

# IA02 : Résolution de Problèmes et Programmation Logique

## Logique du Premier Ordre

Sylvain Lagrue

[sylvain.lagrue@hds.utc.fr](mailto:sylvain.lagrue@hds.utc.fr)



# À propos de ce document...

Information	Valeur
Auteur	Sylvain Lagrue ( <a href="mailto:sylvain.lagrue@utc.fr">sylvain.lagrue@utc.fr</a> )
Licence	Creative Common <a href="#">CC BY-SA 3.0</a>
Version document	1.3.0

## Sources/bibliographie :

- **Artificial Intelligence: A Modern Approach** : Stuart Russell and Peter Norvig, I.S.B.N 0136042597, 2009
- **Intelligence Artificielle et Informatique Théorique (2<sup>e</sup> édition)** : Jean-Marc Alliot, Pascal Brisset, Frederick Garcia, Thomas Schiex, I.S.B.N. 2854285786, 2002

Des coquilles ? [sylvain.lagrue@utc.fr](mailto:sylvain.lagrue@utc.fr) ou sur le [forum du cours moodle](#)

# I. Introduction

## Présentation

- appelé aussi calcul des prédictats
- à la base de la « logique mathématique »
- logique de Platon et de Socrate (syllogismes)
- généralisation de la logique propositionnelle
- utilisation de fonctions, de prédictats et de quantificateurs

## Liens très forts avec :

- la programmation logique (*Prolog*)
- la programmation par contrainte (*CSP*)
- les bases de données
- les bases de données déductives (*Datalog*)
- le web sémantique et les ontologies (*logiques de description*)
- ...

# I. Introduction

## Quelques exemples...

- Tous les Schtroumpfs sont bleus.

$$\forall x \text{ schtroumpf}(x) \rightarrow \text{bleu}(x)$$

Rq: l'implication est utilisée comme *filtrage*

- Il existe un Schtroumpf à lunettes.

Rq: ici, on ne veut surtout pas filtrer...

$$\exists x \text{ schtroumpf}(x) \wedge \text{lunettes}(x)$$

- Il n'existe pas de Schtroumpf qui n'aime pas la salsepareille.

$$\neg(\exists x \text{ schtroumpf}(x) \wedge \neg \text{aimeSalsepareille}(x))$$

$$\forall x \neg(\text{schtroumpf}(x) \wedge \neg \text{aimeSalsepareille}(x))$$

$$\forall x (\neg \text{schtroumpf}(x) \vee \text{aimeSalsepareille}(x))$$

$$\forall x (\text{schtroumpf}(x) \rightarrow \text{aimeSalsepareille}(x))$$

# I. Introduction

- Tout le monde aime tout le monde.

$$\forall x \forall y \text{ aime}(x, y)$$

- Tout le monde aime quelqu'un.

$$\forall x \exists y \text{ aime}(x, y)$$

- Quelqu'un aime tout le monde.

$$\exists x \forall y \text{ aime}(x, y)$$

- Tout le monde s'aime soi-même.

$$\forall x \forall y (x = y) \rightarrow \text{aime}(x, y)$$

$$\forall x \text{ aime}(x, x)$$

- Il existe quelqu'un qui n'aime personne.

$$\exists x \forall y \neg \text{aime}(x, y)$$

# I. Introduction

## Exemple de syllogisme

- Tous les humains sont mortels. Socrate est un humain. Donc Socrate est mortel.

$$((\forall x \text{ humain}(x) \rightarrow \text{mortel}(x)) \wedge \text{humain}(\text{Socrate})) \rightarrow \text{mortel}(\text{Socrate})$$

## Exemple de paralogisme

- Je ne suis pas dans la même pièce que mon frère.
- Ma fille n'est pas dans la même pièce que mon frère.
- **Donc** ma fille et moi sommes dans la même pièce...

# I. Introduction

## Pourquoi « premier ordre » ?

- on ne peut pas avoir de « prédicats de prédicats »
- en particulier, on ne peut pas quantifier les prédicats :

$$\forall p \exists x \ p(x)$$

## Remarque

- la logique du premier ordre est suffisante pour formaliser l'ensemble des preuves de la théorie des ensembles

## II. Le langage

### Les briques de base

- $F$  : un ensemble dénombrable de symboles de fonctions ( $f, g$ , etc.)
- $P$  : un ensemble dénombrable de symboles de prédictats ( $p, q$ , etc.)
- $C$  : un ensemble dénombrable de constantes ( $a, b, c$ , etc.)
- $V$  : un ensemble dénombrable de variables ( $x, y, z$ , etc.)

#### Remarques

- les fonctions et les prédictats possèdent une **arité**
- un prédictat d'arité 0 est une constante propositionnelle
- les constantes peuvent être vues comme des fonctions d'arité 0

#### Connecteurs usuels

- $\rightarrow \vee \wedge \leftrightarrow \neg \perp \top$  et les quantificateurs  $\exists$  et  $\forall$

#### Le prédictat d'égalité

- $=$

## II. Le langage

### Définition : terme

1. une variable est un terme
2. un symbole de constante est un terme
3. si  $f$  est une fonction d'arité  $n$  et si  $t_1, \dots, t_n$  sont des termes, alors  $f(t_1, \dots, t_n)$  est un terme

Tout terme est obtenu par l'application de ces règles un nombre **fini** de fois.

### Exemples

- $1 + 2 > 12$
- $\text{square}(\text{abs}(x))$
- $\text{concat}(x, "toto")$

## II. Le langage

### Définition : atome (formule atomique)

- si  $p$  est un symbole de prédicat d'arité  $n$  et  $t_1, t_2, \dots, t_n$  sont des termes, alors  $p(t_1, t_2, \dots, t_n)$  est un atome (ou formule atomique)

#### Exemples

- $schtroumpf(x)$
- $12 = abs(-12)$
- $\cos(\pi + x) = -\cos(x)$

## II. Le langage

### Définition : formule

1. un atome est une formule
2. si  $A$  et  $B$  sont des formules, alors  $(A \wedge B)$ ,  $(A \vee B)$ ,  $(A \rightarrow B)$ ,  $(A \leftrightarrow B)$  et  $(\neg A)$  sont des formules
3.  $\top$  et  $\perp$  sont des formules
4. Si  $A$  est une formule et  $x$  est une variable, alors  $(\forall x A)$  et  $(\exists x A)$  sont des formules

Une formule ne peut être définie que par un nombre fin de ces règles.

### Exemple

$$(\forall x (\exists y ((p(x, f(g(a), y)) \wedge q) \rightarrow r)))$$

- On peut représenter une formule sous forme d'arbre...
- On utilisera les mêmes priorités qu'en logique propositionnelle,  $\exists$  et  $\forall$  seront les moins prioritaires avec une priorité *droite/gauche*

## II. Le langage

### Variables liées et variables libres

#### Exemple

$$\forall x \ p(x, y)$$

$x$  est liée et  $y$  est libre

#### Définition (inductive) variables liées

Soit  $A$  une formule et  $\text{Linked}(A)$  l'ensemble des variables liées de  $A$

1. si  $A$  est un atome alors  $\text{Linked}(A) = \emptyset$
2.  $\text{Linked}(\perp) = \text{Linked}(\top) = \emptyset$
3. si  $A$  est de la forme  $B * C$  alors  $\text{Linked}(A) = \text{Linked}(B) \cup \text{Linked}(C)$   
(avec  $* \in \{\rightarrow, \vee, \wedge, \leftrightarrow\}$ )
4. si  $A$  est de la forme  $\neg B$  alors  $\text{Linked}(A) = \text{Linked}(B)$
5. si  $A$  est de la forme  $\forall x \ B$  ou de la forme  $\exists x \ B$  alors  
 $\text{Linked}(A) = \{x\} \cup \text{Linked}(B)$

## II. Le langage

### Autres définitions

- une **variable libre** est une variable non liée !
- $Free(A)$  représente l'ensemble des variables libres de la formule  $A$
- une formule **close** (fermée) est une formule ne contenant aucune variable libre

### Exercices :

- quelles sont les variables libres et les variables liées de la formule suivante ?

$$p(x, y) \vee (\forall x \ q(x) \wedge r(x))$$

- donner une définition inductive des variables libres

## II. Le langage

### Définition (inductive) variables libres

Soit  $A$  une formule et  $Free(A)$  l'ensemble des variables libres de  $A$ .

1. si  $A$  est un atome alors  $Free(A) = \{\text{variables apparaissant dans } A\}$
2.  $Free(\perp) = Free(\top) = \emptyset$
3. si  $A$  est de la forme  $B * C$  alors  $Free(A) = Free(B) \cup Free(C)$  (avec  $* \in \{\rightarrow, \vee, \wedge, \leftrightarrow\}$ )
4. si  $A$  est de la forme  $\neg B$  alors  $Free(A) = Free(B)$
5. si  $A$  est de la forme  $\forall x B$  ou de la forme  $\exists x B$  alors  
 $Free(A) = Free(B) \setminus \{x\}$

## II. Le langage

### Clôture d'une formule

- la clôture universelle (resp. existentielle) de  $A$  est telle que, si  $\{x_1, x_2, \dots, x_n\}$  est l'ensemble des variables libres de  $A$  :

$$\forall x_1 \forall x_2 \dots \forall x_n A$$

$$(\text{resp. } \exists x_1 \exists x_2 \dots \exists x_n A)$$

*Attention, il peut être nécessaire de renommer des variables...*

**Remarque :** identité vs égalité conditionnelle...

- $(x + 1)^2 = x^2 + 2x + 1$
- $x^2 - 3x + 1 = 0$

### III. Théorie des modèles

Pour cela, on va avoir besoin de **domaines** pour l'ensemble des variables des prédictats et des fonctions.

**Exemple** : prédictat *bleu* dont le domaine est

$$D = \{Xzulrb\_le\_martien, Ernest, Schtroumpf\_grognon\}$$

x	V(I, bleu(x))
Xzulrb_le_martien	F
Ernest	F
Schtroumpf_grognon	V

**Remarque** : les domaines peuvent être infinis (par exemple les entiers...)

### III. Théorie des modèles

#### Définition d'un interprétation

Une interprétation  $I$  est un triplet  $(\mathcal{D}, \mathcal{I}_c, I_v)$

- $\mathcal{D}$  domaine d'interprétation
- $I_c$  une fonction qui associe à toute variable libre une valeur de  $\mathcal{D}$
- $I_v$  une fonction qui associe :
  - à toute constante une valeur de  $\mathcal{D}$
  - à toute constante fonctionnelle  $f_n$  une fonction  $I_{c(f_n)}$  dans  $\mathcal{D}$
  - à toute constante prédicative  $P_m$  une fonction  $I_{c(P_m)}$  dans  $\{V, F\}$

### III. Théorie des modèles

#### Valuation

Valuation d'une formule  $A$  par l'interprétation  $I$  :

- Si  $x$  est une variable libre, alors  $I(x) = I_v(x)$
- $I(f(t_1, \dots, t_n)) = (I_c(f))(I(t_1), \dots, I(t_n))$
- $I(p(t_1, \dots, t_m)) = (I_c(p))(I(t_1), \dots, I(t_m))$
- si  $A$  et  $B$  sont des formules alors  $\neg A, A \wedge B, A \vee B, A \rightarrow B, A \leftrightarrow B$  s'interprètent comme dans la logique propositionnelle
- si  $A$  est une formule et  $x$  une variable alors  $I(\forall x A) = V$  si  $I_{\{x:=d\}}(A) = V$  pour tout élément  $d \in D$
- si  $A$  est une formule et  $x$  une variable alors  $I(\exists x A) = V$  si  $I_{\{x:=d\}}(A) = V$  pour au moins un élément  $d \in D$

### III. Théorie des modèles

#### Exercice

Traduire les phrases suivantes en logique des prédictats

- A : Toutes les voitures ont exactement un propriétaire
- B : Certains étudiants ont une voiture
- C : Certains étudiants n'ont pas de voiture

Soit  $I = (D, Ic, Iv)$  avec  $D = \{a, b\}$  telle que :

- $I_c(voiture) = p_v$  tq si  $x = a$  alors  $p_v(x) = V$  sinon  $p_v(x) = F$
- $I_c(etudiant) = p_e$  tq si  $x = b$  alors  $p_e(x) = V$  sinon  $p_e(x) = F$
- $I_c(posse de) = p_p$  tq si  $x = b$  et  $y = a$  alors  $p_p(x, y) = V$  sinon  $p_p(x, y) = F$

$I(A) ? I(B) ? I(C) ?$

### III. Théorie des modèles

Traduire la phrase suivante en logique des prédictats

- B : Certains étudiants ont une voiture

$$\exists x \exists y \text{ etudiant}(x) \wedge \text{voiture}(y) \wedge \text{possede}(x, y)$$

### III. Théorie des modèles

Soit  $I = (D, I_c, I_v)$  avec  $D = \{a, b\}$  telle que :

- $I_c(\text{voiture}) = p_v$  tq si  $x = a$  alors  $p_v(x) = V$  sinon  $p_v(x) = F$
- $I_c(\text{etudiant}) = p_e$  tq si  $x = b$  alors  $p_e(x) = V$  sinon  $p_e(x) = F$
- $I_c(\text{possede}) = p_p$  tq si  $x = b$  et  $y = a$  alors  $p_p(x, y) = V$  sinon  $p_p(x, y) = F$
- $I(B) = I(\exists x \exists y \text{ etudiant}(x) \wedge \text{voiture}(y) \wedge \text{possede}(x, y)) = ?$ 
  - $I_{\{x:=a\}}(\exists y \text{ etudiant}(a) \wedge \text{voiture}(y) \wedge \text{possede}(a, y))$ 
    - $I_{\{y:=a\}}(\exists y \text{ etudiant}(a) \wedge \text{voiture}(a) \wedge \text{possede}(a, a))$ 
      - $I(\text{etudiant}(a) \wedge \text{voiture}(a) \wedge \text{possede}(a, a))$
      - $I_c(\text{etudiant})(a) \wedge I_c(\text{voiture})(a) \wedge I_c(\text{possede})(a, a)$
      - $p_e(a) \wedge p_v(a) \wedge p_p(a, a)$
      - $F \wedge V \wedge F$

### III. Théorie des modèles

Soit  $I = (D, Ic, Iv)$  avec  $D = \{a, b\}$  telle que :

- $I_c(\text{voiture}) = p_v$  tq si  $x = a$  alors  $p_v(x) = V$  sinon  $p_v(x) = F$
- $I_c(\text{etudiant}) = p_e$  tq si  $x = b$  alors  $p_e(x) = V$  sinon  $p_e(x) = F$
- $I_c(\text{possede}) = p_p$  tq si  $x = b$  et  $y = a$  alors  $p_p(x, y) = V$  sinon  $p_p(x, y) = F$
- $I_{\{x:=a\}}(\exists y \text{ etudiant}(a) \wedge \text{voiture}(y) \wedge \text{possede}(a, y))$ 
  - $I_{\{y:=b\}}(\exists y \text{ etudiant}(a) \wedge \text{voiture}(b) \wedge \text{possede}(a, b))$
  - $I(\text{etudiant}(a) \wedge \text{voiture}(b) \wedge \text{possede}(a, b))$
  - $I_c(\text{etudiant})(a) \wedge I_c(\text{voiture})(b) \wedge I_c(\text{possede})(a, b)$
  - $p_e(a) \wedge p_v(b) \wedge p_p(a, b)$
  - $F \wedge F \wedge F$

### III. Théorie des modèles

Soit  $I = (D, Ic, Iv)$  avec  $D = \{a, b\}$  telle que :

- $I_c(\text{voiture}) = p_v$  tq si  $x = a$  alors  $p_v(x) = V$  sinon  $p_v(x) = F$
- $I_c(\text{etudiant}) = p_e$  tq si  $x = b$  alors  $p_e(x) = V$  sinon  $p_e(x) = F$
- $I_c(\text{possede}) = p_p$  tq si  $x = b$  et  $y = a$  alors  $p_p(x, y) = V$  sinon  $p_p(x, y) = F$
- $I_{\{x:=b\}}(\exists y \text{ etudiant}(b) \wedge \text{voiture}(y) \wedge \text{possede}(b, y))$ 
  - $I_{\{y:=a\}}(\exists y \text{ etudiant}(b) \wedge \text{voiture}(a) \wedge \text{possede}(b, a))$ 
    - $I(\text{etudiant}(b) \wedge \text{voiture}(a) \wedge \text{possede}(b, a))$
    - $I_c(\text{etudiant})(b) \wedge I_c(\text{voiture})(a) \wedge I_c(\text{possede})(b, a)$
    - $p_e(b) \wedge p_v(a) \wedge p_p(b, a)$
    - $V \wedge V \wedge V$
  - ...

Donc  $I(B) = V$

### III. Théorie des modèles

#### Conséquence logique

Les définitions de conséquence logique, de formules valides, contingentes et contradictoires s'étendent directement depuis celles de la logique propositionnelle.  
Soit  $W$  l'ensemble des interprétations :

- $\models A$  si pour tout  $I \in W$  on a  $I(A) = V$
- $A \models B$  si pour tout  $I \in W$  telle que  $I(A) = V$  on a  $I(B) = V$

### III. Théorie des modèles

#### Quelques propriétés...

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

$$(\forall x A) \vee (\forall x B) \models \forall x (A \vee B)$$

$$\forall x (A \rightarrow B) \models (\forall x A) \rightarrow (\forall x B)$$

$$\forall x (A \leftrightarrow B) \models (\forall x A) \leftrightarrow (\forall x B)$$

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

$$\exists x (A \wedge B) \models (\exists x A) \wedge (\exists x B)$$

$$\exists x (A \rightarrow B) \equiv (\exists x A) \rightarrow (\exists x B)$$

$$\forall x \neg A \equiv \neg \exists x A$$

## IV. Système formel

### Schéma d'axiomes

Soient  $A, B, C, D$  des formules du premier ordre quelconques,  $x$  une variable et  $t$  un terme, tel que  $x \notin \text{Free}(D)$  :

- $A1 : (A \rightarrow (B \rightarrow A))$
- $A2 : ((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$
- $A3 : ((\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B))$
- $A4 : (\forall x A(x) \rightarrow A(t))$
- $A5 : ((D \rightarrow B) \rightarrow (D \rightarrow \forall x B))$

# IV. Système formel

## Règles d'inférences

- *Modus Ponens:*

$$\frac{\vdash A, \quad \vdash A \rightarrow B}{\vdash B}$$

- *Généralisation:*

$$\frac{\vdash A}{\vdash (\forall x \ A)}$$

## IV. Système formel

- *Règle de substitution*

$$\frac{A(x)\{x := t\}}{A(t)}$$

- soit  $A(x)$  une formule contenant  $x$  comme variable libre et  $t$  un terme
- $A(t)$  : obtenue en remplaçant les occurrences libres de  $x$  par  $t$  dans  $A(x)$
- Si  $x$  ou  $t$  apparaissent comme variables liées dans la formule  $A(x)$  alors renommer ces occurrences

# IV. Système formel

**Exemples de substitution :**

$$\frac{(p(x) \vee q(x, y))\{x := z\}}{p(z) \vee q(z, y)}$$

$$\frac{((p(x) \vee q(x, y))\{x := y\})}{p(y) \vee q(y, y)}$$

$$\frac{(\forall x q(x, y))\{y := x, x := z\}}{\forall z q(z, x)}$$

## Exemple

Montrons :  $\vdash (\forall x A \rightarrow A)$

Étape 1 :  $\vdash (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$  (Axiome 2)

Étape 2 : en substituant  $A \rightarrow A$  à  $B$  et  $A \rightarrow A$  à  $C$  on obtient :

$$\vdash (A \rightarrow ((A \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A))$$

Étape 3 :  $\vdash A \rightarrow (B \rightarrow A)$  (Axiome 1)

Étape 4 : en substituant  $A \rightarrow A$  à  $B$  dans 3 on obtient :

$$\vdash A \rightarrow ((A \rightarrow A) \rightarrow A)$$

Étape 5 : *modus ponens* entre 4 et 2 permet d'obtenir :

$$\vdash (A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)$$

*Étape 5 : modus ponens entre 4 et 2 permet d'obtenir :*

$$\vdash (A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)$$

*Étape 6 : En substituant  $A$  à  $B$  dans l'axiome 1 on obtient :*

$$\vdash A \rightarrow (A \rightarrow A)$$

*Étape 7 : modus ponens entre 5 et 6*

$$\vdash A \rightarrow A$$

*Étape 8 : généralisation sur 7*

$$\vdash (\forall x A \rightarrow A)$$

# Théorèmes de la déduction

- Soit  $A$  une formule **close**. Si  $A \vdash B$  alors  $\vdash A \rightarrow B$ .

*Plus généralement :*

- Soient  $A_1, \dots, A_n$  des formules **closes** :

Si  $A_1, \dots, A_n \vdash B$  alors  $A_1, \dots, A_{n-1} \vdash A_n \rightarrow B$

- **théorème (d'adéquation)** : pour toute formule  $A \in L$  si  $\vdash A$  alors  $\models A$
  - **théorème (de complétude faible)** : pour toute formule  $A \in L$ , si  $\models A$  alors  $\vdash A$
  - **théorème (de complétude forte)** : pour tout ensemble de formules  $E \subseteq L$  et pour toute formule  $C \in L$ , si  $E \models C$  alors  $E \vdash C$
- 
- **théorème (de décidabilité)** : la logique des prédictats est semi-décidable. Il n'existe aucun programme qui pour une formule  $A$  indique en un temps fini si  $A$  n'est pas un théorème.

# Propositions (3)

- Toute théorie axiomatique égalitaire ayant :
  - un nombre fini de symboles, un nombre fini de constantes
  - un seul symbole fonctionnel unaire  $f$
  - un nombre fini de prédictats unaires et le prédictat binaire égalité
  - n'ayant pas d'axiomes non logiques

est décidable.

# V. Formes normales et univers de Herbrand

**Objectif** : se ramener à des formes plus facilement calculables... Voir à des formes proches de la logique propositionnelle !

- forme prénexe
- forme de Skolem
- forme normale (conjonctive)

# V. Formes normales et univers de Herbrand

## Forme prénexe

**Définition.** Une *matrice* est une formule de la logique du premier ordre ne comportant aucun quantificateur.

*Exemple.*

$$p(x) \leftrightarrow q(f(g(y)), a)$$

**Définition.** Une formule est sous *forme prénexe* si elle est de la forme :

$$Q_1 x_1 Q_2 x_2 \dots Q_n x_n M$$

où  $Q \in \{\exists, \forall\}$  et  $M$  est une matrice.

*Exemple.*

$$\exists x \forall y (p(x) \leftrightarrow (q(f(g(y)), a)))$$

**Théorème.** Toute formule admet une forme prénexe équivalente.

1. supprimer les connecteurs d'équivalence et d'implication
2. renommer les variables liées afin qu'une même variable ne soit pas quantifiée 2 fois  $\forall x A(x)$  est équivalent à  $\forall y A(y)$
3. supprimer les quantificateurs inutiles (dont la variable quantifiée n'apparaît pas dans leur portée)
4. transférer les négations devant les atomes en utilisant les règles de la logique propositionnelle et les 2 règles suivantes
  - $\neg \exists x A \equiv \forall x \neg A$
  - $\neg \forall x A \equiv \exists x \neg A$
5. faire passer les quantificateur en tête en utilisant les règles suivantes (et en utilisant éventuellement l'associativité, la commutativité et le renommage de variable) :
  - $(\forall x A) \wedge B \equiv \forall x (A \wedge B)$  si  $B$  ne contient pas  $x$
  - $(\exists x A) \wedge B \equiv \exists x (A \wedge B)$  si  $B$  ne contient pas  $x$
  - $(\forall x A) \vee B \equiv \forall x (A \vee B)$  si  $B$  ne contient pas  $x$
  - $(\exists x A) \vee B \equiv \exists x (A \vee B)$  si  $B$  ne contient pas  $x$

## Exemple

$$\begin{aligned} & ((\forall x p(x)) \wedge (\exists y q(y))) \rightarrow (\exists y p(y) \wedge q(y)) \\ & \equiv \neg((\forall x p(x)) \wedge (\exists y q(y))) \vee (\exists y p(y) \wedge q(y)) \\ & \equiv (\neg\forall x p(x)) \vee (\neg\exists y q(y)) \vee (\exists y p(y) \wedge q(y)) \\ & \equiv (\exists x \neg p(x)) \vee (\forall y \neg q(y)) \vee (\exists y p(y) \wedge q(y)) \\ & \equiv (\exists x \neg p(x)) \vee (\forall y \neg q(y)) \vee (\exists z p(z) \wedge q(z)) \\ & \equiv \exists x \forall y \exists z \neg p(x) \vee \neg q(y) \vee (p(z) \wedge q(z)) \end{aligned}$$

# V. Formes normales et saturation

## Forme de Skolem

### Objectif

Faire disparaître les quantificateurs existentiels et se ramener à la forme la plus simple possible pour le calcul.

### Algorithme (de Skolémisation)

Soit une formule  $F$

1. transformer la formule  $F$  en une forme prénexe
2. remplacer toutes les variables quantifiées existentiellement par un symbole de fonction dont les arguments sont les variables quantifiées universellement qui le précèdent
3. on supprime les quantificateurs existentiels

# V. Formes normales et saturation

## Exemple (suite)

$$\exists x \forall y \exists z \neg p(x) \vee \neg q(y) \vee (p(z) \wedge q(z))$$

$$\exists x \forall y \exists z \neg p(x) \vee \neg q(y) \vee (p(f(y)) \wedge q(f(y)))$$

$$\exists x \forall y \exists z \neg p(a) \vee \neg q(y) \vee (p(f(y)) \wedge q(f(y)))$$

$$\forall y \neg p(a) \vee \neg q(y) \vee (p(f(y)) \wedge q(f(y)))$$

# V. Formes normales et saturation

## Forme normale (conjonctive)

### Extension directe des concepts

- littéral
- clause
- cube
- forme normale conjonctive
- forme normale disjonctive

### Définition

Une formule est sous **forme normale conjonctive** si elle est sous forme de Skolem et que sa matrice est composée uniquement de conjonctions de clauses.

N.B. (1) : On peut supprimer les quantificateurs universels.

N.B. (2) : On peut écrire la formule sous forme ensembliste en enlevant les conjonctions.

# V. Formes normales et saturation

## Exemple (suite)

$$\forall y \neg p(a) \vee \neg q(y) \vee (p(f(y)) \wedge q(f(y)))$$

$$\forall x \neg p(a) \vee \neg q(x) \vee (p(f(x)) \wedge q(f(x)))$$

$$\forall x (\neg p(a) \vee \neg q(x)) \vee (p(f(x)) \wedge q(f(x)))$$

$$\forall x (\neg p(a) \vee \neg q(x) \vee p(f(x))) \wedge (\neg p(a) \vee \neg q(x) \vee q(f(x)))$$

$$(\neg p(a) \vee \neg q(x) \vee p(f(x))) \wedge (\neg p(a) \vee \neg q(x) \vee q(f(x)))$$

$$\{\neg p(a) \vee \neg q(x) \vee p(f(x)), \neg p(a) \vee \neg q(x) \vee q(f(x))\}$$

# V. Formes normales et saturation

## Univers de Herbrand

*Objectif.* Restreindre les domaines aux domaines apparaissant réellement dans la formule.

### Terme de base et atome de base

**Définition.** Un terme (resp. un atome) de base est un terme (resp. un atome) qui n'apparaît pas de variable. On parle également de terme (resp. d'atome) complètement instancié.

### Univers de Herbrand

**Définition.** On appelle univers de Herbrand d'une forme normale  $E$  l'ensemble des termes de base que l'on peut construire à partir des symboles de fonction et des symboles de constantes qui apparaissent dans  $E$ .

Si aucun symbole de constante n'apparaît, on introduit arbitrairement une constante afin de ne pas laisser l'univers vide.

# V. Formes normales et saturation

*Exemple.*

$$E = \{p(f(x)) \vee q(a), r(g(x))\}$$

L'univers de Herbrand associé à  $E$  est :

$$\{a, f(a), g(a), f(f(a)), f(g(a)), g(f(a)), g(g(a)), \dots\}$$

# V. Formes normales et saturation

## Base de Herbrand

**Définition.** On appelle base de Herbrand d'un ensemble de clauses  $E$  l'ensemble des atomes de base qui peuvent être construits à partir des symboles de prédictats de  $E$ , appliqués aux termes de l'univers de Herbrand associé.

Sur l'exemple précédent :

$$E = \{p(f(x)) \vee q(a), r(g(x))\}$$

$$\begin{aligned} &\{p(a), q(a), r(a), p(f(a)), q(f(a)), r(f(a)), \dots, \\ &p(f(g(a))), q(f(g(a))), r(f(g(a))), \dots\} \end{aligned}$$

# V. Formes normales et saturation

## Système de Herbrand

Soit  $E$  un ensemble de clauses, le système de Herbrand est l'ensemble des clauses obtenues à partir de  $E$  en remplaçant les variables par des éléments de l'univers de Herbrand.

## Théorème de Herbrand

Un ensemble de clauses  $E$  est satisfiable si et seulement si il a un modèle de Herbrand.

# V. Formes normales et saturation

## Exemple (1)

$$F = \forall x \ p(x) \wedge \neg p(a)$$

$$E = \{p(x), \neg p(a)\}$$

$$D_H = \{a\}$$

$$S = \{\neg p(a), p(a)\}$$

Contradiction !

# V. Formes normales et saturation

## Exemple (2)

$$F = \forall x (p(x) \vee q(x)) \wedge \neg p(a) \wedge \neg q(b)$$

$$E = \{p(x) \vee q(x), \neg p(a), \neg q(b)\}$$

$$D_H = \{a, b\}$$

$$S_1 = \{p(a) \vee q(a), \neg p(a), \neg q(b)\}$$

(qui est vrai si  $q(a)$  est vrai)

$$S_2 = S_1 \cup \{p(b) \vee q(b), \neg p(a), \neg q(b)\}$$

(qui est vrai si  $p(b)$  est vrai)

Tout le domaine ayant été utilisé, on obtient le modèle I suivant :

- $I(p(a)) = F$  et  $I(p(b)) = V$ ,
- $I(q(a)) = V$  et  $I(q(b)) = F$

# V. Formes normales et saturation

## Exemple (3)

$$F = \forall x \forall y (p(x) \vee q(x)) \wedge p(x) \wedge \neg p(f(y)))$$

$$E = \{p(x) \vee q(x), p(x), \neg p(f(y)))\}$$

$$D_H = \{a, f(a), f(f(a)), f(f(f(a))), \dots\}$$

$$S_1 = \{p(a), \neg p(f(a)))\}$$

$$S_2 = S_1 \cup \{p(f(a)), \neg p(f(f(a))))\}$$

Ce qui est impossible !

# VI. Unification et principe de résolution

**Objectif :** généraliser le principe de résolution de la logique propositionnelle !

- substitution
- unification
- résolution

# VI. Unification et principe de résolution

## Substitution

### Définition : substitution

- Une **substitution**  $\sigma$  est une application de l'ensemble des variables  $Var$  dans l'ensemble des termes  $T$  telle que l'ensemble  $\{x \in Var, \sigma(x) \neq x\}$  est fini.

### Définition : instance

- Le terme  $\sigma(t)$  obtenu en remplaçant simultanément dans le terme  $t$  toutes les occurrences des variables  $x_i$  par  $\sigma(x_i)$  est une **instance** de  $t$ .
- Soit  $t_1$  et  $t_2$  des termes,  $t_2$  est une **instance** de  $t_1$  s'il existe une substitution  $\sigma$  telle que  $t_2 = \sigma(t_1)$ .

# VI. Unification et principe de résolution

## Exemple

Soient  $x$  et  $y$  2 variables et  $a$  une constante.

Soit  $\sigma$  tel que :

- $\sigma(x) = a$
- $\sigma(y) = f(x, a)$

Soit le terme  $t = g(x, y, f(x, y))$

$$\sigma(t) = g(a, f(x, a), f(a, f(x, a)))$$

$\sigma(t)$  n'est pas totalement instanciée...

$$\sigma(\sigma(t)) = g(a, f(a, a), f(a, f(a, a)))$$

# VI. Unification et principe de résolution

## Unification

### Définition : instance fondamentale d'une clause

Une instance d'une clause  $C$  dont tous les termes sont complètement instanciés est une instance fondamentale (ou de base) de  $C$ .

*Exemple :*  $p(a, f(b)) \vee q(g(c))$

### Définition : littéraux unifiables

Deux termes sont dits unifiables s'ils possèdent une instance fondamentale commune.

*Exemple :*  $f(x)$  et  $f(g(y, z))$  avec  $f(g(a, b))$

### Définition : unificateur principal

$\sigma'$  est plus général que  $\sigma$ , s'il existe  $\sigma''$  tq  $\sigma = \sigma'' \circ \sigma'$

Un unificateur principal (unificateur le plus général) est un unificateur pour qui il n'existe pas d'unificateur plus général.

# VI. Unification et principe de résolution

*Exemple*

Soit  $p(a, y, z)$  et  $p(x, b, z)$ , on a :

- $\sigma_1 = \{x := a, y := b, z := c\} \rightsquigarrow p(a, b, c)$
- $\sigma_2 = \{x := a, y := b, z := d\} \rightsquigarrow p(a, b, d)$
- $\sigma_3 = \{x := a, y := b, z := f(c)\} \rightsquigarrow p(a, b, f(c))$
- ...

Si on considère  $\sigma = \{x := a, y := b\}$  :

- $\sigma_1 = \{z := c\} \circ \sigma$
- $\sigma_2 = \{z := d\} \circ \sigma$
- $\sigma_3 = \{z := f(c)\} \circ \sigma$
- ...

# VI. Unification et principe de résolution

## Algorithme de Robinson (1965)

**entrée** :  $A_1$  et  $A_2$  deux atomes à unifier

**sortie** : un unificateur principal  $\sigma$

$$\sigma = \{\}$$

tant que  $\sigma(A_1) \neq \sigma(A_2)$ :

- trouver le premier symbole de  $A_1$  différent du symbole correspondant de  $A_2$   
déterminer les termes respectifs  $t_1$  et  $t_2$  de  $A_1$  et  $A_2$  débutant à ce rang
- si  $t_1$  et  $t_2$  ne sont pas des variables : *Échec*
- si l'un des 2 termes est une variable  $x$  contenue dans l'autre terme  $t$  : *Échec*
- sinon composer  $\sigma$  avec  $\{x := t\}$

renvoyer  $\sigma$

# VI. Unification et principe de résolution

Exemple :

$$A_1 = p(x, a) \text{ et } A_2 = p(f(y), y)$$

- **itération 1** :  $\sigma(A_1) \neq \sigma(A_2)$ 
  - premiers symboles non concordants :  $x$  et  $f$
  - termes :  $x$  et  $f(y)$
  - $\sigma = \{x := f(y)\}$
- **itération 2** :  $\sigma(A_1) = p(f(y), a)$  et  $\sigma(A_2) = p(f(y), y)$  donc  $\sigma(A_1) \neq \sigma(A_2)$ 
  - premiers symboles non concordants :  $a$  et  $y$
  - termes :  $a$  et  $y$
  - $\sigma = \{y := a\} \circ \{x := f(y)\}$
- **itération 3** :  $\sigma(A_1) = p(f(a), a) = \sigma(A_2)$
- renvoyer  $\sigma$

# VI. Unification et principe de résolution

Remarques :

- algorithme très simple
- particulièrement coûteux (en temps et en mémoire)
- il existe des algorithmes beaucoup plus performants...

# VI. Unification et principe de résolution

## Principe de résolution

- $S$  : un ensemble de clauses,  $c_1, c_2 \in S$
- $l_1$  apparaît dans  $c_1$  et  $\neg l_2$  apparaît dans  $c_2$
- $\theta$  une substitution de renommage telle que  $\theta(c_1)$  et  $c_2$  n'ont aucune variable libre en commun
- soit  $\sigma$  l'unificateur principal de  $\theta(c_1)$  et  $c_2$  Alors :

$$S \equiv S \cup \{r\}$$

avec  $r = \sigma((\theta(c_1) \setminus \{l_1\}) \vee (c_2 \setminus \{\neg l_2\}))$  appelée résolvante

# VI. Unification et principe de résolution

## Exemple

- (1)  $\neg p(x) \vee p(f(x))$
- (2)  $p(a)$
- (3)  $\neg p(f(z))$

(1) et (2) avec  $\sigma(x) = a$  produit (4)  $p(f(a))$

(3) et (4) avec  $\sigma(z) = a$  produit une contradiction !

# VI. Unification et principe de résolution

## Exemple

- $A$  : les Schtroumpfs aiment la salsepareille
- $B$  : il existe un Schtroumpf grognon
- $C$  : les Schtroumpfs grognons n'aiment rien

Montrer que ces assertions sont incohérentes.

## Vocabulaire

- prédicts (concepts + relation) :  $schtroumpf/1$ ,  $grognon/1$ ,  $aime/2$
- variables :  $x, y$
- constante :  $salsepareille$

## Représentation du problème

- $A = \forall x schtroumpf(x) \rightarrow aime(x, salsepareille)$
- $B = \exists x schtroumpf(x) \wedge grognon(x)$
- $C = \forall x schtroumpf(x) \wedge grognon(x) \rightarrow \neg(\exists y aime(x, y))$

# VI. Unification et principe de résolution

## La formule à traiter

$$\begin{aligned} F = & (\forall x \text{ schtroumpf}(x) \rightarrow \text{aime}(x, \text{salsepareille})) \wedge \\ & (\exists x \text{ schtroumpf}(x) \wedge \text{grognon}(x)) \wedge \\ & (\forall x \text{ schtroumpf}(x) \wedge \text{grognon}(x) \rightarrow \neg(\exists y \text{ aime}(x, y))) \end{aligned}$$

## Mise sous forme prénex

$$\begin{aligned} C \equiv & \forall x \text{ schtroumpf}(x) \wedge \text{grognon}(x) \rightarrow \neg(\exists y \text{ aime}(x, y)) \\ \equiv & \forall x \text{ schtroumpf}(x) \wedge \text{grognon}(x) \rightarrow (\forall y \neg \text{aime}(x, y)) \\ \equiv & \forall x \forall y \text{ schtroumpf}(x) \wedge \text{grognon}(x) \rightarrow \neg \text{aime}(x, y) \end{aligned}$$

Rappel 1 : commutativité de la conjonction

$$\begin{aligned} F \equiv & (\exists x \text{ schtroumpf}(x) \wedge \text{grognon}(x)) \wedge \\ & (\forall x \text{ schtroumpf}(x) \rightarrow \text{aime}(x, \text{salsepareille})) \wedge \\ & (\forall x \forall y \text{ schtroumpf}(x) \wedge \text{grognon}(x) \rightarrow \neg \text{aime}(x, y)) \end{aligned}$$

# VI. Unification et principe de résolution

Rappel 2:  $(\forall x F) \wedge (\forall x G) \equiv \forall x (F \wedge G)$

$$\begin{aligned} F \equiv & (\exists x \text{ schtroumpf}(x) \wedge \text{grognon}(x)) \wedge \\ & \forall x ((\text{schtroumpf}(x) \rightarrow \text{aime}(x, \text{salsepareille})) \wedge \\ & (\forall y \text{ schtroumpf}(x) \wedge \text{grognon}(x) \rightarrow \neg \text{aime}(x, y))) \end{aligned}$$

Après renommage...

$$\begin{aligned} F \equiv & \exists z \forall x (\text{schtroumpf}(z) \wedge \text{grognon}(z)) \wedge \\ & (\text{schtroumpf}(x) \rightarrow \text{aime}(x, \text{salsepareille})) \wedge \\ & (\forall y \text{ schtroumpf}(x) \wedge \text{grognon}(x) \rightarrow \neg \text{aime}(x, y)) \end{aligned}$$

$y$  n'apparaissant pas dans la partie gauche de la formule...

$$\begin{aligned} F \equiv & \exists z \forall x \forall y \text{ schtroumpf}(z) \wedge \text{grognon}(z) \wedge \\ & (\text{schtroumpf}(x) \rightarrow \text{aime}(x, \text{salsepareille})) \wedge \\ & (\text{schtroumpf}(x) \wedge \text{grognon}(x) \rightarrow \neg \text{aime}(x, y)) \end{aligned}$$

# VI. Unification et principe de résolution

## Mise sous forme de Skolem

$$\begin{aligned} F \equiv & \textit{schtroumpf}(a) \wedge \textit{grognon}(a) \wedge \\ & (\textit{schtroumpf}(x) \rightarrow \textit{aime}(x, \textit{salsepareille})) \wedge \\ & (\textit{schtroumpf}(x) \wedge \textit{grognon}(x) \rightarrow \neg \textit{aime}(x, y)) \end{aligned}$$

## Mise sous forme normale conjonctive...

$$\text{Rappel 3 : } F \wedge G \rightarrow H \equiv \neg(F \wedge G) \vee H \equiv \neg F \vee \neg G \vee H$$

$$\begin{aligned} F \equiv & \textit{schtroumpf}(a) \wedge \textit{grognon}(a) \wedge \\ & (\neg \textit{schtroumpf}(x) \vee \textit{aime}(x, \textit{salsepareille})) \wedge \\ & (\neg \textit{schtroumpf}(x) \vee \neg \textit{grognon}(x) \vee \neg \textit{aime}(x, y)) \end{aligned}$$

# VI. Unification et principe de résolution

La base de clauses  $KB$ :

- $C_1 = \text{schtroumpf}(a)$
- $C_2 = \text{grognon}(a)$
- $C_3 = \neg \text{schtroumpf}(x) \vee \text{aime}(x, \text{salsepareille})$
- $C_4 = \neg \text{schtroumpf}(x) \vee \neg \text{grognon}(x) \vee \neg \text{aime}(x, y)$

Applications du principe de résolution

- $$\frac{C_1, C_3, \sigma=\{x:=a\}}{C_5 = \text{aime}(a, \text{salsepareille})}$$
- $$\frac{C_1, C_4, \sigma=\{x:=a\}}{C_6 = \neg \text{grognon}(a) \vee \neg \text{aime}(a, y)}$$
- $$\frac{C_5, C_6, \sigma=\{y:=\text{salsepareille}\}}{C_7 = \neg \text{grognon}(a)}$$
- $$\frac{C_7, C_2, \sigma=\{\}}{\emptyset}$$

D'après le théorème de Herbrand, la formule est inconsistante et l'énoncé est donc incohérent.

# VI. Unification et principe de résolution

## Question...

Comment déduire une formule  $F$  d'un ensemble de clause  $KB$  ?

En montrant que  $KB \cup \{\neg F\}$  est inconsistante !

# Conclusion et synthèse des points abordés

## Le langage, la syntaxe et la sémantique

- définition d'un nouveau langage (ajout de quantificateurs, de prédictats, de fonctions et de constantes)
- si la formule ne contient que des prédictats 0-aire, on retrouve la logique propositionnelle
- définition d'une syntaxe basée sur des règles de réécritures
- définition d'une sémantique basée sur des domaines et des interprétations
- la logique du premier ordre est adéquate et complète
- en revanche elle n'est que semi-décidable

# Conclusion et synthèse des points abordés

## D'un point de vue computationnel

- il est toujours possible de mettre une formule/un ensemble de formules sous forme normale conjonctive (via les formes prénexes et de Skolem)
- le principe de résolution s'étend à la logique du premier ordre via le principe d'unification
- le calcul d'incohérence et d'inférence est rendu possible via le théorème de Herbrand

## La suite ?

- la programmation logique !

