Logique des prédicats

Introduction

Le calcul des propositions est bien trop limite pour décrire des situations réelles. En effet il ne permet que de décrire des phrases dont la vérité ne dépend pas des individus (par exemple < Il pleut >) ; il ne peut pas représenter des phrases qui mettent en jeu des individus ou des objets (par exemple < Si x est le père de y et si z est le père de x alors z est un grand-père de y > ou < Tout individu a un père >).

Le calcul des prédicats (ou Logique du Premier Ordre) permet d'exprimer de telles relations entre individus, il est donc bien plus riche que le calcul propositionnel. En premier lieux, il contient des individus (ou entités) (donnes par des symboles de variables x; y; z; : : :). Il contient des fonctions (f; g; : : : ; s; : : :) permettant de transformer des entités en autres entités (par exemple la fonction qui associe une personne a son père), et des relations (: : : ; P; Q;R; : : :) permettant de lier les individus entre eux.

Introduction

La logique du premier ordre introduit deux *quantificateurs* qui portent sur des variables :

∀, le quantificateur *universel* (« pour tout »)

∃, le quantificateur *existentiel* (« il existe »)

Introduction

Considérons par exemple l'énoncé suivant :

(1) tout chat est un animal

(2) Félix est un chat

donc Félix est un animal.

On pourra le formaliser en logique du premier ordre de la manière suivante :

(1) $\forall x\ (chat(x) \rightarrow animal(x))$ (tout x qui est un chat est un animal)

(2) *chat(Félix),*

De (1) et (2), on peut déduire : *animal(Félix)*

Félix est une *constante*, x une *variable*, chat(.) et animal(.) sont des *prédicats* unaires.

Syntaxe

Définition (langage du premier ordre)

Un *langage du premier ordre* L = ({a1, a2, ..., ak}, {p1, p2, ..., pn}) est formé :

- d'un ensemble de **constantes** : {a1, a2, ..., ak}

- et d'un ensemble de symboles de relation, ou **prédicats** : {p1, p2, ..., pn}.

A chaque prédicat pi est associé un entier ≥ 0, son **arité**, qui fixe son nombre d'arguments.

Syntaxe

Remarque. Du point de vue des langages formels, les formules sont des *mots* construits sur l'alphabet formé de **L**, d'un ensemble de **variables** et des autres symboles que sont les **quantificateurs** (∀,∃), les **connecteurs** (¬, ∧,∨, …) et les **parenthèses**. Donc **L** est une partie de l'alphabet ; toutefois l'utilisation de l'expression « langage » du premier ordre est traditionnelle en logique.

Syntaxe

Définition (terme)

Un *terme* est une variable ou une constante

Syntaxe

Définition (formule bien formée – fbf)

Les *formules bien formées,* fbf, construites sur le langage du premier ordre L peuvent être définies inductivement de la manière suivante :

(base) la base est constituée par les formules atomiques (ou **atomes**) **: p(t1, ..., tn)** où p est un prédicat n-aire et t1, ..., tn sont des termes ; parfois, on ajoute l'atome ⊥ qui symbolise « l'absurde ».

(règle de construction) si A et B sont des fbf et si x est une variable alors les expressions suivantes sont des fbf :

¬A, **(A** ∧ **B)**, [(A ∨ B), (A → B), (A ↔ B)]

∀**xA** [∃xA]

Syntaxe

Définition (arborescence syntaxique d'une fbf)

L'arborescence syntaxique d'une fbf **f** (notation : ARBO(f)) se définit inductivement de la façon suivante :

(base) si f est un atome, **ARBO(f)** est une arborescence réduite à un sommet étiqueté par f (règles)

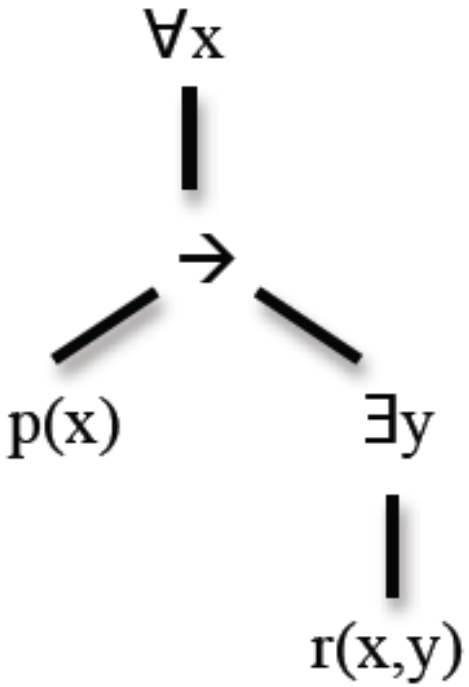
si f = ¬A, ARBO(f) est une arborescence dont la racine est étiquetée par ¬ et a un seul fils qui est la racine de ARBO(A)

si f = QxA, où Q est un quantificateur, ARBO(f) est une arborescence dont la racine est étiquetée par Qx et a un seul fils qui est la racine de ARBO(A)

si f = (A op B), où op est un opérateur binaire, ARBO(f) est une arborescence dont la racine est étiquetée par op et qui a deux fils : le fils gauche est la racine de ARBO(A) et le fils droit est la racine de ARBO(B).

Syntaxe

Exemple : arborescence de la formule
$\forall x(p(x) \rightarrow \exists y\ r(x,y))$



Exemple *(profondeur d'une fbf) :*

Soit f une fbf.

(base) Si f est un atome, profondeur(f) = 0

(règles)

Si f = ¬A ou f = ∃x A ou f = ∀x A, profondeur(f) = 1 + profondeur(A)

Si f = (A *op* B) où *op* est l'un des connecteurs binaires,

profondeur(f) = 1 + max(profondeur(A), profondeur(B)).

Synatxe

Définition (formule fermée)

Une fbf est dite **fermée** (ou **close**) lorsqu'elle n'a **aucune variable libre** *(et pas lorsque toutes les variables sont liées, puisqu'une variable peut être libre et liée ...).*

Syntaxe

Occurrences libres et liées d'une variable

Lorsqu'une variable x appartient à une sous-formule précédée d'un quantificateur, $\forall x$ ou $\exists x$, elle est dite **liée** par ce quantificateur. Si une variable n'est liée par aucun quantificateur, elle est **libre**.

La distinction entre variable libre et variable liée est importante. Une variable liée ne possède pas d'identité propre et peut être remplacée par n'importe quel autre nom de variable qui n'apparaît pas dans la formule. Ainsi, $\exists x(x < y)$ est identique à $\exists z(z < y)$ mais pas à $\exists x(x < z)$ et encore moins à $\exists y(y < y)$.

L'ensemble des variables libres $\mathsf{FV}(\varphi)$ (depuis l'anglais, *free variable*), et l'ensemble des variables liées $\mathsf{BV}(\varphi)$ (depuis l'anglais, *bound variable*) [3] d'une formule φ sont définis par induction sur la structure de φ :

- si φ est une formule atomique, alors tout occurrence d'une variable x dans φ est libre : $\mathsf{FV}(\varphi) = Var(\varphi)$ et $\mathsf{BV}(\varphi) = \varnothing$.
- si φ est $\exists x\psi$ ou $\forall x\psi$, alors $\mathsf{FV}(\varphi) = \mathsf{FV}(\psi) - \{x\}$; $\mathsf{BV}(\varphi) = \mathsf{BV}(\varphi) \cup \{x\}$ (et toute occurrence de x **libre dans** ψ devient liée dans φ par le quantificateur introduit) ;
- si $\varphi = \varphi_1 \circ \varphi_2$ (où $\circ \in \{\wedge, \vee, \Rightarrow\}$) alors $\mathsf{FV}(\varphi) = \mathsf{FV}(\varphi_1) \cup \mathsf{FV}(\varphi_2)$, $\mathsf{BV}(\varphi) = \mathsf{BV}(\varphi_1) \cup \mathsf{BV}(\varphi_2)$;
- si $\varphi = \neg\psi$, alors $\mathsf{FV}(\varphi) = \mathsf{FV}(\psi)$ et $\mathsf{BV}(\varphi) = \mathsf{BV}(\psi)$.

Sémantique

La valeur de vérité d'une fbf (vrai ou faux) dépend de l'**interprétation du langage L** sur lequel elle est construite. On peut voir une interprétation de L comme :

(1) un monde constitué d'entités (ou individus, ou objets, …) : l'ensemble des entités de ce monde s'appelle le *domaine* de l'interprétation ;

(2) une mise en correspondance de L avec ce monde : les constantes de L donnent un nom à certaines entités de ce monde (mais pas forcément à toutes) et les prédicats de L décrivent certaines propriétés des entités du domaine (si ce sont des prédicats unaires) ou certaines relations entre les entités du domaine.

Il existe une infinité d'interprétations possibles de L.

Sémantique

Définition (Interprétation)

Soit L = (C, P) un langage du premier ordre où C est l'ensemble des constantes et P est l'ensemble des prédicats. Une *interprétation* **I** de L est constituée d'un ensemble **non vide D** appelé **domaine** et d'une définition du sens des symboles de L sous forme d'applications dans D ou de relations sur D :

- pour tout a de C, **I(a)** est un **élément de D** *(il y a une application de C dans D)*
- pour tout p de P, d'arité non nulle, **I(p)** $\subseteq$ **Darité(p)**

  Si p est d'arité 0 (p peut-être vu comme un symbole propositionnel) I(p) est égal à vrai ou faux.