

Plan du cours

Logique L3 Informatique

Peter Habermehl

Université Paris Diderot, Seine Paris Cité



- 1 Rappels :
 - ▶ **Induction** : ordres bien fondés, définitions inductives, principe d'induction bien fondée, preuves par induction.
 - ▶ **Calcul propositionnel** : syntaxe, sémantique, tables de vérité.
- 2 Systèmes de preuves syntaxiques pour le calcul propositionnel :
 - ▶ Hilbert.
 - ▶ Dédution naturelle.
 - ▶ Gentzen.
 - ▶ Correction et complétude.
- 3 Calcul des prédicats :
 - ▶ Syntaxe, sémantique.
 - ▶ Unification et résolution.
 - ▶ Théorèmes.

Modalités du cours

Chargés de TD :

- ▶ Alexis Goyet, lundi 13h-15h, salle 4/31F.
- ▶ Arnaud Sangnier, mardi 14h-16h, salle 4/01E.
- ▶ Stéphane Zimmermann, jeudi 14h30 - 16h30, salle 4/01E.

La première session : $\frac{1}{2}$ note partielle + $\frac{1}{2}$ examen.
La deuxième session rattrapage : $\frac{1}{2}$ note partielle + $\frac{1}{2}$ examen.
Après la deuxième session rattrapage, les étudiants ont le droit à la consultation de deux feuilles blanches et strictement personnelles. Tous les autres documents sont interdits.

- Notes finales
- Notes de Maxam rattrapage)
- Pendre sur le mur de la salle de travail : A4 recto-verso : les autres documents ne seront pas acceptés.

Documents du cours

- Transparents du cours
<http://www.liafa.jussieu.fr/~haberm/cours/logique/>
- Tableau (exemples et démonstrations)

Bibliographie

- Logique pour l'info. : introduction à la déduction automatique.
S. Cerrito, VUIBERT.
- Mathématiques pour l'informatique.
A. Arnold et I. Guessarian, MASSON.
- Introduction à la logique.

• Logique et fondements de l'informatique
R. Lassaigne et M. Rougemont, MASSON

• First-Order Logic and Automated Theorem Proving.
M. Fitting, SPRINGER.

• Concrete Mathematics.
R. L. Graham, D. E. Knuth et J. P. Bentley, ADDISON-WESLEY.

Bibliographie

- Logic for Computer Science.
J. Gallier, WILEY. Disponible en ligne :
<http://www.cis.upenn.edu/~jean/gbooks/logic.html>
- Logicomics.
A. Doxiadis, C. Papadimitriou, A. Papadatos, A. Di Donna, VUIBERT.

Ensembles

Notions préliminaires

- l'intersection de A et B est $A \cap B$
- l'union de A et B est $A \cup B$
- la différence de A et B est $A \setminus B$
- le complémentaire de A est $\overline{A} = \mathcal{U} \setminus A$
- $\mathcal{P}(A)$ est l'ensemble des parties de A

Définition. Soient deux ensembles A, B inclus dans \mathcal{U} (Univers).

l'intersection de A et B est $A \cap B = \{e \in \mathcal{U} \mid e \in A \text{ et } e \in B\}$

l'union de A et B est $A \cup B = \{e \in \mathcal{U} \mid e \in A \text{ ou } e \in B\}$

la différence de A et B est $A \setminus B = \{e \in \mathcal{U} \mid e \in A \text{ et } e \notin B\}$

le complémentaire de A est $\overline{A} = \mathcal{U} \setminus A = \{e \in \mathcal{U} \mid e \notin A\}$

$\mathcal{P}(A)$ est l'ensemble des parties (sous-ensembles) de A

Ensembles



(Lois de de Morgan)
 $\overline{A \cup B} = \overline{A} \cap \overline{B}$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

Définition :

- L'ensemble $\mathcal{A}_1 \times \dots \times \mathcal{A}_n$ est l'ensemble de n -uplets $\mathcal{A}_1 \times \dots \times \mathcal{A}_n : \{(a_1, \dots, a_n) \mid \exists \mathcal{A}_i\}$. Si $\mathcal{A}_i = \mathcal{A}$ pour tout i , on note \mathcal{A}^n le produit $\mathcal{A}_1 \times \dots \times \mathcal{A}_n$.
- Un ensemble A est appelé **dénombrable** sseulement si il existe une fonction injective (voir définition plus tard) de \mathbb{N} vers A .

Relations

Définition : Une **relation n -aire** sur $\mathcal{A}_1 \dots \mathcal{A}_n$ est un sous-ensemble de $\mathcal{A}_1 \times \dots \times \mathcal{A}_n$.

Définition : Soit $R \subseteq \mathcal{A} \times \mathcal{A}$ une relation **binaire**.

- R est **réflexive** ssi pour tout $x \in \mathcal{A}$, $(x, x) \in R$.
 R est **irréflexive** ssi pour tout $x \in \mathcal{A}$, $(x, x) \notin R$.
- R est **symétrique** si pour tout $x, y \in \mathcal{A}$, $(x, y) \in R$ implique $(y, x) \in R$.
 R est **anti-symétrique** si pour tout $x, y \in \mathcal{A}$, $(x, y) \in R$ et $(y, x) \in R$ implique $x = y$.
- R est **transitive** si pour tout $x, y, z \in \mathcal{A}$, $(x, y) \in R$ et $(y, z) \in R$ implique $(x, z) \in R$.

Notations

- $(x, y) \in R$ peut s'écrire aussi $x R y$.
- On peut utiliser un symbole à la place de R .
 Par exemple, si \leq est une relation, alors $x \leq y$ s'écrit $x \leq y$.
 - $y \geq x$ lorsque $x \leq y$.

Ainsi
 (x, y)
 On é

Exemples

Exemple : La relation \geq sur les entiers naturels est réflexive, la relation $>$ sur les entiers naturels est irréflexive.

Exemple : La relation $=$ sur les ensembles est symétrique, la relation \supseteq sur les entiers naturels est anti-symétrique.

sur les ensembles est transitive.

Équivalence et Congruence

Définition :

- R est une **équivalence** si elle est réflexive, symétrique et transitive.

Exercice : Montrer que $\sim = \{(x, y) \mid 3 \text{ est diviseur de } x - y\}$ est une équivalence.

Exercice : R est une **congruence** p.r. à une fonction f si f est une équivalence compatible avec f , c'est-à-dire, si $b_1 \dots a_n R b_n$ implique $f(a_1, \dots, a_n) R f(b_1 \dots b_n)$.
Exercice : Montrer que $\sim = \{(x, y) \mid 3 \text{ est diviseur de } x - y\}$ est une congruence par rapport à $+$ et à $*$.

Classes d'équivalence

La **classe d'équivalence** de $a \in \mathcal{A}$ par rapport à une équivalence R est l'ensemble $[a]_R = \{b \in \mathcal{A} \mid aRb\}$.

Composition de relations

Définition : Si $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{B}$ et $\mathcal{S} \subseteq \mathcal{B} \times \mathcal{C}$, alors la **composition** de \mathcal{S} avec \mathcal{R} est une relation dans $\mathcal{A} \times \mathcal{C}$ t.q.

$\mathcal{S} \circ \mathcal{R} = \{(a, c) \mid \exists b \in \mathcal{B} : (a, b) \in \mathcal{R} \text{ et } (b, c) \in \mathcal{S}\}$.

Exercice : Montrer que la composition de \mathcal{R} avec elle-même est définie par la même relation.

$\mathcal{R}^0 = \{(a, a) \mid a \in \mathcal{A}\}$
 $\mathcal{R}^n \circ \mathcal{R} = \mathcal{R} \circ \mathcal{R}^n = \mathcal{R}^{n+1}$

\mathcal{R}^0
 $\mathcal{R}^n \circ \dots \circ \mathcal{R}$
 $n+1 \text{ fois}$

$\{Paris, Lyon, Toulouse\}$ et
 $\mathcal{R} = \{(Paris, Lyon), (Lyon, Paris), (Lyon, Lyon), (Toulouse, Toulouse), (Lyon, Toulouse)\}$.

Exemple : Soit $A = \{Paris, Lyon, Toulouse\}$
 $\mathcal{R} = \{(Paris, Lyon), (Lyon, Paris), (Lyon, Lyon), (Toulouse, Toulouse)\}$
 $\mathcal{R}^2 = \{(Paris, Paris), (Lyon, Toulouse), (Lyon, Lyon), (Toulouse, Toulouse)\}$
 Calculer \mathcal{R}^3 .

Les clôtures

Définition : La **clôture transitive** d'une relation \mathcal{R} est donnée par

$$\mathcal{R}^+ = \bigcup_{n=1}^{\infty} \mathcal{R}^n$$

La **clôture réflexive et transitive** d'une relation \mathcal{R} est donnée par

$$\mathcal{R}^* = \bigcup_{n=0}^{\infty} \mathcal{R}^n = \mathcal{R}^+ \cup \mathcal{R}^0$$

Exemple : Dans l'exemple d'avant, $\mathcal{R}^* = A \times A$.

Fonctions

Définition : Une fonction f entre deux ensembles \mathcal{A} et \mathcal{B} , notée $f : \mathcal{A} \rightarrow \mathcal{B}$, est une relation sur $\mathcal{A} \times \mathcal{B}$ t.q. pour tout x, y, z si $(x, y) \in f$ et $(x, z) \in f$, alors $y = z$.

Notation : On écrit $f(x)$ pour dénoter l'unique élément y t.q. $(x, y) \in f$ et $f(\mathcal{C}) = \{y \in \mathcal{B} \mid \exists x \in \mathcal{C}, f(x) = y\}$.

On note $id_{\mathcal{A}}$ la fonction identité sur \mathcal{A} donnée par $id_{\mathcal{A}}(x) = x$.

Définition : Soit $f : \mathcal{A} \rightarrow \mathcal{B}$ une fonction.

- Le domaine de f est $Dom(f) = \{x \in \mathcal{A} \mid \exists y \in \mathcal{B}, (x, y) \in f\}$
- L'image de f est $Im(f) = \{y \in \mathcal{B} \mid \exists x \in \mathcal{A}, (x, y) \in f\}$
- L'inverse (pas toujours une fonction) de f est $f^{-1} = \{(y, x) \in \mathcal{B} \times \mathcal{A} \mid (x, y) \in f\}$

Composition de fonctions

Définition :

- La composition de $f : \mathcal{B} \rightarrow \mathcal{C}$ avec $g : \mathcal{A} \rightarrow \mathcal{B}$ est la fonction $f \circ g : \mathcal{A} \rightarrow \mathcal{C}$, où $f \circ g(x) = f(g(x))$.

Exemple : $f(x) = x^2$, $g(x) = x + 4$, $f \circ g(x) = (x + 4)^2$,
 $g \circ f(x) = x^2 + 4$.

- La n -composition de f avec elle-même, notée f^n , est défini par récurrence sur n :

- Si $n = 0$, alors $f^0 = id$
- Si $n > 0$, alors $f^n = f \circ f^{n-1}$

Exemple : $f(x) = x + 2$, $f^0(x) = x$, $f^1(x) = x + 2$, $f^2(x) = x + 4$,
 $f^3(x) = x + 6$, \dots , $f^n(x) = x + 2.n$.

Exercice : Soit $n > 0$. Montrer que $f^n = f^{n-1} \circ f$.

Propriétés des fonctions

Définition : Une fonction $f : \mathcal{A} \rightarrow \mathcal{B}$ est injective ssi pour tout $x, y \in \mathcal{A}$, $f(x) = f(y)$ implique $x = y$.

Exemple : $f(x) = x + 2$ sur les entiers est injective. L'ensemble d'entiers n'est pas injective. Ainsi $f(\{2, 3, 4\}) = \{4, 5, 6\}$ mais $\{2, 3, 4\} \neq \{2, 4\}$.

Définition : Une fonction $f : \mathcal{A} \rightarrow \mathcal{B}$ est surjective ssi pour tout $y \in \mathcal{B}$ il existe $x \in \mathcal{A}$ tel que $f(x) = y$.

Exemple : $f(x) = x \text{ div } 2$ sur les entiers naturels est surjective.
 $f(x) = x + 2$ sur les entiers naturels n'est pas surjective.

Définition : Une fonction est bijective ssi elle est injective et surjective.

Exemple : Soit \mathcal{A} l'ensemble de mots de longueur 3 contenant uniquement 0 et 1. Soit $\mathcal{B} = \{0 \dots 7\}$. Soit $f("b_2 b_1 b_0") = b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$. Cette fonction est injective et surjective, donc bijective.

Préordres, ordres

Définition :

Un préordre est une relation réflexive et transitive.

Exemple : \leq sur \mathbb{N} est un préordre. $(2, 2), (3, 3), (4, 4), (3, 2), (2, 3), (2, 4), (3, 4)$.

Un ordre ou ordrediel est une relation réflexive, anti-symétrique et transitive.

Notation : \geq .

Exemple : \mathcal{R} n'est pas un ordre car $(3, 2), (2, 3)$ mais $2 \neq 3$.

$\mathcal{S} = \{(2, 2), (3, 3), (4, 4), (2, 3), (2, 4), (3, 4)\}$ est un ordre.

Ordre stricte

Définition : Un **ordre strict** est une relation irréflé-

Notation : $>$

Exemple : $>$ sur les entiers, \supset sur les ensembles.

Définition : Un ordre strict est **bien fondé** ssi il n'y a pas de suite infinie décroissante (i.e., de la forme $a_0 > a_1 > a_2 > \dots$).

Exemple : $>$ sur les entiers naturels est bien fondé. \supset sur les ensembles finis est bien fondé.

Majorants/minorants et bornes supérieures/inférieures

Soit \mathcal{E} un ensemble muni d'un ordre \leq . Soit $\mathcal{A} \subseteq \mathcal{E}$.

Définition :

Majorant : $x \in \mathcal{E}$ t.q. pour tout $y \in \mathcal{A}$, $y \leq x$.

Minorant : $x \in \mathcal{E}$ t.q. pour tout $y \in \mathcal{A}$, $x \leq y$.

Borne supérieure : $\sup(\mathcal{A})$

notée $\sup(\mathcal{A})$, est le plus petit des majorants de \mathcal{A} alors $\sup(\mathcal{A}) \leq z$.

Borne inférieure : $\inf(\mathcal{A})$, est le plus grand des minorants de \mathcal{A} alors $z \leq \inf(\mathcal{A})$.

Exemple : $\{1, 2, \dots, 10\}$. Tous les entiers dans $\{10, \dots\}$ sont des bornes supérieures.

Exemple : $\{1, 2, \dots, 10\}$. Tous les entiers dans $\{10, \dots\}$ sont des bornes supérieures.

Exemple : $\{1, 2, \dots, 10\}$. Tous les entiers dans $\{10, \dots\}$ sont des bornes supérieures.

Définitions inductives en informatique

Définitions Inductives et preuves par induction

- Syntaxe concrète
- Syntaxe abstraite
- Règles de typage
- Règles d'évaluation

Le principe

Une définition inductive est caractérisée par :

- Une ou plusieurs **assertions**
- Un ensemble de **règles** d'inférence pour dériver ces assertions

Exemple :

• Règles d'inférence :

R1 : 0 est naturel.

R2 : Si n est naturel, (n) est naturel.

• En général $n \geq 0$. Si

Notation

Les règles d'inférence sont notées

$$\frac{\text{Hypothèse}_1 \dots \text{Hypothèse}_n}{\text{Conclusion}} \text{ (Nom de la règle)}$$

- Conclusion est une assertion

thèse_n sont des assertions

$n = 0$ la règle est un **axiome**

Exemple (règle unaire)

Les entiers naturels

$$\frac{}{0 \text{ est naturel}} \text{ (Nat0)} \quad \frac{n \text{ est naturel}}{\text{succ}(n) \text{ est naturel}}$$

est un arbre binaire:
 $\text{node}(A_1, A_2)$

Exemple (règle binaire)

$$\frac{}{\text{un arbre binaire}} \text{ (Abin-nil)}$$

A_1
bin-ind)

est un arbre binaire
 A_2 est un arbre binaire
est un arbre binaire

Exemple

Les mots sur un alphabet A

$$\frac{}{\epsilon \text{ mot}} \quad \frac{a \in A \quad n \text{ mot}}{a.n \text{ mot}} \quad \frac{A_1 \text{ expr} \quad t}{A_1 \vee A_2 \in} \quad \frac{A_1 \text{ expr}}{A_1 \rightarrow A_2 \in}$$

Exemple (plusieurs axiomes, règles unaires et binaires)

Les expressions de la logique propositionnelle sur l'alphabet A

$$\frac{p \in A}{p \text{ expr}} \quad \frac{A_2 \text{ expr}}{\text{expr}} \quad \frac{A_1 \text{ expr} \quad A_2 \text{ expr}}{A_1 \wedge A_2 \text{ expr}} \quad \frac{A_2 \text{ expr}}{\text{expr}} \quad \frac{A \text{ expr}}{\neg A \text{ expr}}$$

Exemple (plusieurs assertions)

Les forêts de type T

$$\frac{}{f \text{ vide} \in} \quad \frac{}{f \text{ vide} \in \text{arbre } T} \quad \frac{}{\exists \text{ foret } T} \quad \frac{t \in T \quad f \in \text{foret } T \quad A \in}{f \in \text{foret } T} \quad \frac{}{(\text{arbre } T, \text{foret } T, A) \in \text{foret } T}$$

Dérivation d'une assertion

Une assertion A est **dérivable** ssi

- A est un axiome

$$\frac{}{A}$$

- ou il y a une règle de la forme

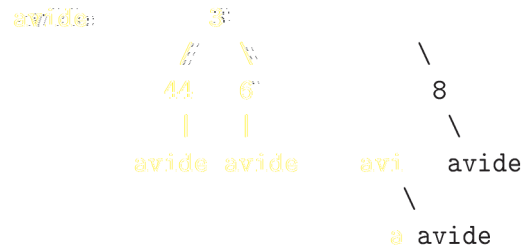
$$\frac{A_1 \quad \dots \quad A_n}{A}$$

telle que A_1, \dots, A_n sont dérivables

Dérivation d'une assertion

Exercice :

- Donner le terme qui dénote la forêt suivante et montrer comment la construire avec les règles précédentes :



Ensemble inductif

Un ensemble inductif est le **plus petit** ensemble engendré par un système de règles d'inférence.

Preuves par Induction

- Induction sur les entiers
 - ▶ Induction mathématique
 - ▶ Induction complète
 - ▶ Équivalence
- Induction bien fondée
- Induction structurale
- Induction sur un ensemble inductif

Théorème : Soit P une propriété sur les entiers. Supposons

(IM1) $P(0)$,

(IM2) pour tout $n \in \mathbb{N}$ on a $P(n)$ implique $P(n+1)$

Formellement, $\forall n \in \mathbb{N}. P(n) \rightarrow P(n+1)$,

alors pour tout $n \in \mathbb{N}$ on a $P(n)$ (formellement $\forall n \in \mathbb{N}, P(n)$)

$$1) \quad \sum_{i=1}^n i = \frac{n * (n + 1)}{2} \qquad 2) \quad n^2 = \sum_{i=1}^n (2i - 1)$$

Mais comment prouver

1 "Tout entier est décomposable en produit de facteurs premiers" \Rightarrow si n est impair, n est divisible par 3, alors $fib(n)$ est pair,

Induction sur les entiers II (induction complète)

Théorème : Soit P une propriété sur les entiers. Supposons

(IC1) $P(0)$,

$$(IC2) \quad \forall n \in \mathbf{N}. ((\forall k \in \mathbf{N}. k < n \rightarrow P(k)) \rightarrow P(n))$$

alors $\forall n \in \mathbf{N}. P(n)$

Équivalence des deux principes

Malgré l'apparente supériorité du deuxième principe, on prouve

Théorème : Induction mathématique et complète sont équivalentes.

Un théorème fondamental

Théorème : Tous les français sont d'accord avec le Président de la République.

Preuve : On montre, par induction sur le nombre de français, que tout groupe de n personnes contenant le Président est d'accord avec lui.

Cas de base : il y a seulement le Président, trivial.

Cas inductif : on suppose l'énoncé vrai pour tout groupe de n personnes, et on le prouve pour tout groupe de $n + 1$.

Numérotons de 1 à $n + 1$ les personnes en question, de façon que le

Président soit le numéro n , et considérons le groupe A des premières n personnes et le groupe B des dernières n personnes.

Les deux groupes contiennent le Président et on peut donc appliquer l'hypothèse d'induction et en déduire qu'ils sont tous d'accord avec le Président (qui est dans les deux) conclure.

vrai ou faux ?

Principe d'induction bien fondée

Un ensemble \mathcal{A} , un ordre strict $>$ et une propriété P sur \mathcal{A}

Principe d'induction :

Si

- 1 "pour tout élément minimal $y \in \mathcal{A}$ on a $P(y)$ "
- 2 "le fait que $P(z)$ soit vérifiée pour tout élément $z < x$ implique $P(x)$ "

alors
"pour tout $x \in \mathcal{A}$ on a $P(x)$ "

On a $P(x)$ pour tout x de taille $n < n + 1$. On déduit qu'ils sont tous d'accord avec le Président, ce qui nous permet de conclure.

Ce principe est-il toujours bien défini ?

Soit $>$ un ordre strict.

Théorème :

Si $>$ est **bien fondé**, alors le principe d'induction est correct.

Théorème :

Si le principe d'induction est correct, alors $>$ est **bien fondé**.

Corollaire : Le principe d'induction est correct pour les ensembles inductifs.

Corollaire : Le principe d'induction structurelle est correct.

Exemples

- **Les mots :**

$P(m)$ est la propriété :

$\text{concat}(\text{concat}(m, v_1), v_2) = \text{concat}(m, \text{concat}(v_1, v_2))$

- **Les arbres binaires :**

$P(a)$ est la propriété : $\text{feuilles}(a) = \text{noeuds_internes}(a) + 1$

Induction sur quelques ordres bien fondés

- Ordre lexicographique
- Ordre multi-ensemble
- Combinaisons

Ordres lexicographiques

Soit $>_{A_i}$ un ordre strict sur l'ensemble \mathcal{A}_i .

Ordre lexicographique sur le produit de 2 ensembles :

$$(x, y) >_{lex} (x', y') \text{ ssi } (x >_{A_1} x') \text{ ou } (x = x' \text{ et } y >_{A_2} y')$$

Exemple :

$$(4, "abc") >_{lex} (3, "abc") >_{lex} (2, "abcde") >_{lex} (2, "bcde") >_{lex} (2, "e") >_{lex} (1, "e") >_{lex} (0, \epsilon)$$

Ordre lexicographique sur le produit de n ensembles

Si chaque $>_{A_i}$ est un ordre strict sur l'ensemble \mathcal{A}_i , alors $>_{lex}$ est un ordre strict qui permet de comparer deux n -uplets de la manière suivante :

$$(x_1, \dots, x_n) >_{lex} (x'_1, \dots, x'_n) \text{ ssi } \exists 1 \leq j \leq n \text{ such that } (x_j >_{A_j} x'_j \text{ and } \forall 1 \leq i < j \ x_i = x'_i)$$

Théorème : Si chaque $>_{A_i}$ est un ordre strict bien fondé sur \mathcal{A}_i , alors l'ordre lexicographique $>_{lex}$ sur le produit de $\mathcal{A}_1 \times \dots \times \mathcal{A}_n$ est un ordre

Avertissement : $>_{lex}$ n'est pas l'ordre du dictionnaire !

Exemple : la fonction d'Ackermann



Montrer par induction que la fonction suivante termine.

$$\begin{aligned} \text{Ackermann}(m, n) &= n+1 \\ \text{Ackermann}(m-1, 0) &= \text{Ackermann}(m, 1) \\ \text{Ackermann}(m-1, n) &= \text{Ackermann}(m, \text{Ackermann}(m-1, n-1)) \end{aligned}$$

Définition : Soit A un ensemble. Un **multi-ensemble** de base A est une fonction $M : A \rightarrow \mathbb{N}$ telle que $M(x) > 0$ seulement pour un nombre fini d'éléments x de A .

Notation : $\{a, a, b\}$.

Définition : $M \succ_{mul} N$ ssi N s'obtient à partir de M en appliquant la règle suivante un nombre fini de fois : enlever un élément x de M et le remplacer par un nombre fini d'éléments plus petits que x (par rapport à l'ordre $>$).

Notation : $\{5, 3, 1, 1\}$

Exemple : $\{5, 3, 1, 1\} \succ_{mul} \{4, 3, 3, 1\}$.
Car $\{5, 3, 1, 1\} \succ_{mul} \{4, 3, 3, 1, 1\} \succ_{mul} \{4, 3, 3, 1\}$

Théorème : Si $>_A$ est un ordre strict bien fondé sur A , alors \succ_{mul} est un ordre strict bien fondé sur les multi-ensembles de base A .

Exemple

Un homme possède une somme d'argent en euros. Chaque jour il procède de la façon suivante :

- soit il jette une pièce de monnaie dans une fontaine,
- ou bien il change l'un de ses billets à la banque par un nombre arbitraire de pièces de monnaie de valeur quelconque.

Montrer que ce processus termine, c'est à dire, qu'un jour fini l'homme est ruiné.

Le calcul propositionnel

Syntaxe

Sémantique

Définissabilité

Systèmes de preuves

- ▷ Systèmes de preuves sémantiques (vérité)
- ▷ Systèmes de preuves syntaxiques

Logiques (tables de vérité)

Syntaxe de la logique propositionnelle

Soit \mathcal{R} un ensemble dénombrable de lettres dites **propositionnelles**.

Définition : L'ensemble de **formules** de la logique propositionnelle est le plus petit ensemble contenant \mathcal{R} et fermé par les opérations binaires \vee , \wedge , \rightarrow et l'opération unaire \neg .

Exemple : $\neg(p)$ $\vee(p, p)$ $\rightarrow (\wedge(p, q), \neg(r))$
Autre notation : $\neg p$ $p \vee p$ $(p \wedge q) \rightarrow \neg r$

Notation : On écrira $\#$ pour \vee , \wedge ou \rightarrow .

Remarque : C'est un ensemble inductif, donc on pourra appliquer le principe d'induction.

$\mathcal{SF}(A)$: sous-formules d'une formule A

- Si A est une lettre p , $\mathcal{SF}(A) = \{p\}$.
- Si A est $\neg B$, $\mathcal{SF}(A) = \{\neg B\} \cup \mathcal{SF}(B)$.
- Si A est $B \# C$, $\mathcal{SF}(A) = \{B \# C\} \cup \mathcal{SF}(B) \cup \mathcal{SF}(C)$.

Sémantique de la logique propositionnelle

Étant donnée une valeur de l'ensemble $\mathbf{BOOL} = \{\mathbf{V}, \mathbf{F}\}$ pour chaque lettre propositionnelle, on veut établir la valeur d'une formule propositionnelle A .

- Définir une interprétation $I : \mathcal{R} \rightarrow \mathbf{BOOL}$ qui donne \mathbf{V} ou \mathbf{F} à chaque lettre propositionnelle.
- Définir la fonction booléenne unaire $\mathcal{FB}_{\neg} : \mathbf{BOOL} \rightarrow \mathbf{BOOL}$ et les fonctions booléennes binaires $\mathcal{FB}_{\vee}, \mathcal{FB}_{\wedge}, \mathcal{FB}_{\rightarrow} : \mathbf{BOOL}^2 \rightarrow \mathbf{BOOL}$.
- Construire la **valeur vérité** de la formule A .

La fonction booléenne unaire

$$\begin{aligned}\mathcal{FB}_{\neg}(V) &= F \\ \mathcal{FB}_{\neg}(F) &= V\end{aligned}$$

Les fonctions booléennes binaires

$$\begin{aligned}\mathcal{FB}_{\vee}(V, V) &= V & \mathcal{FB}_{\wedge}(V, V) &= V \\ \mathcal{FB}_{\vee}(V, F) &= V & \mathcal{FB}_{\wedge}(V, F) &= F \\ \mathcal{FB}_{\vee}(F, V) &= V & \mathcal{FB}_{\wedge}(F, V) &= F \\ \mathcal{FB}_{\vee}(F, F) &= F & \mathcal{FB}_{\wedge}(F, F) &= F\end{aligned}$$

$$\begin{aligned}\mathcal{FB}_{\rightarrow}(V, V) &= V \\ \mathcal{FB}_{\rightarrow}(V, F) &= F \\ \mathcal{FB}_{\rightarrow}(F, V) &= V \\ \mathcal{FB}_{\rightarrow}(F, F) &= V\end{aligned}$$

Valeur de vérité d'une formule A par rapport à une interprétation I

- Si A est une lettre p , $[A]_I = I(p)$.
- Si A est $\neg B$, $[A]_I = \mathcal{FB}_{\neg}([B]_I)$.

Tables de vérité

À quoi ça sert ? Méthode pour raisonner sur les modèles de formules propositionnelles.

Comment ça marche ? Soit A une formule ayant comme lettres propositionnelles l'ensemble $\{p_1, \dots, p_n\}$ et dont l'ensemble de sous-formules est $\{A_1, \dots, A_k\}$.

1. Construire une table où chaque colonne est étiquetée par une lettre p_i et chaque ligne par une sous-formule A_j .

2. Remplir chaque ligne de la table :

1. Calculer la valeur de vérité de la sous-formule A_j par rapport à l'interprétation I_m aux lettres p_1, \dots, p_n .
2. Calculer la valeur de vérité de la formule A par rapport à l'interprétation I_m .

Soit I l'interprétation $I(p) : V, I(q) : F$.
Calculer la valeur de vérité de la formule $(p \vee q) \wedge \neg p$ par rapport à I .

Exercice : Calculer la valeur de vérité de la formule $(p \vee q) \wedge \neg p$ par rapport à I .

Satisfaire et falsifier une formule

Soit I une interprétation, A une formule et Δ un ensemble de formules.

Définition :

I satisfait une formule A si $[A]_I = V$

I falsifie une formule A si $[A]_I = F$.

I satisfait un ensemble de formules Δ si I satisfait toute formule de Δ .

I falsifie un ensemble de formules Δ ssi il existe au moins une formule A dans Δ telle que $[A]_I = F$.

Formules satisfaisables, contradictoires, valides

Définition : Une formule A est satisfaisable s'il existe au moins une interprétation I qui satisfait A . Un ensemble de formules Δ est satisfaisable s'il existe au moins une interprétation I telle que I satisfait Δ , c'est à dire s'il existe au moins une interprétation I telle que I satisfait toutes les formules de Δ en même temps.

Définition : Une formule A est contradictoire ou insatisfaisable si elle n'est pas satisfaisable, c'est à dire si toute interprétation I falsifie A . Un ensemble de formules Δ est contradictoire ou insatisfaisable si il n'existe pas d'interprétation I qui satisfait toutes les formules de Δ en même temps.

A est satisfaisable si il existe une interprétation I qui satisfait A .
 A est contradictoire ou insatisfaisable si il n'existe pas d'interprétation I qui satisfait A .

Comment lire une table de vérité ?

- Si la colonne étiquetée par la formule A (qui est une sous-formule de A) ne contient que de V , alors A est valide.
- Si la colonne de la formule A contient au moins un F , alors A est satisfaisable.
- Sinon, l'interprétation qui rends A fautive est l'interprétation qui rends A satisfaisable.

on est Validité)
t soit p une de ses lettres pr
ntir de A en remplaçant syst
2. Si A est valide, alors A' es

Lemme : (Substitu
Soit A une formule propositionnelles. Soit A' la
formule obtenue de A en remplaçant
formule quelconque valide aussi.

Équivalence logique

Définition : Deux formules A et B sont **équivalentes**, noté $A \equiv B$, ssi $\{A\} \models B$ et $\{B\} \models A$.

Remarque : $A \equiv B$ ssi $(A \rightarrow B) \wedge (B \rightarrow A)$ est valide.

Lemme : (Remplacement équivalent)

Soit A, B, C trois formules et B une sous-formule de A . Si $B \equiv C$ alors $A \equiv A'$ où A' est obtenu à partir de A en remplaçant B par C .

Encore quelques exemples

(Associativité)	$(A \vee B) \vee C \equiv A \vee (B \vee C)$
	$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$
(Commutativité)	$A \vee B \equiv B \vee A$
	$A \wedge B \equiv B \wedge A$
(Idempotence)	$A \vee A \equiv A$
	$A \wedge A \equiv A$
(Lois de De Morgan)	$\neg(A \wedge B) \equiv \neg A \vee \neg B$
	$\neg(A \vee B) \equiv \neg A \wedge \neg B$
(Distributivité)	$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
	$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
(Loi de la double négation)	$\neg\neg A \equiv A$
(Définissabilité de \rightarrow)	$A \rightarrow B \equiv \neg A \vee B$

Remarques

1. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

2. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

3. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

4. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

5. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

6. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

7. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

8. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

9. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

10. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

11. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

12. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

13. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

14. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

15. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

16. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

17. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

18. $\{A \rightarrow B, A\} \models B$ ssi la formule $A \rightarrow B$ est valide pour tout A, B .

Théorème de compacité

Théorème : Un ensemble Δ de formules est **satisfaisable** ssi tout sous-ensemble fini de Δ est **satisfaisable**.