Logique du premier ordre
1ère leçon

Jean-Pierre Jouannaud
École Polytechnique
91400 Palaiseau, France

email: jouannaud@lix.polytechnique.fr
http: //w^3.lix.polytechnique.fr/Labo/Jean-Pierre.Jouannaud

Project LogiCal, Pôle Commun de Recherche en Informatique du Plateau de Saclay, CNRS, École Polytechnique, INRIA, Université Paris-Sud.

February 6, 2006

Outline

Syntaxe

$T(\mathcal{F}, \mathcal{X})$ est l'ensemble des *termes* finis :
- toute variable de $\mathcal{X}$ est un terme ;
- $f(t_1, \ldots, t_n)$ est un *terme* ssi $f \in \mathcal{F}$ est d'arité n et $t_1, \ldots, t_n$ sont des termes.

$Var(t)$ désigne l'ensemble des variables de t. t est *clos* si $Var(t) = \emptyset$. $T(\mathcal{F})$ désigne l'ensemble des termes clos.

$P(t_1, \ldots, t_n)$ est un *atome* ssi P est un symbole de prédicat d'arité n et $t_1, \ldots, t_n$ sont des termes.
Tout atome est une *formule* dite atomique.
Le langage $\mathcal{F}(\mathcal{R}, \mathcal{F}, \mathcal{X})$ des *formules du premier ordre* bati sur le vocabulaire $\mathcal{R}, \mathcal{F}, \mathcal{X}$ est le plus petit ensemble contenant les atomes, les *constantes logiques* 0 et 1, et clos par les opérations de

négation : $\neg A$ *implication* : $A \Rightarrow B$

conjonction : $A \wedge B$ *disjonction* : $A \vee B$

quantification existencielle : $\exists x A$

quantification universelle : $\forall x A$

Pour désambiguer les formules, on adopte la convention de précédence $\neg > \wedge > \vee > \Rightarrow$.

Formules, suite

Le langage des *formules propositionnelles* est obtenu pour $\mathcal{X} = \emptyset, \mathcal{F} = \emptyset, \mathcal{P} = \mathcal{P}_0$.

Un *littéral* est un atome (auquel cas il est dit *positif*) ou la négation d'un atome (auquel cas il est dit *négatif*). Une formule sans variable est dite *close*. Une formule dont les quantificateurs apparaissent tous en tête est dite *prénexe*. Une formule *universelle* (resp. *existentielle*) est une formule prénexe close dont les quantificateurs sont universels (resp. existentiels).

On appelle *clause* toute formule universelle dont les sous formules non quantifiées sont des littéraux ou des disjonctions de littéraux :

$$\forall x_1 \ldots \forall x_n \, A_1 \vee \ldots \vee A_n \vee \neg B_1 \vee \ldots \vee \neg B_q$$

Formules, suite

Les quantificateurs *lient* les variables sur lesquelles ils portent. On définit donc les variables *libres* $\mathcal{V}ar()$ et les variables *liées* $\mathcal{BV}ar()$ d'une formule par récurrence sur la structure de la formule :

Vars $\mathcal{V}ar(x) = \{x\}$ si x est une variable
$\mathcal{BV}ar(x) = \emptyset$ si x est une variable

Terms $\mathcal{V}ar(f(t_1, \ldots, t_n)) = \mathcal{V}ar(t_1) \cup \ldots \cup \mathcal{V}ar(t_n)$
$\mathcal{BV}ar(f(t_1, \ldots, t_n)) = \emptyset$

Atoms $\mathcal{V}ar(P(t_1, \ldots, t_n)) = \mathcal{V}ar(t_1) \cup \ldots \cup \mathcal{V}ar(t_n)$
$\mathcal{BV}ar(P(t_1, \ldots, t_n)) = \emptyset$

Formules, suite

Formes
$$\mathcal{V}ar(A \wedge B) = \mathcal{V}ar(A) \cup \mathcal{V}ar(B)$$
$$\mathcal{B}\mathcal{V}ar(A \wedge B) = \mathcal{B}\mathcal{V}ar(A) \cup \mathcal{B}\mathcal{V}ar(B)$$
$$\mathcal{V}ar(A \vee B) = \mathcal{V}ar(A) \cup \mathcal{V}ar(B)$$
$$\mathcal{B}\mathcal{V}ar(A \vee B) = \mathcal{B}\mathcal{V}ar(A) \cup \mathcal{B}\mathcal{V}ar(B)$$
$$\mathcal{V}ar(A \Rightarrow B) = \mathcal{V}ar(A) \cup \mathcal{V}ar(B)$$
$$\mathcal{B}\mathcal{V}ar(A \Rightarrow B) = \mathcal{B}\mathcal{V}ar(A) \cup \mathcal{B}\mathcal{V}ar(B)$$
$$\mathcal{V}ar(\neg A) = \mathcal{V}ar(A)$$
$$\mathcal{B}\mathcal{V}ar(\neg A) = \mathcal{B}\mathcal{V}ar(A)$$
$$\mathcal{V}ar(\exists x A) = \mathcal{V}ar(A) - \{x\}$$
$$\mathcal{B}\mathcal{V}ar(\exists x A) = \mathcal{B}\mathcal{V}ar(A) \cup \{x\}$$
$$\mathcal{V}ar(\forall x A) = \mathcal{V}ar(A) - \{x\}$$
$$\mathcal{B}\mathcal{V}ar(\forall x A) = \mathcal{B}\mathcal{V}ar(A) \cup \{x\}$$

On représente les formules par des arbres étiquettés.
La *profondeur* d'une formule est la longueur de sa plus longue branche.
Sa *taille* est celle de l'arbre associé.
$\equiv$ désigne l'égalité “syntaxique” entre formules.
On utilise des mots sur le vocabulaire des entiers naturels pour désigner les positions des sous formules.
On note par $\phi|_p$ la sous-formule de ϕ à la position p, et par $\phi[\psi]_p$ la formule obtenue en remplaçant $\phi|_p$ par ψ dans ϕ (en respectant les catégories syntaxiques).

Formules, suite

x a une occurrence libre (position 1.1) et une occurence liée (position 2.1.1) dans la formule $P(x) \vee \forall x Q(x)$, en considérant $\forall x$ comme un opérateur unaire indexé par x :
en pratique, la variable liée x est représentée par un pointeur sur le noeud correspondant. Deux occurrences de x liées par le même quantificateur pointeront sur le même noeud. Deux variables distinctes ou liées par des quantificateurs distincts pointeront sur des noeuds distincts. Les variables libres pointent sur des noeuds fictifs rajoutés en tête de la formule.
Cette représentation est due à De Bruijn.

Substitutions

On appelle substitution toute application (en notation postfixée) de l'ensemble des variables dans les termes.
On appelle *domaine* de la substitution σ l'ensemble $\mathcal{D}om(\sigma) = \{x \in \mathcal{X} \mid x\sigma \neq x\}$.
Une substitution de domaine fini sera noté en extension, sous la forme $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$.
On note par $\sigma_{|X}$ la restriction de σ à $\mathcal{D}om(\sigma) \cap X$.
On appelle *image* de la substitution σ l'ensemble $\Im(\sigma) = \bigcup_{x \in \mathcal{D}om(\sigma)} \mathcal{V}ar(x\sigma)$.
L'application d'une substitution est étendue successivement aux termes, aux atomes, puis aux formules comme suit :

Substitutions, suite

Vars $x\sigma = \sigma(x)$

Terms $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$

Atoms $P(t_1, \ldots, t_n)\sigma = P(t_1\sigma, \ldots, t_n\sigma)$

Forms $(A \wedge B)\sigma = A\sigma \wedge B\sigma$

$(A \vee B)\sigma = A\sigma \vee B\sigma$

$(A \Rightarrow B)\sigma = A\sigma \Rightarrow B\sigma$

$(\neg A)\sigma = \neg(A\sigma)$

$(\exists x A)\sigma = \exists y(A\sigma')$

avec $\sigma' = \sigma_{|\neq x} \cup \{x \mapsto y\}$ pour y fraiche

$(\forall x A)\sigma = \forall y(A\sigma')$

avec $\sigma' = \sigma_{|\neq x} \cup \{x \mapsto y\}$ pour y fraiche

Sémantique

Vérité

x	y	$\neg_{Bool}(x)$	$\wedge_{Bool}(x,y)$	$\vee_{Bool}(x,y)$	$\Rightarrow_{Bool}(x,y)$
T	T	F	T	T	T
T	F		F	T	F
F	T	T	F	T	T
F	F		F	F	T

Une $(\mathcal{F}, \mathcal{P})$-*structure* (ou *structure*) est une paire (S, I) telle que S est un ensemble appelé *domaine de discours* muni
- d'une famille d'opérations, telle que f_I est d'arité n ssi $f \in \mathcal{F}$ d'arité n,
- d'une famille de relations telle que P_I est un sous ensemble de D^n ssi P est d'arité n.

Lorsque le symbole q est d'arité nulle, la relation q_I est une *valeur de vérité*.
On abrégera souvent (S, I) en S lorsque l'interprétation est déterminée sans ambigité par le contexte.

Exemples

Domaine	Opérations	Relations	Nom
N	$0, S, +$	$=, \leq$	Presburger
N	$+, *, 0, S$	$=, \leq$	Peano
R	$+, *, 0, S$	$=, \leq$	Tarski
$\{T, F\}$	$\emptyset$	$\{p_0, \ldots, p_n\}$	Prop.
$T(F)$	$\hat{F}$	$\{=\}$	F-Algèbre
$T(F)$	$\hat{F}$	$\hat{R}$	Herbrand

Dans le cas du modèle de Herbrand :

$$\forall f \in \mathcal{F},\ \hat{f}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$$

$$s = t \text{ est égal à } T \text{ ssi } s \equiv t$$

$$\forall R \in \mathcal{P} \setminus \{=\}, \hat{R}(t_1, \ldots, t_n) \in \{T, F\}$$

Interprétation des formules

valuation : $\quad v : \mathcal{X} \longrightarrow S$

Valeur de vérité de A : $\quad [A]_{I,v}$ (ou $[A]$)

- $[y] = v(y)$
- $[f(t_1, \ldots, t_p)] = f_I([t_1], \ldots, [t_p])$
- $[R(t_1, \ldots, t_p)] = R_I([t_1], \ldots, [t_p])$
- $[0] = F, \quad [1] = T$

  $[\neg A] = \neg_{Bool}[A]$

  $[A \wedge B] = [A] \wedge_{Bool} [B]$

  $[A \vee B] = [A] \vee_{Bool} [B]$

  $[A \Rightarrow B] = [A] \Rightarrow_{Bool} [B]$

  $[\exists x A]_v = T$ si $\exists d \in S$ t.q. $[A]_{v(x)=d} = T$

  $[\forall x A]_v$ si $\forall d \in S$, $[A]_{v(x)=d} = T$

Interprétation des formules

Lemma

Soient v vet v' t.q. $\forall x \in \mathcal{V}ar(A)$, $v(x) = v'(x)$
Alors $[A]_v = [A]_{v'}$.

Proof.

Par récurrence sur la structure de A. □ □

Interprétation et substitutions sont compatibles :

Lemma

Pour toute valuation v, $[A\sigma]_v = [A]_{v'}$, avec $v'(x) = [x\sigma]_v$

Proof.

Par récurrence sur la structure de A. □ □

Interprétation des formules

Lemma

Soient v et v' t.q. $\forall x \in \mathcal{V}ar(A)$, $v(x) = v'(x)$
Alors $[A]_v = [A]_{v'}$.

Proof.

Par récurrence sur la structure de A. $\square$ $\square$

Interprétation et substitutions sont compatibles :

Lemma

Pour toute valuation v, $[A\sigma]_v = [A]_{v'}$, *avec* $v'(x) = [x\sigma]_v$

Proof.

Par récurrence sur la structure de A. $\square$ $\square$

Validé et satisfiabilité

A est *satisfiable* dans (S, I) si $\exists v$ t.q. $[A]_v = T$
A est *satisfiable* s'il existe une structure (S, I) dans laquelle A est satisfiable, et *insatisfiable* dans le cas contraire
A est *valide* dans (S, I) ou (S, I) est un *modèle* de A, noté $(S, I) \models A$, si pour toute valuation v, $[A]_v = T$
A est (universellement) *valide*, noté $\models A$, si A est valide dans toute structure (S, I), et on dit alors également que A est une *tautologie*.

suite

A est satisfiable ssi $\exists \mathcal{V}ar(A)A$ est satisfiable
A est valide si sa clôture universelle $\forall \mathcal{V}ar(A)A$ est valide
A est valide ssi $\neg A$ est insatisfiable.

Conséquence et équivalence sémantiques

B est *conséquence sémantique* de A : $A \models B$ si pour toute structure (S, I) et toute valuation v, $[B]_v = T$ chaque fois que $[A]_v = T$.

La relation de conséquence sémantique s'étend naturellement à des ensembles de formules à droite ou à gauche du symbole $\models$

Pour des formules closes, $A \models B$ ssi $(S, I) \models B$ pour tout modèle (S, I) de A.

La relation de conséquence sémantique est un préordre dont l'équivalence associée notée $\equiv$ est l'*équivalence sémantique*

On obtient une structure d'algèbre de Boole ou d'anneau Booléen suivant le choix des opérateurs.

Algèbre de Boole

$$A \Rightarrow B \equiv B \vee \neg A$$

$$A \oplus B \equiv (A \wedge \neg B) \vee (B \wedge \neg A)$$

$$A \vee (B \vee C) \equiv (A \vee B) \vee C$$

$$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$$

$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$$

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$

$$\neg(A \vee B) \equiv (\neg A) \wedge (\neg B)$$

$$\neg(A \wedge B) \equiv (\neg A) \vee (\neg B)$$

$$A \vee B \equiv B \vee A \qquad A \wedge B \equiv B \wedge A$$

$$A \vee \neg A \equiv 1 \qquad A \wedge \neg A \equiv 0$$

$$A \vee 0 \equiv A \qquad A \vee 1 \equiv 1 \qquad A \vee A \equiv A$$

$$A \wedge 0 \equiv 0 \qquad A \wedge 1 \equiv A \qquad A \wedge A \equiv A$$

$$\neg 1 \equiv 0 \qquad \neg 0 \equiv 1 \qquad \neg\neg A \equiv A$$

Quantificateurs

$$
\begin{array}{rcll}
\neg\forall x A & \equiv & \exists x \neg A & \\
\neg\exists x A & \equiv & \forall x \neg A & \\
(\forall x A) \wedge B & \equiv & \forall x (A \wedge B) & \multirow{2}{*}{} \\
(\forall x A) \vee B & \equiv & \forall x (A \vee B) & \text{if } x \notin \mathcal{V}ar(B) \\
(A \Rightarrow \forall x B) & \equiv & \forall x (A \Rightarrow B) & \\
(A \Rightarrow \exists x B) & \equiv & \exists x (A \Rightarrow B) & \text{if } x \notin \mathcal{V}ar(A) \\
(\forall x A \Rightarrow B) & \equiv & \exists x (A \Rightarrow B) & \\
(\exists x A \Rightarrow B) & \equiv & \forall x (A \Rightarrow B) & \text{if } x \notin \mathcal{V}ar(B) \\
\end{array}
$$

$$\forall x\ A \equiv \forall y\ A\{x \mapsto y\} \qquad \text{if } y \notin \mathcal{V}ar(A)$$

Théorie d'une structure

La *théorie* d'une structure (S, I) est l'ensemble des formules closes de $CP1$ dont (S, I) est un modèle. On la note $Th(S, I)$.

Un ensemble Γ de formules closes est une *théorie* s'il est clos par conséquence sémantique, cad $A \in \Gamma$ ssi $\Gamma \models A$.

Une théorie Γ est *consistante* si elle ne contient pas à la fois une formule A et sa négation $\neg A$.

Une théorie Γ est *complète* si pour toute formule A, soit $A \in \Gamma$ soit $\neg A \in \Gamma$.

Axiomatisations

Une théorie Γ est *finiment axiomatisable* s'il existe un sous-ensemble fini de Γ dont Γ soit conséquence sémantique.

Une théorie Γ est *récursivement axiomatisable* s'il existe un sous-ensemble récursif de Γ dont Γ soit conséquence sémantique.

Une théorie Γ est *décidable* s'il existe une procédure qui termine toujours, telle que pour toute formule A la procédure retourne oui si $A \in \Gamma$ et non dans le cas contraire.

Une théorie Γ est *semi-décidable* s'il existe une procédure telle que pour toute formule A la procédure retourne oui si $A \in \Gamma$ et non ou ne termine pas dans le cas contraire.

L'arithmétique de Presburger (resp. Peano, Tarski) est la théorie de la structure Presburger (resp. Peano, Tarski).

Lemma

L'arithmétique de Presburger est décidable.
L'arithmétique de Peano est indécidable.
La théorie de Tarski est décidable.

La théorie du graphe $(\{1,2\},\{(1,2),(2,1)\})$ (on utilise $R \in \mathcal{P}_2$ pour la relation de succession) contient les formules

$\forall xy(xRy \Rightarrow yRx) \qquad \forall x.\neg xRx$

$\forall xy(xRy \vee \neg xRy)$

Exemples

L'arithmétique de Presburger (resp. Peano, Tarski) est la théorie de la structure Presburger (resp. Peano, Tarski).

Lemma

L'arithmétique de Presburger est décidable.
L'arithmétique de Peano est indécidable.
La théorie de Tarski est décidable.

La théorie du graphe $(\{1,2\},\{(1,2),(2,1)\})$ (on utilise $R \in \mathcal{P}_2$ pour la relation de succession) contient les formules

$\forall xy(xRy \Rightarrow yRx)$ $\quad \forall x.\neg xRx$

$\forall xy(xRy \vee \neg xRy)$

L'arithmétique de Presburger (resp. Peano, Tarski) est la théorie de la structure Presburger (resp. Peano, Tarski).

Lemma

L'arithmétique de Presburger est décidable.
L'arithmétique de Peano est indécidable.
La théorie de Tarski est décidable.

La théorie du graphe $(\{1,2\},\{(1,2),(2,1)\})$ (on utilise $R \in \mathcal{P}_2$ pour la relation de succession) contient les formules

$\forall xy(xRy \Rightarrow yRx) \qquad \forall x.\neg xRx$

$\forall xy(xRy \vee \neg xRy)$

Égalité

La théorie du prédicat d'égalité est finiment axiomatisable si $\mathcal{F}, \mathcal{R}$ sont finis :

$$
\begin{array}{cl}
\forall x \; x = x & \text{réflexivité} \\
\forall xy \; x = y \Rightarrow y = x & \text{symétrie} \\
\forall xyz \; x = y \wedge y = z \Rightarrow x = z & \text{transitivité} \\
\forall \overline{xy} \; \overline{x} = \overline{y} \Rightarrow f(\overline{x}) = f(\overline{y}) & \text{congruence } (\forall f \in \mathcal{F}) \\
\forall \overline{xy} \; \overline{x} = \overline{y} \Rightarrow R(\overline{x}) \leftrightarrow R(\overline{y}) &
\end{array}
$$

Groupes

La théorie des groupes est engendrée par exemple par les trois axiomes suivants :

$$\forall xyz\ x * (y * z) = (x * y) * z$$
$$\forall x\ e * x = x$$
$$\forall x \exists y\ x * y = e$$

La théorie des groupes n'est pas complète, car il existe des groupes commutatifs et d'autres qui ne le sont pas.

Peano

$$\forall x\ \neg(S(x) = 0) \qquad \forall xy\ S(x) = S(y) \Rightarrow x = y$$
$$\forall x\ x + 0 = x \qquad \forall xy\ x + S(y) = S(x + y)$$
$$\forall x\ x \times 0 = 0 \qquad \forall xy\ x \times S(y) = x \times y + x$$
$$\forall x\ \neg(x < 0) \qquad \forall x\ 0 < S(x)$$
$$\forall xy\ S(x) < S(y) \Rightarrow x < y$$
$$(A(0) \wedge \forall x(A(x) \Rightarrow A(S(x)))) \Rightarrow \forall x A(x)$$
$$x \text{ libre dans } A$$

L'arithmétique de Peano est récursive.
L'arithmétique de Peano est incompléte (Gödel).
Montrer que les 2 définition de Peano coincident est laissé à la sagacité du lecteur.

En plus des axiomes de l'égalité :

$$\forall \overline{xy}(f(\bar{x}) = f(\bar{y}) \leftrightarrow \bar{x} = \bar{y}) \qquad \text{Décomposition}$$

$$\forall \overline{xy}(f(\bar{x}) = g(\bar{y}) \leftrightarrow 0) \qquad \text{Conflit}$$

$$\forall xy\ [x[y]_{p \neq \Lambda} = y \leftrightarrow 0] \qquad \text{Occurrence}$$

L'axiomatisation de $x[y]_p$ est laissée au lecteur.

$$\forall x\ (\exists \bar{y}\ x = \bigvee_{f \in \mathcal{F}} f(\bar{y})) \qquad \text{Monde clos}$$

Theorem

La théorie des arbres finis étiquetés est complète.

En plus des axiomes de l'égalité :

$$\forall \overline{x}\overline{y}(f(\overline{x}) = f(\overline{y}) \leftrightarrow \overline{x} = \overline{y}) \qquad \text{Décomposition}$$

$$\forall \overline{x}\overline{y}(f(\overline{x}) = g(\overline{y}) \leftrightarrow 0) \qquad \text{Conflit}$$

$$\forall xy\ [x[y]_{p \neq \Lambda} = y \leftrightarrow 0] \qquad \text{Occurrence}$$

L'axiomatisation de $x[y]_p$ est laissée au lecteur.

$$\forall x\ (\exists \overline{y}\ x = \bigvee_{f \in \mathcal{F}} f(\overline{y})) \qquad \text{Monde clos}$$

Theorem

La théorie des arbres finis étiquetés est complète.

Structures isomorphes

Les structures (S, I) et $(S', I,)$ sont *isomorphes* s'il existe une bijection $\xi : S \longrightarrow S'$ t.q. :
- pour tout symbole de fonction $f \in \mathcal{F}_n$ et tout $\overline{d} \in S^n$, $f_I(\overline{d}) = c$ ssi $f_{I'}(\xi(\overline{d})) = \xi(c)$,
- pour tout symbole de relation $R \in \mathcal{P}_n$ et tout $\overline{d} \in S^n$, $\overline{d} \in R_I$ ssi $\xi(\overline{d}) \in R_{I'}$.

Deux graphes qui ne diffèrent que par le nom des sommets sont isomorphes.
Deux structures isomorphes sont modèles des même formules.

Structures isomorphes

Les structures (S, I) et $(S', I,)$ sont *isomorphes* s'il existe une bijection $\xi : S \longrightarrow S'$ t.q. :
- pour tout symbole de fonction $f \in \mathcal{F}_n$ et tout $\overline{d} \in S^n$, $f_I(\overline{d}) = c$ ssi $f_{I'}(\xi(\overline{d})) = \xi(c)$,
- pour tout symbole de relation $R \in \mathcal{P}_n$ et tout $\overline{d} \in S^n$, $\overline{d} \in R_I$ ssi $\xi(\overline{d}) \in R_{I'}$.

Deux graphes qui ne diffèrent que par le nom des sommets sont isomorphes.

Deux structures isomorphes sont modèles des même formules.

Structures isomorphes

Les structures (S, I) et $(S', I,)$ sont *isomorphes* s'il existe une bijection $\xi : S \longrightarrow S'$ t.q. :
- pour tout symbole de fonction $f \in \mathcal{F}_n$ et tout $\overline{d} \in S^n$, $f_I(\overline{d}) = c$ ssi $f_{I'}(\xi(\overline{d})) = \xi(c)$,
- pour tout symbole de relation $R \in \mathcal{P}_n$ et tout $\overline{d} \in S^n$, $\overline{d} \in R_I$ ssi $\xi(\overline{d}) \in R_{I'}$.

Deux graphes qui ne diffèrent que par le nom des sommets sont isomorphes.
Deux structures isomorphes sont modèles des même formules.

Structures isomorphes, suite

Lemma

Deux structures finies qui valident le même ensemble de formules sont isomorphes.

La réciproque est fausse. Les deux structures $(Q, \{<_Q\})$ et $(R, \{<_R\})$ ne sont pas isomorphes, mais sont modèles des même formules du premier ordre.
Ces deux structures sont distinguables, mais pas par des formules du premier ordre construites sur le seul symbole relationnel $<$ (il faut enrichir la syntaxe avec des symboles d'opération ou passer au second ordre).

Structures isomorphes, suite

Lemma

Deux structures finies qui valident le même ensemble de formules sont isomorphes.

La réciproque est fausse. Les deux structures $(Q, \{<_Q\})$ et $(R, \{<_R\})$ ne sont pas isomorphes, mais sont modèles des même formules du premier ordre.

Ces deux structures sont distinguables, mais pas par des formules du premier ordre construites sur le seul symbole relationnel $<$ (il faut enrichir la syntaxe avec des symboles d'opération ou passer au second ordre).

Lemma

Deux structures finies qui valident le même ensemble de formules sont isomorphes.

La réciproque est fausse. Les deux structures $(Q, \{<_Q\})$ et $(R, \{<_R\})$ ne sont pas isomorphes, mais sont modèles des même formules du premier ordre.
Ces deux structures sont distinguables, mais pas par des formules du premier ordre construites sur le seul symbole relationnel $<$ (il faut enrichir la syntaxe avec des symboles d'opération ou passer au second ordre).

Lemma

La clôture par conséquence sémantique d'un ensemble de formules qui a un modèle unique à isomorphisme près, est une théorie complète.

Problème dual : programmer en logique.

Definition

Une relation R d'arité n sur une $(F, \mathcal{P})$-structure (S, I) est *définissable* dans la logique du premier ordre s'il existe une formule $A \in \mathcal{CP}1(\mathcal{F}, \mathcal{P}, \mathcal{X})$ dépendant des n variables libres $\overline{x}$ telle que $[A]_{I,\overline{x}}(\overline{d}) = T$ ssi $\overline{d} \in R$. La définition s'étend aux classes de structures.

Définissabilité

Lemma

La clôture par conséquence sémantique d'un ensemble de formules qui a un modèle unique à isomorphisme près, est une théorie complète.

Problème dual : programmer en logique.

Definition

Une relation R d'arité n sur une $(F, \mathcal{P})$-structure (S, I) est *définissable* dans la logique du premier ordre s'il existe une formule $A \in \mathcal{CP}1(\mathcal{F}, \mathcal{P}, \mathcal{X})$ dépendant des n variables libres $\overline{x}$ telle que $[A]_{I,\overline{x}}(\overline{d}) = T$ ssi $\overline{d} \in R$. La définition s'étend aux classes de structures.

Exemple : la primalité

La relation de la structure Peano $R(x)$ ssi x est premier est définissable par la formule :

$$\forall yz\ x = y \times z \Rightarrow \begin{array}{c}(y = x \wedge z = S(0)) \\ \vee(z = x \wedge y = S(0))\end{array}$$

L'interprétation de cette formule dans la structure Peano vaut T pour tout nombre premier et F pour tout nombre non premier.

De très nombreuses propriétés élémentaires des structures usuelles comme les graphes ne sont pas exprimables en logique du premier ordre. La logique du second ordre a beaucoup plus d'expressivité.

Exemple : la primalité

La relation de la structure Peano $R(x)$ ssi x est premier est définissable par la formule :

$$\forall yz \; x = y \times z \Rightarrow \begin{array}{l} \;\;(y = x \wedge z = S(0)) \\ \vee(z = x \wedge y = S(0)) \end{array}$$

L'interprétation de cette formule dans la structure Peano vaut T pour tout nombre premier et F pour tout nombre non premier.

De très nombreuses propriétés élémentaires des structures usuelles comme les graphes ne sont pas exprimables en logique du premier ordre. La logique du second ordre a beaucoup plus d'expressivité.

K. R. Apt and M. H. van Emden
Contributions to the theory of logic programming.
Journal of the ACM, 29(3), July 1982.

Alain Colmerauer
Prolog II. Manuel de référence et modèle théorique.
Research report, GIA Luminy, Marseille, March 1982.

Alain Colmerauer
Equations and inequations on finite and infinite trees.
In *FGCS'84 Proceedings*, pages 85–99, November 1984.

Alain Colmerauer
Opening the Prolog III universe.
Byte Magazine, August 1987.

N. Dershowitz and J.-P. Jouannaud
Rewrite systems.
In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–309. North-Holland, 1990.

Gérard Huet
Résolution d'équations dans les langages d'ordre $1, 2, \ldots \omega$.
Thèse d'Etat, Univ. Paris 7, 1976.

Jean-Pierre Jouannaud and Claude Kirchner
Solving equations in abstract algebras: A rule-based survey of unification.
In Jean-Louis Lassez and Gordon Plotkin eds, *Computational Logic: Essays in Honor of Alan*

Robert Kowalski
Semantic trees in automatic theorem-proving.
Machine Intelligence, 4:86–101, 1969.

Robert Kowalski
Search strategies for theorem-proving.
Machine Intelligence, 5:181–201, 1970.

Robert A. Kowalski and M. H. van Emden
The semantics of predicate logic as a programming language.
J. of the Association for Computing Machinery, 23:733–742, October 1976.

Alberto Martelli and Ugo Montanari
An efficient unification algorithm.
ACM Transactions on Programming Languages and Systems, 4(2):258–282, April 1982.

J. A. Robinson
A machine-oriented logic based on the resolution principle.
Journal of the ACM, 12(1):23–41, 1965.

J. A. Robinson
The generalized resolution principle.
Machine Intelligence, 3, 1968.