Logique formelle & Programmation logique

$\exists \Rightarrow \forall$

Dr. Stéphane Lengrand,
Stephane.Lengrand@Polytechnique.edu

Cours 0 :

Motivation, Introduction

Introduction au cours

Présentations

Pourquoi ce cours vous intéresse :

– Vous vous destinés à êtres ingénieurs. . .

. . .en Info, Electronique et Automatique

– Nécessairement, vous allez faire des erreurs

parce que vous êtes humains !

parce que d'autres en ont fait avant vous !

– Certains bugs coûtent plus que d'autres

Crash d'Ariane 5, 04/06/1996 : 500 millions d'euros

Oui mais pourquoi la logique ?

– Développement personnel : Ingénieurs de qualité $\Leftarrow$ Rigueur $\Leftarrow$ Logique

– Science pour la fiabilité des systèmes :

Logique a produit des **outils** pour l'assurance de la qualité

Méthodes formelles

. . .à base de mathématiques

– Intelligence Artificielle

Vous vous dites : est-ce que ça en vaut les efforts ?

Développement personnel : Oui, pour vous

Intelligence artificielle : Oui, la logique en est la base

Méthodes formelles :

Ca dépend, elles sont coûteuses, mais nécessaires dans 2 cas

– quand beaucoup d'argent est en jeu
  (cf Ariane, transactions financières, cryptographie)
– quand des vies humaines sont en jeu
  (cf systèmes embarqués, pilotes automatiques, ligne 14-Meteor)

Sur ce cours

En 3ème année : vous devez être autonomes

APPRENDRE **COMPRENDRE**

Contrôle : Droit aux documents. Pas aux livres.
Pas plus facile, prend du temps, de l'écoute

Venir me voir = votre responsabilité
M'interrompre = votre responsabilité

Contact : `Stephane.Lengrand@Polytechnique.edu`
Page web :
`http ://www.lix.polytechnique.fr/~lengrand/`

Sur ce cours, suite

Transparents disponibles en pdf sur mon site web

Ne contient pas toutes les infos (par ex : les démonstrations)

$\Longrightarrow$ en TDs ou en livres :

– *Logique pour l'informatique : introduction à la déduction automatique*,
  Serenella Cerrito (Ed. Vuibert) (23,75 eur -Amazon)
– *Logique mathématique, tome 1 & 2*,
  René Cori et Daniel Lascar (Ed. Dunod) (38,95 eur)
– *Introduction à la logique : Théorie de la démonstration*,
  Karim Nour, René David et Christophe Raffalli (Ed. Dunod) (30,88 eur)
– *Proof Theory and Automated Deduction*, Jean Goubault-Larrecq et Ian
  MacKie (Kluwer Academic Publisher) (48,57 eur)
– ce qu'il y a de disponible dans la bibliothèque locale

La logique, ça vient d'où ?

C'est vieux (Aristote, Socrate, . . .).

Vérité : confrontation avec la réalité.

Depuis. . .
– diversification des disciplines mathématiques
– éloignement de plus en plus important d'une réalité tangible

XIXème siècle, crise logique. Les mathématiques : quoi, comment ?
– quelle est la théorie universelle qui unifie les mathématiques ?
– comment raisonne-t-on pour tirer des conclusions de cette théorie ?

– Axiomes
– Démonstration

Grossièrement, quelques noms et contributions

Boole (1815-1864) : algèbres de Boole, booléens,. . .ça vous dit qq chose ?

Frege (1848-1925) :
bases -imparfaites- de la théorie des ensembles, formalismes logiques, . . .

Hilbert (1862-1943) : 23 problèmes ouverts, meta-mathématiques,. . .

Zermelo (1871-1953) : théorie des ensembles moderne, avec Fraenkel

Russell (1872-1970) :
célèbre pour son paradoxe trouvé chez Frege, Principia Mathematica

Brouwer (1881-1966) : constructivisme

Goedel (1906-1978) : théorèmes d'incomplétude

Church (1903-1995) : fonctions et calcul, théorèmes d'indécidabilité. . .

Gentzen (1909-1945) : théorèmes de cohérence et formalismes logiques

Méta-mathématiques

Les mathématiques =

outils et méthodes pour étudier rigoureusement des objets

Et si l'objet d'étude était le fonctionnement des maths elles-mêmes ? ? ?

$\Longrightarrow$ Meta-mathématiques

Grande avancée : vérité $\Longrightarrow$ prouvabilité

Une proposition P est-elle vraie ?

$\Downarrow$

Une proposition P est-elle prouvable ?

Questions meta-mathématiques

– Existe-t-il un langage adéquat pour parler de toutes les mathématiques ?
OUI : langage des prédicats = langage du 1er ordre

– Qu'est-ce qu'une démonstration / preuve / deduction ?
Toujours pas d'accord. Raisonnement par l'absurde ? ou pas ?
Brouwer, Heyting, Kolmogorov,. . .le refusent
ensuite, questions de présentations (calcul des séquents,. . .)

– Existe-t-il une collection d'axiomes (si possible la plus petite) à partir desquelles se déduisent toutes les maths ?
Théorie des ensembles, Zermelo-Fraenkel : 9 "axiomes" (ou schémas)

Frege : pour chaque propriété P, autorise la construction $\{x \mid P(x)\}$
Paradoxe :
Soit $F = \{x \mid x \notin x\}$. *Est-ce que* $F \in F$ *ou est-ce que* $F \notin F$ *?*. . .

Questions meta-mathématiques, suite

– Etant donné une proposition P, existe-t-il toujours soit une preuve de P soit une preuve de $\neg P$? (in)complétude1
NON (Goedel)

– Les mathématiques peuvent-elles démontrer qu'elles ne se contredisent pas ? (in)complétude2
NON (Goedel)

– Existe-il un algorithme qui réponde OUI s'il existe une preuve de P, qui réponde NON sinon ? (in)décidabilité
NON (Church, Turing)

Ces trois réponses datent des années 30.

Questions?

Cours 1 :

Syntaxe, Sémantique, Logique propositionnelle

Pas de pensée sans langage

Prenons les mathématiques comme objet d'étude
(faisons des meta-mathématiques !), et analysons leur structure !

Une **proposition** exprime, dans un langage syntaxique, quelque chose (qui a un sens).

structure à base de symboles qui permet d'exprimer qq chose = syntaxe,
signification de cette expression = sémantique

signifiant = syntaxe, signifié = sémantique

Niveau meta (transparent subtile, mais pas vital)

Pendant longtemps : monde réel fournit un cadre pour la sémantique,
on vérifiait qu'une proposition était vraie ou fausse par confrontation avec le réel.

Maintenant : objets mathématiques trop abstraits
(voyez-vous des nombres complexes dans la rue ?)

$\Longrightarrow$ sémantique donnée par le niveau meta-mathématique

Se placer / raisonner dans une logique X pour étudier la logique Y.

Exemples :
X = l'arithmétique de Péano $\Longrightarrow$ sémantique à base d'entiers
X = théorie des ensembles $\Longrightarrow$ sémantique à base d'ensembles

Remarque : le niveau meta-mathématique étant syntaxique, finalement il n'y a jamais que de la syntaxe (sémantique = syntaxe du niveau meta)

Clivage syntaxe/sémantique à deux niveaux

Une proposition parle d'objets (d'études).

Niveau objet : les structures syntaxiques 4 et IV désignent le même objet sémantique.

Niveau proposition : les structures syntaxiques $(x \in y) \wedge (x \in z)$ et $(x \in z) \wedge (x \in y)$ ont la même sémantique.

Syntaxe de la logique propositionnelle

En logique propositionnelle : pas d'objets !

– des variables propositionnelles $p, q, r, \ldots$ désignent des propositions quelconques

– des connecteurs $\wedge$ (et) $\vee$ (ou) $\Rightarrow$ (implique), $\neg$ (non-),. . .construisent des propositions complexes à partir de propositions simples, ou sont des constantes logiques $\top$ (vrai) $\bot$ (faux),. . .

Une manière rapide d'écrire les règles de construction des propositions :

$A, B, C, \ldots ::= p \mid (A \wedge B) \mid (A \vee B) \mid (A \Rightarrow B) \mid (\neg A) \mid \top \mid \bot$

On appelle ça une définition inductive

Propositions = chaînes de symbôles bien-parenthésées, ou arbres ?

les 2 visions sont équivalentes

Sémantique de la logique propositionnelle

Pour donner une sémantique aux propositions, il faut

– un ensemble $\mathcal{B}$ dans lequel on va interpréter les propositions.

– une sémantique pour chaque connecteur $\star$,
c'est-à-dire une fonction $f_\star$ de $\mathcal{B}^n$ dans $\mathcal{B}$
($n = 2$ pour les connecteurs binaires, $n = 1$ pour les connecteurs unaires, $n = 0$ pour les constantes, . . .)

Pour que la sémantique de $\wedge, \vee, \ldots$ corresponde à notre intuition, il faut que $\mathcal{B}$ soit une algèbre de Boole

Exemple d'algèbre de Boole :

$\mathcal{B}$ est l'ensemble des parties d'un ensemble X (quelconque), avec

$f_\wedge(x, y) = x \cap y \quad f_\top = X \quad f_\neg(x) = \overline{x}$

$f_\vee(x, y) = x \cup y \quad f_\bot = \emptyset$

Autre exemple d'algèbre de Boole : les booléens

L'ensemble des booléens $\mathcal{B} = \{\mathsf{T}, \mathsf{F}\}$ à 2 éléments.

x	y	$f_\wedge(x,y)$
T	T	T
T	F	F
F	T	F
F	F	F

x	y	$f_\vee(x,y)$
T	T	T
T	F	T
F	T	T
F	F	F

x	y	$f_\Rightarrow(x,y)$
T	T	T
T	F	F
F	T	T
F	F	T

x	$f_\neg(x)$
T	F
F	T

$f_\top()$
T

$f_\bot()$
F

Sémantique de la logique propositionnelle

Une fois que l'on s'est donné ces fonctions, on peut alors calculer l'interprétation dans $\mathcal{B}$ des propositions

Un paramètre : l'interprétation $\mathcal{I}$ des variables propositionelles $p, q, r, \ldots$, dite valuation. Exemple avec les booléens : $\mathcal{I}(p) = \mathsf{T}$ ou $\mathcal{I}(p) = \mathsf{F}$.

L'interprétation $[A]_{\mathcal{I}}$ d'une d'une proposition A selon la valuation $\mathcal{I}$ est définie par récurrence sur A :

$$\begin{aligned} [p]_{\mathcal{I}} &:= \mathcal{I}(p) \\ [\star(A_1, \ldots, A_n)]_{\mathcal{I}} &:= f_\star([A_1]_{\mathcal{I}}, \ldots, [A_n]_{\mathcal{I}}) \end{aligned}$$

Exemple

$$\begin{aligned} [A \wedge B]_{\mathcal{I}} &:= f_\wedge([A]_{\mathcal{I}}, [B]_{\mathcal{I}}) \\ [\neg A]_{\mathcal{I}} &:= f_\neg([A]_{\mathcal{I}}) \end{aligned}$$

Satisfiable, Valide

A partir de maintenant, $\mathcal{B} = \{\mathsf{T}, \mathsf{F}\}$

Définitions :

- $\mathcal{I}$ est un modèle de A si $[A]_{\mathcal{I}} = \mathsf{T}$
- A est satisfiable s'il y a une valuation qui est un modèle de A
- A est valide si toute valuation est un modèle de A

Théorème : A est satisfiable (resp. valide) si et seulement si $\neg A$ n'est pas valide (resp. satisfiable)

Pour vérifier qu'une proposition est valide ou satisfiable, on regarde toutes les valuations $\mathcal{I}$ possibles sous la forme d'une table de vérité

Si A possède n variables propositionnelles, on a 2^n cas à tester !

Exemple en exercice avec $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$

Conséquence sémantique (parfois dite logique)

Définition

- B est une conséquence sémantique de A, noté $A \models B$ si tout modèle de A est aussi un modèle de B
- A et B sont sémantiquement équivalents, noté $A \equiv B$, si $A \models B$ et $B \models A$

voir exercice sur les lois de De Morgan

Notez que

- $\top \models A$, aussi noté $\models A$, si et seulement si A est valide.
- $A \models B$ si et seulement si $A \Rightarrow B$ est valide (voir TD).
- si $A \models B$, alors
  - A est valide implique B est valide
  - A est satisfiable implique B est satisfiable

  mais l'inverse n'est pas vrai !

Conclusions

Toutes ces propriétés des propositions sont des constatations sémantiques

Cours suivant : on verra comment on peut prouver la conséquence ou la validité par une démonstration, c'est-à-dire par un raisonnement syntaxique

Questions?

Cours 2 :

Logique propositionnelle — La notion de démonstration

Notion de preuve

Rappelez-vous : on cherche à caractériser de manière syntaxique les notions de **conséquence sémantique** et de **validité**

On avait des notion sémantiques $A \models B$ et $\models B$

basées sur la constatation

(passant par valeurs de vérité & interprétation sémantique des formules)

On cherche maintenant des notions syntaxiques $A \vdash B$ et $\vdash B$

basées sur la démonstration (=preuve)

A quoi bon ? La constatation sémantique n'est-elle pas suffisante ?

. . .après tout, si on peut "voir" si une proposition est vraie ou pas. . .

Aha ! tant qu'on parle de choses finies (par ex : logique propositionnelle). . .

. . .à la rigueur. . .

Mais quand l'univers du discours est infini, comment constater des propriétés universelles ? (c.f. logique des prédicats (= du 1er ordre))

Le calcul des séquents

Un séquent est une paire de multisets de formules.

C'est quoi un multiset de formules (que l'on notera $\Gamma, \Delta, \ldots$) ?

Listes : ordre importe, répétitions importent

$$A, B \neq B, A \qquad A, A \neq A$$

Ensembles : ordre n'importe pas, répétitions n'importent pas

$$\{A, B\} = \{B, A\} \qquad \{A, A\} = A$$

Multisets : ordre n'importe pas, répétitions importent

$$\{\!\{A, B\}\!\} = \{\!\{B, A\}\!\} \qquad \{\!\{A, A\}\!\} \neq A$$

Formellement : une fonction f des formules vers les entiers ≥ 0 à support fini (support = les formules A telles que $f(A) > 0$)

$f(A)$ étant le nombre d'occurrences de A dans le multiset

Le calcul des séquents

Une paire de multisets de formules $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ est appelé séquent (on lache les $\{\!\{ \}\!\}$ des multisets)

Ce n'est qu'une construction syntaxique, mais le sens intuitif d'un tel séquent est $A_1 \wedge \ldots \wedge A_n \Rightarrow B_1 \vee \ldots \vee B_m$

Une dérivation (=preuve=démonstration) est un arbre

– dont les noeuds sont étiquetés par des séquents, par exemple :

$$\dfrac{\vdash a \qquad \dfrac{\vdash b}{\vdash b \vee c}}{\vdash a \wedge (b \vee c)}$$

– dont l'étiquetage suit des règles dites d'inférence, par exemple :

$$\dfrac{\vdash A \qquad \vdash B}{\vdash A \wedge B}$$

premisses

conclusion

Le calcul des séquents

Un séquent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ est dérivable dans un système $\mathcal{S}$ de règles d'inférence s'il existe une dérivation dont il est la conclusion (i.e. dont il décore la racine).

On le note $A_1, \ldots, A_n \vdash_{\mathcal{S}} B_1, \ldots, B_m$

L'idée est maintenant de trouver un système $\mathcal{S}$ de règles d'inférence caractérisant la conséquence sémantique

(i.e. tel que $A_1, \ldots, A_n \vdash_{\mathcal{S}} B_1, \ldots, B_m$ si et seulement si $A_1 \wedge \ldots \wedge A_n \models B_1 \vee \ldots \vee B_m$)

Schémas et instances

Schéma :

$$\dfrac{\vdash A \qquad \vdash B}{\vdash A \wedge B}$$

où A, B dénotent des formules arbitraires

Instances (exemples) :

$$\dfrac{\vdash c \qquad \vdash c'}{\vdash c \wedge c'} \qquad \dfrac{\vdash \neg c \qquad \vdash \neg c'}{\vdash \neg c \wedge \neg c'} \qquad \ldots$$

pour les variables propositionnelles particulières c et c'

Système G3 : les règles d'inférence

Règle de base (axiome) : $\dfrac{}{\Gamma, A \vdash A, \Delta}$

Connecteur	Règle d'intro gauche	Règle d'intro droite
$\top$		$\dfrac{}{\Gamma \vdash \top, \Delta}$
$\bot$	$\dfrac{}{\Gamma, \bot \vdash \Delta}$	
$\neg$	$\dfrac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}$	$\dfrac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}$
$\vee$	$\dfrac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}$	$\dfrac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}$
$\wedge$	$\dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}$	$\dfrac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$

Système G3 : les règles d'inférence

Connecteur	Règle d'intro gauche	Règle d'intro droite
$\Rightarrow$	$\dfrac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta}$	$\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta}$

Correspond à l'idée que $A \Rightarrow B$ est la même chose que $(\neg A) \vee B$

(écrivez les règles et vous verrez...)

G3 : Correction et complétude

Théorème

$$A_1, \ldots, A_n \vdash_{G3} B_1, \ldots, B_m$$

si et seulement si

$$A_1 \wedge \ldots \wedge A_n \models B_1 \vee \ldots \vee B_m$$

Démonstration : D'habitude,

du haut vers le bas : par récurrence sur la hauteur de l'arbre de preuve

du bas vers le haut : beaucoup de façons

Souvenez-vous : la récurrence (aussi appelée induction) est au raisonnement ce que la récursion est au calcul / à la programmation.

Questions?

Cours 3 :

Logique du 1er ordre

Pour faire rapide

On rajoute les quantificateurs $\exists x, P$ et $\forall x, P$

Oui mais ils quantifient sur quoi ?

On a besoin d'un univers de discours, dont les éléments sont décrits (dans la syntaxe) par des termes

Exemples d'univers de discours :	Exemples de termes
– les entiers	$0, S(0), 3+4$
– les réels	$3/4, \pi$
– les ensembles	$\emptyset, A \cup B$

Syntaxe : termes

Formellement, on se donne une signature Σ de termes :

ensemble de constructeurs de termes avec, pour chacun, une arité

Exemple : $\Sigma = \{$"0" / 0, "S" / 1, "$+$" /2 $\}$

Comme annoncé, on se donne aussi des variables de termes x, y, z

La syntaxe des termes est alors donnée par :

$$t ::= x \mid f(t_1, \ldots, t_n) \qquad \text{si } f/n \in \Sigma$$

Syntaxe : propositions

Ensuite, on veut dire des choses sur ces objets de l'univers

Exemple : $1 + 1 = 2$ $\quad 3 \leq 4$ $\quad x \in y$ $\quad x \subseteq y$

On se donne donc une signature Ψ de propositions :

ensemble de propositions atomiques avec, pour chacune, une arité

Exemple : $\Psi = \{$"$=$" / 2, "$\leq$" / 2, "IsEven" /1 $\}$

La syntaxe des propositions est alors donnée par :

$P ::= p(t_1, \ldots, t_n) \mid \ldots$ [comme en logique prop.] $\ldots \mid (\exists x, P) \mid (\forall x, P)$

si $p/n \in \Psi$

Syntaxe : propositions

On bazarde les variables propositionnelles !

Elle servaient à ce que les propositions atomiques ne soient pas interprétées de manière constante (que chacune, selon l'interprétation, soit parfois vraie parfois fausse)

Ici, les termes donnés comme arguments des prop. atomiques vont créer cette variation.

Syntaxe : variables liées/muettes 1

Notion de variable muette en maths :

On “sait bien” que $\forall x, P(x)$ c’est la même chose que $\forall y, P(y)$

Intuition : le nom / la variable que j’utilise pour désigner une chose n’a pas d’importance

Plus complexe qu’il n’y parait.

Tâche 1 : définir quelles sont, dans P les variables muettes (=liées)

Celles qui n’y sont pas libres ! (on est bien avancé)

Syntaxe : variables liées/muettes 2

Toute variable (de terme) apparaissant dans t est libre dans t, elles forment l’ensemble $fv(t)$

fv est défini inductivement sur les propositions :

$$
\begin{aligned}
fv(p(t_1, \ldots, t_n)) &:= fv(t_1) \cup \ldots \cup fv(t_n) \\
fv(A \wedge B) &:= fv(A) \cup fv(B) \\
&\ldots \\
fv(\exists x, P) &:= fv(P) \backslash \{x\} \\
fv(\forall x, P) &:= fv(P) \backslash \{x\}
\end{aligned}
$$

Syntaxe : variables liées/muettes 3

Tâche 2 : définir ce qu’est le renommage d’une variable muette

appelé α-conversion

On définit pour ça l’échange de 2 variables sur P (resp. t)

$(xy)P$ (resp. $(xy)t$) :

partout où vous avez écrit x (lié ou libre), vous mettez y, et vice versa

$\exists x, P$ est identifié avec $\exists y, (xy)P$ si $y \notin fv(P)$

$\forall x, P$ est identifié avec $\forall y, (xy)P$ si $y \notin fv(P)$

Pourquoi “si $y \notin fv(P)$” (i.e. y est une variable fraiche) ?

$(y = 0) \wedge (\exists x, x = y)$ n’est pas la même chose que

$(y = 0) \wedge \exists y, y = x$

Syntaxe : variables substituées

Tâche 3 : définir une notion de substitution (variable x par terme t)

sur les termes : $\{t/x\}\, t'$

trivial.

sur les propositions : $\{t/x\}\, P$

ATTENTION quand vous définissez $\{t/x\}\, (\forall y, P)$ et $\{t/x\}\, (\exists y, P)$!!!

Que se passe-t-il si $x = y$?

si $y \in fv(t)$?

Moralité :

$\{t/x\}\, (\forall y, P) := \forall y, \{t/x\}\, P$ et $\{t/x\}\, (\exists y, P) = \exists y, \{t/x\}\, P$

si $x \neq y$ et $y \notin fv(t)$

sinon, renommer y en l’échangeant avec variable fraiche z : $(yz)P$

Sémantique : termes

Il nous faut :

– un univers (sémantique) $\mathcal{U}$ pour interpréter les termes
– une interprétation $\tilde{f}$ de tous les constructeurs de termes f, à savoir si f est d'arité n, une fonction $\tilde{f}$ de $\mathcal{U}^n$ dans $\mathcal{U}$

Sémantique d'un terme t : par récurrence sur t

et si t est une variable x ?

Interprétation $[t]_I$ d'un terme t est parametrée par valuation qui interprète les variables vers $\mathcal{U}$, ce qui donne :

$$\begin{aligned} [x]_I &:= I(x) \\ [f(t_1, \ldots, t_n)]_I &:= \tilde{f}([t_1]_I, \ldots, [t_n]_I) \end{aligned}$$

Sémantique : propositions

Il nous faut :

– une interprétation $\tilde{p}$ de toutes les propositions atomiques p, à savoir si p est d'arité n, une fonction $\tilde{p}$ de $\mathcal{U}^n$ vers $\mathcal{B}$

Sémantique d'une proposition P : par récurrence sur P

dépend toujours de l'interprétation I des variable libres de P

$$\begin{aligned} [p(t_1, \ldots, t_n)]_I &:= \tilde{p}([t_1]_I, \ldots, [t_n]_I) \\ [A \wedge B]_I &:= f_\wedge([A]_I, [B]_I) \\ &\ldots \\ [\forall x, P]_I &:= min\{[P]_{I, x \mapsto u} \mid u \in \mathcal{U}\} \\ [\exists x, P]_I &:= max\{[P]_{I, x \mapsto u} \mid u \in \mathcal{U}\} \end{aligned}$$

Modèles & co.

Un modèle d'une proposition P est maintenant :

– un univers $\mathcal{U}$ non-vide
– une interprétation $\tilde{f}$ pour chaque $f \in \Sigma$ et $\tilde{p}$ pour chaque $p \in \Psi$
– une interprétation $I(x) \in \mathcal{U}$ pour chaque variable $x \in fv(P)$

tels que $[P]_I = T$

Dans le cas particulier où $fv(P) = \emptyset$ (on dit que P est clos),

cela ne dépend que de l'univers $\mathcal{U}$ et de l'interprétation $\tilde{}$

Rebelotte

Définitions :

– A est satisfiable s'il a un modèle
– A est valide, noté $\models A$, si toutes les structures ci-dessus en sont des modèles
– B est une conséquence sémantique de A (noté $A \models B$) si tout modèle de A est aussi un modèle de B
– A et B sont sémantiquement équivalents, noté $A \equiv B$, si $A \models B$ et $B \models A$

Théorème : A est satisfiable (resp. valide) si et seulement si $\neg A$ n'est pas valide (resp. satisfiable)

Système de preuve ! G3 version logique du 1er ordre

Même règles qu'en propositionnel, plus

$$\frac{\Gamma \vdash P, \Delta}{\Gamma \vdash (\forall x, P), \Delta}\, x \notin fv(\Gamma, \Delta) \qquad \frac{\Gamma, (\forall x, P), \{{}^{t}/_{x}\} P \vdash \Delta}{\Gamma, (\forall x, P) \vdash \Delta}$$

$$\frac{\Gamma \vdash \{{}^{t}/_{x}\} P, (\exists x, P), \Delta}{\Gamma \vdash (\exists x, P), \Delta} \qquad \frac{\Gamma, P \vdash \Delta}{\Gamma, (\exists x, P) \vdash \Delta}\, x \notin fv(\Gamma, \Delta)$$

A nouveau : Dualité de De Morgan entre $\forall$ et $\exists$

G3 : Correction et complétude

Théorème

$$A_1, \ldots, A_n \vdash_{G3} B_1, \ldots, B_m$$
si et seulement si
$$A_1 \wedge \ldots \wedge A_n \models B_1 \vee \ldots \vee B_m$$

Démonstration :

– du haut vers le bas : comme d'hab. par récurrence sur la hauteur de la dérivation

– du bas vers le haut : Oula ! ! !
Il faut construire un modèle, avec un $\mathcal{U}$ et un $\tilde{}$! ! !

A partir de quoi ? La syntaxe quotientée. . . modèles syntaxiques. . .

Questions?

Cours 4 :

Résolution

Programmation logique

Un petit point sur le buveur

Si le bar est vide, le théorème est faux.

Deux visions :

Première vision : à la “Théorie des types”

on enregistre à quelles variables libres on a droit :

$$\frac{\Gamma \vdash^{\Phi,x} P, \Delta}{\Gamma \vdash^{\Phi} (\forall x, P), \Delta} \qquad \frac{\Gamma, (\forall x, P), \{{}^{t}/_{x}\} P \vdash^{\Phi} \Delta}{\Gamma, (\forall x, P) \vdash^{\Phi} \Delta} fv(t) \subseteq \Phi$$

$$\frac{\Gamma \vdash^{\Phi} \{{}^{t}/_{x}\} P, (\exists x, P), \Delta}{\Gamma \vdash^{\Phi} (\exists x, P), \Delta} fv(t) \subseteq \Phi \qquad \frac{\Gamma, P \vdash^{\Phi,x} \Delta}{\Gamma, (\exists x, P) \vdash^{\Phi} \Delta}$$

Un petit point sur le buveur

Du coup, $\vdash^{\emptyset} \exists x, (p(x) \Rightarrow \forall y, p(y))$ n’est pas un séquent prouvable. . .

. . .sans constante de terme (=constructeur de terme d’arité 0)

par exemple ici : le barman ?

par contre, $\vdash^{z} \exists x, (p(x) \Rightarrow \forall y, p(y))$ est dérivable !

ainsi que $(\exists z, \top) \vdash^{\emptyset} \exists x, (p(x) \Rightarrow \forall y, p(y))$

Un petit point sur le buveur

Deuxième vision : à la “Logique du premier ordre”

Les règles sont celles que j’ai présentées au cours 3 :

$$\frac{\Gamma \vdash P, \Delta}{\Gamma \vdash (\forall x, P), \Delta} x \notin fv(\Gamma, \Delta) \qquad \frac{\Gamma, (\forall x, P), \{{}^{t}/_{x}\} P \vdash \Delta}{\Gamma, (\forall x, P) \vdash \Delta}$$

$$\frac{\Gamma \vdash \{{}^{t}/_{x}\} P, (\exists x, P), \Delta}{\Gamma \vdash (\exists x, P), \Delta} \qquad \frac{\Gamma, P \vdash \Delta}{\Gamma, (\exists x, P) \vdash \Delta} x \notin fv(\Gamma, \Delta)$$

Du coup, $\vdash \exists x, (p(x) \Rightarrow \forall y, p(y))$ est prouvable.

Un petit point sur le buveur

Les deux versions ne prouvent pas les mêmes théorèmes !

(. . .sauf si la signature possède une constante de terme)

La version “Théorie des types” est correcte et complète vis-à-vis de la notion de modèle :

– un univers $\mathcal{U}$

– une interprétation $\tilde{f}$ pour chaque $f \in \Sigma$ et $\tilde{p}$ pour chaque $p \in \Psi$

– une interprétation $I(x) \in \mathcal{U}$ pour chaque variable $x \in fv(P)$

tels que $[P]_I = T$

Définition

$A_1, \ldots, A_n \models B$ si tout modèle de $A_1, \ldots, A_n$ est un modèle de B

Théorème

$A_1, \ldots, A_n \vdash_{G3} B_1, \ldots, B_m$ ssi $A_1, \ldots, A_n \models B_1 \vee \ldots \vee B_m$

Un petit point sur le buveur

La version “Logique du premier ordre” est correcte et complète vis-à-vis de la notion de modèle :

– un univers $\mathcal{U}$ non-vide

– une interprétation $\tilde{f}$ pour chaque $f \in \Sigma$ et $\tilde{p}$ pour chaque $p \in \Psi$

– une interprétation $I(x) \in \mathcal{U}$ pour chaque variable $x \in fv(P)$

tels que $[P]_I = T$

Définition

$A_1, \ldots, A_n \models B$ si tout modèle de A_1, …, A_n est un modèle de B

Théorème

$A_1, \ldots, A_n \vdash_{G3} B_1, \ldots, B_m$ ssi $A_1, \ldots, A_n \models B_1 \vee \ldots \vee B_m$

Notez que si la signature possède une constante de terme c, les deux notions de modèles coïncident puisque $\tilde{c}$ force $\mathcal{U}$ à ne pas être vide

Inconvénients de G3 au 1er ordre

Contrairement au cas propositionnel,

application (bottom-up) de certaines règles ne font pas diminuer le nombre de connecteurs

$\Longrightarrow$La hauteur des preuves d’un théorème donné n’a pas de borne.

$\Longrightarrow$La prouvabilité est indécidable (Théorème de Church).

procédure de semi-décision : un algorithme qui, étant donné un problème,

– s’arrêtera en répondant oui si sa réponse est oui

– s’arrêtera en répondant non ou ne s’arrêtera pas si sa réponse est non

Un algo. de semi-décision pour la prouvabilité :

on énumère tous les arbres décorés par des séquents,

on vérifie pour chacun s’il s’agit d’une preuve du théorème demandé

Très bête.

Inconvénients de G3 au 1er ordre

Peut-on faire plus intelligent ?

. . .mmm. . .de toute façon, les règles

$$\frac{\Gamma, (\forall x, P), \{^{t}\!/_{x}\} P \vdash \Delta}{\Gamma, (\forall x, P) \vdash \Delta} \qquad \frac{\Gamma \vdash \{^{t}\!/_{x}\} P, (\exists x, P), \Delta}{\Gamma \vdash (\exists x, P), \Delta}$$

nécessitent de sortir t du chapeau. Il semble nécessaire d’énumérer tous les t jusqu’à ce qu’on trouve le bon. . .

$r(986) \vdash (\exists y, r(y))$

Reprenons depuis le début

Soit A une formule du 1er ordre. On veut un algo (pas trop bête) de semi-décision qui réponde à la question :

A est-elle prouvable/valide ?

c’est-à-dire

$\neg A$ est-elle insatisfiable ?

c’est-à-dire

B est-elle insatisfiable ? . . .où B est. . .

une forme clausale de $\neg A$

Forme clausale

une forme prenexe de A :

une formule logiquement équivalente à A, de la forme

$$Q_1 x_1, \ldots, Q_n x_n, C$$

avec tous les quantificateurs $Q_1 \ldots Q_n$ en tête, C sans quantificateurs.

une forme prénexe skolémisée de A :

une formule close B de la forme

$$\forall y_1, \ldots, \forall y_m, D$$

avec D sans quantificateurs, telle que A est satisfiable ssi B satisfiable.
(Toutes les variables libres ou quantifiées existentiellement ont été substituées avec des nouveaux constructeurs de termes)

Forme clausale :

une forme normale conjonctive prénexe skolémisée de A :

la même chose en imposant que D est une grande conjonction de clauses, i.e. une formule de la forme

$$\forall y_1, \ldots, \forall y_m, (l_1^1 \vee \ldots \vee l_{p_1}^1) \wedge \ldots \wedge (l_1^q \vee \ldots \vee l_{p_q}^q)$$

(Rappelons qu'une clause est une disjonction de littéraux, et qu'un littéral l est une formule atomique ou la négation d'une formule atomique)
une forme clausale de A :
une conjonction de clauses closes, logiquement équivalente à une forme normale conjonctive prénexe skolémisée de A, i.e. de la forme :

$$(\forall x_1^1 \ldots \forall x_{k_1}^1, l_1^1 \vee \ldots \vee l_{p_1}^1) \wedge \ldots \wedge (\forall x_1^q \ldots \forall x_{k_q}^q, l_1^q \vee \ldots \vee l_{p_q}^q)$$

avec $x_j^i \in fv(l_1^i \vee \ldots \vee l_{p_i}^i)$

Forme clausale

L'existence d'une telle forme, pour toute formule A, résulte des exercices des TDs.

Exemple : $\neg(r(986) \Rightarrow \exists y, r(y))$ devient $r(986) \wedge \forall y, \neg r(y)$,
la seconde étant bien insatisfiable ssi la première l'est
(c'est-à-dire ssi $r(986) \Rightarrow \exists y, r(y)$ est valide)

Pour économiser de la place, sans perdre d'information logique,
on ne retient de

$$(\forall x_1^1 \ldots \forall x_{k_1}^1, C^1) \wedge \ldots \wedge (\forall x_1^q \ldots \forall x_{k_q}^q, C^q)$$

que l'ensemble des clauses $C_1, \ldots, C_n$, où la disjonction est associative + commutative (les clauses sont des multisets de littéraux)

Méthode de résolution

La méthode consiste à déduire de ces clauses de nouvelles clauses
On enrichit ainsi notre catalogue de clauses connues

Ceci par 2 règles :
La simplification

$$\frac{C \vee l \vee l}{C \vee l}$$

La résolution

$$\frac{C \vee r(t_1, \ldots, t_n) \qquad C' \vee \neg r(t'_1, \ldots, t'_n)}{\sigma(C \vee C')}$$

quel est σ ? une substitution qui unifie t_1 avec t'_1,..., t_n avec t'_n
on la note $\sigma = mgu(t_1 = t'_1, \ldots, t_n = t'_n)$
Exemple : avec les clauses $r(986)$ et $\neg r(y)$,
$\sigma = mgu(986 = y)$ est la substitution qui envoie y sur 986

Méthode de résolution

Conclusion : appliquer la règle de résolution aux $r(986)$ et $\neg r(y)$ vous donne la clause vide, i.e. la clause fausse

(rappel : $C \vee \bot \equiv C$)

. . .et donc notre ensemble de clauses $r(986)$ et $\neg r(y)$ était insatisfiable

. . .et donc la formule $\neg(r(986) \Rightarrow \exists y, r(y))$ était insatisfiable

. . .et donc la formule $r(986) \Rightarrow \exists y, r(y)$ était valide

plus généralement. . .

Correction et complétude

(pour les modèles non vides)

L'ensemble de clauses $C_1, \ldots, C_n$ est insatisfiable

si et seulement si

on peut déduire la clause vide en appliquant les règles de simplification et de résolution

Dernier exemple

Le buveur :

La formule $\exists x, (p(x) \Rightarrow \forall y, p(y))$, à valider, devient :

l'ensemble des clauses $p(x)$ et $\neg p(f(z))$, à insatisfier

J'applique ma règle de résolution :

j'obtiens pour $\sigma = mgu(x = f(z))$ la substitution qui envoie x sur $f(z)$

j'obtiens comme nouvelle clause : la clause vide

cqfd

mgu

Questions :

Existe-il toujours une substitution $mgu(t_1 = t'_1, \ldots, t_n = t'_n)$?

Comment l'obtiens-je dans les cas non-triviaux ?

hehe. . .**l'algorithme d'unification du 1er ordre**

Reste à faire

- L'algo d'unif
- Extension de la logique du 1er ordre avec égalité
- Clauses de Horn et description de Prolog
- Programmation logique d'opérations arithmétiques

Questions?

Cours 5 :

Logique du 1er ordre avec égalité

Logique du 1er ordre avec égalité

Symbole $=$/2 peut faire partie de la signature Ψ des propositions atomiques

Normalement, prédicats atomiques pas concernés par règles d'inférence (mais éventuellement régis par axiomes/hypothèses)
Pour $=$, c'est un peu différent. Connecteur spécial : on veut

$$((t = u) \wedge P(t)) \Rightarrow P(u)$$

pour toute proposition P Égalité de Leibniz

pas une hypothèse, mais **schéma d'hypothèses**

Solution : on traite $=$ par des règles d'inférence spécifiques

Extension de G3 pour $=$

$$\frac{}{\Gamma \vdash (t = t), \Delta} \qquad \frac{\Gamma, (t = u) \vdash (\{^{t}/_{x}\} P), \Delta}{\Gamma, (t = u) \vdash (\{^{u}/_{x}\} P), \Delta}$$

Exemple

Avec ces règles on démontre la symétrie et la transitivité de $=$:

$$\forall x, \forall y, (x = y) \Rightarrow (y = x)$$
$$\forall x, \forall y, \forall z, ((x = y) \wedge (y = z)) \Rightarrow (x = z)$$

Sémantique de cette extension

Très simple. Il faut interpréter le prédicat $=$ dans le monde sémantique.

Une manière canonique :

pour $a, b \in \mathcal{U}$, on définit

$\tilde{=}(a, b) \quad = T$ si a et b sont égaux (dans $\mathcal{U}$)

$\tilde{=}(a, b) \quad = F$ sinon

Le système G3 étendu avec les deux règles pour $=$ est correct et complet vis-à-vis de cette sémantique (où $\tilde{=}$ est fixe)

Questions?