Logique des prédicats

MLO

Catherine Dubois, Julien Narboux (Strasbourg)

Introduction

Soit le syllogisme de Socrate suivant :

Tous les hommes sont mortels
Socrate est un homme
donc Socrate est mortel

Si on modélise ceci en logique des propositions, on arrive à

$$p \wedge q \Rightarrow r$$

avec
p la variable propositionnelle qui représente *Tous les hommes sont mortels*
q la variable propositionnelle qui représente *Socrate est un homme*
et r la variable propositionnelle qui représente *Socrate est mortel*

La proposition $p \wedge q \Rightarrow r$ n'est évidemment pas toujours vraie .....

$\Longrightarrow$ la logique des proposiitons n'est pas assez fine pour rendre compte de ce raisonnement $\Longrightarrow$ logique des prédicats ou logique du 1er ordre

Mettre en évidence ce qui est commun entre les 3 propositions : être mortel et être un homme.
Ce sont les atomes, les briques de base du langage.

Socrate est un objet qui a la propriété d'être un homme et d'être mortel.

Il faut aussi un moyen de parler de tous : énoncer une propriété pour tous les objets $\Rightarrow$ quantificateur universel $\forall$

On arrive à la formalisation suivante :

$$(\forall x, homme(x) \Rightarrow mortel(x)) \wedge homme(Socrate) \Rightarrow mortel(Socrate)$$

Volet Syntaxe

Syntaxe

ou Comment écrire une formule de la logique des prédicats ?

Plus compliqué qu'en logique des prédicats.

On doit d'abord fixer de quoi on veut parler : comment on va construire les objets dont on parle ? quelles sont les proporiétés de base que l'on pourra utiliser ?
$\Rightarrow$ Notion de langage ou de signature

Puis on pourra construire la notion de terme et enfin définir la notion de formule ou de prédicat.

Signature

Une signature (ou langage) est composée de:

- un ensemble $\mathcal{F}$ de symboles de fonctions munies de leur arité
- un ensemble $\mathcal{R}$ de symboles de relations munies de leur arité
- un ensemble dénombrable de variables $\mathcal{V}$ (parfois hors de la signature)

Arité $=$ nombre d'arguments.

Remarque : une constante est un symbole de fonction d'arité 0

Signature

Une signature (ou langage) est composée de:

- un ensemble $\mathcal{F}$ de symboles de fonctions munies de leur arité
- un ensemble $\mathcal{R}$ de symboles de relations munies de leur arité
- un ensemble dénombrable de variables $\mathcal{V}$ (parfois hors de la signature)

Arité $=$ nombre d'arguments.

Remarque : une constante est un symbole de fonction d'arité 0

Exemple du syllogisme de Socrate :
On a deux symboles de relation d'arité 1 : *homme* et *mortel* et une constante *Socrate*.

Terme

Soit Σ une signature.
L'ensemble des Σ-termes (ou plus simplement termes) l'ensemble défini inductivement par les règles suivantes:

- si x est une variable, alors x est un terme ;
- si f est un symbole de fonction d'arité n (dans Σ) et si $t_1, \ldots, t_n$ sont des termes, alors $f(t_1, t_2, \ldots, t_n)$ est un terme.

Terme

Soit Σ une signature.
L'ensemble des Σ-termes (ou plus simplement termes) l'ensemble défini inductivement par les règles suivantes:

- si x est une variable, alors x est un terme ;
- si f est un symbole de fonction d'arité n (dans Σ) et si $t_1, \ldots, t_n$ sont des termes, alors $f(t_1, t_2, \ldots, t_n)$ est un terme.

Quand on écrit $f(t_1, t_2, ..., t_n)$, on sous-entend (sauf mention contraire) que f a la bonne arité (soit n).

Terme

Soit Σ une signature.
L'ensemble des Σ-termes (ou plus simplement termes) l'ensemble défini inductivement par les règles suivantes:

- si x est une variable, alors x est un terme ;
- si f est un symbole de fonction d'arité n (dans Σ) et si $t_1, \ldots, t_n$ sont des termes, alors $f(t_1, t_2, \ldots, t_n)$ est un terme.

Quand on écrit $f(t_1, t_2, ..., t_n)$, on sous-entend (sauf mention contraire) que f a la bonne arité (soit n).

La notation infixe est parfois préférée : on écrit $x + y$ au lieu de $+(x, y)$.

Terme

Soit Σ une signature.
L'ensemble des Σ-termes (ou plus simplement termes) l'ensemble défini inductivement par les règles suivantes:

- si x est une variable, alors x est un terme ;
- si f est un symbole de fonction d'arité n (dans Σ) et si $t_1, \ldots, t_n$ sont des termes, alors $f(t_1, t_2, \ldots, t_n)$ est un terme.

Quand on écrit $f(t_1, t_2, ..., t_n)$, on sous-entend (sauf mention contraire) que f a la bonne arité (soit n).

La notation infixe est parfois préférée : on écrit $x + y$ au lieu de $+(x, y)$.

Un terme peut être vu comme un arbre

Formule

L'ensemble des Σ-formules (ou formules ou encore prédicats) l'ensemble défini inductivement par les règles suivantes:

- si p un symbole de prédicat d'arité n (dans Σ) et si $t_1, \ldots, t_n$ sont n termes, alors $p(t_1, t_2, \ldots, t_n)$ est une formule (atomique) ;
- $\perp$ est une formule ;
- si F est une formule alors $\neg F$ est une formule;
- si F_1 et F_2 sont des formules alors $F_1 \wedge F_2$, $F_1 \vee F_2$, $F_1 \Rightarrow F_2$ sont des formules;
- si x est une variable et F est une formule, alors $\forall x, F$ et $\exists x, F$ sont des formules.

$\forall$ et $\exists$ sont appelés des quantificateurs.

Formule

L'ensemble des Σ-formules (ou formules ou encore prédicats) l'ensemble défini inductivement par les règles suivantes:

- si p un symbole de prédicat d'arité n (dans Σ) et si $t_1, \ldots, t_n$ sont n termes, alors $p(t_1, t_2, \ldots, t_n)$ est une formule (atomique) ;
- $\perp$ est une formule ;
- si F est une formule alors $\neg F$ est une formule;
- si F_1 et F_2 sont des formules alors $F_1 \wedge F_2$, $F_1 \vee F_2$, $F_1 \Rightarrow F_2$ sont des formules;
- si x est une variable et F est une formule, alors $\forall x, F$ et $\exists x, F$ sont des formules.

$\forall$ et $\exists$ sont appelés des quantificateurs.

La quantification ne porte que sur les variables (objets du discours).
En logique du second ordre ou d'ordre supérieur, on peut quantifier sur les fonctions et les relations.

Les priorités ont les mêmes qu'en logique des propositions en ce qui concerne les connecteurs.

On a donc par ordre de priorité décroissante :

- les symboles de relations
- la négation,
- les quantificateurs
- et
- ou
- implique

Ainsi $\forall x, F \wedge G \Rightarrow \neg C \vee D$ s'interpète comme $((\forall x, F) \wedge G) \Rightarrow ((\neg C) \vee D)$

Rappel : $\Rightarrow$ est associatif à droite : $F \Rightarrow G \Rightarrow H$ s'interprète comme $F \Rightarrow (G \Rightarrow H)$

Variables libres/variables liées

La présence des quantificateurs pose des problèmes relatifs au nom des variables.

- Une variable est dite liée quand le nom que l'on utilise pour définir un objet n'a pas d'importance.
- $\forall x, x.y = y.x$ a le même sens $\forall z, z.y = y.z$
  Les deux formules sont les mêmes au renommage près des variables quantifiées : variables liées
- Mais $\forall x, x.y = y.x$ n'a pas le même sens que $\forall x, x.t = t.x$
  1ère formule : propriété sur y, la 2ème une propriété sur t.
  Ces deux variables ne sont pas introduites par un quantificateur : variables libres

Variables libres/variable liées

- Une même variable peut être liée et libre dans une formule, ou plus exactement elle peut avoir des occurrences libres et des occurrences liées.
  une occurrence d'une variable = un endroit où la variable apparaît dans la formule (sauf derrière un quantificateur)

  $(\forall x, R(x)) \wedge P(x)$
  occurrence liée de x occurrence libre de x
- On dit qu'une variable est libre dans une formule si elle possède au moins une occurrence libre dans cette formule.
- On dit qu'une variable est liée dans une formule si toutes les occurrences de la variable dans la formule sont liées.

Variables libres/variable liées

Variables libres d'un terme

$FV(t)$ = ensemble des variables libres de t = ensemble des variables qui apparaissent dans t (FV comme free variables)

$$\begin{array}{rcl} FV(x) & = & \{x\} \\ FV(f(t_2, t_2, \ldots, t_n)) & = & FV(t_1) \cup FV(t_2) \cup \ldots \cup FV(t_n) \end{array}$$

Variables libres/variable liées

Variables libres d'une formule

$$
\begin{array}{rcl}
FV(p(t_1, t_2, \ldots, t_n)) & = & FV(t_1) \cup FV(t_2) \cup \ldots \cup FV(t_n) \\
FV(\neg A) & = & FV(A) \\
FV(A \wedge B) & = & FV(A) \cup FV(B) \\
FV(A \vee B) & = & FV(A) \cup FV(B) \\
FV(A \Rightarrow B) & = & FV(A) \cup FV(B) \\
FV(\exists x, P) & = & FV(P) - \{x\} \\
FV(\forall x, P) & = & FV(P) - \{x\}
\end{array}
$$

Toutes les occurrences de x dans la formule Qx, P sont liées avec $Q \in \{\forall, \exists\}$

Une variable peut être liée dans une formule et libre dans une sous-formule. Exemple y est liée dans $\forall x, x.y = y.x$ mais libre dans $x.y = y.x$.

Vocabulaire

Un terme (ou une formule) sans variable libre est dit clos.
Une théorie est un ensemble de formules closes.

Renommage (ou α-conversion)

On dit que deux formules F et G sont α-équivalentes ssi elles sont identiques au renommage près des **occurrences liées** de leurs variables.

On écrit $F \equiv_\alpha G$.

Renommage (ou α-conversion)

On dit que deux formules F et G sont α-équivalentes ssi elles sont identiques au renommage près des **occurrences liées** de leurs variables.

On écrit $F \equiv_\alpha G$.

Exemple : $\forall x, x.y = y.x$

Renommage (ou α-conversion)

On dit que deux formules F et G sont α-équivalentes ssi elles sont identiques au renommage près des **occurrences liées** de leurs variables.

On écrit $F \equiv_\alpha G$.

Exemple : $\forall x, x.y = y.x \quad \equiv_\alpha$

Renommage (ou α-conversion)

On dit que deux formules F et G sont α-équivalentes ssi elles sont identiques au renommage près des **occurrences liées** de leurs variables.

On écrit $F \equiv_\alpha G$.

Exemple : $\forall x, x.y = y.x \quad \equiv_\alpha \quad \forall z, z.y = y.z$

Renommage (ou α-conversion)

On dit que deux formules F et G sont α-équivalentes ssi elles sont identiques au renommage près des **occurrences liées** de leurs variables.

On écrit $F \equiv_\alpha G$.

Exemple : $\forall x, x.y = y.x \quad \equiv_\alpha \quad \forall z, z.y = y.z$

Mais on ne peut pas renommer x en y : on aurait alors $\forall y, y.y = y.y$

Renommage (ou α-conversion)

On dit que deux formules F et G sont α-équivalentes ssi elles sont identiques au renommage près des **occurrences liées** de leurs variables.

On écrit $F \equiv_\alpha G$.

Exemple : $\forall x, x.y = y.x \quad \equiv_\alpha \quad \forall z, z.y = y.z$

Mais on ne peut pas renommer x en y : on aurait alors $\forall y, y.y = y.y$
$\Rightarrow$ Problème : capture de y par le quantificateur : il devient lié ....

Renommage (ou α-conversion)

On dit que deux formules F et G sont α-équivalentes ssi elles sont identiques au renommage près des **occurrences liées** de leurs variables.

On écrit $F \equiv_\alpha G$.

Exemple : $\forall x, x.y = y.x \quad \equiv_\alpha \quad \forall z, z.y = y.z$

Mais on ne peut pas renommer x en y : on aurait alors $\forall y, y.y = y.y$
$\Rightarrow$ Problème : capture de y par le quantificateur : il devient lié ....

Usuellement on choisit pour renommer une variable fraîche (i.e. une variable qui n'apparaît pas déjà).
Il suffit de choisir une variable non libre dans la formule.

- Codage permettant de rendre syntaxiquement identiques deux formules équivalentes au renommage près.
- Notation très cintéressante pour les programmes (prouveurs par exemple) mais peu lisible pour des humains.
- les noms des variables disparaissent.
- Une occurrence de variable est représentée par le nombre de quantificateurs qu'il faut traverser (dans une représentation arborescente) avant de trouver le quantificateurs qui lie l'occurrence.

Exemple : $\forall x, ((\forall y, x * y = y * x) \wedge x * E = E * x)$.
sous forme arborescente

```
               ∀x
               |
               |
               ∧
             /    |
            /     |
           ∀y    x⋆E=E⋆x
           |     0⋆E=E⋆0
           |
      x⋆y=y⋆x
      1⋆0=0⋆1
```

La formule $\forall x, ((\forall y, x * y = y * x) \wedge x * E = E * x)$ s'écrit $\forall.((\forall.1 * 0 = 0 * 1) \wedge 0 * E = E * 0)$.
La formule $\forall z, ((\forall y, z * y = y * z) \wedge z * E = E * z)$ s'écrit aussi $\forall.((\forall.1 * 0 = 0 * 1) \wedge 0 * E = E * 0)$.

Substitution

Remplacer toutes les occurrences libres d'une variable x par un terme t dans une formule F : $F[x ::= t]$

Attention au renommage !

$$(\forall y, P)[x ::= t] = \forall y, P[x ::= t]$$

que si $x \neq y$ et $y \notin FV(t)$!

On définit formellement d'abord la substitution dans un terme puis dans une formule (Voir le tableau !)
$\Rightarrow$ Eviter la capture de variables libres de t
$\Rightarrow$ Remède : renommage des variables liées de F.