Logique

L3 informatique – Université de la Réunion

J. Diatta et F. Mesnard

Table des matières

Chapitre 1

Systèmes formels

La théorie des *systèmes formels* constitue un cadre général dans lequel on peut exprimer et étudier, de façon rigoureuse, la notion de mécanisme déductif. En particulier, les concepts de démonstration et de théorème deviennent des concepts mathématiques objets, au sujet desquels on peut établir des résultats (qualifiés de *métamathématiques*) au même titre qu'on établit des résultats au sujet des nombres entiers ou des matrices inversibles.

La réalisation d'un système formel fournit un cadre de représentation d'une réalité donnée où des règles automatiques produisent tous les éléments vrais. Le but est d'avoir un petit nombre de vérités initiales (ce sont les *axiomes*) et de disposer de mécanismes de raisonnement (ce sont les *règles d'inférence*) pour révéler des vérités cachées.

Par exemple, un système expert, qui est constitué d'une base de faits et de règles permettant d'inférer de nouvelles connaissances, est un système formel.

I Définitions

I.A Ensembles récursifs et récursivement énumérables

Soit E et F deux ensembles tels que $E \subseteq F$ et x un objet arbitraire de F. Considérons l'assertion "x appartient à E".

– L'ensemble E est dit *reconnaissable* ou *récursif* s'il l'on dispose d'un algorithme, au lieu d'un critère non nécessairement exécutable, pour calculer, quel que soit x, la valeur de vérité de l'assertion précédente.

– L'ensemble E sera dit *semi-reconnaissable* ou *récursivement énumérable* s'il existe un algorithme qui, pour tout x appartenant à E, établira cette propriété en un nombre fini d'étapes (mais on n'exige pas que cet algorithme se termine lorsque x n'est pas élément de E).

Exemple I.1 L'ensemble des entiers naturels et l'ensemble des entiers naturels pairs sont des ensembles récursifs.

Remarque I.1 1) Tout ensemble récursif est récursivement énumérable.
2) Il existe des ensembles récursivement énumérables qui ne sont pas récursifs.

I.B Systèmes formels

On appelle *système formel* (*sf*) S la donnée de :

(a) un alphabet dénombrable Σ_S,

(b) un sous-ensemble récursif $\mathbb{F}_S$ de l'ensemble Σ_S^* des *suites finies d'éléments de* Σ_S, appelé ensemble des *formules bien formées* (*fbf*) de S,

(c) un sous-ensemble récursif $\mathbb{A}_S \subseteq \mathbb{F}_S$, appelé ensemble des *axiomes* de S,

(d) un ensemble fini $\Omega_S = \{r_1, \ldots, r_n\}$ de *règles de d'inférence* (qui permettent d'enrichir le système) tel que, pour chaque r_i il existe un algorithme A_i qui, pour toute donnée $f_1, \ldots, f_m, f \in \mathbb{F}_S$, calcule la valeur de vérité de l'assertion "on peut déduire la formule f à partir des formules $f_1, \ldots, f_m$ par la règle r_i", ce qui s'écrit :

$$f_1, \ldots, f_m \underset{r_i}{\vdash} f.$$

Remarque I.2 La construction d'un sf illustre bien le processus de définition inductive. En effet, étant donné un ensemble E d'objets, sa définition (description) par un schéma d'induction consiste en trois phases :

- en *phase initiale*, on choisit une *base*, i.e. un ensemble initial d'objets de l'ensemble ;

- en *phase inductive*, on définit un ensemble de *règles de production* permettant, à partir d'objets de l'ensemble, d'en produire de nouveaux ;

- en *phase de clôture*, on décide qu'un objet appartient à l'ensemble E si et seulement si on l'obtient à partir d'éléments de la base en enchaînant un nombre fini de règles et en produisant, si nécessaire, des objets intermédiaires.

Exemple I.2 Soit le système formel S_1 défini par :
$\Sigma_{S_1} = \{1, +, =\}$,
$\mathbb{F}_{S_1} = \{1^n + 1^m = 1^p \quad | \quad n, m, p \in \mathbb{N} \setminus \{0\}\}$ où $1^k = \underbrace{1 \cdots 1}_{k \text{ fois}}$,
$\mathbb{A}_{S_1} = \{1 + 1 = 11\}$,
$\Omega_{S_1} = \{r_1, r_2\}$ où :

$$1^n + 1^m = 1^p \underset{r_1}{\vdash} 1^{n+1} + 1^m = 1^{p+1}$$

et

$$1^n + 1^m = 1^p \underset{r_2}{\vdash} 1^n + 1^{m+1} = 1^{p+1}.$$

Remarques I.1 1) L'ensemble des fbf peut être défini par un schéma d'induction ; dans ce cas, on prendra bien garde de ne pas confondre les règles de production de ce schéma et les règles de d'inférence du système.
2) Les règles de d'inférence s'appellent aussi règles de *dérivation* ou de *déduction.*

II Définitions supplémentaires

On se place dans le cadre d'un sf $S = (\Sigma_S, \mathbb{F}_S, \mathbb{A}_S, \Omega_S)$.

– On appelle *déduction* (ou *démonstration* ou *preuve*) à partir de formules $h_1, \ldots, h_n$ toute suite finie de formules $f_1, \ldots, f_p$ telle que, pour tout $i \in \{1, \ldots, p\}$:

(a) ou bien f_i est un axiome

(b) ou bien f_i est l'une des formules $h_1, \ldots, h_n$

(c) ou bien f_i est obtenu par application d'une règle de déduction $r_k \in \Omega_S$ à partir de formules $f_{i_0}, \ldots, f_{i_l}$ placées avant f_i :

$$f_{i_0}, \ldots, f_{i_l} \underset{r_k}{\vdash} f_i \text{ avec } i_0, \ldots, i_l < i.$$

Remarque II.1 Dans le cas de la définition ci-dessus, on dit aussi que $f_1, \ldots, f_p$ est une déduction de f_p à partir des hypothèses $h_1, \ldots, h_n$, ce que l'on note : $h_1, \ldots, h_n \underset{S}{\vdash} f_p$.

– On appelle *théorème* de S toute formule t dont il existe une déduction (à partir de l'ensemble vide). On le note $\underset{S}{\vdash} t$ et l'ensemble des théorèmes de S est désigné par $\mathbb{T}_S$.

Remarque II.2 (i) Tout axiome est un théorème.
(ii) Toute formule située avant un théorème dans une déduction de celui-ci est un théorème.

– On dit que S est *cohérent* s'il existe des fbf de S qui ne sont pas des théorèmes.

– Si S comporte un symbole de négation $\neg$, on dit que S est *consistant* (ou *non-contradictoire*) s'il n'existe aucune fbf $\varphi \in \mathbb{F}_S$ telle que φ et $\neg\varphi$ sont des théorèmes de S. Dans le ces contraire, on dit que S est *inconsistant* ou *contradictoire*.

– On dit que les axiomes de S sont *indépendants* si, à chaque fois que l'on enlève un axiome à S, on obtient un sf ayant moins de théorèmes.

– La question de décidabilité dans un sf est la question qui se pose lorsque l'on cherche à savoir si une formule est un théorème du système ; le sf S est *décidable* si $\mathbb{T}_S$ est récursif.

– Lorsque l'on a en vue la modélisation d'un problème par un sf S et que l'on voudrait obtenir comme ensemble de théorèmes de S un ensemble T fixé à l'avance, on dit que :

S est *correct* si $\mathbb{T}_S \subseteq T$

S est *complet* si $T \subseteq \mathbb{T}_S$.

III Résultats généraux

Proposition III.1 *Soit S un système formel. Si $h \underset{S}{\vdash} g$ et $g \underset{S}{\vdash} f$, alors $h \underset{S}{\vdash} f$.*

Proposition III.2 *L'ensemble des déductions dans un système formel est récursif.*

Proposition III.3 *L'ensemble des théorèmes dans un système formel est récursivement énumérable.*

Proposition III.4 *Il existe des systèmes formels non décidables, i.e., dont l'ensemble des théorèmes n'est pas récursif.*

Remarque III.1 Ce résultat est fondamental car il montre que, bien que tous les objets intervenant dans la définition d'un sf soient récursifs, l'ensemble des théorèmes peut ne pas l'être, alors qu'il existe toujours un moyen algorithmique (i.e. un programme) qui détermine si une suite de formules est une déduction ou pas.

Chapitre 2

Calcul propositionnel

Le *calcul propositionnel* ou *logique des propositions* a pour objet l'étude des formes de raisonnement dont la validité est indépendante de la structure des propositions composantes et résulte uniquement de leurs propriétés d'être vraies ou fausses.

Deux aspects, liés l'un à l'autre, doivent être pris en compte :

– l'aspect *syntaxique* qui revient simplement à définir un système formel dans lequel les déductions qu'on peut faire conduisent à des théorèmes du calcul propositionnel ;

– l'aspect *sémantique* qui est l'interprétation des formules, fondée sur les valeurs de vérité de leurs propositions composantes.

La liaison entre les deux aspects est illustrée par le fait que les formules qui sont les tautologies (i.e. qui sont sémantiquement valides) sont les mêmes que les formules qui sont les théorèmes (i.e. qui sont syntaxiquement valides). Une conséquence de ce résultat est que le calcul propositionnel est décidable.

I Syntaxe du calcul propositionnel

Le langage propositionnel est défini par la donnée de :

- l'alphabet Σ composé de symboles $p_0, p_1, \ldots, p_n, \ldots$, appelés *variables propositionnelles* (ou *propositions atomiques*, ou *atomes*), de symboles de parenthèses (et), et de *connecteurs* $\neg$ (*négation*), $\vee$ (*disjonction*), $\wedge$ (*conjonction*), $\Longrightarrow$ (*implication logique*) et $\Longleftrightarrow$ (*équivalence logique*),
- l'ensemble $\mathbb{F}$ des formules bien formées qui est le plus petit ensemble de formules tel que :
  (1) tout atome est une formule,
  (2) si A et B sont des formules, alors (A), $\neg A$, $A \vee B$, $A \wedge B$, $A \Longrightarrow B$ et $A \Longleftrightarrow B$ sont des formules.

En l'absence de parenthèses, les priorités sont les suivantes : $\neg$ est prioritaire devant $\vee$ et $\wedge$ qui sont prioritaires devant $\Longrightarrow$ et $\Longleftrightarrow$; à priorités égales, l'évaluation se fait de la gauche vers la droite. Toutefois, il est fortement conseillé d'utiliser des parenthèses.

II Système formel du calcul propositionnel (sfcp)

– L'alphabet et l'ensemble des formules bien formées sont respectivement les ensembles Σ et $\mathbb{F}$ définis plus haut.

– L'ensemble des axiomes est l'ensemble $\mathbb{A}$ de toutes les formules de l'une des trois formes suivantes :

SA_1 : $A \Longrightarrow (B \Longrightarrow A)$ (*conséquence de l'hypothèse*)

SA_2 : $(A \Longrightarrow (B \Longrightarrow C)) \Longrightarrow ((A \Longrightarrow B) \Longrightarrow (A \Longrightarrow C))$ (*auto-distributivité de* $\Longrightarrow$)

SA_3 : $(\neg B \Longrightarrow \neg A) \Longrightarrow (A \Longrightarrow B)$ (*contraposée partielle*)

où A, B et C sont des formules quelconques.

– L'ensemble des règles de déduction est l'ensemble Ω réduit à un seul élément appelé le *modus ponens* (*mp*) où :

$$A, A \Longrightarrow B \underset{mp}{\vdash} B.$$

On écrit aussi

$$\frac{\begin{array}{l} A \\ A \Longrightarrow B \end{array}}{\therefore B}$$

Nota Les expressions SA_1, SA_2 et SA_3 s'appellent des *schémas d'axiomes* ; à chacune d'elles correspond une infinité d'axiomes. Par exemple à SA_1 correspond :

$$\begin{array}{c} A \Longrightarrow (B \Longrightarrow A), \\ (A \Longrightarrow B) \Longrightarrow ((C \Longrightarrow B) \Longrightarrow (A \Longrightarrow B)), \\ \ldots \end{array}$$

Proposition II.1 *Pour toute formule $A \in \mathbb{F}$ on a $\vdash (A \Longrightarrow A)$.*

Proposition II.2 ***(Théorème de déduction)**. Soit $A_1, \ldots, A_{n-1}, A_n, B \in \mathbb{F}$. Alors $A_1, \ldots, A_{n-1} \vdash (A_n \Longrightarrow B)$ ssi $A_1, \ldots, A_{n-1}, A_n \vdash B$.*

Exemple II.1 Pour montrer que

$$\vdash (A \Longrightarrow (B \Longrightarrow C)) \Longrightarrow (B \Longrightarrow (A \Longrightarrow C)),$$

il suffit d'établir que

$$(A \Longrightarrow (B \Longrightarrow C)) \vdash (B \Longrightarrow (A \Longrightarrow C))$$

et donc, il suffit d'établir que

$$(A \Longrightarrow (B \Longrightarrow C)), B \vdash (A \Longrightarrow C)$$

et donc, il suffit d'établir que

$$(A \Longrightarrow (B \Longrightarrow C)), B, A \vdash C$$

ce qui est immédiat par :

$f_1 : A \Longrightarrow (B \Longrightarrow C)$	hypothèse
$f_2 : B$	hypothèse
$f_3 : A$	hypothèse
$f_4 : B \Longrightarrow C$	$mp(f_1, f_3)$
$f_5 : C$	$mp(f_2, f_4)$.

III Sémantique du calcul propositionnel

On appelle *interprétation* de $\mathbb{F}$ toute application $I : \{p_0, p_1, \ldots, p_n, \ldots\} \to \{V, F\}$ (*Vrai, Faux*) qui attribue une valeur de vérité à chaque atome. L'application I est alors étendue à $\mathbb{F}$ par les formules :

$I(\neg A) = \neg[I(A)]$ $\qquad$ $I(A \vee B) = \vee[I(A), I(B)]$

$I(A \wedge B) = \wedge[I(A), I(B)]$ $\qquad$ $I(A \Longrightarrow B) = \Longrightarrow[I(A), I(B)]$

$I(A \iff B) = \iff [I(A), I(B)]$

où $\neg[\centerdot]$ (resp. $\vee[\centerdot, \centerdot]$, $\wedge[\centerdot, \centerdot]$, $\Longrightarrow[\centerdot, \centerdot]$, $\iff [\centerdot, \centerdot]$) est l'application de $\{V, F\}$ (resp. $\{V, F\} \times \{V, F\}$) dans $\{V, F\}$ définie par la *table de vérité*

X	$\neg X$
V	F
F	V

respectivement

X, Y	$\vee[X, Y]$	$\wedge[X, Y]$	$\Longrightarrow[X, Y]$	$\iff [X, Y]$
V, V	V	V	V	V
V, F	V	F	F	F
F, V	V	F	V	F
F, F	F	F	V	V

Exemple III.1 Soit I une interprétation de $\mathbb{F}$ telle que $I(p_0) = V$, $I(p_1) = F$, $I(p_2) = V$. Alors :

$$\begin{aligned} I((p_0 \Longrightarrow p_1) \vee \neg p_2) &= \vee[I(p_0 \Longrightarrow p_1), I(\neg p_2)] \\ &= \vee[\Longrightarrow[I(p_0), I(p_1)], \neg[I(p_2)]] \\ &= \vee[\Longrightarrow[V, F], \neg[V]] \\ &= \vee[F, F] \\ &= F \end{aligned}$$

III.A Définitions sur l'interprétation sémantique des formules

Il existe divers types de formules qui se caractérisent par leur interprétation en fonction des valeurs de vérité assignées aux atomes qui les composent. Certaines sont toujours vraies quelle que soit la valeur de vérité des atomes, d'autres toujours fausses, et enfin certaines prennent les deux valeurs. Dans ce dernier cas, il est important de comparer les formules entre elles pour en trouver de plus simples ayant la même interprétation.

– On dit qu'une formule $B \in \mathbb{F}$ est *conséquence sémantique* d'un ensemble de formules $\Gamma \subseteq \mathbb{F}$ si pour toute interprétation I telle que pour tout $A \in \Gamma$ $I(A) = V$, on a aussi $I(B) = V$. On écrit alors $\Gamma \models B$.

– On dit que deux formules $A \in \mathbb{F}$ et $B \in \mathbb{F}$ sont *sémantiquement équivalentes* si pour chaque interprétation I, $I(A) = I(B)$; autrement dit, si $A \models B$ et $B \models A$. On écrit alors $A \equiv B$.

– On dit qu'un ensemble de formules $\Gamma \subseteq \mathbb{F}$ est *satisfaisable* (ou *consistant* ou *non-contradictoire*) s'il existe une interprétation I telle que, pour tout$A \in \Gamma$, on ait $I(A) = V$ (une telle interprétation est appelée *modèle* de Γ). Dans le cas contraire, on dit que Γ est *insatisfaisable* (ou *inconsistant* ou *contradictoire*). Une formule contradictoire est dite une *contradiction* et sera notée F_0.

– Une formule A est dite une *tautologie* si pour toute interprétation I on a $I(A) = V$. On écrit alors $\models A$. Dans le cas contraire, A est dite *réfutable*. Toute tautologie sera notée V_0.

III.B Simplification de formules de $\mathbb{F}$

Les simplifications de formules, ci-après, servent, en particulier, à trouver la valeur de vérité d'une proposition en se ramenant à des formules plus simples.

– Double négation :
$\neg(\neg A) \equiv A$.

– Associativité et commutativité de $\vee$ et $\wedge$:
$A \vee (B \vee C) \equiv (A \vee B) \vee C$ et $(A \vee B) \equiv (B \vee A)$
$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$ et $(A \wedge B) \equiv (B \wedge A)$.

– Idempotence de $\vee$ et $\wedge$:
$(A \vee A) \equiv A$ et $(A \wedge A) \equiv A$.

– Tautologie et contradiction :
$(A \vee \neg A) \equiv V_0$ et $(A \wedge \neg A) \equiv F_0$.

– Double distributivité :
$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$ et $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$.

– Absorption :
$A \vee (A \wedge B) \equiv A$ et $A \wedge (A \vee B) \equiv A$.

– Lois de de Morgan :
$\neg(A \vee B) \equiv (\neg A \wedge \neg B)$ et$\neg(A \wedge B) \equiv (\neg A \vee \neg B)$.

– Lien entre implication, équivalence et négation, disjonction ou conjonction :
$A \Longrightarrow B \equiv \neg A \vee B$
$A \iff B \equiv (A \Longrightarrow B) \wedge (B \Longrightarrow A) \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$.

III.C Quelques remarques

C.1 Négation d'une implication

Considérons les propositions atomiques

p_1 : Jacques va à la plage

p_2 : Marie représente Jacques au CA

et l'implication
$p_1 \Longrightarrow p_2$: Si Jacques va à la plage, alors Marie représentera Jacques au CA.
Nous souhaitons écrire la négation de $p_1 \Longrightarrow p_2$ d'une manière autre qu'en disant simplement : "ce n'est pas le cas que si Jacques va à la plage, alors Marie représentera Jacques au CA." Pour celà, nous considérons les équivalences sémantiques suivantes.

$$\begin{aligned} \neg(p_1 \Longrightarrow p_2) &\equiv \neg(\neg p_1 \vee p_2) \\ &\equiv \neg\neg p_1 \wedge \neg p_2 \\ &\equiv p_1 \wedge \neg p_2 \end{aligned}$$

D'où la négation de $p_1 \Longrightarrow p_2$
$\neg(p_1 \Longrightarrow p_2)$: Jacques va à la plage, mais Marie ne représente pas Jacques au CA.

Remarque III.1 La négation d'une implication n'est pas une autre implication.

C.2 Contraposée, réciproque et inverse

Considérons les propositions atomiques

p_1 : Aujourd'hui est Pâques

p_2 : Demain est lundi

Alors nous obtenons

- (L'implication $p_1 \Longrightarrow p_2$). Si aujourd'hui est Pâques, alors demain est lundi. (VRAI)
- (La contraposée $\neg p_2 \Longrightarrow \neg p_1$). Si demain n'est pas lundi, alors aujourd'hui n'est pas Pâques. (Egalement VRAI)
- (La réciproque $p_2 \Longrightarrow p_1$). Si demain est lundi, alors aujourd'hui est Pâques.
- (L'inverse $\neg p_1 \Longrightarrow \neg p_2$). Si aujourd'hui n'est pas Pâques, alors demain n'est pas lundi.

Remarque III.2 La réciproque est fausse pour tous les lundi de l'année sauf un. L'inverse est la contraposée de la réciproque.

C.3 Equivalence sémantique

Considérons les deux segments (a) et (b) de programme Pascal suivants, où x, y, z et i sont des variables entières :

```
z := 4;                                                    (a)
Pour i := 1 à 10 faire
    Début
        x := z − i;
        y := z + 3 ∗ i;
        Si ((x > 0) et (y > 0)) alors
            Ecrire('La valeur de la somme x + y est', x + y);
    fin;
```

```
z := 4;                                                    (b)
Pour i := 1 à 10 faire
    Début
        x := z − i;
        y := z + 3 ∗ i;
        Si x > 0 alors
            Si y > 0 alors
              Ecrire('La valeur de la somme x + y est', x + y);
    fin;
```

Le segment (a) utilise une structure de contrôle comparable à une formule de la forme $(p \wedge q) \Longrightarrow r$. Le segment (b) utilise, quant à lui, une forme de formule comparable aux implications embotées $p \Longrightarrow (q \Longrightarrow r)$. Ces deux segments conduisent au même résultat puisqu'utilisant des formes de formules sémantiquement équivalentes.

III.D Quelques résultats

Proposition III.1 *Soit $\Gamma \cup \{B\} \subseteq \mathbb{F}$ un ensemble de fbf du calcul propositionnel. Alors $\Gamma \models B$ ssi $\Gamma \cup \{\neg B\}$ est contradictoire.*

Proposition III.2 *Soient A et B deux formules de $\mathbb{F}$. Alors*
(1) $\models (A \Longrightarrow B)$ ssi $A \models B$.
(2) Si $\models A$ et $\models (A \Longrightarrow B)$, alors $\models B$.

Nota On vérifie aisément que les éléments de $\mathbb{A}$ sont des tautologies.

Proposition III.3 *(**Théorème de correction**). Soit $A \in \mathbb{F}$. Si $\vdash A$, alors $\models A$. Autrement dit, les formules qui sont des théorèmes (validité syntaxique) sont des tautologies (validité sémantique).*

Proposition III.4 *(**Théorème de complétude**). Soit $A \in \mathbb{F}$. Si $\models A$, alors $\vdash A$. Autrement dit, toutes les formules qui sont des tautologies sont des théorèmes.*

Corollaire III.1 *Le système formel du calcul propositionnel est décidable.*

Proposition III.5 *(**Théorème de compacité**). Soit Γ un ensemble de formules du calcul propositionnel. Si toute partie finie de Γ a un modèle, $I(A) = V$, alors Γ a un modèle.*

III.E Formes conjonctives

– Un atome est, rappelons le, une fbf réduite à une variable propositionnelle.

– Un *littéral* est soit un atome (*littéral positif*), soit la négation d'un atome (*littéral négatif*).

– Une *paire opposée* de littéraux est composée d'un atome et de sa négation.

– Une *clause* est une disjonction finie de littéraux ; une clause est une tautologie ssi elle contient une paire opposée de littéraux.

– Une *forme conjonctive* est une conjonction finie de clauses.

Remarque III.3 La *clause vide* (sans littéral) est la seule clause inconsistante. On la note $\Box$.

Proposition III.6 *Toute fbf du sfcp admet une forme conjonctive qui lui est sémantiquement équivalente.*

IV Résolution

Un ensemble $\{A_1, \ldots, A_n\}$ de fbf du calcul propositionnel étant donné, on cherche à savoir s'il est satisfaisable ou non. La procédure naturel et simple qui consiste à faire la table de vérité de $A_1 \wedge \cdots \wedge A_n$ puis à regarder s'il y a au moins un V dans la colonne principale de cette table , assure d'obtenir une réponse en temps fini. Toutefois, ce temps de calcul est de l'ordre de 2^n, où n est le nombre d'atomes intervenant dans les A_i.

La méthode de résolution de Robinson est plus simple et se généralise aux formules du calcul des prédicats du premier ordre. Il s'agit d'un algorithme simple de démonstration automatique. Elle évite le recours au modus ponens qui est un mécanisme de démonstration lent et fastidieux dans la plupart des cas. La méthode de résolution sert également à montrer qu'une formule est universellement valide en montrant que sa négation est contradictoire. Pour pouvoir l'appliquer, il convient de transformer la (les) formule(s) donnée(s) en un ensemble de clauses : c'est la *mise sous forme clausale*. Le mécanisme de résolution consiste alors à déduire la clause vide à partir de l'ensemble de clauses ainsi obtenues, dans le sf RSV (résolution sans variable) défini comme suit.

– $\Sigma_{RSV} = \{p_0, p_1, \ldots, p_n, \ldots\} \cup \{\neg, \vee\}$;

– $\mathbb{F}_{RSV}$ est l'ensemble de toutes les clauses construites à partir des variables propositionnelles ; on ne considère que les clauses sans répétition, i.e. ne contenant pas deux fois le même littéral ;

– $\mathbb{A}_{RSV} = \emptyset$;

– $\Omega_{RSV} = \{res\}$ avec

$$C,\, C' \underset{res}{\vdash} res(C, C'),$$

où C et C' sont deux clauses non tautologiques constituant une paire résoluble, i.e. $C = C_1 \vee p_i$ et $C' = C'_1 \vee \neg p_i$ et où $res(C, C')$, appelé résolvante de C et C', est la clause (sans répétition de littéraux) constituée des littéraux de C_1 et C'_1.

Exemple IV.1 On a

$$p_0 \vee p_1 \vee p_2,\ \neg p_0 \vee p_1 \vee p_2,\ \neg p_1 \vee p_2 \underset{RSV}{\vdash} p_2$$

grâce la déduction :

$f_1 : p_0 \vee p_1 \vee p_2$	(hypothèse)
$f_2 : \neg p_0 \vee p_1 \vee p_2$	(hypothèse)
$f_3 : p_1 \vee p_2$	$res(f_1, f_2)$
$f_4 : \neg p_1 \vee p_2$	(hypothèse)
$f_5 : p_2$	$res(f_3, f_4)$.

Remarques IV.1 Ce mécanisme n'est pas déterministe, dans la mesure où, à une étape donnée, il peut exister plusieurs possibilités de combiner des clauses.
2) Un ensemble Γ de fbf est satisfaisable ssi l'ensemble C_Γ des clauses de la forme clausale de Γ est satisfaisable.
3) $\Gamma \vdash B$ ssi $\Gamma \models B$ ssi $\Gamma \cup \{\neg B\}$ insatisfaisable.

Proposition IV.1 *Un ensemble Γ de clauses du calcul propositionnel est insatisfaisable ssi $\Gamma \underset{RSV}{\vdash} \square$.*

Exemple IV.2 On montre que

$$p_0 \vee p_1,\ p_0 \vee \neg p_1,\ \neg p_0 \vee p_1 \vdash p_0 \wedge p_1$$

en montrant que

$$p_0 \vee p_1,\ p_0 \vee \neg p_1,\ \neg p_0 \vee p_1,\ \neg p_0 \vee \neg p_1 \underset{RSV}{\vdash} \square$$

grâce à la déduction :

$f_1 : p_0 \vee p_1$	(hypothèse)
$f_2 : p_0 \vee \neg p_1$	(hypothèse)
$f_3 : p_0$	$res(f_1, f_2)$
$f_4 : \neg p_0 \vee p_1$	(hypothèse)
$f_5 : \neg p_0 \vee \neg p_1$	(hypothèse)
$f_6 : \neg p_0$	$res(f_4, f_5)$.
$f_7 : \square$	$res(f_3, f_6)$.

Chapitre 3

Calcul des prédicats du premier ordre

Dès qu'on veut manipuler des propriétés générales un peu compliquées et des relations entre objets, on est conduit à utiliser des énoncés dont la valeur de vérité dépend de variables, comme par exemple : "lundi il pleut et dimanche il ne pleut pas". De tels énoncés s'appellent des *prédicats* et leur théorie, qui généralise le calcul propositionnel, s'appelle le calcul des prédicats.

Ce calcul a été introduit par les mathématiciens pour répondre à leurs propres besoins, et la preuve de son immense pouvoir d'expression est qu'il leur permet effectivement de représenter les objets et notions dont ils se servent.

C'est à cause de son aptitude très générale pour la représentation et la manipulation des connaissances que le calcul des prédicats a intéressé les informaticiens.

I Syntaxe du calcul des prédicats du premier ordre

Le langage des prédicats du premier ordre est défini par la donnée de :

– l'alphabet Σ' composé de symboles de constantes $a, b, \ldots$, de symboles de variables $x, y, \ldots$, de symboles de parenthèses (et), d'un symbole de virgule ,, de symboles de fonctions $f, g, \ldots$, de symboles de relations ou *prédicats* $P, Q, \ldots$, de connecteurs $\neg, \vee, \wedge, \Longrightarrow, \Longleftrightarrow$, et de quantificateurs $\forall$ (*quantificateur universel*), $\exists$ (*quantificateur existentiel*) ;

– l'ensemble $\mathbb{F}'$ des formules bien formées étant le plus petit ensemble tel que :

- les termes sont définis par :
  - les constantes et les variables sont des termes,
  - si $t_1, \ldots, t_n$ sont des termes et f une fonction d'arité n (une fonction à n arguments), alors $f(t_1, \ldots, t_n)$ est un terme ;
- les formules atomiques sont définies par :
  si $t_1, \ldots, t_n$ sont des termes et P un prédicat d'arité n, alors $P(t_1, \ldots, t_n)$ est une formule atomique ;

- les formules sont définies par :
  - les formules atomiques sont des formules,
  - si A et B sont des formules, (A), $\neg A$, $A \vee B$, $A \wedge B$, $A \Longrightarrow B$ et $A \Longleftrightarrow B$ sont des formules,
  - si A est une formule et x une variable, alors $\forall x\, A$ et $\exists x\, A$ sont des formules.

Le langage des prédicats du premier ordre se démarque du langage propositionnel par l'utilisation de variables. Ces variables sont incluses dans des prédicats dont l'interprétation dépend des individus qu'elles représentent. Par exemple, le prédicat $Ilpleut(x)$ peut signifier "il pleut le jour x". Si l'on substitue une constante à la variable x, le prédicat devient, par exemple, $Ilpleut(lundi)$ et signifie alors "il pleut lundi".

En l'absence de parenthèses, les ordres de priorité sont : $\exists, \forall, \neg$ sont prioritaires devant $\vee, \wedge$ qui sont prioritaires devant $\Longrightarrow$, $\Longleftrightarrow$; à priorités égales, l'évaluation est effectuée de la gauche vers la droite. Il est cependant fortement conseillé d'utiliser des parenthèses pour délimiter la portée des quantificateurs et connecteurs.

I.A Sous-formules d'une formule

L'ensemble des *sous-formules* d'une formule est défini de la manière suivante :

- si A est une formule atomique, alors l'ensemble des sous-formules de A est $\{A\}$;
- si A est une formule, alors l'ensemble des sous-formules de $\neg A$ est $\{\neg A\}$ union l'ensemble des sous-formules de A ;
- si A est une formule, x une variable et si Q désigne l'un des quantificateurs $\forall$ ou $\exists$, alors l'ensemble des sous-formules de $Q\, x\, A$ est $\{Q\, x\, A\}$ union l'ensemble des sous-formules de A ;
- si A et B sont des formules et si *conn* désigne un des connecteurs binaires $\vee$, $\wedge$, $\Longrightarrow$ ou $\Longleftrightarrow$, alors l'ensemble des sous-formules de $A\, conn\, B$ est l'ensemble $\{A\, conn\, B\}$ union l'ensemble des sous-formules de A union l'ensemble des sous-formules de B.

Exemple I.1 Soit le langage L du calcul des prédicats défini par

$$L = \{a, b, c\} \cup \{f/2, g/3, h/1\} \cup \{P/3, Q/2\}$$

Le premier ensemble est composé des constantes qui, dans une définition plus générale, peuvent être considérées comme des fonctions d'arité 0. Le deuxième ensemble est composé des fonctions et de leurs arités. Cette arité est généralement sous-entendue dans l'écriture des formules. Le troisième ensemble est composé des prédicats et de leurs arités. La formule

$$(\forall x\, P(f(x, a), b, y)) \Longrightarrow (\forall u\, Q(z, b))$$

est une formule de L dont $(\forall x\, P(f(x, a), b, y))$ est une sous-formule.

I.B Variables libres ou liées

Les notions de variables libres ou variables liées jouent un rôle crucial en logique des prédicats. En effet, seules les variables libres sont substituables dans une formule. Cette distinction est étroitement liée à la notion de *portée d'un quantificateur* dans une formule, i.e. la sous-formule sur laquelle le quantificateur s'applique.

Définition I.1 *Une occurrence d'une variable x est dite liée dans une formule A si elle appartient à une sous-formule de A qui débute par $\forall x$ ou $\exists x$; sinon elle est dite libre.*

Exemple I.2 Dans la formule $\forall x\ P(x, y, f(x)) \Longrightarrow E(g(x, y), x)$, les deux premières occurrences de x sont liées, les deux dernières sont libres. Dans la formule $\forall x\,(P(x, y, f(x)) \Longrightarrow E(g(x, y), x))$, toutes les occurrences de x sont liées.

Définition I.2 *Une variable libre (resp. liée) d'une formule A est une variable qui a une occurrence libre (resp. liée) dans A. Une formule sans variable libre est dite close.*

I.C Standardisation des variables

Une formule est dite *propre* ou *rectifiée* lorsque l'ensemble de ses variables liées est disjoint de celui des variables libres, et que toutes les occurrences d'une variable liée appartiennent à une même sous-formule de liaison.

Pour transformer une formule non propre en une formule propre, il suffit de *standardiser* les variables en les renommant de la manière suivante :

- renommer les occurrences liées de toute variable libre,
- donner des noms différents à toutes les variables liées se trouvant dans des sous-formules de liaison différentes.

Exemple I.3 Soit la formule non propre

$$A = \forall x\,(\exists y\,P(x, y) \Longrightarrow \forall z\,Q(x, y, z) \wedge \forall y\,\exists x\,R(f(x), y)).$$

Elle se transforme en la formule propre

$$A' = \forall x\,(\exists u\,P(x, u) \Longrightarrow \forall z\,Q(x, y, z) \wedge \forall v\,\exists w\,R(f(w), v)).$$

Soit $\{x_1, \ldots, x_n\}$ l'ensemble des variables libres d'une formule propre A. La formule close $\forall x_1(\ldots(\forall x_n\,A)\ldots)$ est appelée *clôture universelle* de A.

I.D Substitution d'une variable par un terme

Soient A une formule dont x est une variable libre et t un terme. La *substitution* de t à x dans A, notée $(x \,|\, t)\,A$, est la formule obtenue en remplaçant chaque occurrence libre de x dans A par t, et ceci après avoir renommé des variables liées de telle manière que que x ne soit plus variable liée (si x l'était) et que plus aucune variable de t ne soit liée (s'il y en avait).

Exemple I.4 Soit $A = P(x) \vee \forall x\, \exists y\, Q(x,y)$ et $t = f(y,u)$. Pour obtenir $(x \mid t)\, A$, on renomme d'abord les occurrences liées de x et y, ce qui donne :

$$P(x) \vee \forall z_1\, \exists z_2\, Q(z_2, z_1),$$

puis on effectue la substitution, ce qui donne :

$$P\big(f(y,u)\big) \vee \forall z_1\, \exists z_2\, Q(z_1, z_2).$$

Soit x une variable libre d'une fbf A et t un terme. Le terme t est *substituable* à x si aucune occurrence libre de x, lorsque remplacée par t, ne devient liée en ce sens qu'il y apparait des occurrences liées de variables de t.

Exemple I.5 Le terme $f(y,u)$ est substituable à x dans $P(x) \vee \forall x\, \exists y\, Q(x,y)$ mais pas dans $\forall y\, P(x,y) \vee \forall x\, \exists y\, Q(x,y)$

II Système formel du calcul des prédicats du premier ordre (sfcppo)

– L'alphabet et l'ensemble des formules bien formées sont, respectivement, les ensembles Σ' et $\mathbb{F}'$ définis précédemment.

– L'ensemble des axiomes est l'ensemble des formules de $\mathbb{F}'$ de l'une des formes suivantes :

SA_1 : $A \Longrightarrow (B \Longrightarrow A)$

SA_2 : $(A \Longrightarrow (B \Longrightarrow C)) \Longrightarrow ((A \Longrightarrow B) \Longrightarrow (A \Longrightarrow C))$

SA_3 : $(\neg A \Longrightarrow \neg B) \Longrightarrow (B \Longrightarrow A)$

SA_4 : $\forall x\, A(x) \Longrightarrow A(t)$

SA_5 : $(D \Longrightarrow B) \Longrightarrow (D \Longrightarrow \forall x\, B)$

où A, B et C sont des formules quelconques de $\mathbb{F}'$, x une variable, t un terme et D une formule n'ayant pas x comme variable libre.

– L'ensemble des règles de déduction est la paire $\Omega' = \{mp, g\}$ où

$$A, A \Longrightarrow B \underset{mp}{\vdash} B \qquad (\textit{modus ponens})$$

$$A \underset{g}{\vdash} \forall x\, A \qquad (\textit{généralisation})$$

pour toutes formules A, B de $\mathbb{F}'$ et pour toute variable x.

Proposition II.1 *Pour toute formule A du calcul des prédicats du premier ordre, la formule $(A \Longrightarrow A)$ est un théorème du sfcppo.*

Proposition II.2 ***(Théorème de déduction.)*** *Soient $A_1, \ldots, A_{n-1}, A_n$ des formules closes et B une formule quelconque, du calcul des prédicats du premier ordre. Alors $A_1, \ldots, A_{n-1} \vdash (A_n \Longrightarrow B)$ ssi $A_1, \ldots, A_{n-1}, A_n \vdash B$.*

III Sémantique du calcul des prédicats du premier ordre

On appelle *interprétation* I du calcul des prédicats du premier ordre la donnée de :

– un *domaine d'interprétation* (ou *base de l'interprétation*) D : un ensemble non vide de valeurs que peuvent prendre les variables,

– une application qui

- à toute constante c associe un élément c_I de D,
- à toute fonction f d'arité n associe une application $f_I : D^n \Longrightarrow D$,
- à tout prédicat P d'arité n associe une application $P_I : D^n \Longrightarrow \{V, F\}$.

Pour une interprétation I donnée de base D, on appelle *valuation* (ou *assignation*), toute application de l'ensemble des variables dans D.

Définition III.1 *(**Sémantique des termes**). Soit I une interprétation de base D, v une valuation relative à I, et t un terme. Alors l'interprétation $v_I(t)$ de t est l'élément de D défini de la manière suivante :*

- *si t est une constante c, alors $v_I(t) = c_I$*
- *si t est une variable x, alors $v_I(t) = v(x)$*
- *si t est de la forme $f(t_1, \ldots, t_n)$, alors $v_I(t) = f_I(v_I(t_1), \ldots, v_I(t_n))$.*

Définition III.2 *(**Sémantique des formules bien formées**). Soit I une interprétation, v une valuation et φ une formule bien formée. Le fait "φ est vrai dans I relativement à v", noté $\models_I^v \varphi$, est défini comme suit :*

$\models_I^v P(t_1, \ldots, t_n)$ *ssi* $P_I(v_I(t_1), \ldots, v_I(t_n)) = V$

$\models_I^v (\neg A)$ *ssi* $\not\models_I^v A$

$\models_I^v (A \vee B)$ *ssi* $\models_I^v A$ *ou* $\models_I^v B$

$\models_I^v (A \wedge B)$ *ssi* $\models_I^v A$ *et* $\models_I^v B$

$\models_I^v (A \Longrightarrow B)$ *ssi* $\not\models_I^v A$ *ou* $\models_I^v B$

$\models_I^v (A \iff B)$ *ssi* $\models_I^v (A \Longrightarrow B)$ *et* $\models_I^v (B \Longrightarrow A)$

$\models_I^v (\forall x\, A)$ *ssi* $\models_I^w A$ *pour toute valuation w égale à v sauf éventuellement en x*

$\models_I^v (\exists x\, A)$ *ssi* $\models_I^w A$ *pour au moins une valuation w égale à v sauf éventuellement en x*

Remarque III.1 Il ressort clairement de ces définitions que pour une interprétation donnée, la valeur de vérité d'une formule ne dépend que de la valuation de ses variables libres. Ainsi, pour une interprétation donnée, la valeur de vérité d'une formule close ne dépend que de cette interprétation.

Exemple III.1 Soit le langage $L = \{p/2\} \cup \{\mathrm{Div}/2, E/2\}$ et la formule

$$\forall x\,(\exists y\,\mathrm{Div}(x,y) \Longrightarrow \exists z\,E(x,p(z,u))).$$

Soit I une interprétation de base $D = \mathbb{N}$, $p_I(x,y) = xy$, $\mathrm{Div}_I(x,y) = V$ ssi $y \neq x$ et y divise x, et $E_I(x,y) = V$ ssi $x = y$.
La seule variable libre de la formule est u. Pour une valuation assignant la valeur 2 à u, la formule exprime que tout entier naturel divisible par un autre entier naturel est pair ; ce qui est faux.

III.A Définitions sur l'interprétation sémantique des formules

Comme en calcul propositionnel, les formules se caractérisent selon leur valeur de vérité. Celle-ci dépend, d'une part de l'interprétation, et, d'autre part, pour une interprétation donnée, de la valuation des variables (libres).

– On dit qu'une formule B est *conséquence sémantique* d'un ensemble de formules Γ si pour toute interprétation I et toute valuation $v(I)$ telles que pour tout $A \in \Gamma$: $\vDash_I^v A$, on a aussi $\vDash_I^v B$. On écrit alors $\Gamma \models B$.

– On dit que deux formules A et B sont *sémantiquement équivalentes* si $A \models B$ et $B \models A$. On écrit alors $A \equiv B$.

– On dit qu'un ensemble de formules Γ est *satisfaisable* (ou *consistant* ou *non-contradictoire*) s'il existe une interprétation I et une valuation $v = v(I)$ telles que pour toute formule $A \in \Gamma$ on ait $\vDash_I^v A$. Un tel couple (I, v) est appelé *modèle* de Γ. Dans le cas contraire, on dit que Γ est *insatisfaisable* (ou *inconsistant* ou *contradictoire*).

– On dit qu'une formule A est (universellement) *valide* (ou une *tautologie*) si pour toute interprétation I et pour toute valuation $v(I)$, on a $\vDash_I^v A$. On écrit alors $\models A$.

Proposition III.1 ***(Théorème de correction)***. *Soit A une formule bien formée du calcul des prédicats du premier ordre. Si $\vdash A$, alors $\models A$.*

Proposition III.2 ***(Théorème de complétude)***. *Soit A une formule bien formée du calcul des prédicats du premier ordre. Si $\models A$, alors $\vdash A$.*

Proposition III.3 ***(Théorème de complétude généralisé)***. *Si $\Gamma \cup \{B\} \subseteq \mathbb{F}'$ alors $\Gamma \models B$ si et seulement si $\Gamma \vdash B$.*

Remarque III.2 (i) Soit A une formule bien formée et $\overline{A}$ sa clôture universelle. Alors $\models A$ ssi $\models \overline{A}$.
(ii) Contrairement au calcul propositionnel, le nombre d'interprétations différentes pour une formule donnée n'est pas fini. Pour savoir si une formule est une tautologie, il ne peut donc être question d'énumérer toutes les interprétations.
(iii) Le système formel du calcul des prédicats du premier ordre est indécidable. En revanche, le théorème de complétude nous assure que si une formule est une tautologie alors elle admet une déduction dans le système formel.

Proposition III.4 *Soit $\Gamma \cup \{B\}$ un ensemble de formules du calcul des prédicats du premier ordre. Alors $\Gamma \models B$ ssi $\Gamma \cup \{\neg B\}$ est insatisfaisable.*

Remarque III.3 Il est généralement "plus simple" de montrer que $\Gamma \not\models B$ car il suffit, pour cela, d'exhiber un modèle de Γ qui ne soit pas modèle de B.

IV Préparation des formules Modèles de Herbrand

La mise sous forme prénexe et la mise sous forme de Skolem permettent de se débarrasser des quantificateurs d'un ensemble fini de formules du calcul des prédicats du premier ordre.

Le théorème de Herbrand montre que lorsqu'on a un ensemble fini de formules (mises sous bonne forme) l'existence d'un modèle est équivalente à l'existence d'un "modèle syntaxique", i.e. fondé uniquement sur les termes engendrés par les formules.

IV.A Forme prénexe

Soit A une formule du calcul des prédicats du premier ordre. On dit que A est sous forme *prénexe* si A est de la forme :

$$A = Q_1\, x_1 \ldots Q_n\, x_n\, B$$

où chaque Q_i est soit $\forall$, soit $\exists$ et où B ne contient aucun quantificateur.

Proposition IV.1 *Pour toute formule A du calcul des prédicats du premier ordre, il existe une formule A' sous forme prénexe qui est sémantiquement équivalente à A.*

On sait que si dans une formule F comportant une sous-formule B, on remplace B par une formule (sémantiquement) équivalente B', on obtient une formule F' (sémantiquement) équivalente à F. Voici une méthode permettant, à partir d'une formule quelconque, d'obtenir une formule équivalente sous forme prénexe. Le fait que cette méthode se termine et donne effectivement une formule sous forme prénexe équivalente à celle de départ résulte de considérations élémentaires.

(a) Se débarrasser de $\Longrightarrow$ et $\Longleftrightarrow$ en utilisant les équivalences suivantes de gauche à droite :

a1 $(A \Longrightarrow B) \equiv (\neg A \vee B)$

a2 $(A \Longleftrightarrow B) \equiv ((A \wedge B) \vee (\neg A \wedge \neg B))$

(b) Changer le nom de certaines variables liées de manière à n'avoir plus de variable quantifiée deux fois, en utilisant les équivalences suivantes :

b1 $\forall x\, A(x) \equiv \forall y\, A(y)$

b2 $\exists x\, A(x) \equiv \exists y\, A(y)$.

(c) Faire remonter tous les quantificateurs en tête en utilisant les équivalences suivantes de gauche à droite (x n'étant pas variable libre de C) :

c1 $\neg\exists x\, A(x) \equiv \forall x\, \neg A(x)$

c2 $\neg\forall x\, A(x) \equiv \exists x\, \neg A(x)$

c3 $(C \vee \forall x\, A(x)) \equiv \forall x\, (C \vee A(x))$

c4 $(C \vee \exists x\, A(x)) \equiv \exists x\, (C \vee A(x))$

c5 $(C \wedge \forall x\, A(x)) \equiv \forall x\, (C \wedge A(x))$

c6 $(C \wedge \exists x\, A(x)) \equiv \exists x\, (C \wedge A(x))$

Exemple IV.1 $(\forall x\, A(x) \Longrightarrow (\exists y\, B(y) \vee \exists y\, C(y)))$

$\equiv (\neg \forall x\, A(x) \vee (\exists y\, B(y) \vee \exists y\, C(y)))$ (a1)

$\equiv (\neg \forall x\, A(x) \vee (\exists y\, B(y) \vee \exists z\, C(z)))$ (b2)

$\equiv (\exists x\, \neg A(x) \vee (\exists y\, B(y) \vee \exists z\, C(z)))$ (c2)

$\equiv \exists x\, (\neg A(x) \vee (\exists y\, B(y) \vee \exists z\, C(z)))$ (c4 et commutativité de $\vee$)

$\equiv \exists x\, (\neg A(x) \vee \exists y\, (B(y) \vee \exists z\, C(z)))$ (c4 et commutativité de $\vee$)

$\equiv \exists x\, \exists y\, (\neg A(x) \vee (B(y) \vee \exists z\, C(z)))$ (c4)

$\equiv \exists x\, \exists y\, (\neg A(x) \vee \exists z\, (B(y) \vee C(z)))$ (c4)

$\equiv \exists x\, \exists y\, \exists z\, (\neg A(x) \vee (B(y) \vee C(z)))$ (c4)

Remarque IV.1 Pour limiter le nombre de quantificateurs de la formule finale, il peut être intéressant de ne pas mener complètement l'étape (b) et, à l'étape (c), d'utiliser les deux équivalences supplémentaires suivantes :

c7 $(\forall x\, A(x) \wedge \forall x\, B(x)) \equiv \forall x\, (A(x) \wedge B(x))$

c8 $(\exists x\, A(x) \vee \exists x\, B(x)) \equiv \exists x\, (A(x) \vee B(x))$

Exemple IV.2 En reprenant l'exemple précédent :

$(\forall x\, A(x) \Longrightarrow (\exists y\, B(y) \vee \exists y\, C(y)))$

$\equiv (\neg \forall x\, A(x) \vee (\exists y\, B(y) \vee \exists y\, C(y)))$ (a1)

$\equiv (\neg \forall x\, A(x) \vee \exists y\, (B(y) \vee C(y)))$ (c8)

$\equiv (\exists x\, \neg A(x) \vee \exists y\, (B(y) \vee C(y)))$ (c2)

$\equiv (\exists x\, \neg A(x) \vee \exists x\, (B(x) \vee C(x)))$ (b2)

$\equiv \exists x\, (\neg A(x) \vee (B(x) \vee C(x)))$ (c8)

IV.B Forme de Skolem

Soit $Q_1\, x_1 \ldots Q_n\, x_n\, B$ une formule A mise sous forme prénexe. On appelle *forme de Skolem* de A, et on note A^S, la formule obtenue en enlevant tous les $\exists x_i$, en remplaçant chacune des variables x_i quantifiées avec un $\exists$ par $f_i(x_{j_1}, \ldots x_{j_l})$ où $x_{j_1}, \ldots x_{j_l}$ sont les variables quantifiées par des $\forall$ placés avant le $\exists x_i$. Lorsqu'il n'y a aucun quantificateur $\forall$ avant le $\exists x_i$, le symbole que l'on introduit est une constante (une constante est un symbole fonctionnel d'arité nulle). On suppose, bien sûr, que les symboles fonctionnels f_i introduits sont différents de tous ceux utilisés par ailleurs.

Exemple IV.3 *La forme de Skolem de*

$$\exists x_1\, \forall x_2\, \exists x_3\, \forall x_4\, \exists x_5\, P(x_1, x_2, x_3, x_4, x_5)$$

est

$$\forall x_2\, \forall x_4\, P(a, x_2, f_1(x_2), x_4, f_2(x_2, x_4)).$$

Proposition IV.2 ***(Théorème de Skolem)***. *Soit $\{A_1, \ldots, A_n\}$ un ensemble fini de formules du calcul des prédicats du premier ordre. Soit $\{A_1^S, \ldots, A_n^S\}$ l'ensemble des formes de Skolem de ces formules. Alors $\{A_1, \ldots, A_n\}$ admet un modèle de base (domaine) D ssi $\{A_1^S, \ldots, A_n^S\}$ admet un modèle de base D.*

IV.C Forme clausale

– Un *littéral* est une formule atomique (auquel cas il est dit *positif*) ou la négation d'une formule atomique (auquel cas il est dit *négatif*).

– Une *paire opposée* de littéraux est composée d'un littéral positif $P(t_1, \ldots, t_n)$ de prédicat P et d'un littéral négatif $\neg P(t'_1, \ldots, t'_n)$ de même prédicat.

– Une *clause* est une disjonction finie de littéraux.

– Une *clause de Horn* est une clause ayant au plus un littéral positif telle que :

$$\neg P_1 \vee \cdots \vee \neg P_n \vee Q.$$

Cette clause est logiquement équivalente à

$$(P_1 \wedge \cdots \wedge P_n) \Longrightarrow Q.$$

On peut distinguer quatre sous-catégories de clauses de Horn :

- *clause implicative* $(P_1 \wedge \cdots \wedge P_n) \Longrightarrow Q$ (avec un littéral positif et avec au moins un littéral négatif)
- *clause but* $(P_1 \wedge \cdots \wedge P_n) \Longrightarrow$ (sans littéral positif et avec au moins un littéral négatif)
- *clause assertion* (ou *faits*) $\Longrightarrow Q$ (avec un littéral positif et sans littéral négatif)
- *clause vide* $\Longrightarrow$ (sans littéraux).

IV.D Problème de la démonstration automatique

Le problème général de la démonstration automatique peut se formuler par la question : "est-il vrai qu'une formule B est conséquence logique d'un ensemble de formules Γ" ? ou encore, "est-il vrai que l'ensemble $\Gamma \cup \{\neg B\}$ n'a pas de modèle" ?

On est donc ramené au problème de savoir si un certain ensemble de formules Γ_0 possède un modèle ou non. Pour le résoudre, on effectue successivement les traitements suivants :

- mise sous forme prénexe des formules de Γ_0, ce qui donne Γ_1 ;
- mise sous forme de Skolem des formules de Γ_1, ce qui donne Γ_2 ;
- suppression des quantificateurs universels des formules de Γ_2, ce qui donne Γ_3 ;
- mise sous forme clausale des formules de Γ_3, ce qui donne Γ_4.

Exemple IV.4 Soit le problème de savoir si la formule

$$B = \exists x \, \exists y \, Q(x, y)$$

est conséquence de l'ensemble des formules

$$\Gamma = \{\forall x \, (P(x) \Longrightarrow \exists y \, (R(y) \wedge Q(x, y))), \exists x \, P(x)\}.$$

L'ensemble Γ_0 est alors :
$\Gamma_0 = \{\forall x\,(P(x) \Longrightarrow \exists y\,(R(y) \wedge Q(x,y))), \exists x\,P(x), \neg\exists x\,\exists y\,Q(x,y)\}$.
On obtient successivement les ensembles de formules :
$\Gamma_1 = \{\forall x\,\exists y\,(\neg P(x) \vee (R(y) \wedge Q(x,y))), \exists x\,P(x), \forall x\,\forall y\,\neg Q(x,y)\}$,
$\Gamma_2 = \{\forall x\,(\neg P(x) \vee (R(f(x)) \wedge Q(x,f(x)))), P(a), \forall x\,\forall y\,\neg Q(x,y)\}$,
$\Gamma_3 = \{\neg P(x) \vee (R(f(x)) \wedge Q(x,f(x))), P(a), \neg Q(x,y)\}$,
$\Gamma_4 = \{\neg P(x) \vee Q(x,f(x)), \neg P(x) \vee R(f(x)), P(a), \neg Q(x,y)\}$

Si Γ_4 n'a pas de modèle, alors la formule B est conséquence logique de l'ensemble de formules Γ ; autrement, la formule B n'est pas conséquence logique de l'ensemble de formules Γ.

IV.E Modèles de Herbrand

A priori, pour savoir si un ensemble de formules du calcul des prédicats du premier ordre admet un modèle il faut réaliser une infinité d'essais : essayer s'il y a un modèle ayant une base à un élément, essayer s'il y a un modèle ayant une base à deux éléments, etc.$\cdots$, chacun de ces essais se divisant lui-même en un très grand nombre d'essais.

L'intérêt du théorème de Herbrand est qu'il permet de se ramener à un seul essai : pour savoir si un ensemble fini Γ de formules possède des modèles, il suffit de savoir si Γ possède un "modèle syntaxique", i.e. construit de manière standard à partir du vocabulaire utilisé dans les formules de Γ. De plus, savoir si ce modèle existe se ramène à l'étude d'un ensemble de formules du calcul propositionnel.

Soit $\{A_1, \ldots, A_n\}$ un ensemble de formules du calcul des prédicats du premier ordre dont le vocabulaire contient au moins un symbole de constante.

– On appelle *univers de Herbrand* associé à $\{A_1, \ldots, A_n\}$ l'ensemble, noté $U_H(\{A_1, \ldots, A_n\})$, de tous les termes sans variables construits à partir du vocabulaire des formules $A_1, \ldots, A_n$.

Exemple IV.5 Reprenons l'exemple précédent :
$A_1 = \neg P(x) \vee Q(x, f(x))$
$A_2 = \neg P(x) \vee R(f(x))$
$A_3 = P(a)$
$A_4 = \neg Q(x,y)$
Alors $U_H(\{A_1, \ldots, A_4\}) = \{a, f(a), f(f(a)), \ldots, f(\ldots(f(a))\ldots), \ldots\}$

Remarque IV.2 Dès qu'il y a un symbole fonctionnel, l'univers de Herbrand est infini.

– On appelle *base de Herbrand* associée à $\{A_1, \ldots, A_n\}$ l'ensemble, noté $B_H(\{A_1, \ldots, A_n\})$, de tous les atomes sans variables construits à partir du vocabulaire des formules $A_1, \ldots, A_n$, i.e. de toutes les formules de la forme $P(t_1, \ldots, t_r)$ où P est un des symboles de prédicat de l'une des formules $A_1, \ldots, A_n$ et où $t_1, \ldots, t_r$ sont des éléments de $U_H\{A_1, \ldots, A_n\}$.

Exemple IV.6 En reprenant l'exemple précédent,
$B_H(\{A_1, \ldots, A_4\}) = \{P(a), R(a), Q(a,a), P(f(a)), R(f(a)), Q(a, f(a)), Q(f(a), a)$,
$Q(f(a), f(a)), P(f(f(a))), \ldots\}$

– On appelle *système de Herbrand* associé à $\{A_1, \ldots, A_n\}$ l'ensemble, noté $S_H(\{A_1, \ldots, A_n\})$, de toutes les formules obtenues à partir des formules A_i en remplaçant dans les A_i les variables par des éléments de l'univers de Herbrand.

Exemple IV.7 En reprenant l'exemple précédent, $S_H(\{A_1, \ldots, A_4\}) = \{\neg P(a) \vee Q(a, f(a)), \neg P(a) \vee R(f(a)), P(a), \neg Q(a, a), \neg P(f(a)) \vee Q(f(a), f(f(a))), \ldots\}$.

Proposition IV.3 *(**Théorème de Herbrand**). Soit $\{A_1, \ldots, A_n\}$ un ensemble fini de clauses. Les trois assertions suivantes sont équivalentes.*

(a) $\{A_1, \ldots, A_n\}$ possède un modèle.

(b) $S_H(\{A_1, \ldots, A_n\})$ considéré comme un ensemble de formules du calcul propositionnel dont les atomes sont les éléments de $B_H(\{A_1, \ldots, A_n\})$ possède un modèle.

(c) $\{A_1, \ldots, A_n\}$ possède un modèle dont la base est $U_H(\{A_1, \ldots, A_n\})$.

Corollaire IV.1 *Soit $\{A_1, \ldots, A_n\}$ un ensemble fini de clauses et soit $S_H(\{A_1, \ldots, A_n\}) = \{F_0, \ldots, F_m, \ldots\}$. L'ensemble $\{A_1, \ldots, A_n\}$ est inconsistant ssi une des formules $F_0 \wedge \cdots \wedge F_m$ est insatisfaisable.*

IV.F Illustration du théorème de Herbrand

En reprenant l'exemple précédent, on a :

$F_0 = \neg P(a) \vee Q(a, f(a))$ $\quad$ $F_1 = \neg P(a) \vee R(f(a))$
$F_2 = P(a)$ $\quad$ $F_3 = \neg Q(a, a)$
$F_4 = \neg P(f(a)) \vee Q(f(a), f(f(a)))$ $\quad$ $F_5 = \neg P(f(a)) \vee R(f(f(a)))$
$F_6 = P(f(a))$ $\quad$ $F_7 = \neg Q(a, f(a))$
. . . . . .

Pour savoir si l'ensemble des formules $\{A_1, \ldots, A_4\}$ que nous avions au départ possède un modèle, nous étudions successivement $F_0, F_0 \wedge F_1, \ldots, F_0 \wedge \cdots \wedge F_m, \ldots$, en cherchant à savoir, à chaque étape, si la formule est satisfaisable ou non. Jusqu'à $F_0 \wedge \cdots \wedge F_5$, on constate que oui (par exemple grce à l'interprétation $I(P(a)) = V$, $I(Q(a, f(a))) = V$, $I(R(f(a))) = V$, $I(Q(a, a)) = F$, $I(P(f(a))) = F$).

Une fois arrivé à $F_0 \wedge \cdots \wedge F_6$, on constate qu'il est impossible de construire un modèle car F_0, F_2 et F_6 ne peuvent être satisfaites simultanément. On en conclut que $\{A_1, \ldots, A_4\}$ n'a pas de modèle, et donc que la formule $B = \exists x \, \exists y \, Q(x, y)$ se déduit de l'ensemble des formules

$$\Gamma = \{\forall x \, (P(x) \Longrightarrow \exists y \, (R(y) \wedge Q(x, y))), \exists x \, P(x)\}.$$

Pour vérifier si un ensemble fini de formules du calcul propositionnel est satisfaisable la méthode la plus simple est de calculer la table de vérité de la conjonction des formules et de regarder si il y a au moins un V dans la colonne principale. Il existe bien d'autres méthodes.

IV.G Arbres sémantiques

Lorsqu'on travaille à la main, on utilise souvent la méthode des *arbres sémantiques* que nous allons décrire à l'aide d'un exemple.

Imaginons qu'on cherche à savoir si la formule

$$B = (\exists x\, \neg Q(x) \Longrightarrow \forall y\, P(y))$$

est conséquence des deux formules

$$B_1 = (\exists x\, P(x) \Longrightarrow \forall y\, P(y))$$

et

$$B_2 = \forall x\, (P(x) \vee Q(x)).$$

La mise sous forme clausale de $\{B_1, B_2, \neg B\}$ donne :
$A_1 = \neg P(x) \vee P(y)$ $\quad A_2 = P(x) \vee Q(x)$
$A_3 = \neg Q(a)$ $\quad A_4 = \neg P(b)$
L'univers de Herbrand est $U_H = \{a, b\}$.
Les atomes de Herbrand sont $B_H = \{Q(a), P(a), Q(b), P(b)\}$.
Le système de Herbrand est $S_H = \{\neg P(a) \vee P(a), \neg P(a) \vee P(b), \neg P(b) \vee P(b), \neg P(b) \vee P(a), P(a) \vee Q(a), P(b) \vee Q(b), \neg Q(a), \neg P(b)\} = \{F_1, \ldots, F_8\}$.

On dessine alors un arbre binaire, chaque niveau correspondant à un atome de Herbrand et chaque branche entre la racine et un nœud correspondant à une interprétation de tous les atomes de Herbrand des niveaux par où passe cette branche.

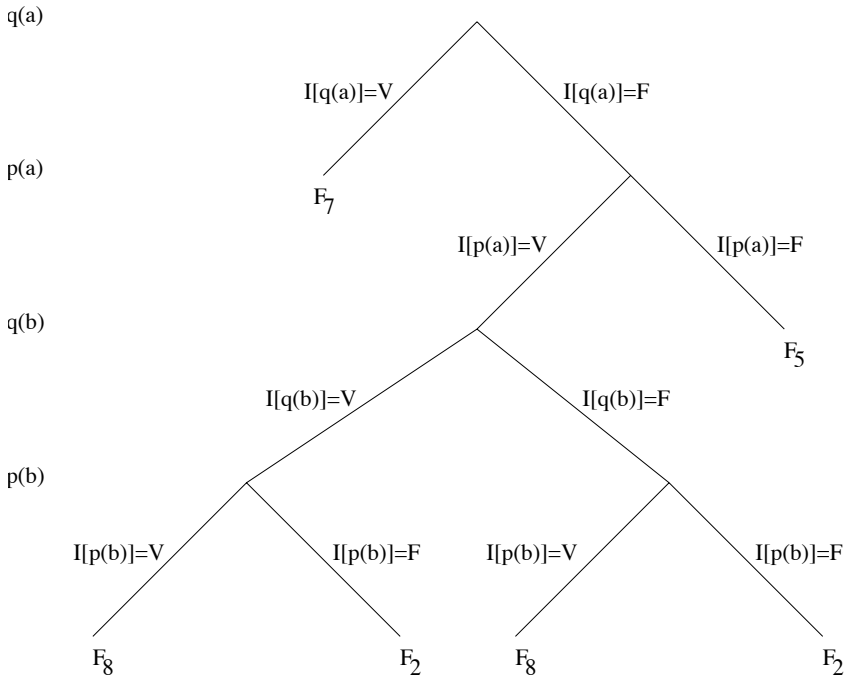
Dans cet arbre, on interrompt le développement d'une branche dès que l'interprétation correspondant à cette branche contredit une formule de S_H. On marque alors le nœud avec la formule qui a donné la contradiction.

Trois cas peuvent se produire :

- toutes les branches sont interrompues et marquées, ce qui signifie qu'il n'existe aucun modèle de S_H ;

- l'arbre est fini et l'une des branches n'a pas été marquée, ce qui signifie que l'interprétation correspondant à cette branche est un modèle de S_H ;

- l'arbre est infini alors il existe une branche infinie (lemme de Koenig) à laquelle correspond un modèle de S_H.



Dans cet exemple, l'arbre est fini et entièrement marqué, ce qui donne comme conclusion que la formule B est conséquence des formules B_1 et B_2.

V Unification

L'unification est la mise en coïncidence des atomes par un bon choix des termes substitués aux variables. Il s'agit là d'une idée qui est assez ancienne en logique mathématique. Son utilisation comme élément fondamental d'un algorithme de démonstration automatique est due à Robinson (1965).

V.A Substitutions

Les formules considérées dans ce sous-paragraphe ne sont pas quantifiées.

– On appelle *composant de substitution* toute expression de la forme $(x \mid t)$ où x est une variable et t un terme quelconque du calcul des prédicats. Si A est une formule du calcul des prédicats, on note $(x \mid t)\, A$ la formule obtenue en remplaçant dans A toutes les occurrences de x par t.

Remarque V.1 Les formules envisagées dans cette section n'étant pas quantifiées, le renommage des variables n'est pas nécessaire.

Une *substitution* est une application σ de l'ensemble des fbf du calcul des prédicats du premier ordre dans lui-même, de la forme :

$$\sigma : A \mapsto c_1 \dots c_k\, A$$

où $c_1, \dots, c_k$ sont des composants de substitution. La suite finie $c_1, \dots, c_k$ est appelée *décomposition* de la substitution σ, ce que l'on note $\sigma = [c_1 \dots c_k]$. La *substitution identique* sera notée $[\,]$ ou ε.

Exemple V.1 Soit $\sigma = [(x \mid f(a))\,(y \mid f(x))]$ une substitution. Alors

$$\begin{array}{rcl} \sigma\, P(x,y) & = & (x \mid f(a))\, P(x, f(x)) \\ & = & P(f(a), f(f(a))) \end{array}$$

Remarque V.2 La décomposition en composants de substitution n'est en général ni commutative (se vérifie à partir de l'exemple ci-dessus), ni unique ($[(x \mid y)\,(z \mid y)] = [(z \mid y)\,(x \mid y)]$).

– Soit $\Gamma = \{A_1, \ldots, A_n\}$ un ensemble fini de formules atomiques du calcul des prédicats du premier ordre. On appelle *unificateur* de Γ toute substitution σ telle que

$$\sigma\, A_1 = \cdots = \sigma\, A_n.$$

Exemple V.2 Soit $\Gamma = \{A_1, A_2, A_3\}$ avec $A_1 = P(x, z)$, $A_2 = P(f(y), g(a))$ et $A_3 = P(f(u), z)$. Alors $\sigma = [(x \mid f(u))\,(y \mid u)\,(z \mid g(a))]$ est un unificateur de Γ.

Remarque V.3 1) Si σ est un unificateur d'un ensemble fini de formules Γ, alors pour toute substitution α : $\alpha\,\sigma$ est un unificateur de Γ.
2) Il est possible qu'un ensemble de formules n'admette aucun unificateur ($\Gamma = \{P(x, f(x)), P(f(y), y)\}$).

Soit σ un unificateur d'un ensemble de formules Γ. On dit que σ est un *plus grand unificateur* (ou un *unificateur le plus général*) de Γ si pour tout unificateur α de Γ il existe une substitution β telle que $\sigma = \beta\,\alpha$.

Exemple V.3 En reprenant l'exemple précédent, σ est un plus grand unificateur de Γ. La substitution $\rho = [(x \mid f(v))\,(y \mid v)\,(z \mid g(a))\,(u \mid v)]$ en est un autre et on a $\sigma = (v \mid u)\,\rho$ et $\rho = (u \mid v)\,\sigma$.

V.B Algorithme d'unification de deux atomes A et B

- $\theta := \epsilon$;
- tant que $\theta\, A \neq \theta\, B$, faire
  - déterminer le symbole le plus à gauche de $\theta\, A$ qui soit différent du symbole de même rang de $\theta\, B$;
  - déterminer t_1 et t_2 les sous-termes respectifs de $\theta\, A$ et $\theta\, B$ qui commencent à ce symbole ;
  - si "aucun n'est une variable" ou "l'un est une variable contenue dans l'autre", alors imprimer "A et B ne sont pas unifiables" ; aller à *fin.*
  - sinon faire
    - déterminer x une variable parmi t_1 et t_2 ;
    - déterminer t celui de t_1 et t_2 qui n'est pas x ;
    - $\theta = (x \mid t)\,\theta$;
  - fin-de-si ;
- fin-de-tant-que ;
- imprimer "θ est un plus grand unificateur de A et B" ;

- fin.

Exemple V.4

$\theta\, A$	$\theta\, B$	θ
$\underset{\uparrow}{P}(x, f(x), a)$	$\underset{\uparrow}{P}(u, w, w)$	ϵ
$P(\underset{\uparrow}{x}, f(x), a)$	$P(\underset{\uparrow}{u}, w, w)$	$[(x \mid u)]$
$P(u, f(\underset{\uparrow}{u}), a)$	$P(u, \underset{\uparrow}{w}, w)$	$[(w \mid f(u))\,(x \mid u)]$
$P(u, f(u), \underset{\uparrow}{a})$	$P(u, f(u), \underset{\uparrow}{f(u)})$	échec car a et $f(u)$ non variables

Exemple V.5

$\theta\, A$	$\theta\, B$	θ
$\underset{\uparrow}{P}(x, f(g(x)), a)$	$\underset{\uparrow}{P}(b, y, z)$	ϵ
$P(\underset{\uparrow}{x}, f(g(x)), a)$	$P(\underset{\uparrow}{b}, y, z)$	$[(x \mid b)]$
$P(b, f(\underset{\uparrow}{g(b)}), a)$	$P(b, \underset{\uparrow}{y}, z)$	$[(y \mid f(g(b)))\,(x \mid b)]$
$P(b, f(g(b)), \underset{\uparrow}{a})$	$P(b, f(g(b)), \underset{\uparrow}{z})$	$[(z \mid a)\,(y \mid f(g(b)))\,(x \mid b)]$

Succès, A et B sont unifiables et un plus grand unificateur est :

$$[(z \mid a)\,(y \mid f(g(b)))\,(x \mid b)].$$

VI Résolution

La méthode de résolution étendue au calcul des prédicats est plus complexe que sa version en calcul propositionnel puisqu'elle doit prendre en compte l'existence de variables.

Pour pouvoir appliquer la méthode de résolution, les formules sont mises sous forme clausale. La résolution consiste, comme dans le cas du calcul propositionnel, à apparier des clauses satisfaisant à certaines conditions pour produire de nouvelles clauses jusqu'à obtenir la clause vide.

Soient C_1 et C_2 deux clauses non tautologiques. Elles forment une paire *résoluble* si elles contiennent une paire opposée de littéraux $P(t_1, \ldots, t_n)$ et $\neg P(t'_1, \ldots, t'_n)$ telle qu'il existe un unificateur σ qui égalise les termes correspondants des deux littéraux. Dans ce cas, on appelle *résolvante* de C_1 et C_2, la clause, notée $res(C_1, C_2)$, obtenue en prenant la réunion des littéraux de C_1 et C_2, en effectuant σ et en supprimant la paire opposée.

Exemple VI.1 Soient les deux clauses $C_1 = P(x) \vee Q(g(x))$ et $C_2 = \neg P(f(y))$. La paire $\{C_1, C_2\}$ est résoluble en prenant la substitution $\sigma = [(x \mid f(y))]$.

VI.A Système formel de résolution avec variables

Le système formel de résolution avec variables RAV est défini par:

– Σ_{RAV} est l'alphabet Σ' du calcul des prédicats du premier ordre privé de $\wedge$, $\Longrightarrow$, $\Longleftrightarrow$ et des quantificateurs ;

– $\mathbb{F}_{RAV}$ est l'ensemble de toutes les clauses du calcul des prédicats du premier ordre ; on ne considère que les clauses sans répétition, i.e. ne contenant pas deux fois le même littéral ;

– $\mathbb{A}_{RAV} = \emptyset$;
– $\Omega_{RAV} = \{res, fac\}$ où

$$f, g \underset{res}{\vdash} h$$

ssi

f est de la forme $l \vee f_1$

g est de la forme $\neg l' \vee g_1$

h est de la forme $\sigma(\theta f_1 \vee g_1)$

où l et l' sont deux atomes de mme symbole de prédicat, θ est une substitution telle que θf et θg n'aient aucune variable commune (θ est appelé substitution de renommage) et où σ est un plus grand unificateur des atomes θl et l', f et g étant deux clauses non tautologiques ; de telles clauses f et g, rappelons le,constituent une paire résoluble et la clause h est leur résolvante ;

$$f \underset{fac}{\vdash} h$$

ssi

f est de la forme $l \vee l' \vee f_1$

h est de la forme $\sigma l \vee \sigma f_1$

où l et l' sont deux atomes de mme symbole de prédicat et où σ est un plus grand unificateur de l et l'.

Exemple VI.2 1) On a

$$P(x, c) \vee R(x), \neg P(c, c) \vee Q(x) \underset{res}{\vdash} R(c) \vee Q(x)$$

avec $l = P(x, c)$, $l' = P(c, c)$, $\theta = [(x|y)]$ et $\sigma = [(y|c)]$.
2) On a

$$P(x, g(y)) \vee P(f(c), z) \vee R(x, y, z) \underset{fac}{\vdash} P(f(c), g(y)) \vee R(f(c), y, g(y))$$

avec $\sigma = [(x|f(c))(z|g(y))]$.

Proposition VI.1 *Un ensemble* Γ *de clauses du calcul des prédicats du premier ordre est insatisfaisable ssi* $\Gamma \underset{RAV}{\vdash} \square$.

Références bibliographiques

1. J. P. Delahaye. Outils logiques pour l'Intelligence Artificielle, *Eyrolles*, 1986.

2. C. Jacquemin. Logique et mathématiques pour l'informatique et l'I. A., *Masson*, 1994.