La Logique du premier ordre LPO

Lacune de la logique des propositions : ne permet pas de parler d'individus (dans l'exemple ''Sophie est coupable'' S désignait la proposition, et non la personne '' Sophie''). Or la plupart de nos connaissances portent sur des individus, sur des classes d'individus, sur des relations entre individus. Il existe un outil plus puissant qui permet d'exprimer de telles connaissances la logique du premier ordre : coupable (x) avec x peut être l'individu Sophie ou Jean ou autre.

L'idée remonte aux grecs anciens, et se formule ainsi : dans une phrase , le verbe établit une relation entre son sujet et ses compléments. La proposition jean prend son manteau s'exprime prend(jean, manteau). Nous allons décrire les trois volets de la LPO.

Langage

$\Sigma = \{ \supset, \forall \} \cup$ <u>variables</u>$\{x, y, \ldots\} \cup$ Symboles de fonction $\{f, g, \ldots\} \cup$ Symboles de prédicats $\{\boldsymbol{\ell}, p, q, \ldots\}$.

A chaque fonction et à chaque prédicat est associé un entier naturel, son **arité ; *ℓ*** est le prédicat d'arité 0.

<u>*Formule*</u> est l'axiome de ce langage.

Les réécritures sont :

Formule $\rightarrow$ $pred_0$ / $pred_1$(terme) / $pred_2$(terme, terme) /…/ (<u>*formule*</u> $\supset$ <u>*formule*</u>)/ ($\forall$var)(formule).

Terme $\rightarrow$ <u>$fonct_0$</u> / $fonct_1$(terme) / $fonct_2$(terme, terme) / …/ var

Var $\rightarrow$ x / y /…

Pour chaque i, $pred_i$ (respectivement $fonct_i$) correspond au symbole de prédicat (fonction) d'arité i

$\{v, \neg , \wedge, \vee, \equiv\}$: mêmes règles données pour la logique propositionnelle.

V désigne la formule $(\boldsymbol{f} \supset \boldsymbol{f})$

$\forall$x, $\neg$ x désigne la formule $(x \supset \boldsymbol{f})$

$\forall$x,y, $x \vee y$ ssi $((x \supset y) \supset y)$

$\forall$x,y, $x \wedge y$ ssi $((x \supset (y \supset \boldsymbol{f})) \supset \boldsymbol{f})$

$\forall$x,y, $x \equiv y$ ssi $(((x \supset y) \supset ((y \supset x) \supset \boldsymbol{f})) \supset \boldsymbol{f})$

Pour $\exists$ / $\forall$x et $\forall$f , $(\exists x)\ (f) \equiv \neg(\forall x)\ (\neg f)$

Notons enfin que les fonctions d'arité 0 sont également appelées constantes

Parmi les symboles de variables, il importe de faire la distinction entre ceux qui représentent des variables libres et liées.

Système de déduction

- Il y a une infinité d'axiomes dans cette logique. On les représente par des schémas d'axiomes, i.e., les axiomes sont obtenus par substitution (la règle R1 de la logique propositionnelle) à partir des 5 formules ci-dessous :
- (A1), (A2), (A3) : les axiomes correspondants de la logique propositionnelle

Plus

- (A4) $((\forall x)((a \supset b)) \supset (a \supset (\forall x)(b)))$
- (A5) $((\forall x)(a) \supset b)$ où b est une formule obtenue en substituant dans a toutes les occurrences de la variable x par une autre variable y ou par une constante.

<u>Lemme1</u> : pour toute formule a et toute variable x,

$((\forall x)(a) \supset a)$ est un axiome.

<u>Preuve</u> : on obtient la formule cherchée à partir de (A5) :

$((\forall x)(a) \supset b)$; ■

Les règles d’inférence sont :

(R2) [modus ponens] : comme en logique propositionnelle.

(R4) [généralisation] : si a est une formule, R4(a) est l’ensemble des formules ayant la forme $(\forall x)$ (a) où x est un symbole de variable.

<u>Lemme 2</u> : les théorèmes de la logique propositionnelle et les formules du premier ordre obtenues par R1 appliquées à ces théorèmes sont des théorèmes de la logique du premier ordre.

Règles de valuation

En logique du premier ordre, l'ensemble V des valeurs de vérité comprend les mêmes symboles vrai et faux. L'évaluation (dans V) d'une formule du langage s'effectue dans un modèle. Celui-ci se compose d'un **domaine**, d'une **interprétation**, et d'une **assignation**

- Un domaine est un ensemble D non vide, absolument quelconque.

- Une interprétation fait correspondre à chaque symbole de prédicat $pred_n$(respectivement fonction $fonct_n$) d'arité n une application p_n (respectivement f_n) de D^n dans D(V);

  le prédicat *f* s'interpréte comme la valeur faux.

- Une assignation fait correspondre à chaque symbole de variable un élément de D

Comme en logique propositionnelle, les formules dont la valeur est vrai quel que soit le modèle sont appelées **tautologies**. Les règles de valuation étant compatibles, les tautologies du calcul propositionnel sont des tautologies du premier ordre.

Plus généralement, tout théorème est une tautologie, c'est-à-dire que le système déductif à la propriété de correction.

Exercice [correction de LPO] : montrer que tout théorème de la logique du premier ordre (c-à-d toute formule obtenue à partir de (A1-A5), de R2 et R4) est une tautologie .

La propriété réciproque, la complétude, est beaucoup plus délicate à démontrer. La preuve a été apportée par K Gödel en 1930.

Connaissances du premier ordre

Pour de nombreux auteurs, le langage du premier ordre est le langage idéal pour la représentation des connaissances : ‘’ si l’on voit l’univers comme composé d’entité **stables**, sur lesquels on dispose d’une connaissance **complète**, alors la logique du premier ordre est un excellent système de représentation.

Mais la notion d’assertion ‘’généralement’’ vraie n’y a toujours pas sa place.

Exemple sur les nautiles

Les nautiles sont des céphalopodes;

Les céphalopodes sont des mollusques;

Les mollusques ont généralement une coquille;

Les céphalopodes n’en ont généralement pas;

Les nautiles en ont une.

a est un nautile,

b est un céphalopode,

c est un mollusque.

$(\forall x)\ (naut(x) \supset céph(x));$

$(\forall x)\ ((céph(x) \supset mol(x));$

$(\forall x)((mol(x) \wedge \neg(céph(x) \wedge \neg naut(x)) \supset a\text{-}coq(x)));$

$(\forall x)\ ((céph(x) \wedge \neg naut(x) \supset \neg a\text{-}coq(x)));$

$(\forall x)((naut(x) \supset a\text{-}coq(x)));$ naut(a); céph(b); mol(c);

Comme précédemment, on peut conclure a-coq(a), mais il existe des modèles satisfaisant toutes ces formules et a-coq(b)

[il suffit de supposer que b est un naut?]; de même pour ¬a-coq(c).

Les conséquences attendues ne sont toujours pas dérivables. Le problème vient de ce que l'on n'a pas de connaissance complète sur b et c : on ne sait pas, par exemple, si b est ou non un nautile.

Décidabilité

On dit qu’une logique est décidable s’il existe un procédé de calcule qui pour toute formule, indique en un temps fini s’il s’agit ou non d’un théorème de cette logique.

La logique du premier ordre n’est que semi-décidable. Il existe un algorithme qui, si une formule est démontrable, le dira en un temps fini, mais qu’on ne peut borner à priori. Il n’existe en revanche pas d’algorithme permettant, pour toute formule, de déterminer en un temps fini qu’elle n’est pas démontrable.