

Logique formelle & Programmation logique

$$\exists \Rightarrow \forall$$

Dr. Stéphane Lengrand,

`Stephane.Lengrand@Polytechnique.edu`

Cours 6 :
Forme clause, Prolog

Litéral, clause, forme clausale

Définitions :

- **litéral** (noté l, l', \dots) = formule atomique (litéral positif)
ou négation de formule atomique (litéral négatif)

- **clause** (notée C, C', \dots) = disjonction de littéraux

(en général) universellement quantifiée

$$\forall x_1 \dots \forall x_k, l_1 \vee \dots \vee l_p \quad \text{avec } \{x_1, \dots, x_k\} = fv(l_1 \vee \dots \vee l_p)$$

Soit A une formule du 1er ordre (close).

forme clausale de A = ensemble fini de clauses C_1, \dots, C_n telles que

$$A \vdash \perp \text{ ssi } C_1, \dots, C_n \vdash \perp.$$

Beaucoup de techniques de raisonnement automatique utilisent ces formes clausales

Question : A possède-t-elle toujours une telle forme clausale ?

Mise sous formes clauseale

Concrètement, on cherche une formule B de la forme

$$(\forall x_1^1 \dots \forall x_{k_1}^1, l_1^1 \vee \dots \vee l_{p_1}^1) \wedge \dots \wedge (\forall x_1^q \dots \forall x_{k_q}^q, l_1^q \vee \dots \vee l_{p_q}^q)$$

avec $\{x_1^i, \dots, x_{k_i}^i\} = fv(l_1^i \vee \dots \vee l_{p_i}^i)$ et telle que $A \vdash \perp$ ssi $B \vdash \perp$.

4 étapes

Mise sous formes clausale

une **forme prenexe** de A :

une formule logiquement équivalente à A , de la forme

$$Q_1x_1, \dots, Q_nx_n, C$$

avec tous les quantificateurs $Q_1 \dots Q_n$ en tête, C sans quantificateurs.

une **forme prénexé skolémisée** de A :

une formule close B de la forme

$$\forall y_1, \dots, \forall y_m, D$$

avec D sans quantificateurs, telle que $A \vdash \perp$ ssi $B \vdash \perp$.

(Toutes les variables libres ou quantifiées existentiellement ont été substituées avec des nouveaux constructeurs de termes)

Mise sous formes clauseale

une **forme normale conjonctive prénexe skolémisée** de A :

la même chose en imposant que D est une grande conjonction de disjonctions, i.e. une formule de la forme

$$\forall y_1, \dots, \forall y_m, (l_1^1 \vee \dots \vee l_{p_1}^1) \wedge \dots \wedge (l_1^q \vee \dots \vee l_{p_q}^q)$$

une **forme clauseale** de A :

une conjonction de clauses closes, logiquement équivalente à une forme normale conjonctive prénexe skolémisée de A , i.e. de la forme :

$$(\forall x_1^1 \dots \forall x_{k_1}^1, l_1^1 \vee \dots \vee l_{p_1}^1) \wedge \dots \wedge (\forall x_1^q \dots \forall x_{k_q}^q, l_1^q \vee \dots \vee l_{p_q}^q)$$

avec $\{x_1^i, \dots, x_{k_i}^i\} = fv(l_1^i \vee \dots \vee l_{p_i}^i)$

Mise sous forme clauseale

Les transformations permettant chacune des 4 étapes, pour toute formule A , sont dans les exercices des TDs.

Exemple : $\neg(r(986) \Rightarrow \exists y, r(y))$ devient $r(986) \wedge \forall y, \neg r(y)$,
la seconde étant bien insatisfiable ssi la première l'est
(c'est-à-dire ssi $r(986) \Rightarrow \exists y, r(y)$ est valide)

Pour économiser de la place, sans perdre d'information logique,
on ne retient de

$$(\forall x_1^1 \dots \forall x_{k_1}^1, C^1) \wedge \dots \wedge (\forall x_1^q \dots \forall x_{k_q}^q, C^q)$$

que l'ensemble des clauses C_1, \dots, C_n , où la disjonction est associative
+ commutative (les clauses sont des multisets de littéraux)

ProLog (pour Programmation Logique)

ProLog : prouveur qui implémente G3 avec variables existentielles,

restreint aux séquents de la forme $C_1, \dots, C_n \vdash l_1 \wedge \dots \wedge l_p$

où C_1, \dots, C_n sont des **Clauses de Horn** qui possèdent 1 littéral positif

l_1, \dots, l_p sont des **littéraux positifs** avec variables existentielles

Clause de Horn : clause où au plus un littéral est positif

Syntaxe :

$l : -l_1, \dots, l_n.$

pour $(l \vee (\neg l_1) \vee \dots \vee (\neg l_n))$

ou encore $(l_1 \wedge \dots \wedge l_n) \Rightarrow l$

Conjonction de littéraux $l_1 \wedge \dots \wedge l_n$ à prouver, la **requête**, est écrite

$? : -l_1, \dots, l_n.$

Propriété de ce fragment de G3 :

La partie gauche du séquent reste invariante tout au long de la preuve.

C'est le **programme**. Il est donné dès le début dans un fichier `.pl`

Questions?