Cours de Logique

Master 1 Informatique

Jean-Charles Régin

Organisation

- Cours valant 2 ECTS
- 6 séances incluant les contrôles
- 2 contrôles
- 1 séance = 1 cours (1h) + TD (2h)

Remerciements

- Hélène Collavizza
- Michel Rueher

Plan du cours

Une théorie est la donnée d’un langage et d’un ensemble d’axiomes

Théorie : syntaxe

- langage
- axiome

Théorie : sémantique

- interprétation
- validité

Théorie : preuves

- déductives
- inductives

Cas particulier des théories dont le langage est propositionnel (algèbre de boole)

Calcul des prédicats

Cadre de la logique équationnelle

Introduction

- Informatique :
  - stocker, gérer, afficher des données
  - formaliser, raisonner ou calculer de façon mécanique !
- Logique :
  - Passage de la sémantique vers la syntaxe
  - Symboles pour formaliser, règles d'inférence pour transformer
    - automatisation du raisonnement
- Limites de la logique : limites de l'informatique ?

A quoi peut servir la logique ?

- Formaliser le **raisonnement humain** (philosophie, ...)
- Formaliser **les sciences** (mathématiques,informatique…)
- **Démontrer** (mathématiques, preuves de propriétés de programmes, ...)

Aristote (385-322) : syllogisme

Exemple 1: l’antiquité

Tous les hommes sont mortels.
Socrate est un homme.
Donc Socrate est mortel.

Tous les étudiants sont sérieux.
Les M1 sont des étudiants.
Donc les M1 sont sérieux.

Les poules ont trois pattes.
Bécassine est une poule.
Donc Bécassine a trois pattes.

Ce qui est rare est cher.
Un 4x4 bon marché est rare.
Donc un 4x4 bon marché est cher.

Syllogisme

- Le mécanisme du syllogisme est juste
- La déduction est bonne
- Mais attention : les informations peuvent être fausses

Exemple 2

- Le BDE a organisé une grande soirée dansante étudiants/enseignants. Mais les disques de -M- ont disparu ! C'est sûr, les étudiants ont fait le coup. N'écoutant que son courage, le coordonnateur mène l'enquête.

- Les L3 déclarent:
  - si nous sommes coupables alors les professeurs sont coupables ou les M2 sont coupables ou les M1 sont innocents

- Les M1 déclarent:
  - les L3 sont coupables et si les M2 sont coupables alors les M1 sont innocents

- Les étudiants M2 déclarent:
  - si les M1 sont innocents alors les professeurs sont coupables

En calcul propositionnel

Axiomes :

A1: $L3 \Rightarrow \neg M1 \vee M2 \vee P$

A2 : $L3$

A3 : $M2 \Rightarrow \neg M1$

A4 : $\neg M1 \Rightarrow P$

$\Rightarrow$ implication, $\neg$ négation, $\vee$ ou

Procédé de Démonstration syntaxique déductif, noté $\vdash$

Procédé de déduction **syntaxique** :

$$P \vee T, P \Rightarrow Q \vdash Q \vee T$$

"$Q \vee T$ se déduit syntaxiquement de $P \vee T$ et $P \Rightarrow Q$"

En calcul propositionnel

Axiomes :

- A1: $L3 \Rightarrow \neg M1 \vee M2 \vee P$
- A2 : L3
- A3 : $M2 \Rightarrow \neg M1$
- A4 : $\neg M1 \Rightarrow P$

On peut déduire de ces axiomes :

A2

Les L3 sont coupables

A1 et A2 : $L3, L3 \Rightarrow \neg M1 \vee M2 \vee P \vdash \neg M1 \vee M2 \vee P$

Les M1 sont innocents ou les M2 ou les professeurs sont coupabless

A3 et $\neg M1 \vee M2 \vee P$: $M2 \Rightarrow \neg M1, \neg M1 \vee M2 \vee P \vdash \neg M1 \vee P$

Les M1 sont innocents ou les professeurs sont coupables

A4 et $\neg M1 \vee P$: $\neg M1 \Rightarrow P, \neg M1 \vee P \vdash P$

Les professeurs sont coupables

On ne peut rien déduire sur M1 et M2

Exemple

$A1 : \forall x \forall y \forall z\ (x + (y + z) = (x + y) + z)$

$A2 : \forall x \quad (x + 0 = x)$

$A3 : \forall x\ \exists y \quad (x + y) = 0$

- Que représentent-elles ?
- La formule suivante peut-elle être prouvée à partir de A1, A2, A3 ?

$$\phi : \forall x\ \forall z\ \exists y \quad x + y = \quad z$$

Exemple

la méthode triInsertion est-elle correcte ?

```
/*@ public normal_behavior
@ ensures (\forall int i; 0 <= i && i < t.length-1; t[i] <= t[i+1])
@*/
```

```
static void triInsertion(int [] t) {
for (int i = 1; i < t.length; i++) {
   int x = t[i];
   int j = i - 1;
   while ((j >= 0) && (x < t[j])) {
         t[j + 1] = t[j];
         j--;
   }
   t[j] = x;
}
```

Le défi de la logique

Un monde réel

Des faits, des données
Une façon de raisonner

PROUVER

sémantique

Le monde logique

Un écrit
Des règles pour transformer l'écrit

DEMONTRER

syntaxe

PROUVER =? DEMONTRER

Démontrer $\stackrel{?}{=}$ Prouver

Démontrer :

- une théorie τ (langage + axiomes)

- une formule Φ

Φ est-elle une conséquence logique de la théorie τ ?

- Etant donné un monde réel, existe-t-il une théorie pour le décrire?
- Une formule prouvée (sémantiquement) peut-elle être démontrée (syntaxiquement) ?
- Quand on démontre syntaxiquement une formule, dans quels domaines sémantiques est-elle vraie ?
- Quels procédés syntaxiques peuvent être automatisés (et donc programmés) ?

Exemple

- le programme termine ou boucle
- si le programme termine alors il affiche « don't worry »
- si le programme boucle ou affiche « don't worry » alors l'ingénieur informaticien est bien payé

- Montrer que l'ingénieur informaticien est bien payé

sémantique

Exemple : formalisation

On peut formaliser ces énoncés par :

- A1 : $P \vee Q$
- A2 : $P \Rightarrow R$
- A3 : $Q \vee R \Rightarrow T$

On peut déduire T par la règle de déduction syntaxique

$$X \vee Y,\ X \Rightarrow Z \quad \vdash \quad Z \vee Y$$

- A1, A2 $\vdash$ $Q \vee R$
- $Q \vee R$, A3 $\vdash$ T

syntaxe

Exemple

- Si on pose :
  - P : l'eau est potable
  - Q : l'eau est abondante
  - R : les hommes sont heureux
  - T : la paix règne sur terre

- On obtient :
  - L'eau est potable ou abondante
  - Si l'eau est potable alors les hommes sont heureux
  - Si l'eau est abondante ou les hommes sont heureux, alors la paix règne sur terre

- Peut-on en déduire que la paix règne sur terre ?

Exemple : formalisation

On a gardé les mêmes axiomes, mais on a donné une autre sémantique

On a démontré que T se déduit de :

- A1 : $P \vee Q$
- A2 : $P \Rightarrow R$
- A3 : $Q \vee R \Rightarrow T$

syntaxe

en appliquant la règle de déduction $\vdash$

Déduction : validité

- On peut conclure que la paix règne sur terre si deux conditions sont vérifiées :
  - le domaine **sémantique est un modèle** de {A1, A2, A3}
    - A1, A2 et A3 sont vrais quand on interprète
      P par « l'eau est potable », Q par « l'eau est abondante »,
      R par « les hommes sont heureux » et T par « la paix règne sur terre »
  - la règle de déduction $\vdash$ a la propriété :
    - « si A $\vdash$ T, alors T est valide dans **tous les modèles** de A »

- **Tous les modèles** de A : quel que soit le domaine des variables, quel que soit le sens des fonctions et des prédicats, qui valident A

Modèle

- Modèle de {A1,A2,A3} si les axiomes A1, A2, A3 sont vrais
- Modèle : vérification des axiomes

Démonstration syntaxique

- Questions à se poser sur les procédés de démonstration syntaxique :

- **Question 1** : Quand on déduit une formule Φ d'un ensemble d'axiomes A par un procédé de démonstration syntaxique, peut-on en conclure que Φ est valide dans tous les domaines sémantiques qui sont des modèles de A ?

correction

- **Question 2** : Quand une formule Φ se déduit d'un ensemble d'énoncés A dans un domaine sémantique, le procédé de déduction syntaxique permet-t-il de démontrer Φ ?

complétude

Démonstration syntaxique : prudence

Où il faut être prudent avant d'énoncer une vérité

Monde syntaxique S

A1: $f(\perp, x) = x$

A2 : $f(g(x), y) = g(f(x, y))$

Monde sémantique M1

domaine : les entiers naturels

opérations :

f : + (addition)

g : s (successeur)

$\perp$: 0 (la constante 0)

M1 est un « modèle » de S :

A1 et A2 sont valides dans le monde M1

$0+x = x$ et $s(x) + y = s(x+y)$ sont vrais

Monde syntaxique S (rappel)

A1: $f(\perp, x) = x$

A2 : $f(g(x), y) = g(f(x, y))$

Monde sémantique M2

domaine : les listes de a et b

opérations :

f : append (concaténation)

g : $cons_a$ (rajoute a en tête de liste)

$\perp$: null (la constante liste vide)

M2 est un « modèle » de S : A1 et A2 sont vrais dans le monde M2
append(null,l) = l et append($cons_a$ (l1),l2) = $cons_a$ (append(l1,l2)) sont vrais

Monde sémantique M3

domaine : les entiers naturels

opérations :

f : * (multiplication)

g : s (successeur)

$\perp$: 0 (la constante 0)

M3 n'est pas un « modèle » de S : x * 0 = x est faux dans M3

Modèle ou pas ?

- Tous les mondes sémantiques ne sont pas des modèles du monde syntaxique

Procédés de démonstration syntaxique

Un peu de magie :

On suppose connus deux procédés syntaxiques P1 et P2 qui permettent

de faire des démonstrations dans S (vu en fin du cours de logique)

P1 (déduction syntaxique) : utiliser les égalités des axiomes de gauche à droite

P2 (induction syntaxique) : utiliser les égalités des axiomes de gauche à

droite puis la formule à prouver

- On peut démontrer (syntaxiquement) par le procédé P1 que

$$f (g (g (x)) , y) = g (g (f (x , y)))$$

- On peut démontrer (syntaxiquement) par le procédé P2 que

$$f (x , y) = f (y , x)$$

Procédés de démonstration syntaxique

D'un point de vue sémantique

Validité de $f (g (g (x)) , y) = g (g (f (x , y)))$

- Dans M1 : $s (s (x)) + y = s (s (x + y))$ est vrai
- Dans M2 : $append(cons_a (cons_a (l1)) , l2) = cons_a (cons_a (append (l1, l2)))$ est vrai

Validité de $f (x , y) = f (y , x)$

- Dans M1 : $x + y = y + x$ est vrai

MAIS

- Dans M2 : append (l1, l2) = append (l2, l1) est faux!!!!!

contre-exemple: - append (aaaba , baa) vaut aaababaa

- append (baa , aaaba) vaut baaaaaba

Où est la faille ?

- Les axiomes A1 et A2 permettent « de construire de façon unique » tout entier (i.e toute donnée du monde M1).
  - **M1 est un modèle particulier « modèle_initial » de S**

- Les axiomes A1 et A2 ne disent rien sur les listes contenant des b!!! (i.e certaines données du monde M2 ne sont pas représentées par S)
  - Exemple : les éléments aaaba et aab ne peuvent pas être dérivés à partir de null, $cons_a$ et append.
  - **M2 est un modèle quelconque**

Procédés de démonstration syntaxique

- Ce qui est démontré par le procédé P1 est vrai dans tous_les_modèles
  - $f (g (g (x)) , y) = g (g (f (x , y)))$ est donc valide dans M1 et M2

- Ce qui est démontré par le procédé P2 est vrai dans les modèles_initiaux
  - $f (x, y) = f (y , x)$ est donc valide dans M1
  - Le procédé P2 ne nous dit rien sur la validité de $f (x, y) = f (y , x)$ dans M2

Procédés de démonstration syntaxique

- Questions à se poser sur les procédés de démonstration syntaxique :

- **Question 3 :** Dans quels types de modèles le procédé de démonstration syntaxique assure-t-il la validité des formules déduites ?

Procédés de démonstration syntaxique

- Dans ce cours nous verrons trois procédés de démonstration syntaxique :
  - la résolution
  - le remplacement d'égaux (déduction)
  - l'induction.

- Les deux premiers assurent la validité dans tous_les_modèles, l'induction assure la validité dans les modèles_initiaux

Logique du premier ordre

- Syntaxe et sémantique : que nous apporte la logique (du premier ordre) ?

> • Des méthodes de démonstration (syntaxique) qui assurent :
>
> - **ce qui est démontré est vrai dans tous_les_modèles**
> - **ce qui est vrai dans tous_les_modèles est démontrable**
>
> (théorème de complétude de Gödel 1930)

Logique du premier ordre

- **Les formules qui sont vraies dans tous_les_modèles ne sont pas les plus intéressantes pour l'informatique.**

- Ce sont les formules qui correspondent à un raisonnement logique.
- Par exemple, la formule « M(a) » est vraie dans tous_les_modèles de
  - A1 : $\forall$ x (H(x) $\Rightarrow$ M(x))
  - A2 : H(a)

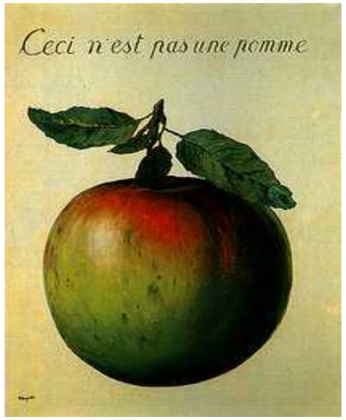
Logique du premier ordre

- **Les formules qui nous intéressent le plus en informatique, sont les formules qui sont vraies dans les modèles_initiaux**
- Ces formules nécessitent une connaissance de la structure des éléments.
- Par exemple, la formule « $x + y = y + x$ » est vraie dans les modèles_initiaux de
  - A1 : $0 + x = x$
  - A2 : $s(x) + y = s(x+y)$

- Elle n'est pas vraie dans tous_les_modèles de {A1,A2}
(on a vu que l'interprétation M2 dans les listes de a et b est un modèle de {A1,A2} mais que $x + y = y + x$ est fausse dans M2)

Logique

- Il existe des méthodes de démonstration (syntaxique) qui assurent :
  ce qui est démontré est vrai dans le modèle_initial
- **MAIS** il n'existe pas de procédé de démonstration qui permette de démontrer n'importe quelle formule vraie du *modèle_initial*



- Quel que soit le système finiment axiomatisé cohérent et capable de formaliser l'arithmétique, on peut toujours construire une formule vraie de l'arithmétique que l'on ne peut pas démontrer (théorème d'incomplétude de Gödel, 1931)
  Basé sur un paradoxe : « ***Le barbier rase tous ceux et seulement ceux qui ne se rasent pas eux-mêmes*** »