Éléments de Logique Formelle et du Raisonnement Mathématique

Introduction à la logique propositionnelle

Stéphane Devismes

Université de Picardie Jules Verne

30 janvier 2026

Organisation

Séances :

- Cours magistral : 5×2 heures
- Travaux dirigés : 8×2 heures

Support :

- Slides (à trous)
- Sujets de TD
- Annales d'examen, partiel et rattrappage disponibles sur MesCoursJv

Moodle :

`https://mescoursjv.u-picardie.fr/moodle/course/view.php?id=717`

Note

Évaluations

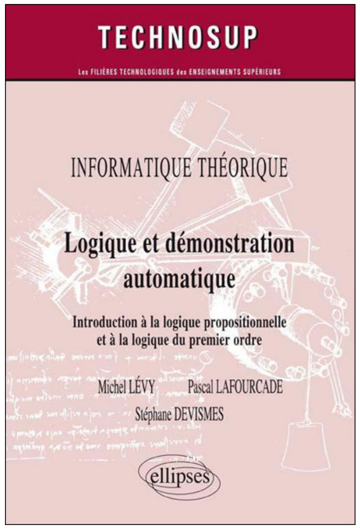
1. Partiel
2. Examen

Pour le partiel et l'examen :

- 1 exo choisi dans un sujet de TD
- 1 feuille RV A4 de notes MANUSCRITES autorisée.

$$\textbf{Note finale} = \max(\textbf{Examen}, \frac{\textbf{Examen+Partiel}}{2})$$

Bibliographie



Plan

Logique

Définition 1.1

- La **logique** précise ce qu'est un raisonnement correct, indépendemment du domaine d'application.
- Un **raisonnement** est un moyen d'obtenir une conclusion à partir d'hypothèses données.
- Un raisonnement **correct** ne dit rien sur la vérité des hypothèses, il dit seulement que **de la vérité des hypothèses, on peut déduire la vérité de la conclusion**.

Exemple

Exemple 1.1

- **Hypothèse I :** Tous les hommes sont mortels
- **Hypothèse II :** Socrate est un homme
- **Conclusion :** Socrate est mortel

Exemple 1.2

- **Hypothèse I :** Tout ce qui est rare est cher
- **Hypothèse II :** Un cheval bon marché est rare
- **Conclusion : Un cheval bon marché est cher !**

Le raisonnement est correct mais l'Hypothèse I de l'exemple 1.2 est fausse !

Applications

- **Hardware** : L'unité arithmétique et logique (UAL) est construite à partir de « portes logiques »
- **Vérification et correction des programmes** :
  - **Meteor** (ligne 14)
  - Outils : prouveurs COQ, PVS, Prover9, MACE, . . .
- **Problème SAT** :
  - Codage de problème de décision sous la forme d'une expression booléenne
  - Applications en planification, model checking, diagnostic, . . .
  - Solveurs : **zchaff**, **satz**, . . .
- **Programmation** : **Prolog** est utilisé dans de nombreux programmes d'intelligence artificielle et dans le traitement de la linguistique par ordinateur
- **Preuves mathématiques**, **Sécurité**, **. . .**

Objectifs du cours

- **Comprendre un raisonnement**, en particulier, être capable de déterminer si un raisonnement logique est correct ou non.
- **Raisonner**, c'est-à-dire, construire un raisonnement correct utilisant les outils de la logique propositionnelle et du premier ordre.
- **Modéliser et formaliser un problème**.
- **Écrire une preuve rigoureuse**.

Plan de ce cours

Logique propositionnelle

Définition 1.2

La **logique propositionnelle** est la logique *sans quantificateurs* qui s'intéresse aux seules lois gouvernant les opérations logiques suivantes :

- $\neg$ (négation),
- $\wedge$ (conjonction, autrement dit le “et”),
- $\vee$ (disjonction, autrement dit le “ou”),
- $\Rightarrow$ (implication) et
- $\Leftrightarrow$ (équivalence).

Remarque 1.1

Nous restreindrons notre étude à la logique **classique**, qui est la logique à deux valeurs de vérité : VRAI et FAUX

Formalisation : exemple

Hypothèses :

- (H1) : Si Pierre est grand, alors Jean n'est pas le fils de Pierre
- (H2) : Si Pierre n'est pas grand, alors Jean est le fils de Pierre
- (H3) : Si Jean est le fils de Pierre alors Marie est la soeur de Jean

Conclusion (C) : Marie est la soeur de Jean ou Pierre est grand ou les deux.

- p : "Pierre est grand"
- j : "Jean est le fils de Pierre"
- m : "Marie est la soeur de Jean"

- (H1) : $p \Rightarrow \neg j$
- (H2) : $\neg p \Rightarrow j$
- (H3) : $j \Rightarrow m$

(C) : $m \vee p$

On peut déduire la conclusion C des hypothèses $H1$, $H2$ et $H2$ si $H1 \wedge H2 \wedge H3 \Rightarrow C$ est toujours vraie, càd

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

est vraie quelque soit la vérité des propositions p, j, m.

Vocabulaire du langage

- **Les constantes :** $\top$ et $\bot$ représentant respectivement le *vrai* et le *faux*.
- **Les variables :** une variable est un identificateur, avec ou sans indice, par exemple x, y_1.
- **Les parenthèses :** ouvrante (et fermante).
- **Les connecteurs :** $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$ respectivement appelés négation, disjonction (ou), conjonction (et), implication et équivalence.

Formule (stricte)

Définition 1.3

Une **formule stricte** est définie de manière inductive par :

- $\top$ et $\bot$ sont des formules strictes.
- Une variable est une formule stricte.
- Si A est une formule stricte alors $\neg A$ est une formule stricte.
- Si A et B sont des formules strictes et si $\circ$ est une des opérations $\vee, \wedge, \Rightarrow, \Leftrightarrow$ alors $(A \circ B)$ est une formule stricte.

Exemple 1.3

$(a \vee (\neg b \wedge c))$ est une formule stricte, mais pas $a \vee (\neg b \wedge c)$, ni $(a \vee (\neg(b) \wedge c))$.

Taille d'une formule

Définition 1.4

La **taille d'une formule** A, notée $|A|$, est définie inductivement par :

- $|\top| = 0$ et $|\bot| = 0$.
- Si A est une variable alors $|A| = 0$.
- $|\neg A| = 1 + |A|$.
- $|(A \circ B)| = |A| + |B| + 1$.

Exemple 1.4

$|(a \vee (\neg b \wedge c))| =$

Arbre

Exemple 1.5

La structure de la formule $(a \vee (\neg b \wedge c))$ est mise en évidence par l'arbre suivant :

Autre exemple

Exemple 1.6

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

Sous-formule

Définition 1.5

Nous appelons **sous-formule** d'une formule (stricte) A tout facteur de A qui est une formule (stricte).

Exemple 1.7

$(\neg b \wedge c)$ est une sous-formule de $(a \vee (\neg b \wedge c))$.

Une sous-formule de la formule A pourra être identifiée comme un sous-arbre de l'arbre représentant la formule A.

Remarque 1.2

Si B est une sous-formule de A telle que $B \neq A$ alors $|B| < |A|$.

Premier résultat

Les formules strictes **se décomposent de manière unique** en leurs sous-formules : d'après la définition de la taille d'une formule (définition 1.4) et la remarque 1.2, **la décomposition donne un schéma de récurrence !**

Théorème 1.1

Pour toute formule A, un et un seul de ces cas se présente :

- A est une variable,
- A est une constante,
- A s'écrit d'une unique façon sous la forme $\neg B$ où B est une formule,
- A s'écrit d'une unique façon sous la forme $(B \circ C)$ où B et C sont des formules.

Preuve (admise) simple mais fastidieuse

Formule à priorité

Définition 1.6

Une **formule à priorité** est définie inductivement par :

- $\top$ et $\bot$ sont des formules à priorité,
- une variable est une formule à priorité,
- si A est une formule à priorité alors $\neg A$ est une formule à priorité,
- si A et B sont des formules à priorité alors $A \circ B$ est une formule à priorité,
- si A est une formule à priorité alors (A) est une formule à priorité.

Exemple 1.8

$a \vee \neg b \wedge c$ est une formule à priorité mais pas une formule (stricte).

Règles de priorité

Définition 1.7

Dans l'ordre des priorités décroissantes : $\neg$, $\wedge$, $\vee$, $\Rightarrow$ et $\Leftrightarrow$.

À priorité égale, le connecteur gauche est prioritaire, **sauf pour l'implication (priorité à droite)**.

Une formule à priorité est **l'abréviation** de la formule (stricte) associé.

Exemple de formule à priorité

Exemple 1.9

- $a \wedge b \wedge c$ est l'abréviation de

- $a \wedge b \vee c$ est l'abréviation de

- $a \vee b \wedge c$ est l'abréviation de

Remarque 1.3

Nous identifions toujours une formule et son abréviation. Autrement dit, ce qui nous intéresse dans une formule, ce n'est pas son écriture superficielle, c'est sa **structure**, qui est mise en évidence par la syntaxe « stricte » : on va donc naturellement écrire en formulation à priorité mais **on raisonnera toujours sur la formule stricte correspondante !**

Tables de base

0 désigne le faux et 1 le vrai.

La constante $\top$ vaut 1 et la constante $\bot$ vaut 0

Dans la suite, nous identifierons les constantes à leurs valeurs et utiliserons indifféremment $\top$ et 1, ainsi que $\bot$ et 0.

Table 1.1 – Table de vérité des connecteurs :

x	y	$\neg x$	$x \vee y$	$x \wedge y$	$x \Rightarrow y$	$x \Leftrightarrow y$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Assignation d'une formule

Définition 1.8

Une **assignation** est une application de l'ensemble de toutes les variables d'une formule dans l'ensemble $\{0, 1\}$. Soit A une formule et v une assignation, $[A]_v$ dénote la valeur de la formule A dans l'assignation v.

Exemple : Soit v une assignation telle que $v(x) = 0$ et $v(y) = 1$

Appliquer l'assignation v à $x \vee y$ s'écrit $[x \vee y]_v$

Cela vaut $0 \vee 1 = 1$

Conclusion : $x \vee y$ est vrai pour l'assignation v

Valeur d'une formule (stricte)

Définition 1.9

Soient A, B des formules, x une variable et v une assignation.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \vee B)]_v = max\{[A]_v, [B]_v\}$
- $[(A \wedge B)]_v = min\{[A]_v, [B]_v\}$
- $[(A \Rightarrow B)]_v =$ si $[A]_v = 0$ alors 1 sinon $[B]_v$
- $[(A \Leftrightarrow B)]_v =$ si $[A]_v = [B]_v$ alors 1 sinon 0

Table de vérité

Définition 1.10

Une **table de vérité** d'une formule A est un tableau qui représente la valeur de A pour toutes les valeurs possibles des variables de A.

- ligne de la table de vérité = une assignation
- colonne = valeur d'une formule.

Exemple :

Exemple 1.10

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1	1	1
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	1	1	0	1	1	1

Autre exemple :

Exemple 1.11

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

Formules équivalentes

Définition 1.11

Deux formules A et B sont **équivalentes** (noté $A \equiv B$ ou simplement $A = B$) si elles ont la même valeur pour toute assignation.

Exemple 1.12

$x \Rightarrow y = \neg x \vee y$

Remarque 1.4

Le connecteur logique $\Leftrightarrow$ ne signifie pas $A \equiv B$.

Valide, tautologie (1/2)

Définition 1.12

- Une formule est **valide** si elle a la valeur 1 pour toute assignation.
- Une formule valide est aussi appelée une **tautologie**.
- Nous notons le fait que A soit valide par $\models A$.

Exemple 1.13

- la formule $(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$ est valide ;
- la formule $x \Rightarrow y$ n'est pas valide car

Valide, tautologie (2/2)

Propriété 1.1

Les formules A et B sont équivalentes si et seulement si la formule $A \Leftrightarrow B$ est valide.

Démonstration.

La propriété est une conséquence de la table 1.1 et des définitions précédentes. □

Modèle d'une formule

Définition 1.13

Une assignation v qui donne la valeur 1 à une formule est un **modèle** de la formule.

v **satisfait** A ou v rend A **vraie**.

Exemple 1.14

Un modèle de $x \Rightarrow y$ est :

Modèle d'un ensemble de formules

Définition 1.14

Une assignation est **un modèle d'un ensemble de formules** si et seulement si elle est un modèle de chaque formule de l'ensemble.

Exemple 1.15

Un modèle de $\{a \Rightarrow b, b \Rightarrow c\}$ est :

Propriété d'un modèle d'un ensemble de formules

Propriété 1.2

Une assignation est un modèle d'un ensemble de formules si et seulement si elle est un modèle de la conjonction des formules de l'ensemble.

Exemple 1.16

L'ensemble de formules $\{a \Rightarrow b, b \Rightarrow c\}$ et la formule $(a \Rightarrow b) \wedge (b \Rightarrow c)$ ont les mêmes modèles.

Contre-modèle

Définition 1.15

Une assignation v qui donne la valeur 0 à une formule est un **contre-modèle** de la formule.

v ne satisfait pas la formule ou v rend la formule **fausse**.

Exemple 1.17

Un contre-modèle de $x \Rightarrow y$ est :

Remarque 1.5

La notion de contre-modèle s'étend aux ensembles de formules de la même manière que la notion de modèle.

Formule satisfaisable

Définition 1.16

Une formule (respectivement un ensemble de formules) est **satisfaisable** s'il existe une assignation qui en est un modèle.

Définition 1.17

Une formule est **insatisfaisable** (resp. un ensemble de formules) si elle (resp. s'il) n'est pas satisfaisable.

Exemple 1.18

Remarque 1.6

Les logiciens utilisent le mot **consistant** comme synonyme de satisfaisable et **contradictoire** comme synonyme d'insatisfaisable.

Conséquence

Définition 1.18

A est **conséquence** de l'ensemble Γ d'hypothèses ($\Gamma \models A$) si tout modèle de Γ est modèle de A.

Remarque 1.7

Nous notons A est valide par $\models A$, car A est valide si et seulement si A est conséquence de l'ensemble vide.

Exemple de Conséquence

Exemple 1.19

$a \Rightarrow b, b \Rightarrow c \models a \Rightarrow c$.

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$
0	0	0	**1**	**1**	**1**
0	0	1	**1**	**1**	**1**
0	1	0	1	0	1
0	1	1	**1**	**1**	**1**
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	**1**	**1**	**1**

Propriété INCONTOURNABLE

Constamment utilisé dans les exercices, EXAMENS.

Propriété 1.3

Soient $n+1$ formules $A_1, \ldots, A_n$, B. Soit H_n la **conjonction** des formules $A_1, \ldots, A_n$. Les trois formulations suivantes sont équivalentes :

1. $A_1, \ldots, A_n \models B$, c'est-à-dire B est conséquence des hypothèses $A_1, \ldots, A_n$.
2. La formule $H_n \Rightarrow B$ est valide.
3. $H_n \wedge \neg B$ est insatisfaisable.

Démonstration.

La propriété est une conséquence de la table 1.1 □

Preuve (1/3)

Preuve (2/3)

Preuve (3/3)

Illustration de la propriété

Exemple 1.20

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$	$(a \Rightarrow b) \wedge (b \Rightarrow c)$ $\Rightarrow (a \Rightarrow c)$	$(a \Rightarrow b) \wedge (b \Rightarrow c)$ $\wedge \neg(a \Rightarrow c)$
0	0	0	**1**	**1**	**1**	**1**	**0**
0	0	1	**1**	**1**	**1**	**1**	**0**
0	1	0	1	0	1	**1**	**0**
0	1	1	**1**	**1**	**1**	**1**	**0**
1	0	0	0	1	0	**1**	**0**
1	0	1	0	1	1	**1**	**0**
1	1	0	1	0	0	**1**	**0**
1	1	1	**1**	**1**	**1**	**1**	**0**

$$a \Rightarrow b, b \Rightarrow c \models a \Rightarrow c$$

La disjonction

- **associative** $x \vee (y \vee z) \equiv (x \vee y) \vee z$
- **commutative** $x \vee y \equiv y \vee x$
- **idempotente** $x \vee x \equiv x$

La conjonction

- **associative** $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$
- **commutative** $x \wedge y \equiv y \wedge x$
- **idempotente** $x \wedge x \equiv x$

Distributivité

- Le produit est distributif sur la somme $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$
- La somme est distributive sur le produit $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$

Neutralité et Absortion

- 0 est l'élément neutre de la disjonction $0 \vee x \equiv x$
- 1 est l'élément neutre de la conjonction $1 \wedge x \equiv x$
- 1 est l'élément absorbant de la disjonction $1 \vee x \equiv 1$
- 0 et l'élément absorbant de la conjonction $0 \wedge x \equiv 0$

Négation

- Les lois de la négation :
  - $x \wedge \neg x \equiv 0$.
  - $x \vee \neg x \equiv 1$ (Le **tiers-exclus**).
- $\neg\neg x \equiv x$.
- $\neg 0 \equiv 1$.
- $\neg 1 \equiv 0$.

Les lois de De Morgan

- $\neg(x \wedge y) \equiv \neg x \vee \neg y$.
- $\neg(x \vee y) \equiv \neg x \wedge \neg y$.

Lois de simplification

Propriété 1.4

Pour tout x, y nous avons :

- $x \vee (x \wedge y) \equiv x$
- $x \wedge (x \vee y) \equiv x$
- $x \vee (\neg x \wedge y) \equiv x \vee y$
- $x \wedge (\neg x \vee y) \equiv x \wedge y$

Démonstration.

La preuve est demandée en TD. □

Conclusion : Aujourd'hui

- Logique propositionnelle
- Syntaxe
- Sens des formules
- Equivalence remarquables

Conclusion : La prochaine fois

- Substitutions et remplacements
- Formes normales
- Algèbre de Boole
- Fonctions booléennes

The more I study, the more I know
The more I know, the more I forget
The more I forget, the less I know