Logique

L3 informatique

Delia Kesner
PPS, Université Paris-Diderot

Email : kesner@pps.jussieu.fr
www.pps.jussieu.fr/~kesner

Plan du cours

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
   - Déduction naturelle
   - Gentzen.
   - Correction et complétude.

3. Calcul des prédicats :
   - Syntaxe, sémantique.
   - Unification et résolution.
   - Théories équationnelles.
   - Théorème d'incomplétude de Gödel.
   - Introduction à la théorie des ensembles.

Modalités du cours

- Chargés de TD :
  Isabelle Fagnot, Peter Habermehl, Severine Maingaud
- Examen partiel obligatoire : début mars.
- Note 1ère session : $\frac{1}{2}$ note partiel + $\frac{1}{2}$ examen final
- Note session rattrapage :

  Max(exam rattrapage, $\frac{1}{2}$ note partiel + $\frac{1}{2}$ exam rattrapage)

Documents du cours

- Transparents du cours

  http://www.pps.jussieu.fr/~kesner/enseignement/licence/logique/
- Tableau (exemples et démonstrations)

Bibliographie

- Logique pour l'informatique : introduction à la déduction automatique.
  S. Cerrito, VUIBERT.
- Mathématiques pour l'informatique.
  A. Arnold et I. Guessarian, MASSON.
- Introduction à la logique.
  R. David, K. Nour et C. Ra alli, DUNOD.
- Logique Mathématique I.
  R. Cori et J-L. Krivine, MASSON.
- Logique et fondements de l'informatique.
  R. Lassaigne et M. Rougemont, HERMES.
- First-Order Logic and Automated Theorem Proving.
  M. Fitting, SPRINGER.
- Concrete Mathematics.
  R. L. Graham, D. E. Knuth et O. Patashnik , ADDISON-WESLEY.

- Logic for Computer Science.
  J. Gallier, WILEY. Disponible en ligne :
  http://www.cis.upenn.edu/~jean/gbooks/logic.html

Notions préliminaires

Préordres, ordres

Définition :

– Un préordre est une relation réflexive et transitive.
  Exemple :
  $R = \{(2,2),(3,3),(4,4),(3,2),(2,3),(2,4),(3,4)\}$.
– Un ordre ou ordre partiel est une relation réflexive, anti-symétrique et transitive.

Notation : $\geq$

Exemple : R n'est pas un ordre car $(3,2),(2,3)$ mais $2 = 3$.
$S = \{(2,2),(3,3),(4,4),(2,3),(2,4),(3,4)\}$ est un ordre.

Définition : Un ordre strict est une relation irréflexive et transitive.

Notation : $>$

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

Définition : Un ordre strict est bien fondé ssi il n'existe aucune chaîne infinie décroissante (i.e., de la forme $a_0 > a_1 > a_2 > \ldots$).

Exemple : $>$ sur les entiers naturels est bien fondé. $>$ sur tous les entiers n'est pas bien fondé. sur les ensembles est bien fondé.

Majorants/minorants et bornes supérieures/inférieures

Soit E un ensemble muni d'un ordre $\leq$. Soit A E.

Définition :

Un majorant de A est un x E t.q. pour tout y A, $y \leq x$.

Un minorant de A est un x E t.q. pour tout y A, $x \leq y$.

La borne supérieure de A, notée $sup(A)$, est le plus petit des majorants de A (si z est un majorant de A alors $sup(A) \leq z$).

La borne inférieure de A, notée $inf(A)$, est le plus grand des minorants de A (si z est un minorant de A alors $z \leq inf(A)$).

Exemple : Soit $A = \{1,\ldots,10\}$. Tous les entiers dans $\{10,\ldots\}$ sont des majorants de A et 10 est la borne supérieure.

Exemple : Si $a \leq c$, $a \leq d$, $b \leq c$, $b \leq d$, alors a et b sont des minorants, mais comme ils sont incomparables il n'y a pas de borne inférieure.

Exemple : Si E_i sont des ensembles dans $P(E)$, alors le sup est $\bigcup_i E_i$ et le inf $\bigcap_i E_i$.

Fonctions monotones et points fixes

Définition : Soit $f : A \rightarrow B$ une fonction et soient $\leq_A, \leq_B$ deux ordres sur A et B respectivement.

La fonction f est monotone ssi $x \leq_A y$ implique $f(x) \leq_B f(y)$.

Exemple : $f(x) = x + 3$.

Définition : Soit $f : A \rightarrow A$ une fonction.

Un point fixe de f est un élément $x \quad A$ t.q. $f(x) = x$.

Exemple : Soit $f(x) = x^2$. Alors $x = 1$ est un point fixe.

Le plus petit point fixe de f est $inf(\{x \quad A \mid f(x) = x\})$.

Le plus grand point fixe de f est $sup(\{x \quad A \mid f(x) = x\})$.

Définitions inductives en informatique

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

Définitions Inductives et preuves par induction

Le principe

Une définition inductive est caracterisée par :

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

Exemple :

- Assertion : "`X est naturel`" ou "`X nat`"
- Règles d'inférence :

  R1 : `0 est naturel`

  R2 : Si `n est naturel`, alors `succ(n) est naturel`.

Notation

Les règles d'inférence sont notées

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

- `Conclusion` est une assertion
- $\texttt{Hypothèse}_1 \ldots \texttt{Hypothèse}_n$ sont des assertions
- En général $n \geq 0$. Si $n = 0$ la règle est un axiome

Exemple (règle unaire)

Les entiers naturels

$$\frac{}{0\ \texttt{est naturel}}\ (Nat0) \qquad \frac{n\ \texttt{est naturel}}{succ(n)\ \texttt{est naturel}}\ (Nat+)$$

Exemple (règle binaire)

Les arbres binaires

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

$$\frac{A_1\ \texttt{est un arbre binaire} \qquad A_2\ \texttt{est un arbre binaire}}{node(A_1, A_2)\ \texttt{est un arbre binaire}}\ (\text{Abin-ind})$$

Exemple

Les mots sur un alphabet A

$$\frac{}{\texttt{mot}} \qquad \frac{a \quad A \quad n\ \texttt{mot}}{a.n\ \texttt{mot}}$$

Exemple (plusieurs axiomes, règles unaires et binaires)

Les expressions de la logique propositionnelle sur l'alphabet A

$$\frac{p \quad A}{p \text{ expr}}$$

$$\frac{A_1 \text{ expr} \qquad A_2 \text{ expr}}{A_1 \quad A_2 \text{ expr}} \qquad \frac{A_1 \text{ expr} \qquad A_2 \text{ expr}}{A_1 \quad A_2 \text{ expr}}$$

$$\frac{A_1 \text{ expr} \qquad A_2 \text{ expr}}{A_1 \to A_2 \text{ expr}} \qquad \frac{A \text{ expr}}{\neg A \text{ expr}}$$

Exemple (plusieures assertions)

Les forêts de type T

$$\frac{}{\mathit{avide} \quad \texttt{arbre T}} \qquad \frac{}{\mathit{fvide} \quad \texttt{foret T}}$$

$$\frac{t \quad \texttt{T} \quad f \quad \texttt{foret T}}{\mathit{node}(t, f) \quad \texttt{arbre T}} \qquad \frac{A \quad \texttt{arbre T} \qquad f \quad \texttt{foret T}}{\mathit{add}(A, f) \quad \texttt{foret T}}$$

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

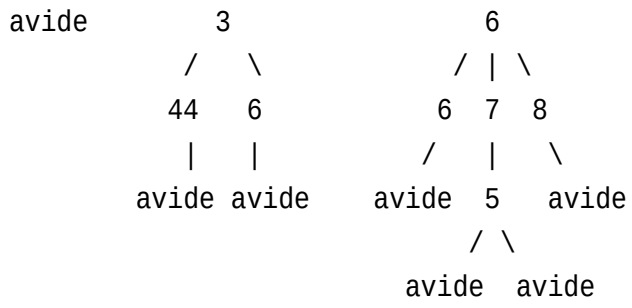
$$\frac{A_1 \qquad A_n}{A}$$

telle que $A_1, \ldots, A_n$ sont dérivables

Exercice :

– Montrer que $\mathit{succ}(\mathit{succ}(0))$ nat est dérivable.

– 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 structurelle
- Induction sur un ensemble inductif

Preuves par Induction

Induction sur les entiers I (induction mathématique)

Théorème : Soit P une propriété sur les entiers. Supposons

(IM1) $P(0)$,

(IM2) $n \in \mathbb{N}. P(n) \rightarrow P(n+1)$,

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

Exemples

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

Mais comment prouver

1. "Tout entier est décomposable en produit de nombres premiers"
2. "Si n est divisible par 3, alors $fib(n)$ est pair, sinon $fib(n)$ est impair".

Induction sur les entiers II (induction complète)

Théorème : Soit P une propriété sur les entiers. Supposons

$$(IC) \qquad n \quad IN.((\ k < n \rightarrow P(k)) \rightarrow P(n))$$

alors $n \quad IN.P(n)$

Équivalence des deux principes

Malgré l'apparente supériorité du deuxième principe, on prouve

Théorème : Induction mathématique et complète sont équivalentes.

Théorème fondamental du cours

Théorème : Tous le monde est d'accord avec le professeur.

Preuve : On montre, par induction sur le nombre de personnes dans l'amphi, que tout groupe de n personnes contenant le professeur est d'accord avec lui.
Cas de base : il y a seulement le professeur, trivial.
Cas inductif : on suppose l'enoncé 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 professeur soit le numéro n, et considérons le groupe A des premières n et le groupe B des dernières n personnes.
Les deux groupes contiennent le professeuruxeourailleuveET0g0.00G1.000.0

$n < n+1$, donc on peut appliquer l'hypothèse d'induction et en déduire qu'ils sont tous d'accord avec le professeur (qui est dans les deux), ce qui nous permet de conclure.

Corollaire : Le professeur a toujours raison.

vrai ou faux ?

Principe d'induction bien fondée

Un ensemble A, un ordre strict $>$ et une propriété P sur A

Principe d'induction :

Si

1. "pour tout élément minimal y 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 A on a $P(x)$"

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 :
  $concat(concat(m, v_1), v_2), = concat(m, concat(v_1, v_2))$
- Les arbres binaires :
  $P(a)$ est la propriété : $feuilles(a) = noeuds\_internes(a) + 1$

Induction sur quelques sur d'ordres bien fondés

- Ordre lexicographique
- Ordre multi-ensemble
- Combinaisons

Ordres lexicographiques

Soit $>_{A_i}$ un ordre strict sur l'ensemble 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 :

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

Ordre lexicographique sur le produit de n ensembles

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

$$(x_1, \ldots, x_n) >_{lex} (x_1, \ldots, x_n) \text{ ssi } \quad 1 \leq j \leq n$$
$$(x_j >_{A_j} x_j' \text{ and } \quad 1 \leq i < j \quad x_i = x_i')$$

Théorème : Si chaque $>_{A_i}$ est un ordre strict bien fondé sur A_i, alors l'ordre lexicographique $>_{lex}$ sur le produit de $A_1 \times \ldots \times A_n$ est un ordre strict bien fondé sur $A_1 \times \ldots \times A_n$.

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

Exemple : la fonction d'Ackerman

Montrer par induction que la fonction suivante termine.

Ackerman(0,n)	=	n+1
Ackerman(m+1,0)	=	Ackerman(m,1)
Ackerman(m+1,n+1)	=	Ackerman(m,Ackerman(m+1,n))

Les multi-ensembles

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

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

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, que dans un temps fini l'homme est ruiné.

Ordres multi-ensembles

Définition : $M >_{mul} N$ 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 : $\{\!\{5, 3, 1, 1\}\!\} \;_{mul}\; \{\!\{4, 3, 3, 1\}\!\}$.

Car $\{\!\{5, 3, 1, 1\}\!\} \;_{mul}\; \{\!\{4, 3, 3, 1, 1\}\!\} \;_{mul}\; \{\!\{4, 3, 3, 1\}\!\}$

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