LOGIQUE
L3 Informatique

Salim Lardjane
Université Bretagne Sud

5. Logique du premier ordre (LP1)

5.8 Automatisation de la LP1

Introduction

Le théorème de Löwenheim-Skolem permet d'envisager un test mécanique de validité (puisqu'il pourra se faire sur des domaines dénombrables).

Restent encore deux questions :

- faut-il envisager de tester toutes les interprétations sur un certain domaine dénombrable ?

- faut-il envisager de tester sur tous les domaines dénombrables ?

La réponse à ces deux questions est (heureusement) négative.

Automatisation de la LP1

Théorème 1. *On peut tester la validité d'une fbf de la LP1 sur un domaine dénombrable sans tenir compte des interprétations.*

Preuve. Admis.

Théorème 2. *F fbf de la LP1 est satisfaisable si et seulement si elle est satisfaisable sur le domaine H(sk(F)).*

Preuve. Admis.

Automatisation de la LP1

Le théorème suivant, qui fusionne le théorème 1 et le théorème 2, est capital pour l'automatisation de la LP1.

Théorème 3 (de Herbrand) *F fbf de la LP1 est valide si et seulement si il existe un ensemble fini d'instances de sk$(\neg F)$ insatisfaisable (contradictoire).*

Le théorème de Herbrand permet de concevoir une procédure de semi-décision pour la LP1 :

Automatisation de la LP1

procédure PROC_SEMI_DEC_L1O
entrée : sk($\neg F$) $Var(sk(\neg F)) = \{x_1, \ldots, x_n\}$
sortie : F valide ou F non valide ou $\perp$
test d'arrêt : détection contradiction ou test de toutes les IH (si univers **fini**)
début
• Engendrer H(sk($\neg F$)) % univers de Herbrand
• Énumérer les n-uplets de H(sk($\neg F$)) : $\overline{t_i}$ ($1 \leq i$; $i \in \mathbb{N}$)
• Obtenir sk($\neg F$)[$\overline{x} \mid \overline{t_i}$] % IH : instances de Herbrand
• Tester (propositionnellement) $\bigcup_i$ { sk($\neg F$)[$\overline{x} \mid \overline{t_i}$] } ($1 \leq i$; $i \in \mathbb{N}$)
• **Si** contradiction détectée **alors** retourner 'F valide'
• **Sinon si** tests épuisés (H(sk($\neg F$)) fini) **alors** retourner F non valide
• **Sinon** continuer
fin

IH : *instances de Herbrand*.

Remarques (complétude de la LP1).

1. Le *théorème de complétude de Gödel* pour la LP1 peut être énoncé de la façon suivante : *il existe un système formel dans lequel toute formule valide de la LP1 a une preuve*.

Le système suivant en est un exemple.

$$\mathcal{S}_{LP1} = \langle \mathcal{L}_1, \mathcal{R}_1, \mathcal{A}_1 \rangle$$

avec

$\mathcal{L}_1$: langage de la LP1

Automatisation de la LP1

$\mathcal{R}_1$: le schéma de règle d'inférence MP du système formel $\mathcal{S}_1$ (LP0) auquel il faut ajouter les schémas d'inférence G et P ci-dessous :

$$G : \frac{F \Rightarrow G(x)}{F \Rightarrow \forall x G(x)}$$

$$P : \frac{G(x) \Rightarrow F}{\exists x G(x) \Rightarrow F}$$

$x \notin Var(F)$

Automatisation de la LP1

$\mathcal{A}_1$: les schémas d'axiomes (A_1), (A_2), (A_3) de $\mathcal{S}_1$ auxquels il faut ajouter (A_4) et (A_5) ci-dessous :

$(A_4) : \forall x F(x) \Rightarrow F(t) \quad t :$ terme

$(A_5) : F(t) \Rightarrow \exists x F(x) \quad t :$ terme

Automatisation de la LP1

2. On peut, du point de vue conceptuel, caractériser un système formel comme « une machine à produire (et à tester) des théorèmes dans une certaine logique », donc on peut voir PROC_SEMLDEC_LIO comme un système formel pour la LP1 et le théorème de Herbrand comme un théorème de complétude pour la LP1.

Ce théorème permet de remplacer un test sur toutes les interprétations (notion sémantique de validité) par un test d'existence d'une preuve (notion syntaxique de preuve).

Automatisation de la LP1

3. Les preuves en LP1 peuvent être **très, très longues**.

Pour fixer les idées nous considérons $\mathcal{S}_{LP1}$, mais la propriété énoncée ci-dessous vaut pour tout système $\mathcal{S}$ pour lequel $\vdash_{\mathcal{S}} T$? est indécidable.

On appelle *longueur d'un théorème* T, noté $long(T)$, le nombre de symboles de son énoncé. De manière analogue, on appelle *longueur d'une preuve* le nombre de symboles de la preuve (on peut en donner d'autres définitions, nombre des pas, etc.).

Automatisation de la LP1

Propriété : Pour toute fonction récursive (c'est-à-dire calculable par algorithme) f, il existe des fbf T de la LP1, telles que $\vdash_{\mathcal{S}_{LP1}} T$ avec $long(T) = n$ ($n \in \mathbb{N}$) et dont la preuve la plus courte est de longueur au moins $f(n)$.

AInsi, il existe des théorèmes dans $\mathcal{S}_{LP1}$, de longueur, disons N, tels que leur preuve la plus courte est de longueur par exemple $10^{100 \times N}$.