

Introduction à la logique

Benjamin Wack (benjamin.wack@univ-grenoble-alpes.fr)

Notes de cours par

Stéphane Devismes Pascal Lafourcade Michel Lévy

Université Grenoble Alpes

Janvier 2022

Organisation

12 semaines

- ▶ Cours magistral : 1h30 par semaine
- ▶ TD : $2 \times 1h30 = 3h$ par semaine

Cette semaine

- ▶ Cours magistral : aujourd'hui ! et jeudi 13h30 comme d'habitude
- ▶ TDs : 1 TD seulement cette semaine (horaires exceptionnels)
2 à partir de la semaine prochaine

<https://wackb.gricad-pages.univ-grenoble-alpes.fr/inf402/>

Note (sous réserve d'organisation des examens)

Evaluations

- ▶ Contrôle continu **60%** : interros **10%**, partiel **20%** et projet **30%**
- ▶ Examen : **40%**

Projet par groupes de 3 (ou 4).

- ▶ Phase 1 : Modélisation d'un problème en logique (avec interface logicielle)
- ▶ Phase 2 : Utilisation d'un solveur SAT pour résoudre ce problème
- ▶ Optionnel : Programmation d'un solveur SAT

Exemples de problèmes : N reines, grilles de type Sudoku...

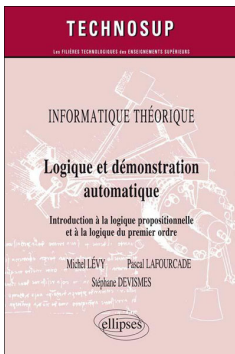
Planning

Dates importantes

- ▶ **Pré-rapport projet** : 11 mars
- ▶ **Partiels** : 14-18 mars
- ▶ **Rapport de Projet** : 29 avril
- ▶ **Soutenance de Projet** : 2-13 mai
- ▶ **Examen** : 16-25 mai

Matériel

- ▶ Poly + version pdf en ligne
- ▶ Sujet du projet + ressources en ligne



Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules (sémantique)

Conclusion

Logique

Définitions

- ▶ La **logique** précise ce qu'est un raisonnement correct, indépendamment 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.**

Exemples

Exemple I

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

Exemple II

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

Ajout d'une hypothèse

Exemple III

- ▶ **Hypothèse I** : Tout ce qui est rare est cher
- ▶ **Hypothèse II** : Un cheval bon marché est rare
- ▶ **Hypothèse III** : Tout ce qui est bon marché n'est pas cher
- ▶ **Conclusion** : Hypothèses contradictoires ! Car :
 - ▶ **Hypothèse I + hypothèse II** : Un cheval bon marché est cher
 - ▶ **Hypothèse III** : Un cheval bon marché n'est pas cher

Applications

- ▶ **Hardware** (portes logiques)
- ▶ **Vérification et correction des programmes, sécurité :**
 - ▶ prouveurs COQ, PVS, Prover9, MACE, ...
 - ▶ applications industrielles (Meteor, Airbus...)
- ▶ **Intelligence artificielle :**
 - ▶ système expert (*MyCin*), ontologie
- ▶ **Programmation :** Prolog
 - ▶ intelligence artificielle
 - ▶ traitement de la langue
- ▶ **Vérification de preuves mathématiques**

Objectifs du cours

- ▶ **Modéliser et formaliser un problème** décrit en langage naturel.
- ▶ **Comprendre un raisonnement** présenté sous forme symbolique, en particulier être capable de déterminer s'il est correct.
- ▶ **Démontrer**, c'est-à-dire construire un raisonnement correct utilisant les règles et/ou les algorithmes de la logique propositionnelle et du premier ordre.
- ▶ **Écrire une preuve rigoureuse**, en particulier par récurrence.

Plan du Semestre

AUJOURD'HUI

- ▶ Logique propositionnelle
- ▶ Résolution propositionnelle
- ▶ Dédution naturelle propositionnelle

PARTIEL

- ▶ Logique du premier ordre
- ▶ Base de la démonstration automatique
(« résolution au premier ordre »)
- ▶ Dédution naturelle au premier ordre

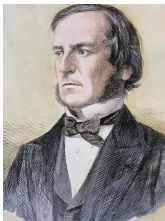
EXAMEN

Notion de logique formelle

Idee déjà formulée par Leibniz (voire par les Stoïciens) : pour systématiser le raisonnement, il faut formaliser : les énoncés puis le raisonnement lui-même.

George Boole (1854) suit une approche calculatoire :

- ▶ traduire les énoncés par des expressions algébriques
- ▶ leur appliquer des règles de calcul qui traduisent le raisonnement humain
- ▶ et réinterpréter le résultat



La proposition de Boole donne le véritable départ de la logique mathématique, et en particulier de la logique propositionnelle.

Logique propositionnelle

Définition

La **logique propositionnelle** est la logique *sans quantificateurs*.

Seules opérations logiques considérées :

- ▶ \neg (négation)
- ▶ \wedge (conjonction “et”)
- ▶ \vee (disjonction “ou”)
- ▶ \Rightarrow (implication)
- ▶ \Leftrightarrow (équivalence)

Exemple : Raisonnement formel

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.

- | | |
|--------------------------------------|---------------------------------|
| ▶ p : "Pierre est grand" | ▶ (H1) : $p \Rightarrow \neg j$ |
| ▶ j : "Jean est le fils de Pierre" | ▶ (H2) : $\neg p \Rightarrow j$ |
| ▶ m : "Marie est la soeur de Jean" | ▶ (H3) : $j \Rightarrow m$ |
| | ▶ (C) : $m \vee p$ |

Il s'agira de montrer que $H1 \wedge H2 \wedge H3 \Rightarrow C$:

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

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

Vocabulaire du langage

- ▶ Les constantes : \top (vrai) et \perp (faux)
- ▶ Les variables : par exemple x, y_1
- ▶ Les parenthèses
- ▶ Les connecteurs : $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$

Formule (stricte)

Définition 1.1.1

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

- ▶ \top et \perp 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.

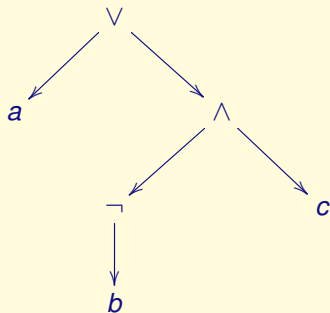
Exemple 1.1.2

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

Arbre

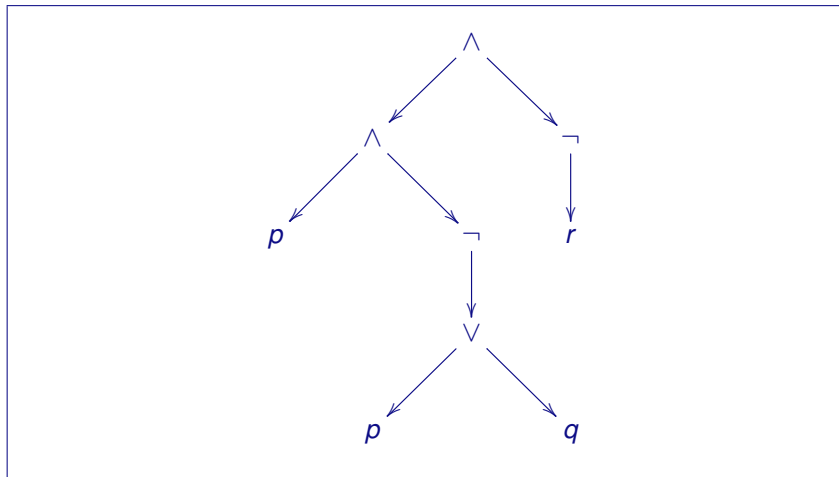
Exemple 1.1.3

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



Exercice

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



Taille d'une formule

Définition 1.1.10

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

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

Exemple 1.1.11

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

3.

Premier résultat

Les formules strictes **se décomposent de manière unique** en leurs sous-formules.

Théorème 1.1.13

Toute formule A s'écrit sous une et une seule de ces formes :

- ▶ une variable,
- ▶ une constante,
- ▶ $\neg B$ où B est une formule,
- ▶ $(B \circ C)$ où B et C sont des formules et \circ un connecteur binaire.

Ce qui autorise :

- ▶ la démonstration *par cas*
- ▶ la récurrence sur la *structure* des formules et non sur leur taille.

Formule à priorité

Définition 1.1.14

Idem mais :

- ▶ 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 1.1.15

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

Règles de priorité

Définition 1.1.16

Par ordre de priorités décroissantes : \neg , \wedge , \vee , \Rightarrow et \Leftrightarrow .

Associativité à gauche

Pour deux connecteurs identiques $A \circ B \circ C = (A \circ B) \circ C$

sauf pour l'implication : $A \Rightarrow B \Rightarrow C = A \Rightarrow (B \Rightarrow C)$

Exemples de formules à priorité

Exemple 1.1.17

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

$$((a \wedge b) \wedge c)$$

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

$$((a \wedge b) \vee c)$$

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

$$(a \vee (b \wedge c))$$

Assignation d'une formule

Définition 1.2.1

Une **assignation** est une fonction qui, à chaque variable d'une formule, associe une valeur dans $\{0, 1\}$.

$[A]_v$ 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 v à $x \vee y$ s'écrit $[x \vee y]_v$

$$[x \vee y]_v = 0 \vee 1 = 1$$

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

Valeur d'une formule

Définition 1.2.2

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

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = 1, [\perp]_v = 0$
- ▶ $[\neg A]_v = 1 - [A]_v$
- ▶ $[(A \vee B)]_v = \max\{[A]_v, [B]_v\}$
- ▶ $[(A \wedge B)]_v = \min\{[A]_v, [B]_v\}$
- ▶ $[(A \Rightarrow B)]_v =$ 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 1.2.3

Une **table de vérité** donne la valeur d'une formule pour **chaque** choix de valeurs des variables de A .

- ▶ une ligne de la table de vérité = une assignation
- ▶ une colonne = toutes les valeurs d'une formule.

Tables de base

On associe à chaque formule une *valeur* : 0 (faux) ou 1 (vrai).
La constante \top vaut 1 et la constante \perp vaut 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

Exemple :

Exemple 1.2.4

Donner la table de vérité des formules suivantes.

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

Formules équivalentes

Définition 1.2.5

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 1.2.6

$$x \Rightarrow y \equiv \neg x \vee y$$

Remarque :

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

Valide, tautologie (1/2)

Définition 1.2.8

- ▶ Une formule est **valide** si elle a la valeur 1 pour toute assignation.
- ▶ Aussi appelée une **tautologie**.
- ▶ Noté $\models A$.

Exemple 1.2.9

- ▶ $(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$ est valide ;
- ▶ $x \Rightarrow y$ n'est pas valide car
elle est fausse pour $x = 1$ et $y = 0$.

Valide, tautologie (2/2)

Propriété 1.2.10

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

Cf table de vérité de \Leftrightarrow .

Modèle d'une formule

Définition 1.2.11

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

On dit aussi que v **satisfait** A ou v rend A **vraie**.

Exemple 1.2.12

Un modèle de $x \Rightarrow y$ est :

$x = 1, y = 1$ (il y en a d'autres).

Par contre $x = 1, y = 0$ n'est pas un modèle de $x \Rightarrow y$.

Modèle d'un ensemble de formules

Définition 1.2.13

v est un **modèle de l'ensemble** $\{A_1, \dots, A_n\}$
si et seulement si
elle est un modèle de chacune de ces formules.

Exemple 1.2.14

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

$a = 0, b = 0$ (et c quelconque).

Propriété d'un modèle d'un ensemble de formules

Propriété 1.2.15

v est un modèle de $\{A_1, \dots, A_n\}$
si et seulement si
 v est un modèle de $A_1 \wedge \dots \wedge A_n$.

Exemple 1.2.16

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 1.2.17

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

On dit que v **ne satisfait pas** A ou que v rend la formule **fausse**.

Exemple 1.2.18

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

$$x = 1, y = 0.$$

Formule satisfaisable

Définition 1.2.20

Un (ensemble de) formule(s) est **satisfaisable** s'il admet un modèle.

Définition 1.2.21

Un (ensemble de) formule(s) est **insatisfaisable** s'il n'est pas satisfaisable.

Exemple 1.2.22

$x \wedge \neg x$ est insatisfaisable, mais $x \Rightarrow y$ est satisfaisable.

Attention

insatisfaisable = 0 modèle

satisfaisable = 1 modèle ou plus

invalides = 1 contre-modèle ou plus

valides = 0 contre-modèle

Aujourd'hui

- ▶ Pourquoi définir et utiliser la logique **formelle** ?
- ▶ Formules de logique propositionnelle :
 - ▶ **1 variable = 1 proposition** (une information) vraie ou fausse
 - ▶ 5 connecteurs pour articuler ces propositions
- ▶ Sens des formules :
 - ▶ **assignment** = choix d'une valeur de vérité pour chaque variable
 - ▶ une formule peut être vraie pour **0, 1, plusieurs ou toutes** les assignments

La prochaine fois

Exercice : étudier “Pierre, Jean et Marie” à l’aide des tables de vérité.

- ▶ Équivalences remarquables
- ▶ Substitutions et remplacements
- ▶ Formes normales

Devise d'Oxford

Plus il y a de gruyère, plus il y a de trous
Plus il y a de trous, moins il y a de gruyère
Donc : plus il y a de gruyère, moins il y a de gruyère