous les cas possibles et qui sont en fait à la fois en forme normale conjonctive et disjonctive.

On suppose la propriété vraie pour $n - 1$ variables propositionnelles. Soit f la fonction valeur de vérité associée à la formule $F(p_1, \cdots, p_n)$. Comme dans la dernière preuve, on peut construire une formule qui représente f, en écrivant une formule de la forme

$$(\neg p_n \wedge G(p_1, \cdots, p_{n-1})) \vee (p_n \wedge H(p_1, \cdots, p_{n-1})).$$

Par hypothèse de récurrence, G et H sont équivalentes à des formules en forme normale disjonctive

$$G \equiv (G_1 \vee G_2 \vee \cdots \vee G_k)$$
$$H \equiv (H_1 \vee H_2 \vee \cdots \vee H_\ell)$$

On peut alors écrire

$$(\neg p_n \wedge G) \equiv (\neg p_n \wedge G_1) \vee (\neg p_n \wedge G_2) \vee \cdots \vee (\neg p_n \wedge G_k)$$

qui est en forme normale disjonctive et

$$(p_n \wedge H) \equiv (p_n \wedge H_1) \vee (p_n \wedge H_2) \vee \cdots \vee (p_n \wedge H_k)$$

qui est aussi en forme normale disjonctive. La fonction f est donc représentée par la disjonction de ces deux formules, et donc par une formule en forme normale disjonctive.

Si l'on veut obtenir F en forme normale conjonctive, alors l'hypothèse d'induction produit deux formes normales conjonctives G et H. L'équivalence que l'on utilise est alors

$$F \equiv ((\neg p_n \vee H) \wedge (p_n \vee G)).$$

□

Remarque 3.5 *Notre lecteur assidu aura remarqué que le théorème précédent, comme la proposition 3.5 peuvent aussi se voir comme la conséquence de cette preuve.*

3.8.2 Méthodes de transformation

En pratique, il existe deux méthodes pour déterminer une forme normale disjonctive, ou conjonctive équivalente à une formule donnée. La première méthode consiste à transformer la formule par équivalences successives à l'aide des règles suivantes appliquées dans cet ordre :

1. élimination des connecteurs $\Rightarrow$ par

$$(F \Rightarrow G) \equiv (\neg F \vee G)$$

2. entrée des négations le plus à l'intérieur possible :

$$\neg(F \wedge G) \equiv (\neg F \vee \neg G)$$

$$\neg(F \vee G) \equiv (\neg F \wedge \neg G)$$

3. distributivité de $\vee$ et $\wedge$ l'un par rapport à l'autre

$$F \wedge (G \vee H)) \equiv ((F \wedge H) \vee (F \wedge H))$$

$$F \vee (G \wedge H)) \equiv ((F \vee H) \wedge (F \vee H))$$

Exemple 3.4 *Mettre la formule $\neg(p \Rightarrow (q \Rightarrow r)) \vee (r \Rightarrow q)$ sous forme normale disjonctive et conjonctive.*

On utilise les équivalences successives

$$\neg(\neg p \vee (\neg q \vee r)) \vee (\neg r \vee q)$$

$$(p \wedge \neg(\neg q \vee r)) \vee (\neg r \vee q)$$

$$(p \wedge q \wedge \neg r) \vee (\neg r \vee q)$$

qui est une forme normale disjonctive.

$$(p \wedge q \wedge \neg r) \vee (\neg r \vee q)$$

$$(p \wedge \neg r \vee q) \wedge (\neg r \vee q)$$

L'autre méthode consiste à déterminer les valuations v telles que $\overline{v}(F) = 1$, et à écrire une disjonction de conjonction, chaque conjonction correspondant à une valuation pour laquelle $\overline{v}(F) = 1$.

La détermination d'une forme normale conjonctive suit le même principe, en échangant les valuations donnant la valeur 1 avec celles donnant la valeur 0, et en échangant conjonction et disjonction.

Exercice 3.13. *Montrer que la forme normale conjonctive et disjonctive d'une formule peut être exponentiellement plus longue que la taille de la formule. La taille d'une formule est définie comme la longueur de la formule vue comme un mot.*

3.9 Théorème de compacité

3.9.1 Satisfaction d'un ensemble de formules

On se donne cette fois un ensemble Σ de formules. On cherche à savoir quand est-ce qu'on peut satisfaire toutes les formules de Σ.

Commençons par fixer la terminologie.

Définition 3.9 *Soit Σ un ensemble de formules.*

- *Une valuation* satisfait Σ *si elle satisfait chaque formule de Σ. On dit aussi dans ce cas que cette valuation est* un modèle *de Σ.*
- *Σ est dit* consistant *(on dit aussi* satifiable*) s'il possède un modèle. En d'autres termes, s'il existe une valuation qui satisfait Σ.*
- *Σ est dit* inconsistant*, ou* contradictoire*, dans le cas contraire.*

Définition 3.10 (Conséquence) *Soit F une formule. La formule F est dite* une conséquence *de Σ si tout modèle de Σ est un modèle de F. On note alors $\Sigma \models F$.*

Exemple 3.5 *La formule q est une conséquence de l'ensemble de formules $\{p, p \Rightarrow q\}$. L'ensemble de formules $\{p, p \Rightarrow q, \neg q\}$ est inconsistant.*

On peut déjà se convaincre du résultat suivant, qui relève d'un jeu sur les définitions.

Proposition 3.6 *Toute formule F est une conséquence d'un ensemble Σ de formules si et seulement si $\Sigma \cup \{\neg F\}$ est inconsistant.*

Démonstration : Si toute valuation qui satisfait Σ satisfait F, alors il n'y a pas de valuation qui satisfait $\Sigma \cup \{\neg F\}$. Réciproquement, par l'absurde : s'il y a une valuation qui satisfait Σ et qui ne satisfait pas F, alors cette valuation satisfait Σ et $\neg F$. □

Exercice 3.14. *Montrer que pour toutes formules F et F', $\{F\} \models F'$ si et seulement si $F \Rightarrow F'$ est une tautologie.*

Plus fondamentalement, on a le résultat surprenant et fondamental suivant.

Théorème 3.4 (Théorème de compacité (1ière version)) *Soit Σ un ensemble de formules construites sur un ensemble dénombrable $\mathcal{P}$ de variables propositionnelles.*

Alors Σ est consistant si et seulement si toute partie finie de Σ est consistant.

Remarque 3.6 *Remarquons que l'hypothèse $\mathcal{P}$ dénombrable n'est pas nécessaire, si l'on accepte d'utiliser l'hypothèse de Zorn (l'axiome du choix). On se limitera au cas $\mathcal{P}$ dénombrable dans ce qui suit.*

En fait, ce théorème peut se reformuler sous la forme suivante

Théorème 3.5 (Théorème de compacité (2ième version)) *Soit Σ un ensemble de formules construites sur un ensemble dénombrable $\mathcal{P}$ de variables propositionnelles.*

Alors Σ est inconsistant si et seulement si Σ possède une partie finie inconsistante.

Ou encore sous la forme suivante :

Théorème 3.6 (Théorème de compacité (3ième version)) *Pour tout ensemble Σ de formules propositionnelles, et pour toute formule propositionnelle F construite sur un ensemble dénombrable $\mathcal{P}$ de variables propositionnelles, F est une conséquence de Σ si et seulement si F est une conséquence d'une partie finie de Σ.*

L'équivalence des trois formulations n'est qu'un simple exercice de manipulations de définitions. Nous allons prouver la première version du théorème.

Une des implications est triviale : si Σ est consistant, alors toute partie de Σ est consistant, et en particulier les parties finies.

Nous allons donner deux preuves de l'autre implication.

Une première preuve qui fait référence à des notions de topologie, en particulier de compacité, et qui s'adresse à ceux qui connaissent ces notions, et qui sont amateurs de topologie.

Démonstration :[Preuve topologique] L'espace topologique $\{0,1\}^{\mathcal{P}}$ (muni de la topologie produit) est un espace compact, car il s'obtient comme un produit de compacts (Théorème de Tychonoff).

Pour chaque formule propositionnelle $F \in \Sigma$, l'ensemble $\overline{F}$ des valuations qui la satisfont est un ouvert dans $\{0,1\}^{\mathcal{P}}$, car la valeur de vérité d'une formule ne dépend que d'un nombre fini de variables, celles qui apparaissent dans la formule. Il est également fermé puisque celles qui ne satisfont pas F sont celles qui satisfont $\neg F$.

L'hypothèse du théorème entraîne que toute intersection finie de $\overline{F}$ pour $F \in \Sigma$ est non-vide. Comme $\{0,1\}^{\mathcal{P}}$ est compact, l'intersection de tous les $\overline{F}$ pour $F \in \Sigma$ est donc non-vide. □

Voici une preuve qui évite la topologie.

Démonstration :[Preuve directe] Considérons $\mathcal{P} = \{p_1, p_2, \cdots, p_k, \cdots\}$ une énumération de $\mathcal{P}$.

Nous allons prouver le lemme suivant : supposons qu'il existe une application v de $\{p_1, p_2, \cdots, p_n\}$ dans $\{0,1\}$ telle que tout sous-ensemble fini de Σ ait un modèle dans lequel $p_1, \cdots, p_n$ prennent les valeurs $v(p_1)$, …, $v(p_n)$. Alors on peut étendre v à $\{p_1, p_2, \cdots, p_{n+1}\}$ avec la même propriété.

En effet, si $v(p_{n+1}) = 0$ ne convient pas, alors il existe un ensemble fini U_0 de Σ qui ne peut pas être satisfait quand $p_1, \cdots, p_n, p_{n+1}$ prennent les valeurs respectives $v(p_1)$, …, $v(p_n)$ et 0. Si U est un sous-ensemble fini quelconque de Σ, alors d'après l'hypothèse faite sur v, $U_0 \cup U$ a un modèle dans lequel $p_1, \cdots, p_n$ prennent les valeurs $v(p_1), \cdots, v(p_n)$. Dans ce modèle, la proposition p_{n+1} prend donc la valeur 1. Autrement dit, tout sous-ensemble fini U de Σ a un modèle dans lequel $p_1, \cdots, p_n, p_{n+1}$ prennent les valeurs respectives $v(p_1)$, …, $v(p_n)$ et 1. Dit encore d'une autre façon, soit $v(p_{n+1}) = 0$ convient auquel cas on peut fixer $v(p_{n+1}) = 0$, soit $v(p_{n+1}) = 0$ ne convient pas auquel cas on peut fixer $v(p_{n+1}) = 1$ qui convient.

En utilisant ce lemme, on définit ainsi une valuation v telle que, par récurrence sur n, pour chaque n, tout sous-ensemble fini de Σ a un modèle dans lequel $p_1, \cdots, p_n$ prennent les valeurs $v(p_1)$, …, $v(p_n)$.

Il en résulte que v satisfait Σ : en effet, soit F une formule de Σ. F ne dépend que d'un ensemble fini $p_{i_1}, p_{i_2}, \cdots, p_{i_k}$ de variables propositionnelles (celles qui apparaissent dans F). En considérant $n = \max(i_1, i_2, \cdots, i_k)$, chacune de ces variables

p_{i_j} est parmi $\{p_1, \cdots, p_n\}$. Nous savons alors que le sous ensemble fini $\{F\}$ réduit à la formule F admet un modèle dans lequel $p_1, \cdots, p_n$ prennent les valeurs $v(p_1)$, ..., $v(p_n)$, i.e. F est satisfaite par v.

□

3.10 Exercices

Exercice 3.15. *Relier les propositions équivalentes.*

1. $\neg(p \wedge q)$	a. $(\neg p \wedge \neg q)$
2. $\neg(p \vee q)$	b. $q \rightarrow (\neg p)$
3. $p \rightarrow (\neg q)$	c. $(\neg p \vee \neg q)$
4. $\neg(p \rightarrow q)$	d. $p \wedge (\neg q)$

Exercice 3.16. *En additionnant deux nombres dont l'écriture dans le système binaire utilise au plus deux chiffres, soit ab et cd, on obtient un nombre d'au plus trois chiffres pqr. Par exemple, $11 + 01 = 100$. Donner une expression de p, q et r en fonction de a, b, c et d à l'aide des connecteurs usuels.*

Exercice 3.17 (corrigé page ??). *Soient F et G deux formules n'ayant aucune variable propositionnelle en commun. Montrer que les deux propriétés suivantes sont équivalentes :*

- *La formule $(F \Rightarrow G)$ est une tautologie*
- *L'une au moins des formules $\neg F$ et G est une tautologie.*

***Exercice 3.18.** *[Théorème d'interpolation] Soient F et F' telle que $F \Rightarrow F'$ soit une tautologie. Montrer qu'il existe une formule propositionnelle C, dont les variables propositionnelles apparaissent dans F et F', telle que $F \Rightarrow C$ et $C \Rightarrow F'$ soient deux tautologies. (on pourra raisonner par récurrence sur le nombre de variables qui ont au moins une occurence dans F sans en avoir dans F').*

Exercice 3.19 (corrigé page ??). *[Application de la compacité au coloriage de graphes] Un graphe $G = (V, E)$ est k-coloriable s'il existe une application f de V dans $\{1, 2, \ldots, k\}$ telle que pour tout $(x, y) \in E$, $f(x) \neq f(y)$. Montrer qu'un graphe est k-coloriable si et seulement si chacun de ses sous-graphes finis est k-coloriable.*

***Exercice 3.20.** *[Applications de la compacité à la théorie des groupes] Un groupe G est dit totalement ordonné si on a sur G une relation d'ordre total telle que $a \leq b$ implique $ac \leq bc$ et $ca \leq cb$ pour tous $a, b, c \in G$. Montrer que pour qu'un groupe abélien G puisse être ordonné, il faut et il suffit que tout sous-groupe de G engendré par un ensemble fini d'éléments de G puisse être ordonné.*

3.11 Notes bibliographiques

Lectures conseillées Pour aller plus loin sur les notions évoquées dans ce chapitre, nous suggérons [Cori and Lascar, 1993] et [Lassaigne and de Rougemont, 2004].

Bibliographie Ce chapitre a été rédigé en s'inspirant essentiellement de ces deux ouvrages : [Cori and Lascar, 1993] et [Lassaigne and de Rougemont, 2004].

Index

Bibliographie

[Cori and Lascar, 1993] Cori, R. and Lascar, D. (1993). *Logique mathématique. Volume I*. Mason.

[Lassaigne and de Rougemont, 2004] Lassaigne, R. and de Rougemont, M. (2004). *Logic and complexity*. Discrete Mathematics and Theoretical Computer Science. Springer.