Logique formelle & Programmation logique

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

Bug du Pentium P5, 1995 : 475 millions de dollars

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 plusieurs cas

- quand des vies humaines sont en jeu

  (cf systèmes embarqués, pilotes automatiques, ligne 14-Meteor)
- quand beaucoup d'argent est en jeu

  (cf Ariane, transactions financières)
- quand les informations sont confidentielles

  (eg correction de protocole cryptographique)

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 0.5 :

Pré-requis à ce cours

Définition par récurrence (faible) sur les entiers

Exemple : La suite géométrique

$$\begin{array}{lll} f(0) & := \ldots & u_0 := 0 \\ f(n+1) & := \ldots f(n) \ldots & u_{n+1} := 2 + u_n \end{array}$$

définit une fonction sur tous les entiers (aussi appelée suite).

Correspond à des programmes récursifs. Le même exemple en Java ou C :

```
int geo2(int n){
   if (n==0) return 0;
   return 2+geo2(n-1);
}
```

définit bien une fonction sur tous les entiers car les appels récursifs terminent.

Principe de récurrence (faible) sur les entiers

Soit $\mathcal{P}$ une propriété qu'un entier peut avoir ou ne pas avoir.

Exemple : $\mathcal{P}$ est "être pair"

"n satisfait la propriété $\mathcal{P}$" se note $\mathcal{P}(n)$

Principe de récurrence "faible"

Si $\mathcal{P}(0)$

et si pour tout entier n, $\mathcal{P}(n)$ implique $\mathcal{P}(n+1)$

alors pour tout entier n, on a $\mathcal{P}(n)$.

Exemple :

Prouvez que pour tout entier n, on a $u_n = 2 * n$

Définition par récurrence (forte) sur les entiers

$f(n) := \ldots f(i) \ldots$ où $i < n$

Exemple : La suite

$$\begin{aligned} v_0 &:= 0 \\ v_n &:= v_{n/2} + 1 \quad n \geq 1 \end{aligned}$$

définit aussi une fonction sur tous les entiers (aussi appelée suite).

Correspond à des programmes récursifs. Le même exemple en Java ou C :

```
int suite(int n){
   if (n==0) return 0;
   return suite(n/2)+1;
}
```

définit bien une fonction sur tous les entiers car les appels récursifs terminent.

Principe de récurrence (fort) sur les entiers

Si pour tout entier n, $(\forall i < n, \mathcal{P}(i))$ implique $\mathcal{P}(n)$

alors pour tout entier n, on a $\mathcal{P}(n)$.

Exemple :

Prouvez que pour tout entier n, on a $v_n \leq n$

Les entiers comme structure inductive (libre)

Induction = récurrence en anglais.

Les 5 axiomes de Peano

- 0 est un entier
- Si n est un entier,

  alors $S(n)$ est un entier.

"Construction inductive" des entiers

- 0 n'est le successeur d'aucun entier
- Si $S(n) = S(m)$ alors $n = m$

Structure **libre**

= injectivité des constructeurs

- Principe de récurrence (faible ou fort)

. . .définissent le comportement des entiers naturels.

Autres structures inductives (libres)

Ce qu'on peut faire avec les entiers, on peut le faire avec autre chose.

Définition des listes d'entiers :

– `nil`, la liste vide, est une liste.
– Si l est une liste et n un entier, alors $n :: l$ est une liste
  (de tête n et de queue l)

Définition de la taille d'une liste, "par récurrence sur la liste" :

– la taille de `nil` est 0.
– la taille de $n :: l$ est la taille de l plus 1.

On pourrait poser des principes de récurrence sur les listes
mais en général, on se ramène à ceux sur les entiers via la notion de taille.

Autres structures inductives (libres)

Définition des arbres étiquetés :

– une feuille, étiquetée par un symbole s, est un arbre

– Si A_1,..., A_n sont des arbres, et f est un symbole,
  alors $f(A_1, \ldots, A_n)$ est un arbre.

Voir dessin au tableau.

(Exemple de) définition de la taille d'un arbre, "par récurrence sur l'arbre" :

– la taille d'une feuille est 1.

– la taille de $f(A_1, \ldots, A_n)$ est la somme des tailles de A_1,..., A_n,
  plus 1.

On pourrait poser des principes de récurrence sur les structures inductives mais en général, on se ramène à ceux sur les entiers.

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 :

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

où A, B dénotent des formules arbitraires

Instances (exemples) :

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

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 :

- les entiers
- les réels
- les ensembles

Exemples de termes

$0, S(0), 3 + 4$

$3/4, \pi$

$\emptyset, A \cup B$

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

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

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

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

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 : difficile pour ce cours
Il faut construire un modèle, avec un $\mathcal{U}$ et un $\tilde{}$! ! !

A partir de quoi ? la syntaxe elle-même. . .

Questions?

Cours 4 :

Logique du 1er ordre : les bonnes questions à se poser

Un petit point sur le buveur : et si le bar est vide ?

Si le bar est vide, le théorème est faux.

Deux visions :

Premième vision : à la “Logique du premier ordre”

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

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

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

Le théorème du buveur $\vdash \exists x, (p(x) \Rightarrow \forall y, p(y))$ est prouvable.

Un petit point sur le buveur : variante

Deuxième vision : à la “Théorie des types”

on enregistre à quelles variables libres on a droit :

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

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

Un petit point sur le buveur : variante

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

Les deux versions ne prouvent pas les mêmes théorèmes !

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

Sémantique de la première version

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

$A \models B$ si tout modèle de A est un modèle de B

Théorème

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

Sémantique de la première version

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

$A \models B$ si tout modèle de A est un modèle de B

Théorème

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

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

Décidabilité

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 décision :

un algorithme qui, étant donné une instance d'un problème,

– s'arrêtera en répondant oui si la réponse est oui

– s'arrêtera en répondant non si la réponse est non

Procédure de semi-décision :

un algorithme qui, étant donné une instance d'un problème,

– s'arrêtera en répondant oui si la réponse est oui

– s'arrêtera en répondant non ou ne s'arrêtera pas si la réponse est non

Algorithmes 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 très bête.

Peut-on faire plus intelligent ? de toute façon, les règles

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

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

– On applique les règles de G3 en énumérant tous les témoins possibles

jusqu'à trouver une preuve du théorème demandé Très bête.

Peut-on faire mieux ? example : $r(986) \vdash (\exists y, r(y))$

Variables existentielles

On retarde le choix des témoins jusqu'à avoir plus d'informations pour les choisir

Variables existentielles...

Questions?

Cours 5 :

Variables existentielles et Unification

Système G3 avec variables existentielles : 1ère tentative

$$\frac{\Gamma \vdash P, \Delta}{\Gamma \vdash (\forall x, P), \Delta}\, x \notin fv(\Gamma, \Delta) \qquad \frac{\Gamma, (\forall x, P), \left\{{}^{?\mathbf{x}}/_{x}\right\} P \vdash \Delta}{\Gamma, (\forall x, P) \vdash \Delta}\, ?\mathbf{x} \notin ev(\Gamma, \Delta)$$

$$\frac{\Gamma \vdash \left\{{}^{?\mathbf{x}}/_{x}\right\} P, (\exists x, P), \Delta}{\Gamma \vdash (\exists x, P), \Delta}\, ?\mathbf{x} \notin ev(\Gamma, \Delta) \qquad \frac{\Gamma, P \vdash \Delta}{\Gamma, (\exists x, P) \vdash \Delta}\, x \notin fv(\Gamma, \Delta)$$

On doit donc enrichir notre syntaxe de terms :

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

$ev(t)$ et $ev(\Gamma)$ sont les équivalents de $fv(t)$ et $fv(\Gamma)$ pour les variables existentielles.

Axiome et Substitution

Exemple : $\dfrac{r(986) \vdash r(?y), (\exists y, r(y))}{r(986) \vdash \exists y, r(y)}$

On a envie de dire : j'ai gagné la branche en instanciant $?y$ par 986

Plus généralement, que faire lors d'un axiome ?

$$\frac{}{\Gamma, p(t_1, \ldots, t_n) \vdash p(u_1, \ldots, u_n), \Delta}$$

Ce serait bien d'instancier toutes les variables existentielles afin que,

pour tout i tel que $1 \leq i \leq n$, on ait $t_i = u_i$.

Substitution : fonction partielle des variables existentielles vers les termes

($\sigma(?x) = t$), que l'on étend facilement aux termes ($\sigma(u) = t$) ainsi :

$$\begin{array}{lll} \sigma(f(t_1, \ldots, t_n)) & = f(\sigma(t_1), \ldots, \sigma(t_n)) & \\ \sigma(x) & = x & \\ \sigma(?x) & = ?x & \text{si } ?x \notin \text{domaine}(\sigma) \end{array}$$

Unificateur et Propagation

Formellement, on cherche donc une substitution σ tel que pour tout i tel que $1 \leq i \leq n, \sigma(t_i) = \sigma(u_i)$.

σ est une solution du problème d'**unification** $t_1 = u_1, \ldots, t_n = u_n$

σ est un **unificateur**

Exemple : Soit $A = \forall x, p(x, x)$ and $B = \exists y, (p(y, 0) \wedge p(y, S(0)))$

$$\dfrac{\dfrac{\dfrac{\text{ok avec } \sigma(?y) = \sigma(?x) = 0}{A, p(?x, ?x) \vdash p(?y, 0), B}}{A \vdash p(?y, 0), B} \quad \dfrac{\dfrac{\text{ok avec } \sigma'(?y) = \sigma'(?x') = S(0)}{A, p(?x', ?x') \vdash p(?y, S(0)), B}}{A \vdash p(?y, S(0)), B}}{\dfrac{A \vdash p(?y, 0) \wedge p(?y, S(0)), B}{A \vdash B}}$$

σ et σ' incompatibles : impossible de reconstruire de preuve dans G3.

Dès qu'on choisit l'un, il faut propager ce choix dans l'autre branche.

Système G3 avec variables existentielles : 2ème tentative

On regroupe l'état de toutes les branches ouvertes
$(\Gamma_1 \vdash \Delta_1) \ldots (\Gamma_n \vdash \Delta_n)$ dans une structure de données :

$$\Gamma_1 \vdash \Delta_1 \;\wr\; \ldots \;\wr\; \Gamma_n \vdash \Delta_n$$

$$\frac{\mathcal{S} \wr \Gamma \vdash P, \Delta}{\mathcal{S} \wr \Gamma \vdash (\forall x, P), \Delta}\, x \notin fv(\Gamma, \Delta) \qquad \frac{\mathcal{S} \wr \Gamma, (\forall x, P), \{{}^{?\mathbf{x}}/_x\} P \vdash \Delta}{\mathcal{S} \wr \Gamma, (\forall x, P) \vdash \Delta}\, ?\mathbf{x} \notin ev(\Gamma, \Delta)$$

$$\frac{\mathcal{S} \wr \Gamma \vdash \{{}^{?\mathbf{x}}/_x\} P, (\exists x, P), \Delta}{\mathcal{S} \wr \Gamma \vdash (\exists x, P), \Delta}\, ?\mathbf{x} \notin ev(\Gamma, \Delta) \qquad \frac{\mathcal{S} \wr \Gamma, P \vdash \Delta}{\mathcal{S} \wr \Gamma, (\exists x, P) \vdash \Delta}\, x \notin fv(\Gamma, \Delta)$$

$$\frac{\sigma(\mathcal{S})}{\mathcal{S} \wr \Gamma, p(t_1, \ldots, t_n) \vdash p(u_1, \ldots, u_n), \Delta}$$

pour tout i tel que $1 \leq i \leq n$,
$\sigma(t_i) = \sigma(u_i)$
pour une certaine substitution σ
qui instancie les variables existentielles

. . .et les règles propositionnelles sont adaptées ainsi

Connect.	Règle d'intro gauche	Règle d'intro droite
$\top, \bot$	$\dfrac{\mathcal{S}}{\mathcal{S} \wr \Gamma, \bot \vdash \Delta}$	$\dfrac{\mathcal{S}}{\mathcal{S} \wr \Gamma \vdash \top, \Delta}$
$\neg$	$\dfrac{\mathcal{S} \wr \Gamma \vdash A, \Delta}{\mathcal{S} \wr \Gamma, \neg A \vdash \Delta}$	$\dfrac{\mathcal{S} \wr \Gamma, A \vdash \Delta}{\mathcal{S} \wr \Gamma \vdash \neg A, \Delta}$
$\vee$	$\dfrac{\mathcal{S} \wr \Gamma, A \vdash \Delta \quad \wr \quad \Gamma, B \vdash \Delta}{\mathcal{S} \wr \Gamma, A \vee B \vdash \Delta}$	$\dfrac{\mathcal{S} \wr \Gamma \vdash A, B, \Delta}{\mathcal{S} \wr \Gamma \vdash A \vee B, \Delta}$
$\wedge$	$\dfrac{\mathcal{S} \wr \Gamma, A, B \vdash \Delta}{\mathcal{S} \wr \Gamma, A \wedge B \vdash \Delta}$	$\dfrac{\mathcal{S} \wr \Gamma \vdash A, \Delta \quad \wr \quad \Gamma \vdash B, \Delta}{\mathcal{S} \wr \Gamma \vdash A \wedge B, \Delta}$
$\Rightarrow$	$\dfrac{\mathcal{S} \wr \Gamma \vdash A, \Delta \quad \wr \quad \Gamma, B \vdash \Delta}{\mathcal{S} \wr \Gamma, A \Rightarrow B \vdash \Delta}$	$\dfrac{\mathcal{S} \wr \Gamma, A \vdash B, \Delta}{\mathcal{S} \wr \Gamma \vdash A \Rightarrow B, \Delta}$

Unification

Revenons aux unificateurs.

Questions :

Existe-il toujours un unificateur au problème $t_1 = t'_1, \ldots, t_n = t'_n$?

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

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

Algorithme d'Unification (Robinson)

$$mgu(f(t_1,\ldots,t_n) = f(t'_1,\ldots,t'_n), E) \quad = mgu(t_1 = t'_1,\ldots,t_n = t'_n, E)$$

$$mgu(f(t_1,\ldots,t_n) = g(t'_1,\ldots,t'_m), E) \quad = \text{Fail}$$

$$mgu(t = t, E) \quad = mgu(E)$$

$$mgu(?x = t, E) \quad = (?x \mapsto \sigma(t)) \cup \sigma$$

où $\sigma = mgu(\{^{t}/_{?x}\}\, E)$

si $?x \notin ev(t)$

$$mgu(?x = t, E) \quad = \text{Fail}$$

sinon

$$mgu(t = ?x, E) \quad = mgu(?x = t, E)$$

si t n'est pas une variable existentielle

$$mgu() \quad = \emptyset$$

Est-ce fini ?

Exemple : Soit $P_1 = \forall z, p(z, z)$ and $P_2 = \exists x, \forall y, p(x, S(y))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P_1, p(?z, ?z) \vdash (p(?x, S(y))), P_2}}{P_1 \vdash (p(?x, S(y))), P_2}}{P_1 \vdash (\forall y, p(?x, S(y))), P_2}}{P_1 \vdash P_2}}{\vdash P_1 \Rightarrow P_2}$$

avec

$$mgu(?z = ?x, ?z = S(y)) : \quad ?z \mapsto S(y)$$
$$?x \mapsto S(y)$$

Le témoin pour x ne pouvait pas utiliser y, libéré plus tard !

L'astuce qui tue

Exemple : Soit $P_1 = \forall z, p(z, z)$ and $P_2 = \exists x, \forall y, p(x, S(y))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P_1, p(?z, ?z) \vdash_{?x} (p(?x, S(y(?x)))), P_2}}{P_1 \vdash_{?x} (p(?x, S(y(?x)))), P_2}}{P_1 \vdash_{?x} (\forall y, p(?x, S(y))), P_2}}{P_1 \vdash P_2}$$

$$\vdash P_1 \Rightarrow P_2$$

pas ok, car aucun unificateur pour $?z = ?x, ?z = S(y(?x))$

$(mgu(?z = ?x, ?z = S(y(?x))) =$ Fail)

Système G3 avec variables existentielles

Cette fois-ci c'est la bonne !

$$\frac{\mathcal{S} \wr \Gamma \vdash_\Phi \left\{{}^{x(\Phi)}/_{x}\right\} P, \Delta}{\mathcal{S} \wr \Gamma \vdash_\Phi (\forall x, P), \Delta}\, x \notin fv(\Gamma, \Delta)$$

$$\frac{\mathcal{S} \wr \Gamma, (\forall x, P), \left\{{}^{?\mathbf{x}}/_{x}\right\} P \vdash_{\Phi, ?\mathbf{x}} \Delta}{\mathcal{S} \wr \Gamma, (\forall x, P) \vdash_\Phi \Delta}\, ?\mathbf{x} \notin ev(\Gamma, \Delta)$$

$$\frac{\mathcal{S} \wr \Gamma \vdash_{\Phi, ?\mathbf{x}} \left\{{}^{?\mathbf{x}}/_{x}\right\} P, (\exists x, P), \Delta}{\mathcal{S} \wr \Gamma \vdash_\Phi (\exists x, P), \Delta}\, ?\mathbf{x} \notin ev(\Gamma, \Delta)$$

$$\frac{\mathcal{S} \wr \Gamma, \left\{{}^{x(\Phi)}/_{x}\right\} P \vdash_\Phi \Delta}{\mathcal{S} \wr \Gamma, (\exists x, P) \vdash_\Phi \Delta}\, x \notin fv(\Gamma, \Delta)$$

$$\frac{\sigma(\mathcal{S})}{\mathcal{S} \wr \Gamma, p(t_1, \ldots, t_n) \vdash_\Phi p(u_1, \ldots, u_n), \Delta}\, \sigma = mgu(t_1 = u_1, \ldots t_n = u_n)$$

Questions?

Cours 6 :

Forme clausale, Prolog

Litéral, clause, forme clausale

Définitions :

– litéral (noté $l, l', \ldots$) = formule atomique (litéral positif)

ou négation de formule atomique (litéral négatif)

– clause (notée $C, C', \ldots$) = disjonction de litéraux

(en général) universellement quantifiée

$\forall x_1 \ldots \forall x_k, l_1 \vee \ldots \vee l_p$ avec $\{x_1, \ldots x_k\} = fv(l_1 \vee \ldots \vee l_p)$

Soit A une formule du 1er ordre (close).

forme clausale de A = ensemble fini de clauses $C_1, \ldots, C_n$ telles que $A \vdash \bot$ ssi $C_1, \ldots, C_n \vdash \bot$.

Beaucoup de techniques de raisonnement automatique utilisent ces formes clausales

Question : A possède-t-elle toujours une telle forme clausale ?

Mise sous formes clausale

Concrètement, on cherche une formule B 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 $\{x_1^i, \ldots, x_{k_i}^i\} = fv(l_1^i \vee \ldots \vee l_{p_i}^i)$ et telle que $A \vdash \bot$ ssi $B \vdash \bot$.

4 étapes

Mise sous formes 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 $A \vdash \bot$ ssi $B \vdash \bot$.

(Toutes les variables libres ou quantifiées existentiellement ont été substituées avec des nouveaux constructeurs de termes)

Mise sous formes clausale

une forme normale conjonctive prénexe skolémisée de A :

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

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

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 $\{x_1^i, \ldots, x_{k_i}^i\} = fv(l_1^i \vee \ldots \vee l_{p_i}^i)$

Mise sous forme clausale

Les transformations permettant chacune des 4 étapes, pour toute formule A, sont dans les 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)

ProLog (pour Programmation Logique)

ProLog : prouveur qui implémente G3 avec variables existentielles,

restreint aux séquents de la forme $C_1, \ldots, C_n \vdash l_1 \wedge \ldots \wedge l_p$

où $C_1, \ldots, C_n$ sont des Clauses de Horn qui possèdent 1 litéral positif

$l_1, \ldots, l_p$ sont des litéraux positifs avec variables existentielles

Clause de Horn : clause où au plus un litéral est positif

Syntaxe :

`l:-l1,...,ln.` pour $(l \vee (\neg l_1) \vee \ldots \vee (\neg l_n))$

ou encore $(l_1 \wedge \ldots \wedge l_n) \Rightarrow l$

Conjonction de litéraux $l_1 \wedge \ldots \wedge l_n$ à prouver, la requête, est écrite

`?:-l1,...,ln.`

Propriété de ce fragment de G3 :

La partie gauche du séquent reste invariante tout au long de la preuve.

C'est le programme. Il est donné dès le début dans un fichier `.pl`

Questions?