# Logique et Principe de résolution

D.Pastre

Université René Descartes - Paris 5
UFR de mathématiques et informatique

SOMMAIRE

# CALCUL PROPOSITIONNEL

Etude des formules sans variables. Manipulation d'énoncés qui sont soit vrais, soit faux.

## 1 Définition du langage

On a un ensemble de variables propositionnelles $\mathcal{V}$ et un ensemble de connecteurs $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$

L'ensemble $\mathcal{F}$ des formules bien formées (wff) est l'ensemble des expressions $F$ telles que :
- $\mathcal{V}$ est inclus dans $\mathcal{F}$
- si $F$ est un élément de $\mathcal{F}$ , alors $(\neg F)$ est un élément de $\mathcal{F}$
- si $F1$ et $F2$ sont des éléments de $\mathcal{F}$, alors $(F1 \wedge F2)$, $(F1 \vee F2)$, $(F1 \rightarrow F2)$ et $(F1 \leftrightarrow F2)$ sont des éléments de $\mathcal{F}$

*Exemple* : $(p \vee (\neg q)) \leftrightarrow r$ que l'on écrira en notation partiellement parenthésée $p \vee \neg q \leftrightarrow r$ en utilisant les priorités suivantes : $\neg$, puis $\wedge, \vee$, puis $\rightarrow, \leftrightarrow$

On peut aussi utiliser une notation polonaise $\rightarrow \vee p \neg q\ r$ ou une notation arborescente.

## 2 Sémantique

On définit des fonctions de vérité $f_\neg$, $f_\wedge$, $f_\vee$, $f_\rightarrow$, $f_\leftrightarrow$ sur l'ensemble des valeurs de vérité $\{v, f\}$ ou sur $\{v, f\} \times \{v, f\}$ à valeurs dans $\{v, f\}$ :

| | $f_\neg$ |
|---|---|
| $v$ | $f$ |
| $f$ | $v$ |

| $f_\wedge$ | $v$ | $f$ |
|---|---|---|
| $v$ | $v$ | $f$ |
| $f$ | $f$ | $f$ |

| $f_\vee$ | $v$ | $f$ |
|---|---|---|
| $v$ | $v$ | $v$ |
| $f$ | $v$ | $f$ |

| $f_\rightarrow$ | $v$ | $f$ |
|---|---|---|
| $v$ | $v$ | $f$ |
| $f$ | $v$ | $v$ |

| $f_\leftrightarrow$ | $v$ | $f$ |
|---|---|---|
| $v$ | $v$ | $f$ |
| $f$ | $f$ | $v$ |

Une **valuation** ou **interprétation** $I$ d'un sous-ensemble $V$ de $\mathcal{V}$ est une application de $V$ dans $\{v, f\}$.
*Exemple* de valuation sur $\mathcal{V} = \{p, q, r\}$ : $I(p) = f, I(q) = I(r) = v$

Une **interprétation** $I$ d'une formule $F$ de variables propositionnelles $\{v_1, v_2, ...v_n\}$ est le prolongement d'une interprétation $I$ de $\{v_1, v_2, ...v_n\}$ tel que
- si $F = \neg F_1$, $I(F) = f_\neg(I(F_1))$
- si $F = F1 \wedge F2$ [resp. $F1 \vee F2$, $F1 \rightarrow F2$, $F1 \leftrightarrow F2$],
  $I(F) = f_\wedge(I(F1), I(F2))$ [resp. $f_\vee(I(F1), I(F2))$, $f_\rightarrow(I(F1), I(F2))$, $f_\leftrightarrow(I(F1), I(F2))$ ]

*Exemple* : avec la valuation $I$ précédente, $I(p \vee \neg q \rightarrow r) = v$

Deux formules $F$ et $G$ sont **équivalentes** $(\equiv)$ si et seulement si, pour toute interprétation $I$, on a $I(F) = I(G)$.
*Exemples* : $p \rightarrow q \equiv \neg p \vee q$
$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$
Cette dernière équivalence montre que le $\wedge$ est associatif (d'un point de vue sémantique), on pourra

donc écrire cette formule sans parenthèses, soit $p \wedge q \wedge r$, de même pour le $\vee$, mais pas pour $\rightarrow$ .

On vérifie aussi les lois de Morgan et la distributivité du $\wedge$ par rapport au $\vee$ , et du $\vee$ par rapport au $\wedge$.

Une formule $F$ est une **tautologie** si, pour toute interprétation $I$, $I(F) = v$.
*Exemples* : $p \vee \neg p$
$p \wedge (p \rightarrow q) \rightarrow q$
$(p \vee r) \wedge (q \vee \neg r) \rightarrow p \vee q$

Une formule $F$ est une **antilogie** si, pour toute interprétation $I$, $I(F) = f$.
*Exemples* : $p \wedge \neg p$
$p \wedge (p \rightarrow q) \wedge \neg q$

Si $I(F) = v$, on dit que $I$ **satisfait** $F$.

Une formule $F$ est **conséquence sémantique** d'un ensemble $\Sigma$ de formules si toute interprétation qui satisfait $\Sigma$ (c'est-à-dire qui satisfait toutes les formules de $\Sigma$) satisfait aussi $F$.
*Notation* : $\Sigma \models F$
*Exemples* : $\{p,\ p \rightarrow q\} \models q$
$\{p \vee r,\ q \vee \neg r\} \models p \vee q$

Un ensemble de formules est **satisfaisable** s'il existe au moins une interprétation qui le satisfait.
*Exemple* : $\{p \rightarrow q,\ q \rightarrow \neg p,\ q\}$ est satisfait par $I(p) = f, I(q) = v$

Un ensemble de formules est **insatisfaisable** ou **contradictoire** si aucune interprétation ne le satisfait.
*Exemples* : $\{p, \neg p\}$
$\{p \wedge \neg p\}$
$\{p,\ p \rightarrow q,\ \neg q\}$
$\{p \vee r,\ q \vee \neg r,\ \neg p,\ \neg q\}$
*Notation* : $\Sigma \models f$

**Propriété** : On a $\Sigma \models F$ si et seulement si $\Sigma \cup \{\neg F\}$ est contradictoire.

# 3 Formes normales

Un **littéral** est une formule qui est, soit une variable propositionnelle, soit une négation de variable propositionnelle.
*Exemples* : $p$, $q$, $\neg p$, $\neg q$.

Une formule qui est une disjonction de conjonctions de littéraux est dite sous **forme normale disjonctive (FND)**
*Exemple* : $(p \wedge q) \vee (\neg p \wedge q)$.

Une formule qui est une conjonction de disjonctions de littéraux est dite sous **forme normale conjonctive (FNC)**.
*Exemple* : $(\neg p \vee q) \wedge (p \vee \neg q))$

**Théorème** : Toute formule du Calcul Propositionnel est équivalente à une formule sous forme normale disjonctive.

*Démonstration* par récurrence sur le nombre de variables ou *c*onstructive de la façon suivante :
Soit $\{p_1, p_2, ..., p_n\}$ l'ensemble des variables d'une formule $F$.
Soit $H_i = q_1 \wedge ... \wedge q_n$ avec

- $q_j = p_j$ si dans la ligne $i$ du tableau de vérité, $p_j$ vaut $v$
- $q_j = \neg p_j$ si dans la ligne $i$ du tableau de vérité, $p_j$ vaut $f$

$H_i$ est vraie dans l'interprétation de la ligne $i$ uniquement.
$F$ est équivalente à la FND obtenue en prenant la disjonction des $H_i$ pour tous les $i$ tels que $F$ est vraie à la ligne $i$.

*Exemple* :
$p \wedge ((p \rightarrow q) \rightarrow r)$
qui est vraie si $p, q, r$ prennent les valeurs $v, v, v$ ou $v, f, v$ ou $v, f, f$ est équivalente à
$(p \wedge q \wedge r) \vee (p \wedge \neg q \wedge r) \vee (p \wedge \neg q \wedge \neg r)$
que l'on peut simplifier en
$(p \wedge r) \vee (p \wedge \neg q \wedge \neg r)$
puis en
$(p \wedge r) \vee (p \wedge \neg q)$ .
*Remarquer* la non unicité de la FND.

De même,
**Théorème** : Toute formule du Calcul Propositionnel est équivalente à une formule sous forme normale conjonctive.

*Démonstration* :
On peut prendre la FND de la négation et nier le résultat.
On peut aussi considérer les $G_i = q_1 \vee ... \vee q_n$ avec

- $q_j = \neg p_j$ si dans la ligne $i$ du tableau de vérité, $p_j$ vaut $v$
- $q_j = p_j$ si dans la ligne $i$ du tableau de vérité, $p_j$ vaut $f$

$G_i$ est vraie sauf dans l'interprétation de la ligne $i$.
$F$ est équivalente à la FNC obtenue en prenant la conjonction des $G_i$ pour tous les $i$ tels que $F$ est fausse à la ligne $i$.

*Exemple* :
$p \wedge ((p \rightarrow q) \rightarrow r)$
qui est fausse si $p, q, r$ prennent les valeurs $v, v, f$ ou $f, v, v$ ou $f, v, f$ ou $f, f, v$ ou $f, f, f$ est équivalente à
$(\neg p \vee \neg q \vee r) \wedge (p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee \neg r) \wedge (p \vee q \vee r)$
que l'on peut simplifier en
$(\neg q \vee r) \wedge (p \vee \neg q) \wedge (p \vee q)$
puis en
$(\neg q \vee r) \wedge p$ .

Non unicité de la FNC.

Pour mettre une formule sous FND ou FNC, on peut utiliser la démarche constructive précédente, on peut aussi :

- remplacer $p \leftrightarrow q$ par la formule équivalente $(p \rightarrow q) \wedge (q \rightarrow p)$ ;
- remplacer $p \rightarrow q$ par la formule équivalente $\neg p \vee q$ ;
- faire descendre les négations par les formules de Morgan, le plus à l'intérieur de la formule ;
- appliquer la distributivité du $\wedge$ par rapport au $\vee$, ou l'inverse, jusqu'à obtenir la forme normale cherchée.

*Exemple* :
$p \wedge ((p \rightarrow q) \rightarrow r) \equiv p \wedge (\neg p \vee q \rightarrow r) \equiv p \wedge (\neg(\neg p \vee q) \vee r) \equiv p \wedge ((p \wedge \neg q) \vee r)$
$\equiv (p \wedge \neg q) \vee (p \wedge r)$ (FND)
$\equiv p \wedge (p \vee r) \wedge (\neg q \vee r) \equiv (p \wedge (\neg q \vee r)$ (FNC)

# PRINCIPE DE RESOLUTION
# dans le
# CALCUL PROPOSITIONNEL

Une **clause** est une disjonction de littéraux.
On travaillera maintenant uniquement avec des clauses.

La propriété de FNC s'exprime de la façon suivante :
**Théorème** : Toute formule est équivalente à une conjonction de clauses.

On notera $\mathcal{C}(F)$ un ensemble de clauses dont la conjonction est équivalente à $F$.
*Exemple* : $\mathcal{C}((p \leftrightarrow q) = \{\neg p \vee q,\ p \vee \neg q\}$

Une interprétation **satisfait** $F$ si et seulement si elle satisfait $\mathcal{C}(F)$. En particulier $\mathcal{C}(F)$ est contradictoire si et seulement si $F$ est une antilogie.

## 1 Règle de Résolution

*Notations* : Si $u$ est un littéral d'une clause $C$, on notera $C \setminus u$ la clause dont tous les littéraux sont ceux de $C$ sauf $u$.
*Exemple* : $(p \vee q \vee r) \setminus q = p \vee r$

La clause $C$ est une **résolvante** des clauses $C1$ et $C2$ s'il existe un littéral $u$ tel que
- $u$ est un littéral de $C1$,
- $\neg u$ est un littéral de $C2$,
- $C = (C1 \setminus u) \vee (C2 \setminus \neg u)$

*Exemples* :
- $p \vee r$ et $q \vee \neg r$ donnent la résolvante $p \vee q$
- $p \vee q \vee r$ et $\neg r \vee s \vee \neg t \vee q$ donnent $p \vee q \vee s \vee \neg t$
- $p \vee q \vee r$ et $\neg p \vee \neg q \vee s$ donnent $q \vee r \vee \neg q \vee s$ et $p \vee r \vee \neg p \vee s$
  Attention, elles ne donnent **pas** $r \vee s$
- $\neg p \vee q$ et $p$ donnent $q$
- $\neg p$ et $p$ donnent la clause vide notée $\square$.

Une **preuve par Résolution** de la clause $F$ à partir de l'ensemble de clauses $\mathcal{C}$ est une suite finie de clauses $C_1, C_2, ...C_n$ telles que $C_i \in \mathcal{C}$ ou bien $C_i$ est une résolvante de deux $C_j$ prédédents, et $C_n = F$.
*Notation* : $\mathcal{C} \vdash F$
*Exemple* : $\{p,\ \neg p \vee q,\ \neg q \vee r\} \vdash r$ en prenant

$C_1 = p$ , $C_2 = \neg p \vee q$, dans l'ensemble de départ,
$C_3 = q$ est une résolvante de $C_1$ et $C_2$,
$C_4 = \neg q \vee r$, dans l'ensemble de départ,
$C_5 = r$ est une résolvante de $C_3$ et $C_4$.

Une **réfutation** d'un ensemble de clauses $\mathcal{C}$ est une preuve de la clause vide à partir de $\mathcal{C}$.
*Notation* : $\mathcal{C} \vdash \square$
*Remarque* : $\mathcal{C} \vdash F$ implique $\mathcal{C} \cup \mathcal{C}(F) \vdash \square$ mais $\mathcal{C} \cup \mathcal{C}(F) \vdash \square$ n'implique **pas** $\mathcal{C} \vdash F$
*Contrexemple* : $\mathcal{C} = \{p, q\}$, $F = p \vee q$

*Exemple de réfutation* :

| de l'ensemble de clauses | | on obtient successivement | | |
|---|---|---|---|---|
| | (1) $\neg p \vee \neg q \vee r$ | | (5) $\neg p \vee \neg q$ | (1) et (4) |
| | (2) $\neg p \vee q$ | | (6) $q$ | (2) et (3) |
| | (3) $p$ | | (7) $\neg p$ | (5) et (6) |
| | (4) $\neg r$ | | (8) $\square$ | (3) et (7) |

# 2 Validité de la règle de Résolution

**Propriété** : La règle de Résolution est **saine** (sound), c'est-à-dire toute résolvante est conséquence sémantique des clauses dont elle est issue.

**Corollaire 1** : Toute clause $F$ déduite d'un ensemble $\mathcal{C}$ de clauses par Résolution est conséquence sémantique de $\mathcal{C}$.
Si $\mathcal{C} \vdash F$ alors $\mathcal{C} \models F$

**Corollaire 2** : Un ensemble de clauses admettant une réfutation est contradictoire.
Si $\mathcal{C} \vdash \square$ alors $\mathcal{C} \models f$

# 3 Complétude de la méthode de Résolution

**Théorème** : La méthode de Résolution est **complète** pour la réfutation, c'est-à-dire, si $\mathcal{C}$ est un ensemble de clauses contradictoire, alors il existe une réfutation de $\mathcal{C}$.
*Remarque* : Elle n'est pas complète pour la déduction.

*En résumé* : on a
$$\begin{array}{ccc} \mathcal{C} \vdash F & \Rightarrow & \mathcal{C} \models F \\ \Downarrow & & \Updownarrow \\ \mathcal{C} \cup \mathcal{C}(\neg F) \vdash \square & \Leftrightarrow & \mathcal{C} \cup \mathcal{C}(\neg F) \models f \end{array}$$

*Remarque* : la méthode est non déterministe, le théorème dit que la réfutation existe, il ne dit pas comment la trouver, ni si la recherche se termine.

# 4 Quelques stratégies

## 4.1 Résolution positive

On n'autorise la Résolution qu'entre deux clauses dont l'une est *positive* (c'est-à-dire n'a aucun littéral négatif)

*Remarque* : Il y a au moins une clause positive dans un ensemble contradictoire. (Sinon l'ensemble serait satisfait par l'interprétation toujours fausse.)

*Propriété* : La Résolution positive est complète pour la réfutation.

*Exemple de réfutation positive* :

| de l'ensemble de clauses | | on obtient successivement | | |
|---|---|---|---|---|
| | (1) $\neg p \vee \neg q \vee r$ | | (5) $\neg q \vee r$ | (1) et (3) |
| | (2) $\neg p \vee q$ | | (6) $q$ | (2) et (3) |
| | (3) $p$ | | (7) $r$ | (5) et (6) |
| | (4) $\neg r$ | | (8) $\square$ | (4) et (7) |

## 4.2 Résolution négative

On n'autorise la Résolution qu'entre deux clauses dont l'une est négative (c'est-à-dire n'a aucun littéral positif)
*Remarque* : Il y a au moins une clause négative dans un ensemble contradictoire. (Sinon l'ensemble serait satisfait par l'interprétation toujours vraie.)

*Propriété* : La Résolution négative est complète pour la réfutation.

*Exemple de réfutation négative* :

| de l'ensemble de clauses | | on obtient successivement | | |
|---|---|---|---|---|
| | (1) $\neg p \vee \neg q \vee r$ | | (5) $\neg p \vee \neg q$ | (1) et (4) |
| | (2) $\neg p \vee q$ | | (6) $\neg p$ | (2) et (5) |
| | (3) $p$ | | (7) $\square$ | (3) et (6) |
| | (4) $\neg r$ | | | |

## 4.3 Résolution linéaire et Résolution linéaire par entrée

Une preuve par Résolution *linéaire* d'une clause $F$ à partir d'un ensemble $\mathcal{C}$ de clauses est une suite $C_0$, $C_1$, ..., $C_n$ telle que $C_0 \in \mathcal{C}$ et pout tout $i$, $C_i$ est une résolvante de $C_{i-1}$ et d'une clause de $\mathcal{C}$ ou d'un $C_j$ précédent.
La preuve est *linéaire par entrée* si on prend toujours une clause de $\mathcal{C}$ (clause d'entrée).

*Exemple de réfutation linéaire par entrée* :

| de l'ensemble de clauses | | on obtient successivement | | |
|---|---|---|---|---|
| | (1) $\neg p \vee \neg q \vee r$ | | (5) $q$ | (2) et (3) |
| | (2) $\neg p \vee q$ | | (6) $\neg p \vee r$ | (5) et (1) |
| | (3) $p$ | | (7) $r$ | (6) et (3) |
| | (4) $\neg r$ | | (8) $\square$ | (7) et (4) |

*Propriété* : La Résolution linéaire est complète pour la réfutation. La Résolution linéaire par entrée n'est pas complète.

*Contrexemple* :

| de l'ensemble de clauses | | on obtient la réfutation linéaire | | |
|---|---|---|---|---|
| | (1) $p \vee q$ | | (5) $p$ | (1) et (2) |
| | (2) $p \vee \neg q$ | | (6) $q$ | (5) et (3) |
| | (3) $\neg p \vee q$ | | (7) $\neg p$ | (6) et (4) |
| | (4) $\neg p \vee \neg q$ | | (8) $\square$ | (7) et (5) |

On ne peut avoir de réfutation linéaire par entrée car les clauses d'entrée ont toutes deux littéraux et ne peuvent donner directement la clause vide

Une **clause de Horn** est une clause qui a au plus un littéral positif. Les clauses négatives sont des clauses de Horn.

*Propriété* : La Résolution linéaire par entrée est complète pour la réfutation des clauses de Horn.

Les clauses du premier exemple étaient des clauses de Horn, on était donc assuré de l'existence d'une réfutation linéaire par entrée. Celles du deuxième exemple n'étaient ps toutes des clauses de Horn.

### 4.4 Stratégie de l'ensemble support

Soit $T$ un sous-ensemble de $\mathcal{C}$ tel que $\mathcal{C} \setminus T$ soit satisfaisable et qui sera appelé l'*ensemble support*. Dans la stratégie de l'ensemble support, on interdit la Résolution entre deux clauses de $\mathcal{C} \setminus T$, c'est-à-dire on n'autorise la Résolution qu'entre deux clauses dont l'une au moins est une clause de $T$ ou est une descendante d'une clause de $T$.

*Exemple de stratégie de l'ensemble support*, avec T $= \{p, \neg r\}$

| de l'ensemble de clauses | | on obtient successivement | | |
|---|---|---|---|---|
| | (1) $\neg p \vee \neg q \vee r$ | | (5) $q$ | (3) et (2) |
| | (2) $\neg p \vee q$ | | (6) $\neg p \vee \neg q$ | (4) et (1) |
| | (3) $p$ | | (7) $\neg p$ | (5) et (6) |
| | (4) $\neg r$ | | (8) $\square$ | (3) et (7) |

*Propriété* : La stratégie de l'ensemble support est complète pour la réfutation.

*Intérêt* : Elle permet de se focaliser sur un sous-ensemble de clauses qui joue un rôle particulier.

## 5 Application

Pour montrer qu'une formule $F$ est une tautologie, on cherchera une réfutation de $\mathcal{C}(\neg F)$.
En effet :

$F$ tautologie $\Leftrightarrow \neg F$ antilogie $\Leftrightarrow \mathcal{C}(\neg F)$ contradictoire $\Leftrightarrow \mathcal{C}(\neg F) \vdash \square$

*Exemple* :
Soit $F = (p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$
alors

$\neg F \equiv (\neg p \vee \neg q \vee r) \wedge (\neg p \vee q) \wedge p \wedge \neg r$
$\mathcal{C}(\neg F) = \{\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p,\ \neg r\}$
c'est l'exemple étudié en 4, on a vu que
$\mathcal{C}(\neg F) \vdash \square$
donc $F$ est une tautologie.

# CALCUL DES PREDICATS DU PREMIER ORDRE

Par rapport au Calcul propositionnel, on rajoute des variables, des quantificateurs, des relations et des fonctions.

## 1 Définition du langage

On a
- l'ensemble de connecteurs $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ et les quantificateurs $\{\forall, \exists\}$
- un ensemble $\mathcal{C}$ fini ou dénombrable de constantes ;
- un ensemble $\mathcal{F}$ fini ou dénombrable de symboles de fonctions et une application, appelée arité, de $\mathcal{F}$ dans $\mathbb{N}^*$ ;
- un ensemble $\mathcal{R}$ fini ou dénombrable de symboles de relations et une application, appelée arité, de $\mathcal{R}$ dans $N*$ ;
- un ensemble $\mathcal{V}$ dénombrable de variables.

Un **terme** est
- ou bien un élément de $\mathcal{C} \cup \mathcal{V}$ ;
- ou bien de la forme $f(t_1, ..., t_n)$ où $f \in \mathcal{F}$ est d'arité $n$ et $t_1, ..., t_n$ sont des termes.

*Exemples* : $t = g(y, f(h(x, x)))$
$t' = c$

où $c$ est une constante,
$f$ un symbole fonctionnel unaire,
$g$ et $h$ des symboles fonctionnels binaires.

Une **formule atomique** est de la forme $R(t_1, ..., t_n)$ où $R \in \mathcal{R}$, d'arité $n$, et $t_1, ..., t_n$ sont des termes.

Une **formule du premier ordre** est
- ou bien une formule atomique ;
- ou bien de la forme $\neg A$ où $A$ est une formule ;
- ou bien de la forme $A1 \wedge A2$, $A1 \vee A2$, $A1 \rightarrow A2$ ou $A1 \leftrightarrow A2$ où $A1$ et $A2$ sont des formules ;
- ou bien de la forme $\forall x A$ où $A$ est une formule ;
- ou bien de la forme $\exists x A$ où $A$ est une formule.

*Exemple* : $\forall y(R(y, c) \rightarrow \exists x(R(t, c) \wedge R(c, t)))$
où $c$ est une constante,
$R$ une relation binaire
et $t$ le terme précédent,
soit $\forall y(R(y, c) \rightarrow \exists x(R(g(y, f(h(x, x))), c) \wedge R(c, g(y, f(h(x, x))))))$

## 2 Sémantique

Une **interprétation** $I$ se compose de

- un ensemble $D_I$, non vide, appelé **domaine** de $I$ ;
- pour chaque constante $c$, un élément de $D_I$, noté $I(c)$ ou $c^I$, appelé interprétation de $c$ ;
- pour chaque symbole fonctionnel $f$ d'arité $n$, une application de ${D_I}^n$ dans $D_I$, notée $I(f)$ ou $f^I$, appelée interprétation de $f$ ;
- pour chaque symbole de relation $R$ d'arité $n$, une application de ${D_I}^n$ dans $\{v, f\}$, notée $I(R)$ ou $R^I$, appelée interprétation de $R$.

Une **assignation** $\sigma$ est une application d'un sous-ensemble fini de $\mathcal{V}$ dans $D_I$.

L'**interprétation d'un terme** $t$, pour une assignation $\sigma$, notée $I_\sigma(t)$ ou $t^{I_\sigma}$, est définie de la façon suivante :

- si $t$ est une constante $c$, $t^{I_\sigma} = c^I$
- si $t$ est une variable $x$, $t^{I_\sigma} = \sigma(x) = x^\sigma$
- si $t = f(t_1, t_2, ...t_n)$, $t^{I_\sigma} = f^I({t_1}^{I_\sigma}, {t_2}^{I_\sigma}, ...{t_n}^{I_\sigma})$

L'**interprétation d'un formule** $F$, pour une assignation $\sigma$ contenant toutes les variables libres de $F$, notée $I_\sigma(F)$ ou $F^{I_\sigma}$, est définie de la façon suivante :

- si $F$ est une formule atomique $R(t_1, t_2, ...t_n)$, $F^{I_\sigma} = R^I(t_1^{I_\sigma}, t_2^{I_\sigma}, ...t_n^{I_\sigma})$ ;
- si $F = \neg F1$, $F^{I_\sigma} = f_\neg(F1^{I_\sigma})$ ;
- si $F = F1 \wedge F2$ [resp. $F1 \vee F2$, $F1 \rightarrow F2$, $F1 \leftrightarrow F2$],
  $F^{I_\sigma} = f_\wedge(F1^{I_\sigma}, F2^{I_\sigma})$ [resp. $f_\vee(F1^{I_\sigma}, F2^{I_\sigma}), f_\rightarrow(F1^{I_\sigma}, F2^{I_\sigma}), f_\leftrightarrow(F1^{I_\sigma}, F2^{I_\sigma})$] ;
- si $F = \exists x F1$,
  . $F^{I_\sigma} = v$ s'il existe un élément $a$ de $D_I$ tel que $F^{I_{\sigma_a}} = v$ avec $\sigma_a$ égale à $\sigma$ sauf en $x$ où $\sigma_a(x) = a$ ;
  . $F^{I_\sigma} = f$ sinon ;
- si $F = \forall x F1$,
  . $F^{I_\sigma} = v$ si pour tout élément $a$ de $D_I$ on a $F^{I_{\sigma_a}} = v$ ;
  . $F^{I_\sigma} = f$ sinon.

*Exemple* : Soit $F$ la formule précédente
$\forall y(R(y,c) \rightarrow \exists x(R(t,c)\ R(c,t)))$ avec $t = g(y, f(h(x,x)))$.
Soit $I$ l'interprétation de domaine l'ensemble des réels,
où $c$ est interprétée par 0,
$f$ par l'opposé dans les réels,
$g$ et $h$ par l'addition et la multiplication dans les réels,
$R$ par la relation $\geq$,
alors $t^\sigma = y^\sigma - (x^\sigma)^2$
$R(t,c)^{I_\sigma} = (y^\sigma - (x^\sigma)^2 \geq 0)$
$(R(t,c) \wedge R(c,t))^{I_\sigma} = (y^\sigma - (x^\sigma)^2 = 0)$
$R(y,c)^{I_\sigma} = (y^\sigma \geq 0)$
$(R(y,c) \rightarrow \exists x(R(t,c) \wedge R(c,t)))^{I_\sigma} = v$
car si $y^\sigma$ est $\geq 0$, il existe un réel $r$ tel que $y^{\sigma_r} = r^2$
$(\forall y(R(y,c) \rightarrow \exists x(R(t,c) \wedge R(c,t))))^{I_\sigma} = F^{I_\sigma} = v$
car pour tout réel $s, (R(y,c) \rightarrow \exists x(R(t,c) \wedge R(c,t)))^{I_{\sigma_s}} = v$

Si on prend comme domaine l'ensemble des entiers ou des rationnels, et les autres interprétations inchangées (dans les entiers ou les rationnels),

alors $(R(y,c) \rightarrow \exists x(R(t,c) \wedge R(c,t)))^{I_\sigma} = \begin{cases} f & \text{si } y^\sigma \text{ est } \geq 0 \text{ et n'est pas un carré parfait} \\ v & \text{sinon} \end{cases}$

$(\forall y(R(y,c) \rightarrow \exists x(R(t,c) \wedge R(c,t))))^{I_\sigma} = F^{I_\sigma} = f$

Une **formule close** (c'est-à-dire sans variable libre) a une interprétation indépendante de toute assignation et sera notée $I(F)$ ou $F^I$. C'est le cas de la formule $F$ précédente.

On dit que $I$ **satisfait** $F$ si, pour toute assignation $\sigma$, $F^{I_\sigma} = v$.

Si les variables libres de $F$ sont $x_1$, $x_2$, ... $x_n$, la formule $\forall x_1 \forall x_2 ... \forall x_n F$ est close et est appelée **cloture universelle** de $F$.

Une formule est **valide** si elle est satisfaite par toute interprétation.
*Exemple* : $\forall x(A(x) \rightarrow B(x)) \wedge A(a) \rightarrow B(a)$

Deux formules $F$ et $G$ sont **équivalentes** ($\equiv$) si et seulement si $F \leftrightarrow G$ est valide, c'est-à-dire si pour toute $I$ et toute $\sigma$, $F^{I_\sigma} = G^{I_\sigma}$
*Exemple* : $\forall x(A(x) \wedge B(x)) \equiv \forall x A(x) \wedge \forall x B(x)$
*Contrexemple* : $\forall x(A(x) \vee B(x))$ et $\forall x A(x) \vee \forall x B(x)$ ne sont pas équivalentes

# 3 Formules prénexes

Une formule $F$ est **prénexe** si elle est de la forme $Q_1 x_1 Q_2 x_2 ... Q_n x_n G$ avec $G$ sans quantificateur et chaque $Q_i$ étant soit $\exists$ soit $\forall$.

**Théorème** : Toute formule est équivalente à une formule prénexe.

*Démonstration* constructive par une suite de transformations ayant pour effet de pousser tous les quantificateurs à l'extérieur et utilisant les propriété suivantes.

Pour toute formule $F$, toute variable $x$ et toute formule $G$ dans laquelle $x$ n'a pas d'occurrence libre, si $Q$ est un quantificateur quelconque et $Q'$ l'autre quantificateur, on a :

$\neg Qx F \equiv Q'x \neg F$
$Qx F \wedge G \equiv Qx(F \wedge G)$
$G \wedge Qx F \equiv Qx(G \wedge F)$
$Qx F \vee G \equiv Qx(F \vee G)$
$G \vee Qx F \equiv Qx(G \vee F)$
$G \rightarrow Qx F \equiv Qx(G \rightarrow F)$
$Qx F \rightarrow G \equiv Q'x(F \rightarrow G)$ (en effet $Qx F \rightarrow G \equiv \neg Qx F \vee G \equiv Q'x \neg F \vee G$
$\equiv Q'x(\neg F \vee G) \equiv Q'x(F \rightarrow G)$ )

*Remarque* : Il peut ne pas y avoir unicité.

*Exemples* :
$\exists x P(x) \wedge \forall y Q(y) \equiv \exists x \forall y(P(x) \wedge Q(y)) \equiv \forall y \exists x(P(x) \wedge Q(y))$
Attention $\exists x \forall y(P(x,y))$ n'est pas en général équivalente à $\forall y \exists x(P(x,y))$
$\exists x P(x) \rightarrow \forall y Q(y) \equiv \forall x \forall y(P(x) \rightarrow Q(y))$

# 4 Formules normales

Une **clause** est une disjonction finie de formules atomiques ou de négations de formules atomiques. Si $F$, sans quantificateurs, est une conjonction finie de clauses, alors la formule $\mathrm{Q}x_1 \mathrm{Q}x_2 ... \mathrm{Q}x_n F$ est dite sous **forme normale conjonctive (FNC)**.

**Théorème** : Toute formule est équivalente à une formule prénexe sous FNC.

*Démonstration* : utiliser le résultat analogue du Calcul Propositionnel et le théorème précédent.

Dans la pratique, on élimine les $\leftrightarrow$ et les $\leftrightarrow$, on fait descendre les négations, on supprime les doubles négations, on fait monter les conjonctions et les quantificateurs.

# 5 Skolémisation

On rajoute au langage une infinité de constantes et, pour chaque $n$, une infinité de symboles de fonctions d'arité $n$, appelées **constantes** et **fonctions de Skolem**.

Une **transformation de Skolem** associe à une formule prénexe close de la forme
$\forall x_1 \forall x_2 ... \forall x_n \exists x F$
la formule
$\forall x_1 \forall x_2 ... \forall x_n F(x|f(x_1, x_2, ..., x_n))$
où
$F(x|f(x_1, x_2, ..., x_n))$
est la formule obtenue à partir de $F$ en remplaçant toutes les occurences de $x$ par le terme $f(x_1, x_2, ..., x_n)$ où $f$ est une nouvelle fonction de Skolem d'arité $n$.

*Cas particulier* : la formule
$\exists x F$
est transformée en
$F(x|c)$
qui est la formule obtenue à partir de $F$ en remplaçant toutes les occurences de $x$ par la constante de Skolem $c$.

On appelle **skolémisée** de $F$ une formule universelle obtenue en appliquant itérativement des transformations de Skolem jusqu'à de qu'il n'y ait plus de $\exists$.

*Exemple* : la formule
$\exists x \forall y \exists z \forall u \exists v P(x, y, z, u, v)$
est skolémisée en
$\forall y \forall u P(c, y, f(y), u, g(y, u))$
où $f$ et $g$ sont des fonctions de Skolem d'arités respectives 1 et 2 et $c$ une constante de Skolem.

*Remarques* : Il n'y a pas unicité et la skolémisée $F'$ d'une fonction $F$ ne lui est pas équivalente. Mais $F$ est conséquence sémantique de $\{F'\}$ ($\{F'\} \models F$) et on a $\models F' \rightarrow F$.

*Exemple* :
$\exists x P(x) \wedge \forall y Q(y)$
se skolémise en
$\forall y (P(a) \wedge Q(y))$
ou en
$\forall y (P(f(y)) \wedge Q(y))$
qui ne lui sont pas équivalentes.

Mais on a le
**Théorème** : Soient $\Sigma$ un ensemble de formules prénexes closes et $\Sigma$' obtenu en skolémisant les formules de $\Sigma$. Alors $\Sigma$ est satisfaisable (resp. contradictoire) si et seulement si $\Sigma'$ l'est.

*Esquisse de démonstration* :
Si une interprétation $I$ satisfait $\Sigma'$, elle satisfait aussi $\Sigma$.
Si une interprétation $I$ satisfait $\Sigma$, on construit une interprétation $I'$ qui satisfait $\Sigma'$.

# 6 Unification

Une substitution $\sigma$ est une application de l'ensemble des variables dans l'ensembles des termes telle que $\sigma$ est égale à l'identité sauf pour une nombre fini de variables $\{x_1, x_2, ...x_n\}$ et sera notée $(x_1|t_1, x_2|t_2, ..., x_n|t_n)$.

On prolonge en une application de l'ensemble des termes (puis des formules) dans eux-même.
*Exemple* : avec $\sigma = (x|a, y|f(x))$,
$R(f(x), f(y))^\sigma = R(f(a), f(f(x)))$

Le produit $\tau_\circ\sigma$ de deux substitutions $\sigma$ et $\tau$ est la substitution obtenue en applicant $\sigma$ puis $\tau$.
*Exemple* : si $\sigma = (y|f(x))$ et $\tau = (x|a)$,
$R(f(x), f(y))^{\tau_\circ\sigma} = R(f(a), f(f(a)))$
$R(f(x), f(y))^{\sigma_\circ\tau} = R(f(a), f(f(x)))$

Un ensemble de littéraux $\{l_1, l_2, ...l_n\}$ est **unifiable** s'il existe une substitution $\sigma$ telle que
$\sigma(l_1) = \sigma(l_2) = ... = \sigma(l_n)$.
$\sigma$ est appelé unificateur de $\{l_1, l_2, ..., l_n\}$.

Un unificateur $\sigma$ est **principal** si tout autre unificateur est de la forme $\tau_\circ\sigma$ pour une substitution convenable $\tau$. Le résultat de l'application d'un tel unificateur est appelé **facteur**.

*Exemple* :
$\{R(x, g(x)), R(f(z), y), R(x, u)\}$ est unifiable par
$\tau = (x|f(f(c)), y|g(f(f(c))), z|f(c), u|g(f(f(c))))$, unificateur non principal
( facteur $R(f(f(c)), g(f(f(c))))$ )
et par
$\sigma = (x|f(z), y|g(f(z)), u|g(f(z))))$, unificateur principal.
( facteur $R(f(z), g(f(z)))$ )
On a $\tau = (z|f(c)_\circ\sigma$.

*Propriété* : L'unificateur principal, s'il existe, est unique à un changement de variables près.

**Algorithme d'unification** de deux littéraux $l_1$ et $l_2$ :

On cherche, dans un parcours préfixe, la position du premier symbole qui n'est pas le même dans les deux littéraux.
- Si les deux symboles correspondants sont des constantes ou des symboles fonctionnels (différents), il y a échec.
- Sinon l'un au moins est une variable $x$.
  Soit $t$ le terme commençant à cette position dans l'autre littéral.
  . Si $x$ apparait dans $t$, il y a aussi échec ("test d'occurence"),
  . sinon on considère la substitution $\sigma = (x|t)$ et on réitère le processus avec $\sigma(l_1)$ et $\sigma(l_2)$ jusqu'à ce que l'on obtienne deux littéraux identiques.

On généralise facilement à un ensemble de plus de deux littéraux.

**Théorème** : L'algorithme d'unification se termine et conduit, si on l'applique à deux littéraux unifiables, à une suite de substitutions $\sigma_1, \sigma_2, ..., \sigma_k$ dont le produit $\sigma_k \circ ... \circ \sigma_1$ est un unificateur principal et le dernier littéral obtenu un facteur des littéraux initiaux.

*Exemple* :

| littéraux | | | substitution |
|---|---|---|---|
| $R(\underline{x}, g(c,z))$ | et | $R(\underline{f(y)}, g(y,k(x))$ | $(x\|f(y))$ |
| $R(f(y), g(\underline{c},z))$ | et | $R(\underline{f(y)}, g(\underline{y},k(f(y)))$ | $(y\|c)$ |
| $R(f(c), g(c,\underline{z}))$ | et | $R(f(c), g(\underline{c},\underline{k(f(c))})$ | $(z\|k(f(c)))$ |
| $R(f(c), g(c,k(f(c))))$ | | | $(x\|f(c)), (y\|c), (z\|k(f(c)))$ |
| facteur principal | | | unificateur principal |

*Autre exemple* :

| littéraux | | | substitution |
|---|---|---|---|
| $R(\underline{x}, f(x))$ | et | $R(\underline{f(y)}, y)$ | $(x\|f(y))$ |
| $R(f(y), \underline{f(f(y))})$ | et | $R(f(y), \underline{y})$ | échec (test d'occurence) |

*Autre exemple* :

| littéraux | | | |
|---|---|---|---|
| $P(\underline{f(x)})$ | et | $P(\underline{c})$ | échec ($c$ est une constante) |

# PRINCIPE DE RESOLUTION dans le CALCUL DES PREDICATS du PREMIER ORDRE

## 1 Règle de Résolution

On a vu que toute formule universelle est équivalente à une formule $\forall x_1 \forall x_2 ... \forall x_n (F_1 \wedge ... \wedge F_k)$ où chaque $F_i$ est une disjonction de clauses.
Cette formule est aussi équivalente à $\forall x_1 \forall x_2 ... \forall x_n F_1 \wedge ... \wedge \forall x_1 \forall x_2 ... \forall x_n F_n$.

On remplacera un ensemble de formules closes universelles par un ensemble de clauses où il est implicite que les variables sont quantifiées universellement.

*Notations* : Si $S$ est un sous-ensemble de l'ensemble des littéraux de la clause $C$, on notera $C \setminus S$ la clause dont les littéraux sont ceux de $C$ sauf les éléments de $S$.

*Exemple* : $P(x) \vee Q(y) \vee R(x,y) \vee \neg T(z) \setminus \{Q(y), \neg T(z)\} = P(x) \vee R(x,y)$

Si $S$ est un ensemble de littéraux, on notera $\neg S$ l'ensemble $\{\neg l \mid l \in S\}$.
*Exemple* : $\neg\{Q(y), \neg T(z)\} = \{\neg Q(y), T(z)\}$

La clause $C$ est une **résolvante** des clauses $C1$ et $C2$ s'il existe des clauses $C'1$ et $C'2$, des ensembles de littéraux $S1$ et $S2$ et une substitution $\sigma$ tels que
- $C'1$ et $C'2$ sont des variantes de $C1$ et $C2$ qui n'ont pas de variable en commun (une variante d'une formule est une formule qui lui est identique à un renommage des variables près) ;
- $S1$ et $S2$ sont des sous-ensembles des ensembles de littéraux de $C'1$ et $C'2$ respectivement ;
- $S1 \cup \neg S2$ est unifiable et $\sigma$ en est un unificateur principal ;
- $C = \sigma((C'1 \setminus S1) \cup (C'2 \setminus S2))$

*Exemples* :
$\underline{\neg P(x)} \vee Q(x)$
et
$\underline{P(a)}$
où $a$ est une constante, donnent la résolvante
$Q(a)$

$\underline{\neg P(f(x), y)} \vee Q(x,y)$
et
$\underline{P(x, f(y))} \vee R(x,y)$ réécrit en $\underline{P(x', f(y'))} \vee R(x', y')$
donnent
$Q(x, f(y')) \vee R(f(x), y')$
avec $\sigma = (x'|f(x), y|f(y'))$

$\underline{\neg P(f(x), y)} \vee Q(x, y) \vee \underline{\neg P(z, z)}$
et
$P(x, f(y)) \vee R(x, y) \vee P(z, z)$ réécrit $\underline{P(x', f(y'))} \vee R(x', y') \vee \underline{P(z', z')}$
donnent
$Q(x, f(x)) \vee R(f(x), x)$
avec $\sigma = (x'|f(x), y|f(x), z|f(x), y'|x, z'|f(x))$

On généralise au Calcul des Prédicats les notions de **preuve** et **réfutation** vues dans le Calcul propositionnel.

## 2 Validité et complétude

On a les mêmes résultats que dans le Calcul propositionnel : la méthode de Résolution est **saine** et **complète** pour la réfutation dans le Calcul des Prédicats.

Pour *démontrer* la complétude, à partir d'un ensemble contradictoire de clauses du Calcul des prédicats du premier ordre, on construit un ensemble contradictoire de clauses du Calcul Propositionnel qui a une réfutation à partir de laquelle on construit une réfutation de l'ensemble initial de clauses grâce à un théorème connu sous le nom de "lemme de relèvement".

## 3 Stratégies

On définit, comme dans le Calcul propositionnel, les stratégies positive, négative, linéaire, linéaire par entrée, de l'ensemble support. On a les mêmes résultats de complétude.

## 4 Applications

**4.1** Pour montrer qu'une formule $F$ est valide, on cherchera une réfutation de $\mathcal{C}(\neg F)$.

En effet :
$F$ valide $\Leftrightarrow \mathcal{C}(\neg F)$ contradictoire $\Leftrightarrow \mathcal{C}(\neg F) \vdash \square$

*Exemple* :
Soit $F = \forall x(A(x) \rightarrow B(x)) \wedge \forall x(B(x) \rightarrow C(x)) \rightarrow \forall x(A(x) \rightarrow C(x))$

$\neg F \equiv \forall x(A(x) \rightarrow B(x)) \wedge \forall x(B(x) \rightarrow C(x)) \wedge \exists x(A(x) \wedge \neg C(x))$
mise sous forme prénexe :
$\neg F \equiv \exists z \forall x \forall y((A(x) \rightarrow B(x)) \wedge (B(y) \rightarrow C(y)) \wedge A(z) \wedge \neg C(z))$
skolémisation :
$\forall x \forall y((A(x) \rightarrow B(x)) \wedge (B(y) \rightarrow C(y)) \wedge A(a) \wedge \neg C(a))$
où $a$ est une constante de Skolem.

clauses :

| | |
|---|---|
| (1) | $\neg A(x) \vee B(x)$ |
| (2) | $\neg B(y) \vee C(y)$ |
| (3) | $A(a)$ |
| (4) | $\neg C(a)$ |

réfutation positive et linéaire par entrée :

| | | |
|---|---|---|
| (5) | $B(a)$ | (1) et (3) |
| (6) | $C(a)$ | (2) et (5) |
| (7) | $\square$ | (4) et (6) |

réfutation négative et linéaire par entrée :

| | | |
|---|---|---|
| (5') | $\neg B(a)$ | (2) et (4) |
| (6') | $\neg A(a)$ | (1) et (5') |
| (7') | $\square$ | (3) et (6') |

stratégie de l'ensemble support avec $\{A(a), \neg C(a)\}$ comme ensemble support, c'est-à-dire les clauses issues de la négation de la conclusion $\forall x(A(x) \rightarrow C(x))$ :

| | |
|---|---|
| (5") $B(a)$ | (3) et (1) |
| (6") $\neg B(a)$ | (4) et (2) |
| (7") $\square$ | (5") et (6") |

**4.2** Pour montrer $\Sigma \models F$ , on montrera $\mathcal{C}(\Sigma \cup \{\neg F\}) \vdash \square$

En effet :
$\Sigma \models F \Leftrightarrow \mathcal{C}(\Sigma \cup \{\neg F\})$ contradictoire $\Leftrightarrow \mathcal{C}(\Sigma \cup \{\neg F\}) \vdash \square$

*Exemple* :
Soit $\Sigma = \{\forall x \exists y P(x,y), \forall x \forall y(\exists z(P(x,z) \wedge P(z,y)) \rightarrow Q(x,y))\}$
et $F = \forall x \exists y Q(x,y)$
$\neg F = \exists x \forall y \neg Q(x,y)$

clauses :

| | |
|---|---|
| (1) $P(x, f(x))$ où $f$ est une fonction de Slolem | |
| (2) $\neg P(x,z) \vee \neg P(z,y)) \vee Q(x,y)$ | |
| (3) $\neg Q(a,y)$ | |

réfutation positive et linéaire par entrée :

| | |
|---|---|
| (4) $\neg P(f(x),y) \vee Q(x,y)$ | (1) et (2) |
| (5) $Q(x, f(f(x)))$ | (1') : $P(x', f(x'))$ et (4) |
| (6) $\square$ | (3) et (5) |

réfutation négative, linéaire par entrée, et c'est aussi la stratégie de l'ensemble support avec $\{\neg Q(a,y)\}$ comme ensemble support c'est-à-dire la clause issue de la négation du *théorème* $F$ dans la *théorie* $\Sigma$ :

| | |
|---|---|
| (4') $\neg P(a,z) \vee \neg P(z,y)$ | (2) et (3) |
| (5') $\neg P(f(a),y)$ | (1) et (4') |
| (6') $\square$ | (1) et (5') |

# PROBLEMES LIES A L'EGALITE
# DEMODULATION
# PARAMODULATION

A partir des connaissances suivantes :

Alain est le père de Bernard.
Bernard est le père de Claude.
Le père du père d'une personne est son grand-père parternel.
Claude joue avec son grand-père parternel.

Peut-on démontrer, en utilisant le Principe de résolution que

Claude joue avec Alain ?

## 1 Formalisation en notation prédicative

A partir des énoncés suivants :

est-pere-de$(a, b)$
est-pere-de$(b, c)$
$\forall x($ est-pere-de$(y, z) \rightarrow$ est-grand-pere-de$(x, z))$
$\forall x($ est-grand-pere-de$(x, c) \rightarrow$ joue$(c, x))$

où $a$, $b$, $c$ sont des constantes et $x$, $y$, $z$ des variables, on peut démontrer facilement que

joue$(c, a)$

en montrant que l'ensemble de clauses[1] suivant est contradictoire.

(1.1) est-pere-de$(a, b)$
(1.2) est-pere-de$(b, c)$
(1.3) est-pere-de$(y, z) \rightarrow$ est-grand-pere-de$(x, z)$
(1.4) est-grand-pere-de$(x, c) \rightarrow$ joue$(c, x)$
(1.5) $\neg$joue$(c, a)$ [négation du fait à démontrer]

| | | |
|---|---|---|
| Réfutation : | (1.6) est-grand-pere-de$(a, c)$ | (1.1), (1.2) et (1.3) |
| | (1.7) joue$(c, a)$ | (1.6) et (1.4) |
| | (1.8) $\square$ | (1.7) et (1.5) |

Remarquer que l'on a, dans cet exemple, non seulement une réfutation, mais aussi une déduction directe de la propriété joue$(c, a)$ (clause (1.7)).

## 2 Formalisation en notation fonctionnelle

A la place des prédicats on utilise des symboles fonctionnels et on a besoin aussi d'un prédicat *egal* que l'on notera ici $=$ mais qui ne possède aucune propriété implicite de l'égalité.

[1] Pour plus de lisibilité dans les applications en mettant en évidence les littéraux *positifs* et les littéraux *négatifs*, on écrit souvent les clauses de la forme
$\neg A_1 \vee ... \vee \neg A_n \vee B_1 \vee ... \vee B_p$
sous la forme
$A_1 \wedge ... \wedge A_n \rightarrow B_1 \vee ... \vee B_p$

| | |
|---|---|
| A partir des clauses suivantes | (2.1) $a = \text{pere}(b)$<br>(2.2) $b = \text{pere}(c)$<br>(2.3) $\text{grand-pere}(x) = \text{pere}(\text{pere}(x)$<br>(2.4) $\text{joue}(c, \text{grand-pere}(c))$<br>(2.5) $\neg\text{joue}(c, a)$ [négation du fait à démontrer] |

on ne peut rien déduire car on ne peut pas unifier les clause $\text{joue}(c, a)$ et $\text{joue}(c, \text{grand-pere}(c))$ puisque la constante $a$ et le terme $\text{grand-pere}(c))$ ne peuvent pas s'unifier.

Pour résoudre ce problème, on a deux solutions :
- ajouter des *axiomes égalitaires* qui expriment explicitement toutes les propriétés implicites de l'égalité ;
- ajouter de nouvelles techniques : la **démodulation** et la **paramodulation**

# 3 Axiomes égalitaires

Pour chaque prédicat $P$ $n$-aire, on ajoute l'axiome

$\forall x_1 \forall x_2 ... \forall x_n \forall y_1 \forall y_2 ... \forall y_n (x_1 = y_1 \wedge x_2 = y_2 \wedge ... \wedge x_n = y_n \rightarrow (\text{pere}(x_1, x_2, ...x_n) \rightarrow \text{pere}(y_1, ..., y_n)))$

Pour chaque symbole fonctionnel $f$ $n$-aire, on ajoute l'axiome

$\forall x_1 \forall x_2 ... \forall x_n \forall y_1 \forall y_2 ... \forall y_n (x_1 = y_1 \wedge x_2 = y_2 \wedge ... \wedge x_n = y_n \rightarrow f(x_1, x_2, ...x_n) = f(y_1, ..., y_n))$

c'est-à-dire, pour l'exemple précédent, les clauses

(3.1) $x_1 = y_1 \wedge x_2 = y_2 \wedge \text{joue}(x_1, x_2) \rightarrow \text{joue}(y_1, y_2)$

(3.2) $x = y \rightarrow \text{pere}(x) = \text{pere}(y)$

On ajoute de plus les axiomes

(3.3) $\forall x \ (x = x)$

et

(3.4) $\forall x \ (x = y \rightarrow y = x)$

On a alors la réfutation suivante

| | |
|---|---|
| (3.5) $x_2 = y_2 \wedge \text{joue}(x_1, x_2) \rightarrow \text{joue}(x_1, y_2)$ | (3.1) et (3.3) |
| (3.6) $\text{joue}(c, \text{pere}(\text{pere}(c)))$ | (2.3), (2.4) et (3.5) |
| (3.7) $\text{pere}(b) = \text{pere}(\text{pere}(c))$ | (2.2) et (3.2) |
| (3.8) $\text{pere}(\text{pere}(c)) = \text{pere}(b)$ | (3.4) et (3.7) |
| (3.9) $\text{joue}(c, \text{pere}(b))$ | (3.8), (3.6) et (3.5) |
| (3.10) $\text{pere}(b) = a$ | (2.1) et (3.2) |
| (3.11) $\text{joue}(c, a)$ | (3.10), (3.9) et (3.5) |
| (3.12) $\square$ | (2.5) et (3.11) |

# 4 Démodulation

Cette nouvelle technique est la suivante

Si on a une clause unitaire égalitaire positive

$r = s$

et une clause

$C = P(..., t, ...) \vee H$

où $t$, un terme appparaissant dans un des arguments de $P(...)$ est une instance de $r$ [ou $s$], c'est-à-dire $t$ peut s'unifier avec $r$ [ou $s$], mais seules les variables de $r$ [ou $s$] peuvent être substituées, par une substitution $\sigma$, soit $t = \sigma(r)$ [ou $t = \sigma(s)$],
alors on **remplace** $t$ par $\sigma(s)$ [ou $\sigma(t)$] dans $C$

*Terminologie* :

$r = s$ est le **démodulateur** ("**de**").
$C$ est la clause **démodulée** ("**dans**").
On a **démodulé de** $r = s$ **dans** $C$.
La nouvelle clause est le **démodulant**.

*Remarque* : la clause $P(..., \sigma(r), ...) \vee H$ n'est pas *ajoutée* mais elle **remplace** la clause $P(...t...) \vee H$.
On n'augmente donc pas le nombre de clauses.
Ceci est possible car ces deux clauses sont équivalentes.

Pour l'exemple précédent, on a la réfutation

| | |
|---|---|
| (4.1) joue($c$, pere(pere($c$)) | **de** (2.3) **dans** (2.4) |
| (4.2) joue($c$, pere($b$) | **de** (2.2) **dans** (4.1) |
| (4.3) joue($c, a$) | **de** (2.1) **dans** (4.2) |
| (4.4) $\square$ | (2.5) et (4.3) |

**Propriété** : Le principe de résolution auquel on adjoint la démodulation est une règle saine, mais non complète.

On constatera la non complétude sur l'exemple de la section suivante.

# 5 Paramodulation

A partir des connaissances suivante :

Alain est le père de Bernard.
Toute personne est plus jeune que son père.

Peut-on conclure que Bernard est plus jeune qu'Alain ?

## 5.1 Notation prédicative

clauses

| |
|---|
| pere($a, b$) |
| pere($x, y$) $\rightarrow y < x$ |
| $\neg(b < a)$ [négation de la conclusion] |

La réfutation est immédiate.

## 5.2 Notation fonctionnelle

clauses

| |
|---|
| (5.1) $a =$ pere($b$) |
| (5.2) $x <$ pere($x$) |
| (5.3) $\neg(b < a)$ [négation de la conclusion] |

On ne peut pas unifier $a$ et pere($b$).

On doit donc rajouter les axiomes

| |
|---|
| (5.4) $x_1 = y_1 \wedge x_2 = y_2 \wedge x_1 < x_2 \rightarrow y_1 < y_2$ |
| (5.5) $x = x$ |
| (5.6) $x = y \rightarrow y = x$ |

pour obtenir la réfutation

| | |
|---|---|
| (5.7) $x_2 = y_2 \wedge x < x_2 \rightarrow x < y_2$ | (5.4) et (5.5) |
| (5.8) pere($x$) $= y_2 \rightarrow x < y_2$ | (5.2) et (5.7) |
| (5.9) pere($b$) $= a$ | (5.1) et (5.6) |
| (5.10) $b < a$ | (5.8) et (5.9) |
| (5.11) $\square$ | (5.3) et (5.10) |

## 5.3 Paramodulation

La démodulation ne permet pas d'éviter de rajouter les axiomes égalitaires.

La paramodulation est une généralisation de la démodulation dans laquelle
- on peut faire des substitutions dans la clause $C$
- la clause où se trouve le littéral égalitaire n'est pas nécessairement unitaire
- on ne remplace pas la clause $C$ mais on ajoute la nouvelle clause

Soit
Si on a une clause égalitaire positive, c'est-à-dire ayant un littéral égalitaire positif $r = s[$ ou $s = r]$,
soit $CE = CE1 \vee (r = s) \vee CE2$
et une clause
$C = C1 \vee P(..., t, ...) \vee C2$
n'ayant pas de variable commune avec $CE$ et où $t$, un terme appparaissant dans un des arguments de $P(...)$ peut s'unifier avec $r$ , par une substitution $\sigma$, soit $\sigma(t) = \sigma(r)$,
alors on **ajoute** la clause
$(CE1 \vee CE2 \vee C1 \vee P(..., s, ...) \vee C2)_\sigma$

*Exemple* ($a$ et $b$ sont des constantes, $x$ et $y$ des variables)
$CE = (f(x, a) = g(x, a)) \vee Q(x)$
$C = P(f(b, y)) \vee R(y)$
avec $\sigma = (x|b,\ y|a)$ on ajoute la clause
$Q(b) \vee P(g(b, a) \vee R(a)$

*Terminologie* :
$CE$ est le **paramodulateur** ("**de**").
$C$ est la clause **paramodulée** ("**dans**").
la nouvelle clause est le **paramodulant**.

Pour l'exemple précédent
- (5.1) $a = \text{pere}(b)$
- (5.2) $x < \text{pere}(x)$
- (5.3) $\neg(b < a)$ [négation de la conclusion]

on a la réfutation
- (5.12) $b < a$ de (5.1) dans (5.2)
- (5.13) $\square$ (5.3) et (5.12)

*Remarque* : $x < \text{pere}(x)$ est plus générale que $b < a$, on n'aurait donc pas pu supprimer $x < \text{pere}(x)$ sans perdre de l'information.

**Propriété** : Le principe de résolution auquel on adjoint la paramodulation est une règle saine et complète.

# EXERCICES

## I - Calcul propositionnel

1. Pour chacune des formules suivantes, dire si c'est une **tautologie** ou une **antilogie**.
   Sinon, dire pour quelles valeurs des variables propositionnelles la formule est vraie.
   $p \vee q$
   $p \rightarrow q$
   $p \vee \neg p$
   $p \rightarrow p$
   $p \wedge \neg p$
   $p \wedge (p \rightarrow q) \wedge \neg q$
   $p \wedge (p \rightarrow q) \rightarrow q$
   $(p \vee r) \wedge (q \vee \neg r) \rightarrow p \vee q$

2. $F \equiv G$ signifie que les formules $F$ et $G$ sont **équivalentes.**
   Est-ce qu'on a
   $p \rightarrow q \equiv \neg p \vee q$ ?
   $p \rightarrow (q \rightarrow r) \equiv p \wedge q \rightarrow r$ ?
   $\neg(p \vee q) \equiv \neg p \vee \neg q$ ?
   $\neg(p \vee q) \equiv \neg p \wedge \neg q$ ?
   $\neg(p \wedge q) \equiv \neg p \vee \neg q$ ?
   $p \rightarrow q \equiv \neg p \rightarrow \neg q$ ?
   $p \rightarrow q \equiv \neg q \rightarrow \neg p$ ?
   $\neg(p \rightarrow q) \equiv p \wedge \neg q$ ?

3. $\Sigma \models F$ signifie que la formule F est **conséquence sémantique** (ou logique) de l'ensemble de formules $\Sigma$
   Montrer que
   $\{p,\ p \rightarrow q\} \models q$
   $\{p \vee r,\ q \vee \neg r\} \models p \vee q$
   $\models p \rightarrow q$ si et seulement si $\{p\} \models q$

4. Mettre les formules suivantes sous **forme normale conjonctive (FNC)** et sous **forme normale disjonctive (FND)**
   $p \wedge (q \vee (r \wedge s))$
   $p \leftrightarrow q$
   $p \wedge q$
   $p \vee q$
   $p \wedge q \rightarrow r \wedge s$
   $p \vee q \rightarrow r \vee s$
   $p \wedge (p \rightarrow q) \rightarrow r$
   $p \wedge ((p \rightarrow q) \rightarrow r)$

# II - Calcul des Prédicats du premier ordre

1. Pour chacune des formules suivantes, dire si elle est **valide** (toujours vraie)
   $\forall x(A(x) \to B(x)) \wedge A(a) \to B(a)$ [a est une constante]
   $\forall x(A(x) \to B(x)) \ \wedge \forall x(B(x) \to C(x)) \to \forall x(A(x) \to C(x))$
   $\forall x A(x) \to \exists x A(x)$
   $\exists x A(x) \to \forall x A(x)$
   $A(a) \to \exists x A(x)$
   $\exists x A(x) \to A(a)$
   $\forall x \exists y P(x,y) \to \exists y \forall x P(x,y)$
   $\exists y \forall x P(x,y) \to \forall x \exists y P(x,y)$

2. Les formules suivantes sont-elles **équivalentes** ?
   $\neg\forall x A(x) \equiv \forall x \neg A(x)$ ?
   $\neg\forall x A(x) \equiv \exists x \neg A(x)$ ?
   $\neg\forall x(A(x) \to B(x)) \equiv \exists x(A(x) \wedge \neg B(x))$ ?
   $\forall x(A(x) \wedge B(x)) \equiv \forall x A(x) \wedge \forall x B(x)$ ?
   $\exists x(A(x) \wedge B(x)) \equiv \exists x A(x) \wedge \exists x B(x)$ ?
   $\forall x(A(x) \vee B(x)) \equiv \forall x A(x) \vee \forall x B(x)$ ?
   $\exists x(A(x) \vee B(x)) \equiv \exists x A(x) \vee \exists x B(x)$ ?
   $H \to \forall x A(x) \equiv \forall x(H \to A(x))$ ?
   $H \to \exists x A(x) \equiv \exists x(H \to A(x))$ ?
   $\forall x A(x) \to C \equiv \forall x(A(x) \to C)$ ?
   $\exists x A(x) \to C \equiv \exists(A(x) \to C)$ ?
   $\forall x A(x) \to C \equiv \exists x(A(x) \to C)$ ?
   $\exists x A(x) \to C \equiv \forall x(A(x) \to C)$ ?
   $\forall x \exists y P(x,y) \equiv \exists y \forall x P(x,y)$ ?
   $\forall x \exists y(A(x) \wedge B(y)) \equiv \exists y \forall x(A(x) \wedge B(y))$ ?

3. Est-ce qu'on a les **conséquences sémantiques** suivantes ?
   $\{\forall x(A(x) \to B(x)),\ A(a)\} \models B(a)$ ?
   $\{\forall x(A(x) \to B(x)), \forall x(B(x) \to C(x))\} \models \forall x(A(x) \to C(x))$ ?
   $\{\forall x(A(x) \to B(x)),\ \forall x(B(x) \to C(x))\} \models \forall x(A(x) \to C(x))$ ?
   $\{\forall x \exists y P(x,y),\ \forall x \forall y \exists z(P(x,z) \wedge P(z,y)) \to Q(x,y)\} \models \forall x \exists z Q(x,y)$ ?

4. Soit I l'interprétation de domaine $\mathbb{N}$, $\mathbb{Z}$ ou $\mathbb{R}$, où $R$ est interprété par $\leq$, $\geq$, $<$ ou $>$.
   Quelle est l'**interprétation des formules** $\forall x \exists y R(x,y)$ et $\exists x \forall y R(x,y)$ ?

# III - Application - Calcul des Prédicats

1. Formaliser les connaissances suivantes dans le calcul des prédicats

   Un dragon est heureux si tous ses enfants peuvent voler.
   Les dragons verts peuvent voler.
   Un dragon est vert si au moins un de ses parents est vert.

2. En déduire que les dragons verts sont heureux.
   Trouver une réfutation de cette propriété en utilisant le Principe de résolution.

3. Que peut faire un dragon rose pour être heureux ?

# IV - Démonstration automatique de théorèmes

On a en théorie des groupes le théorème suivant :
**Théorème** : Si, pour tout élément d'un groupe, le carré de cet élément est égal à l'élément neutre,

alors le groupe est commutatif.

On rappelle les axiomes de groupe :
Un groupe est un ensemble muni d'un opération associative, ayant un élément neutre, et tel que tout élément à un inverse.

C'est-à-dire, si l'on note * l'opération :
- pour tous $x$, $y$, $z$, on a $x * (y * z) = (x * y) * z$
- $e$ l'élément neutre est tel que pour tout x, $e * x = x * e = x$
- pour tout $x$ il existe un inverse $y$ tel que $x * y = y * x = e$

1. Démontrer le théorème ci-desssus par un raisonnement mathématique habituel.
2. En utilisant le prédicat $P$ qui sera défini comme
   $$P(x, y, z) \leftrightarrow z = x * y$$
   trouver une réfutation utilisant le principe de résolution.
3. En utilisant le symbole fonctionnel $f$ qui sera défini comme
   $$f(x, y) = x * y$$
   trouver une réfutation utilisant le principe de résolution avec démodulation et/ou paramodulation.

# BIBLIOGRAPHIE

E. Mendelson, *Introduction to mathematicial logic*, Van Nostrand (1964)
J.A. Robinson, A machine oriented logic based on the resolution principle, J. ACM 12 (1965), 23-41
C.L. Chang, R. C. T. Lee, *Symbolic Logic and Mechanical Theorem Proving*, Academic Press (1973)
R. Kowalski, *Logic for Problem Solving*, North Holland (1979)
J. Stern, *Fondements Mathématiques de l'Informatique*, Mc Graw Hill (1990)
J. Duffy, *Principles of Automated Theorem proving*, Wiley (1991)
W. Bibel, *Deduction, Automated Logic*, Academic press (1993)
R. Lassaigne, M. de Rougemont, *Logique et Fondements de l'Informatique*, Hermès (1993)
J.M. Alliot, T. Scheix, *Intelligence Artificielle et Informatique Théorique*, Cepadues (1994)