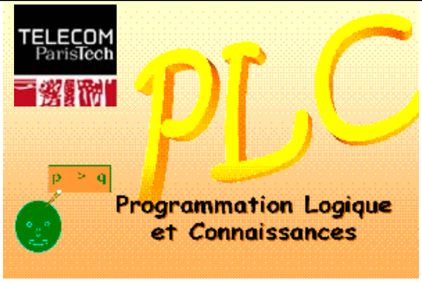

Logique des prédicats (ordre 1)

Jean-Louis Dessalles

INF-221

janvier 15

« Tous les hommes sont mortels »

<h1_humain, h2_humain, h3_humain, h4_humain, h5_humain, h6_humain, h7_humain, h8_humain, …>

$(\forall h)\ (humain(h) \supset mortel(h))$

Quantification

$(\forall x)\ (\exists y)$

Gottlob Frege

- « 4 est précédé par 2 »
  « 2 précède 4 »
- « tout instant est précédé par un instant »
  « un instant précède tout instant »
- « aucun homme n'est immortel »

« Deux individus quelconques communiquent s'il existe un interprète »

$(\forall x)\ (\forall y)\ (\forall l_1)\ (\forall l_2)\ ((p(x, l_1) \wedge (p(y, l_2) \wedge (\exists z)\ (p(z, l_1) \wedge p(z, l_2)))) \supset c(x, y))$

$(\exists x)\ (\exists y)\ (\exists l_1)\ (\exists l_2)\ ((p(x, l_1) \wedge (p(y, l_2) \wedge (\exists z)\ (p(z, l_1) \wedge p(z, l_2)))) \wedge \neg c(x, y))$

« Il y a deux individus qui ne communiquent pas malgré l'existence d'un interprète »

Lien avec les logiques de description

$\exists r.C = (\exists y)\ (r(x,y) \wedge C(y))$

$\forall r.C = (\forall y)\ (r(x,y) \supset C(y))$

$\text{Mother} = \text{Woman} \sqcap \exists\text{hasChild.Person}$

$\text{Mother}(x) \equiv \text{Woman}(x) \wedge (\exists y)\ (\text{hasChild}(x,y) \wedge \text{Person}(y))$

Langage d'ordre 1

Un langage du premier ordre ***L*** est la donnée de :

un ensemble de constantes c_1, c_2, … ;
un ensemble de foncteurs f, symboles de fonction à n places ;
un ensemble de prédicats P, symboles de relation à n places ;
variables v_1, v_2, … ;
les quantificateurs $\forall$, $\exists$;
les connecteurs de la logique propositionnelle $\neg \wedge \vee \supset \subset \uparrow \downarrow \not\subset \not\supset$.

Syntaxe: termes et formules

Dans un langage du premier ordre ***L*** :

Pour toute constante c, c est un terme ;
Pour toute variable v, v est un terme ;
Pour toutes série de termes $t_1, t_2, \ldots,$ et tout foncteur f à n places
$f(t_1, t_2, \ldots, t_n)$ est un terme ;

T et $\perp$ sont des formules atomiques ;
pour toutes série de termes $t_1, t_2, \ldots,$ et tout prédicat P à n places, $P(t_1, t_2, \ldots, t_n)$ est une formule atomique ;

si F est une formule et si ◊ est un quantificateur alors (◊v)F est une formule;
si F $\in$ ***F***, alors $\neg$F est une formule ;
si F_1 et F_2 sont des formules, et si o est un connecteur à deux places, alors F_1 o F_2 est une formule.

Syntaxe

Un terme est *fermé* si et seulement s'il ne contient aucune variable.

Une variable v est libre dans une formule atomique F si et seulement si v a une occurrence dans F.
Soit ◊ un quantificateur, une variable v est libre dans (◊v')F si et seulement si v est libre dans F et v est différente de v'.
Une variable v est libre dans $\neg$F si et seulement si v est libre dans F.
Soit o un connecteur à deux places, une variable v est libre dans F_1 o F_2 si et seulement si v est libre dans F_1 ou dans F_2.

Une variable v est liée dans une formule F si et seulement si elle n'y a aucune occurrence libre.
Une formule est *fermée* si et seulement si elle ne contient aucune variable libre.

Sémantique: modèle

Un modèle ***M***(***D,I***) pour un langage du premier ordre ***L*** est la donnée d'un ensemble non vide ***D***, appelé *domaine*, et d'une fonction ***I***, appelée *interprétation*, qui associe:

à toute symbole de constante c de ***L*** un individu du domaine ***I***(c) = $c^I \in \boldsymbol{D}$;
à chaque foncteur f de ***L*** à n places une fonction d'arité n du domaine ***I***(f) = $f^I : \boldsymbol{D}^n \rightarrow \boldsymbol{D}$;
à chaque prédicat P de ***L*** à n places une relation d'arité n du domaine ***I***(f) = $\boldsymbol{P}^I \subset \boldsymbol{D}^n$.

Pour un langage ***L***, une *assignation* ***A*** dans un modèle ***M***(***D,I***) associe à chaque terme t la valeur $t^{I,A}$ telle que :

pour une variable v, $v^{I,A} = v^A$;
pour une constante c, $c^{I,A} = c^I$;
pour chaque foncteur f, $[f(t_1, , \ldots t_n)]^{I,A} = f^I(t_1^{I,A}, t_2^{I,A}, \ldots t_n^{I,A})$.

Attitudes

Étant donnée une assignation ***A*** dans un modèle ***M(D,I)*** pour un langage du premier ordre ***L*** , on associe à chaque formule *F* de ce langage la valeur de vérité $F^{I,A} \in \{\mathbf{v},\mathbf{f}\}$ telle que :

$[P(t_1, , \ldots t_n)]^{I,A} = \mathbf{v}$ si et seulement si le n-uplet $<t_1^{I,A}, t_2^{I,A}, \ldots t_n^{I,A}> \in P^I$;
$T^{I,A} = \mathbf{v}$; $\perp^{I,A} = \mathbf{f}$;
$[(\forall v)\ F]^{I,A} = \mathbf{v}$ si et seulement si $F^{I,B} = \mathbf{v}$ pour toute assignation ***B*** coïncidant avec ***A*** sauf éventuellement en *v* ;
$[(\exists v)\ F]^{I,A} = \mathbf{v}$ si et seulement si $F^{I,B} = \mathbf{v}$ pour une assignation ***B*** coïncidant avec ***A*** sauf éventuellement en *v* ;
$[\neg F]^{I,A} = \mathbf{non}(F^{I,A})$;
$[F_1\ o\ F_2]^{I,A} = F_1^{I,A} \bullet F_2^{I,A}$ pour o et $\bullet$ associés.

Satisfiabilité, Vérité, Validité

Une formule F d'un langage du premier ordre ***L*** est *vraie* dans un modèle ***M***(***D***,***I***) si et seulement si $F^{I,A} = \mathbf{v}$ pour toute assignation ***A*** des symboles de ce langage dans ce modèle.

Une formule F d'un langage du premier ordre ***L*** est *valide* si et seulement si F est vraie dans tout modèle de ce langage.

Un ensemble *S* de formules d'un langage du premier ordre ***L*** est *satisfiable dans* un modèle ***M***(***D***,***I***) si et seulement s'il existe une assignation ***A*** des symboles de ce langage dans ce modèle telle que $F^{I,A} = \mathbf{v}$ pour toute formule $F \in S$.

Un ensemble *S* de formules d'un langage du premier ordre ***L*** est *satisfiable* si et seulement si il existe un modèle de ce langage dans lequel *S* est satisfiable.

Noter qu'une formule Φ est *valide* si et seulement si $\{\neg\Phi\}$ n'est pas satisfiable.

Modèle de Herbrand

Un modèle ***M***(***D***,***I***) pour un langage du premier ordre ***L*** est un modèle de Herbrand si et seulement si :

D ne contient que des termes fermés de ***L*** ;
pour tout terme fermé *t*, $t^I = t$.

Dans un modèle de Herbrand :

Pour toute formule *F*, $(\forall v)\ F$ est vraie si et seulement si $F\{v/d\}$ est vraie pour tout $d \in D$;
Pour toute formule *F*, $(\exists v)\ F$ est vraie si et seulement si $F\{v/d\}$ est vraie pour un $d \in D$.

$F\{v/d\}$ est le résultat de la substitution de *v* par *d* dans *F*.
Noter que *F(x/d)* reste une formule !

Remplacement des Quantificateurs

Soit $\boldsymbol{M}(\boldsymbol{D},\boldsymbol{I})$ un modèle de Herbrand pour un langage du premier ordre $\boldsymbol{L}$:

Si γ est une formule de $\boldsymbol{L}$, γ est vraie dans $\boldsymbol{M}$ si et seulement si $\gamma(d)$ est vraie pour tout $d \in \boldsymbol{D}$;
Si δ est une formule de $\boldsymbol{L}$, δ est vraie dans $\boldsymbol{M}$ si et seulement si $\delta(d)$ est vraie pour un $d \in \boldsymbol{D}$.

γ	$\gamma(t)$
$(\forall v)\ \Phi$	$\Phi\{v/t\}$
$\neg(\exists\ v)\ \Phi$	$\neg\Phi\{v/t\}$

δ	$\delta(t)$
$(\exists\ v)\ \Phi$	$\Phi\{v/t\}$
$\neg(\forall v)\ \Phi$	$\neg\Phi\{v/t\}$

Procédures de preuve par réfutation

Les arbres de preuve et l'algorithme de résolution sont parfaitement applicables en logique de premier ordre 1, à condition de permettre les insertions suivantes :

Si une formule δ est présente, on peut insérer δ(c) où c est une nouvelle constante additionnelle, qui n'existe pas dans le langage.

Si une formule γ est présente, on peut insérer γ(t) où t est n'importe quel terme fermé (pouvant comporter des constantes additionnelles).

Exemple : Résolution de $(\forall x)\ (P(x) \vee Q(x)) \supset ((\exists x)\ P(x) \vee (\forall x)\ Q(x))$

Résolution de $((\forall x)\ (P(x) \vee Q(x)) \supset ((\exists x)\ P(x) \vee (\forall x)\ Q(x)))$

1. $[\ \neg((\forall x)\ (P(x) \vee Q(x)) \supset ((\exists x)\ P(x) \vee (\forall x)\ Q(x)))\ ]$
2. $[\ (\forall x)\ (P(x) \vee Q(x))\ ]$
3. $[\ \neg((\exists y)\ P(y) \vee (\forall z)\ Q(z))\ ]$
4. $[\ \neg(\exists y)\ P(y)\ ]$
5. $[\ \neg(\forall z)\ Q(z)\ ]$
6. $[\ \neg\ Q(c)\ ]$ introduction d'une constante additionnelle
7. $[\ \neg P(c)\ ]$ utilisation de la même constante additionnelle
8. $[\ P(c), Q(c)\ ]$ à partir de 2.
9. $[\ Q(c)\ ]$ résolvante de 7. et 8.
10. $[\ ]$ résolvante de 6. et 9.

Systèmes axiomatiques en logique d'ordre 1

Au système axiomatique de Hilbert qui engendre les tautologies de la logique propositionnelle, on adjoint une règle d'inférence et un schéma d'axiome :

- sh. axiome 10 : $\gamma \supset (\gamma(t))$ pour tout terme fermé t comportant éventuellement des constantes additionnelles.
- règle d'inférence : **généralisation universelle**
  Si c est une constante additionnelle qui n'apparaît pas dans γ, à partir de $\gamma(c)$, produire γ.

Exemple : Preuve de $(\forall x)\ (P(x) \wedge Q(x)) \supset (\forall x)\ P(x)$

Exemple : Preuve de $(\forall x)\ (P(x) \wedge Q(x)) \supset (\forall x)\ P(x)$

1.	$(\forall x)\ (P(x) \wedge Q(x)) \supset (P(c) \wedge Q(c))$	sh. axiome 10.
2.	$(P(c) \wedge Q(c)) \supset P(c)$	théorème propositionnel
3.	$(\forall x)\ (P(x) \wedge Q(x)) \supset P(c)$	résulte de 1. et 2. en logique propositionnelle
4.	$(\forall x)\ (P(x) \wedge Q(x)) \supset (\forall x)\ P(x)$	règle de généralisation universelle

Preuves en logique d'ordre 1

- Il existe des systèmes de preuve corrects et complets pour la logique des prédicats (Gödel 1930)
  - Systèmes axiomatiques
  - Arbres
  - Résolution
  - ...
- Il n'existe que des algorithmes semi-décidables (l'ensemble des théorèmes est récursivement énumérable, mais n'est en général pas récursif)

Lemme d'Hintikka

Un ensemble S de formules fermées d'un langage du premier ordre est un ensemble d'Hintikka si et seulement si :

Pour toute formule fermée F, soit $F \notin H$, soit $\neg F \notin H$;
$\perp \notin H$; $\neg T \notin H$;
Si $\neg\neg F \in H$, alors $F \in H$;
Si $\alpha \in H$, alors $\alpha_1 \in H$ et $\alpha_2 \in H$;
Si $\beta \in H$, alors $\beta_1 \in H$ ou $\beta_2 \in H$;
Si $\gamma \in H$, alors $\gamma\ (t) \in H$ pour tout terme fermé t ;
Si $\delta \in H$, alors $\delta\ (t) \in H$ pour un terme fermé t ;

Un ensemble de Hintikka H est satisfiable dans un modèle de Herbrand ***M***.

Forme prénexe

Dans les formules qui suivent, la variable v est supposée ne pas avoir d'occurrence dans la partie de la formule qui est hors de la portée du quantificateur ; ceci peut être obtenu par un renommage des variables.
Les formules qui suivent sont valides :

$\neg (\exists v)\ A \equiv (\forall v) \neg A$
$\neg (\forall v)\ A \equiv (\exists v) \neg A$
$((\forall v)\ A \wedge B) \equiv (\forall v)\ (A \wedge B)$
$(A \wedge (\forall v)\ B) \equiv (\forall v)\ (A \wedge B)$
$((\exists v)\ A \wedge B) \equiv (\exists v)\ (A \wedge B)$
$(A \wedge (\exists v)\ B) \equiv (\exists v)\ (A \wedge B)$
$((\forall v)\ A \supset B) \equiv (\exists v)\ (A \supset B)$
$(A \supset (\forall v)\ B) \equiv (\forall v)\ (A \supset B)$
$((\exists v)\ A \supset B) \equiv (\forall v)\ (A \supset B)$
$(A \supset (\exists v)\ B) \equiv (\exists v)\ (A \supset B)$

Mettre sous forme prénexe:
$(\exists x)\ (\forall y)\ R(x,y) \supset (\forall y)\ (\exists x)\ R(x,y)$

Skolémisation

Une formule sous forme prénexe :
$(Q_1 v_1)\ (Q_2 v_2) \ldots (\exists_k v_k) \ldots (Q_n v_n)\ F$
est transformée en
$(Q_1 v_1)\ (Q_2 v_2) \ldots\ldots (Q_n v_n)\ F\{v_k/f(v_1, v_2, \ldots v_{k-1})\}$
où f est un nouveau foncteur appelée la fonction de Skolem.
Les variables libres doivent également être incluses dans la fonction de Skolem.

La première formule est satisfiable si et seulement si la deuxième formule l'est.

Mettre sous forme prénexe et skolémisée :
$(\forall x)\ (\forall y)\ (p(x) \wedge p(y)) \supset (\forall x)\ (\forall y)\ (p(x) \vee p(y))$

Procédure de Preuve Pratique

Pour prouver la validité d'une formule F, on effectue généralement les opérations suivantes :

renommer les variables si nécessaire
mettre $\neg F$ sous forme prénexe,
skolémiser $\neg F$,
supprimer les quantificateurs,
mettre sous forme normale,
appliquer une procédure de preuve par réfutation,
appliquer l'algorithme d'unification.

Tester la validité de la formule : $(\forall x)\ (\exists y)\ (\forall z)\ (\exists w)\ (R(x,y) \vee \neg R(w,z))$

Substitutions

Projection des variables sur des termes $\sigma: V \rightarrow T$ dont on peut étendre l'action sur l'ensemble des termes:
$c\sigma = c$;
$[f(t_1, t_2, \ldots, t_n)]\sigma = f(t_1\sigma, t_2\sigma, \ldots, t_n\sigma)$;

Composition : $v(\sigma\tau) = (v\sigma)\tau$.
Pour tout terme t, $t(\sigma\tau) = (t\sigma)\tau$.
La composition des substitutions est associative $(\sigma\tau)\omega = \sigma(\tau\omega)$.

Pour σ une substitution et v une variable on définit une substitution σ_v identique à σ pour toute variable v' différent de v ($v'\sigma_v = v'\sigma$ si $v' \neq v$) et qui ne change pas la variable v ($v\sigma_v = v$).

Substitutions

On peut aussi étendre l'action d'une substitution sur l'ensemble des formules:

$[P(t_1, t_2, \ldots, t_n)]\sigma = P(t_1\sigma, t_2\sigma, \ldots, t_n\sigma)$;
$T\sigma = T$; $\bot\sigma = \bot$;
$[(\Diamond v)\ F]\sigma = (\forall\Diamond)F\sigma_v$;
$[\neg F]\sigma = \neg(F\sigma)$;
$[F_1 \circ F_2]\sigma = F_1\sigma \circ F_2\sigma$.

Toute substitution σ est libre pour une formule atomique F.
Soit ◊ un quantificateur, une substitution σ est libre pour (◊v)F si et seulement si σ_v est libre pour F et pour toute variable v' différente de v et libre dans F, v'σ ne contient pas v.
Une substitution σ est libre pour ¬F si et seulement si elle est libre dans F.
Soit o un connecteur à deux places, une substitution σ est libre pour F_1 o F_2 si et seulement si σ est libre pour F_1 et elle est libre pour F_2.

Unification

σ est plus *générale* que τ s'il existe une substitution ω telle que $\tau = \sigma\omega$.
Une substitution σ est un *unificateur* de t_1 et de t_2 si $t_1\sigma = t_2\sigma$.
L'*unification* t_1 et de t_2 consiste à trouver l'unificateur le plus général entre t_1 et de t_2.

Algorithme d'unification :

commencer par la substitution d'identité ;
tant que $t_1\sigma \neq t_2\sigma$:
  déterminer un désaccord $\{d_1, d_2\}$;
  si ni d_1, ni d_2 ne sont des variables, ECHEC.
  soit v la variable, t l'autre terme ;
  si v est présent dans t, ECHEC.
  $\sigma = \sigma\{v/t\}$;