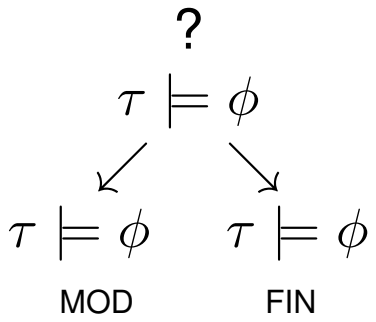
Théories & Preuves

1. Le problème

- ◇ **Données :** une théorie τ et une formule ϕ
- ◇ **Question :** est ce que ϕ est une conséquence logique de τ ?



Conséquence déductive Conséquence inductive

Deux approches :

- ◇ Enumérer les modèles de τ et vérifier la validité de ϕ dans chaque modèle (approche **sémantique)**
- ◇ Appliquer une méthode de preuve **syntaxique**
  Déduction (e.g.,*résolution*,...), Induction

2. APPROCHE SÉMANTIQUE

- ◇ Enumérer les modèles de τ
- ◇ Vérifier la validité de ϕ dans chaque modèle

Contraintes : le nombre de modèles de τ doit être fini !

Exemple 1.

Monsieur X, professeur d'informatique, rentre chez lui et découvre sa femme dans les bras d'un beau brun ténébreux. Surpris, celui-ci s'enfuit en s'envolant par la fenètre. On sait que :

- ◇ *Marc, Batman ou Olivier plaisent à la femme de Monsieur X*
- ◇ *A chaque fois qu'Olivier fait quelque chose, son copain Marc l'imite*
- ◇ *Batman ne sait pas voler*

Marc est-il l'amant de Madame X ?

Formalisation : ce problème peut se formaliser dans la théorie τ suivante :

- ◇ Langage :
  - — pas de variables
  - — pas de fonctions
  - — propositions : m, b et o
- ◇ Axiomes :
  - — A1 : m $\vee$ b $\vee$ o
  - — A2 : o $\Rightarrow$ m
  - — A3 : $\neg$ b
- ◇ ϕ : m

On veut montrer : $\tau \models \phi$

c'est à dire : $\models \tau \Rightarrow \phi$

c'est à dire : $I \models \tau \Rightarrow \phi$ pour toute interprétation I

Vérification de la validité : énumérer toutes les interprétations τ

- ◇ Pour le langage de τ (sans variables ni fonctions), une interprétation est une affectation des valeurs **“vrai”** ou **“faux”** aux propositions.
- ◇ Enumérer toutes les interprétations revient à construire tous les triplets (m, b, o) à valeur dans $\{0,1\}$
- ◇ Montrer que $I \models \tau \Rightarrow \phi$ pour toute interprétation I revient à construire la **table de vérité** de $\tau \Rightarrow \phi$ et montrer que $\tau \Rightarrow \phi$ vaut 1 dans tous les cas.

Table d vérité de τ :

m	b	o	$((m \vee b \vee o) \wedge (o \Rightarrow m) \wedge \neg b) \Rightarrow m$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

... et donc Marc est bien l'amant de Madame X !

LIMITES DE L'APPROCHE SÉMANTIQUE

- ◇ Énumérer toutes les interprétations c'est coûteux : pour un langage sans variables ni fonctions et **avec n propositions il y a 2^n interprétations** possibles
- ◇ Comment énumérer tous les modèles quand on a un langage du premier ordre (avec variables) ?

Exemple

langage : prédicat E (binaire)
fonction : f (unaire)
axiomes : $E(x,y) \Leftrightarrow E(y,x)$
$E(x,y) \wedge E(y,z) \Rightarrow E(x,z)$
$E(x,x)$

Comment prouver E(f(x),f(x)) ?

3. APPROCHES SYNTAXIQUES

Appliquer une **méthode syntaxique m** telle que :

$$\tau \vdash_{\mathbf{m}} \phi \quad \equiv \quad \tau \models \phi$$

Système de règles d'inférence :

système de Hilbert système de Gentzen système de **résolution**	$\models_{MOD}$	Preuve déductive	**correct** & **complet**
système **d'induction**	$\models_{FIN}$	Preuve inductive	**correct**

4. PREUVES DÉDUCTIVES

Une théorie τ , une formule ϕ , une méthode g

$$\tau \underset{\text{g}}{\vdash} \phi$$

La formule ϕ est **prouvable (dérivable)** à partir de τ en appliquant la méthode g

ou

ϕ est un **g–théorème** de τ

NOTATIONS (RAPPEL)

x, y, z	:	variables
a, b, c	:	constantes
p, q, r	:	symboles de prédicat
f, g	:	symboles de function
A, B, C, D, F ϕ , ψ	:	formules

4.1. Equivalences de base

(Herbrand 1931, Robinson 1965)

$\tau \underset{\text{MOD}}{\models} \phi$	ϕ est une conséquence de τ
$\Updownarrow$	$\Updownarrow$
$\tau \underset{\text{MOD}}{\models} \forall\phi$	La cloture universelle de ϕ est une conséquence de τ
$\Updownarrow$	$\Updownarrow$
$\alpha \not\models \tau \wedge \neg\forall(\phi)$ ($\forall$ structure α)	La conjonction $\tau \wedge \neg\forall(\phi)$ est insatisfiable
$\Updownarrow$	$\Updownarrow$
$\alpha \not\models \text{FP}[\tau \wedge \neg\forall(\phi)]$ ($\forall$ structure α)	La forme prénexe de $\tau \wedge \neg\forall(\phi)$ est insatisfiable
$\Updownarrow$	$\Updownarrow$

EQUIVALENCES DE BASE (suite)

$\alpha \not\models$ SK[FP[$\tau \wedge \neg\forall(\phi)$]]	La forme de Skolem de FP[. . .] est insatisfiable
($\forall$ structure α)	
$\Updownarrow$	$\Updownarrow$
$\alpha \not\models$ FNC[SK[FP[$\tau \wedge \neg\forall(\phi)$]]]	La forme normale conjonctive de SK[. . .] est insatisfiable
$\Updownarrow$	$\Updownarrow$
$\overline{\alpha} \not\models$ FNC[SK[FP[$\tau \wedge \neg\forall(\phi)$]]]	La forme normale conjonctive n'a pas de modèle de Herbrand
($\overline{\alpha}$ structure de Herbrand)	
$\Updownarrow$	$\Updownarrow$
FNC[SK[FP[$\tau \wedge \neg\forall(\phi)$]]] $\vdash_{RG} \square$	La clause vide est déductible par résolution

4.2. UNE PROCÉDURE DE PREUVE

Pour prouver que $\tau \models_{MOD} \phi$:

1. Mettre la formule ϕ sous forme $\forall\ \phi$ (**cloture universelle)**
2. Ajouter $\neg\forall\ \phi$ à τ . On obtient $F_0 = \{\tau \wedge \neg\forall(\phi)\}$
3. Mettre F_0 sous **forme prénexe** On obtient F_1
4. Mettre les formules de F_1 sous **forme de Skolem** On obtient F_2
5. Mettre les formules de F_2 sous **forme de clauses** On obtient F_3
6. Appliquer la **résolution** pour dériver la **clause vide** de F_3

5. UNE PROCÉDURE DE PREUVE POUR LE CALCUL PROPOSITIONNEL

Pour prouver que $\tau \models_{MOD} \phi$:

1. Ajouter $\neg\ \phi$ à τ . On obtient $\tau \wedge \neg\phi$
2. Mettre les formules de $\tau \wedge \neg\phi$ sous **forme de clauses.** On obtient C
3. Appliquer à C la **0- résolution** pour dériver la **clause vide** de F_1 :

$$\frac{\neg \mathbf{A} \vee \mathbf{F_1} \ \wedge \ \mathbf{A} \vee \mathbf{F_2}}{\mathbf{F_1} \vee \mathbf{F_2}}$$

   où A est un atome, F_1 et F_2 des clauses
4. Si on infère la clause vide, alors ϕ est valide dans MOD

5.1. 0- RÉSOLUTION : INTUITION

- **Règle :** $(\neg \mathbf{A} \vee \mathbf{F_1}) \wedge (\mathbf{A} \vee \mathbf{F_2}) \vdash \mathbf{F_1} \vee \mathbf{F_2}$
  Si $(\neg \mathbf{A} \vee \mathbf{F_1}) \wedge (\mathbf{A} \vee \mathbf{F_2})$ est vrai, alors $\mathbf{F_1} \vee \mathbf{F_2}$ est vrai car on ne peut pas avoir $\neg \mathbf{A} \wedge \mathbf{A}$

- **Principe de la démonstration : preuve par l'absurde**
  Si $\tau \wedge \neg\phi \vdash \square$alors ϕ est vrai
  (car $\neg\phi$ introduit une contradiction)

Pour les langages sans variables ni symboles fonctionnels la 0-résolution est complète et correcte

5.2 Forme Clausale

Définition 1.
Soit ϕ une formule. On dit que ϕ est sous forme clausale si ϕ est une formule $F_1 \wedge F_2 \wedge \ldots \wedge F_m$ où chaque F_i est de la forme :

$$p_{l1} \vee p_{l2} \vee \ldots p_{li} \vee \neg p_{m1} \vee \neg p_{m2} \vee \ldots \vee \neg p_{mj}$$

Définition 2.
Une disjonction de littéraux $p_{l1} \vee p_{l2} \vee \ldots p_{li} \vee \neg p_{m1} \vee \neg p_{m2} \vee \ldots \vee \neg p_{mj}$ *est appelée* **clause**

Théorème 1.
Pour toute formule ϕ il existe une formule équivalente ϕ ' qui est sous forme de clauses

5.3. MISE SOUS FORME CLAUSALE

L'utilisation des équivalences suivantes de gauche à droite permet de mettre une formule sous forme de clauses :

$c_1 : \neg(A \vee B) \equiv \neg A \wedge \neg B$
$c_2 : \neg(A \wedge B) \equiv \neg A \vee \neg B$
$c_3 : A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
$c_4 : (A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C)$

5.4 Reprise de l'exemple (1)– Marc ?

O-résolution : $(\neg \boldsymbol{A} \vee \boldsymbol{F_1}) \wedge (\boldsymbol{A} \vee \boldsymbol{F_2}) \vdash \boldsymbol{F_1} \vee \boldsymbol{F_2}$

Axiomes : A1 : $m \vee b \vee o$ A2 : $o \Rightarrow m$ A3 : $\neg b$
On veut prouver $\boldsymbol{\phi : m}$

1. Négation de ϕ : $\neg m$
2. Mise sous forme clausale
   - ◇ A1 : $m \vee b \vee o$
   - ◇ A2' : $\neg o \vee m$
   - ◇ A3 : $\neg b$
   - ◇ $\neg \phi$: $\neg m$
3. Résolution
   - ◇ $\neg m \ , \ \neg o \vee m \vdash \neg o$
   - ◇ $\neg o \ , \ m \vee b \vee o \vdash m \vee b$
   - ◇ $m \vee b \ , \ \neg b \vdash m$
   - ◇ $\neg m \ , \ m \vdash \square$

Donc $\tau \models_{MOD} \phi$, c'est à dire ϕ est valide dans tous les modèles de τ.

5.5 Reprise de l'exemple (1)– Olivier ?

O-résolution : $(\neg \mathbf{A} \vee \mathbf{F_1}) \wedge (\mathbf{A} \vee \mathbf{F_2}) \vdash \mathbf{F_1} \vee \mathbf{F_2}$

Axiomes : A1 : $m \vee b \vee o$ A2 : $o \Rightarrow m$ A3 : $\neg b$
On veut prouver $\phi : o$

1. Négation de ϕ : $\neg o$
2. Mise sous forme clausale :
   - ◇ A1 : $m \vee b \vee o$
   - ◇ A2' : $\neg o \vee m$
   - ◇ A3 : $\neg b$
   - ◇ $\neg\phi$: :$\neg o$
3. Résolution
   - ◇ $\neg\phi, A1 \vdash m \vee b$
   - ◇ $m \vee b, A3 \vdash m$

On ne peut inférer la clause vide. **Comme la 0-résolution est complète, o n'est pas une conséquence logique de $\{$A1,A2,A3$\}$**

5.6 Reprise de l'exemple (1), preuve sémantique

m	b	o	$((m \vee b \vee o) \wedge (o \Rightarrow m) \wedge \neg b) \Rightarrow o$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	**0**
1	0	1	1
1	1	0	1
1	1	1	1

... et donc o n'est pas une conséquence logique de $\{$A1,A2,A3$\}$

6. UNE PROCÉDURE DE PREUVE POUR LE CALCUL DES PRÉDICATS DU PREMIER ORDRE (RAPPEL)

Pour prouver que $\tau \models_{\overline{MOD}} \phi$:

1. Mettre la formule ϕ sous forme $\forall\ \phi$ (**cloture universelle)**
2. Ajouter $\neg\forall\ \phi$ à τ . On obtient $F_0 = \{\tau \wedge \neg\forall(\phi)\}$
3. Mettre F_0 sous **forme prénexe** On obtient F_1
4. Mettre les formules de F_1 sous **forme de Skolem** On obtient F_2
5. Mettre les formules de F_2 sous **forme de clauses** On obtient F_3
6. Appliquer la **résolution** pour dériver la **clause vide** de F_3

Exemple 2.

◇ Sémantique

Si x est pair, alors x+1 est impair
Il existe un nombre pair

Montrez qu'il existe y tel que y est impair

◇ Syntaxe

Symboles

- *prédicats :* *p(x) : x pair*
  i(x) : x impair
- *fonctionnel :* *f(x) : x +1*

τ : $(A_1)\ \forall x(p(x) \Rightarrow i(f(x))$
$(A_2)\ \exists x\, p(x)$
ϕ : $\exists y\, i(y)$

Preuve

- **Appliquer Cloture universelle de ϕ**
- **Ajouter $\neg\forall\ \phi$ à τ**

$$(A_1)\ \ \forall x(p(x) \Rightarrow i(f(x))$$
$$(A_2)\ \ \exists x\, p(x) \qquad\qquad (\neg\phi)\quad \neg\exists y\, i(y)$$

- **Mettre sous forme prénexe**

$$(A_1)\ \ \forall x(\neg p(x) \vee i(f(x))$$
$$(A_2)\ \ \exists x\, p(x) \qquad\qquad (\neg\phi)\quad \forall y\, \neg i(y)$$

- **Mettre sous forme de Skolem**

$$(A_1)\ \ \neg p(x) \vee i(f(x))$$
$$(A_2)\ \ p(a) \qquad\qquad (\neg\phi)\quad \neg i(y)$$

- **Mettre sous forme de clauses**

◇ **Appliquer la résolution pour dériver la clause vide**

Résolution entre A1 et A2 : **(unification $x \leftarrow a$)**

$$\frac{\neg \mathbf{p(x)} \vee \mathbf{i(f(x))} \quad \mathbf{p(a)}}{\mathbf{i(f(a))}}$$

Résolution entre i(f(a)) et A3 **(unification $y \leftarrow f(a)$)**

$$\frac{\mathbf{i(f(a))} \quad \neg \mathbf{i(y)}}{\square}$$

contradiction

... donc ϕ est vraie (dans MOD), c'est à dire qu'il existe un nombre impair (les unifications nous donne sa valeur : a + 1)

6.1. Forme prénexe

Définition 3.
Une formule ϕ est sous **forme prénexe** *si elle est de la forme :*

$$\mathbf{Q_1} x_1 \, \mathbf{Q_2} x_2 \, \ldots \mathbf{Q_n} x_n \, \psi$$

où chaque $\mathbf{Q_i}$ est $\forall$ ou $\exists$ et où ψ ne contient aucun quantificateur

Exemple 3.
$\forall \, x \, \exists \, y \, p(x, y)$ *est sous forme prénexe*
$\forall \, x \, q(x) \exists \, y \, p(x, y)$ *n'est pas sous forme prénexe*

Théorème 2.
Pour toute formule ϕ il existe une formule ϕ' sous forme prénexe équivalente à ϕ

a) Elimination des $\Rightarrow$ et des $\Leftrightarrow$ en utilisant les équivalences suivantes de gauche à droite :

$p_1 : (A \Rightarrow B) \equiv (\neg A \vee B)$

$p_2 : (A \Leftrightarrow B) \equiv ((\neg A \wedge \neg B) \vee (A \wedge B))$

b) Renommer certaines variables liées de manière à n'avoir plus de variable quantifiée deux fois en utilisant les équivalences :

$p_3 : \forall x\, A(x) \equiv \forall y\, A(y)$

$p_4 : \exists x\, A(x) \equiv \exists y\, A(y)$

c) Faire remonter les quantificateurs en utilisant les équivalences suivantes de gauche à droite (x $\notin$ varlib(C)) :

p_5 : $\neg\exists x\, A(x) \equiv \forall x \neg A(x)$
p_6 : $\neg\forall x\, A(x) \equiv \exists x \neg A(x)$
p_7 : $\neg\neg A(x) \equiv A(x)$
p_8 : $(C \vee \forall x\, A(x)) \equiv \forall x (C \vee A(x))$
p_9 : $(C \vee \exists x\, A(x)) \equiv \exists x (C \vee A(x))$
p_{10} : $(\forall x\, A(x) \vee C) \equiv \forall x (A(x) \vee C)$
p_{11} : $(\exists x\, A(x) \vee C) \equiv \exists x (A(x) \vee C)$
p_{12} : $(C \wedge \forall x\, A(x)) \equiv \forall x (C \wedge A(x))$
p_{13} : $(C \wedge \exists x\, A(x)) \equiv \exists x (C \wedge A(x))$
p_{14} : $(\forall x\, A(x) \wedge C) \equiv \forall x (A(x) \wedge C)$
p_{15} : $(\exists x\, A(x) \wedge C) \equiv \exists x (A(x) \wedge C)$

Remarque 1.
Pour limiter le nombre de quantificateurs de la formule finale il peut être intéressant d'utiliser les équivalences suivantes :

q_1 : $(\forall x\, A(x) \Rightarrow C) \equiv \exists x\, (A(x) \Rightarrow C)$
q_2 : $(\exists x\, A(x) \Rightarrow C) \equiv \forall x\, (A(x) \Rightarrow C)$
q_3 : $(\forall x\, A(x) \wedge \forall x\, B(x)) \equiv \forall x\, (A(x) \wedge B(x))$
q_4 : $(\exists x\, A(x) \vee \exists x\, B(x)) \equiv \exists x\, (A(x) \vee B(x))$

Remarque 2.
Une forme prénexe peut aussi être obtenue en éliminant les $\Leftrightarrow$ et les $\Rightarrow$ avant de faire remonter les quantificateurs en tête de la formule.

6.2. FORME DE SKOLEM

Définition 4.
Soit $\phi \equiv Q_1 x_1 \, Q_2 x_2 \ldots Q_n x_n \, \psi(x_1 \, x_2 \ldots x_n)$ *une formule mise sous forme prénexe*
On appelle **forme de Skolem** *de* ϕ *la formule* ϕ^S *obtenue en enlevant tous les quantificateurs* $\exists x_i$ *et en remplaçant chacune des variables* x_i *quantifiée avec* $\exists$ *par* $f_i(x_{j1}, x_{j2}, \ldots x_{jn})$ *où* $x_{j1}, x_{j2}, \ldots x_{jn}$ *sont les variables quantifiées par des* $\forall$ *placés devant le* $\exists x_i$

Remarque 3.
Les symboles fonctionnels introduits doivent être **tous différents** *et être* **différents de ceux qui sont utilisés dans** ϕ

Remarque 4.
Lorsqu'il n'y a pas de quantificateur $\forall$ *devant le* $\exists x_i$ *on introduit* **un symbole de constante** *(fonction 0-aire)*

6.2. Forme de Skolem –Propriétés

Théorème 3.
Soit $\{\phi_1, \phi_2, \ldots, \phi_n\}$ *un ensemble de formules sous forme prénexe et* $\{\phi_1^S, \phi_2^S, \ldots, \phi_n^S\}$ *l'ensemble des formes de Skolem de ces formules, alors* $\phi_1, \phi_2, \ldots, \phi_n$ *admet un modèle ssi* $\{\phi_1^S, \phi_2^S, \ldots, \phi_n^S\}$ *admet un modèle*

Intuition de la démonstration :

Considérons $\phi = \forall x, \exists y \;\; p(x, y)$. On a $\phi^s = \forall x \;\; p(x, f(x))$
Soit $i = (E, \overline{p})$ avec $\overline{p} : E \times E \rightarrow \{V, F\}$ un modèle de ϕ dans la base E. Si **i** est un modèle, alors pour tout $\alpha \in E$, il existe $\beta \in E$ tel que $\overline{p}(\alpha, \beta)$ soit vrai.
Soit $\overline{f} : E \rightarrow E$ la fonction tel que $\overline{f}(\alpha) = \beta$, alors $i' = (E, \overline{p}, \overline{f})$ **est un modèle de ϕ^S dans** la base E.
Réciproquement si $i' = (E, \overline{p}, \overline{f})$ est un modèle de ϕ^S dans la base E, alors on vérifie aisément que $i = (E, \overline{p})$ est un modèle ϕ

6.2. Forme de Skolem – Remarque

Remarque 5.
La forme de skolem ϕ ' n'est pas nécessairement équivalente à la formule ϕ *à partir de laquelle elle a été générée : certaines transformations peuvent créer des dépendances parasites dans la forme de skolem*

Exemple :

$$\begin{array}{ccc} & & P(x) \vee Q(f(x)) \\ & \nearrow & \\ \forall x\, P(x) \vee \exists y\, Q(y) & & \\ & \searrow & \\ & & Q(a) \vee P(x) \end{array}$$

6.3. RÈGLE D'INFÉRENCE DE LA RÉSOLUTION

$$\frac{A \vee F_1 \qquad \neg B \vee F_2}{\sigma(\Phi(F_1) \vee F_2)}$$

où

- A et B sont deux atomes avec le même symbole de prédicat et la même arité
- Φ est une substitution telle que $\Phi(A \vee F_1)$ et $\neg B \vee F_2$ n'aient aucune variable commune ; F_1 et F_2 sont des clauses
- σ est un plus grand unificateur de $\Phi(A)$ et B

Exemple 4.

$$\frac{p(x,c) \vee r(x) \qquad\qquad \neg p(c,c) \vee q(x)}{r(c) \vee q(x)}$$

$A = p(x, c)$ $\qquad$ $F_1 = r(x)$

$\Phi = (x|y)$ $\qquad$ $\Phi(A \vee F_1) = p(y, c) \vee r(y)$

$B = p(c, c)$ $\qquad$ $\sigma = (y|c)$

6.4 Règle d'inférence de diminution

$$\frac{A \quad \vee \quad B \quad \vee \quad F_1}{\sigma(A) \quad \vee \quad \sigma(F_1))}$$

où

- A et B sont deux atomes avec le même symbole de prédicat et la même arité
- σ est un plus grand unificateur de A et B

Exemple 5.

$$\frac{p(x,g(y)) \quad \vee \quad p(f(c),z) \quad \vee \quad r(x,y,z)}{p(f(c),g(y)) \quad \vee \quad r(f(c),y,g(y))}$$

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

7. UNIFICATION

$\rightarrow$ **Trouver un représentant commun à deux atomes**

Substitution

Définition 5.
Si A est une formule du calcul des prédicats, on note $(x|t)A$ la formule obtenue en remplaçant toutes les occurrences libres de x dans A par t

Exemple 6.
$(x|f(y,g(a))\ (p(z) \Rightarrow r(x)) \quad = \quad (p(z) \Rightarrow r(f(y,g(a))))$

Remarque 6.
En général $[c_1 c_2] \neq [c_2 c_1]$

Exemple 7.
$\sigma_1 = [(x|f(a))\ (y|f(x))] \quad \sigma_2 = [(y|f(x))\ (x|f(a)))]$

$$\begin{aligned}
\sigma_1(p(x,y)) &= (x|f(a))(p(x,f(x)) \\
&= p(f(a), f(f(a))) \\
\sigma_2(p(x,y)) &= (y|f(x))(p(f(a),y) \\
&= (p(f(a), f(x))
\end{aligned}$$

7.1 UNIFICATEUR

Définition 6.
Soit $S = \{A_1, A_2, \ldots, A_n\}$ un ensemble fini de formules atomiques du calcul des prédicats, on appelle **unificateur** *de S toute substitution σ telle que :*

$$\sigma A_1 = \sigma A_2 = \ldots = \sigma A_n$$

Exemple 8.

$$S = \{A_1, A_2, A_3\} \quad A_1 = p(x, z) \quad A_2 = p(f(y), g(a)) \quad A_3 = p(f(u), z)$$

$$\sigma_1 = [(x|f(u))\ (y|u)\ (z|g(a))] \quad \sigma_1 A_1 = \sigma_1 A_2 = \sigma_1 A_3 = p(f(u), g(a))$$
$$\sigma_2 = [(u|f(a))]\sigma_1 \quad \sigma_2 A_1 = \sigma_2 A_2 = \sigma_2 A_3 = p(f(f(a)), g(a))$$

7.2 UNIFICATEUR LE PLUS GÉNÉRAL

Définition 7.
Soit U_S l'ensemble des unificateurs de S, on dit que σ est **un plus grand unificateur** *de S (ou unificateur le plus général de S) si σ est une substitution de U_S telle que :*

$$\forall \alpha \in U_S \; \exists \beta \in U_S \; : \; \alpha = \beta\sigma$$

Exemple 9.
$S = \{A_1, A_2, A_3\} \quad A_1 = p(x, z) \quad A_2 = p(f(y), g(a)) \quad A_3 = p(f(u), z)$

Quelques plus grands unificateurs :

$\sigma_1 = [(x|f(u))\ (y|u)\ (z|g(a))]$
$\sigma_3 = [(x|f(v))\ (y|v)\ (z|g(a))\ (u|v)]$

et on a $\sigma_1 = (v|u)\sigma_3$ *et* $\sigma_3 = (u|v)\sigma_1$

7.3 ALGORITHME D'UNIFICATION DE DEUX ATOMES A ET B

- $\Phi \leftarrow []$ % Φ est un PGU de A et B
- **tant que** $\Phi A \neq \Phi B$ faire
  - déterminer le symbole le plus à gauche ΦA qui soit différent du symbole de même rang de ΦB
  - déterminer les sous-termes t_1 et t_2 de ΦA et ΦB qui commencent à ce symbole
  - **si** "aucun des deux n'est une variable" ou "l'un des deux est une variable contenue dans l'autre"
    - **alors** imprimer "A et B ne sont pas unifiables " ; **arrêt**
    - **sinon** faire
      - $x \leftarrow$ une variable parmi t_1 et t_2 ; $t \leftarrow$ l'autre terme
      - $\Phi \leftarrow (x|t)\Phi$
- **imprimer** Φ est un plus grand unificateur de A et B

8. STRATÉGIES DE RÉSOLUTION

◇ **Techniques de gestion d'ensembles de clauses**

Enrichissement par résolution & simplication jusqu'à détection de la clause vide ou saturation

◇ **Techniques d'exploration de l'arbre des déductions**

Parcours en largeur ou en profondeur d'abord de l'arbre de déduction pour trouver la clause vide

8.1 STRATÉGIES DE SATURATION

◇ **Algorithme :**

S_0 : ensemble initial de clauses

$i \leftarrow 1$

Faire

- $S_i \leftarrow \cup \{$clauses obtenues en effectuant toutes les résolutions et diminutions possibles sur $S_{i-1}\}$
- $i \leftarrow i + 1$

jusqu'à ce que $\square \in S_i$ ou $S_i = S_{i-1}$

◇ **Problème :** Les ensembles S_i augmentent de manière **exponentielle**

8.2 STRATÉGIE DE SATURATION AVEC SIMPLIFICATION

- **Elimination des tautologies**
  $(A \vee \neg A \vee C)$ où A est un atome et C une clause

- **Elimination des clauses subsumées**
  la clause D est subsumée par la clause C s'il existe une substitution σ telle que $D = \sigma C \vee F$ avec F une clause
  e.g. $p(f(a)) \vee q(a)$ est subsumée par $p(x)$

8.3 STRATÉGIES LINÉAIRES

Définition 8. *On appelle* **déduction linéaire** *de racine C_0 à partir de l'ensemble de clauses $\mathcal{C}$ toute déduction $F_0 F_1 \ldots F_n$ telle que :*

- ◇ *F_0 est obtenue par résolution ou diminution à partir de clauses dont l'une est C_0*
- ◇ *$F_i,\ i > 0$ est obtenue par résolution ou diminution à partir de clauses dont l'une est F_{i-1}*

Exemple 10.

$\mathcal{C} = \{\neg A \vee \neg B,\ A \vee \neg C,\ C,\ B \vee \neg D,\ D \vee B\}$ $\qquad C_0 = \neg A \vee \neg B$

Déduction :

$F_0 : \neg B \vee \neg C$ *(résolution entre* C_0 *et* $A \vee \neg C$*)*

$F_1 : \neg B$ *(résolution entre* F_0 *et* C*)*

$F_2 : \neg D$ *(résolution entre* F_1 *et* $B \vee \neg D$*)*

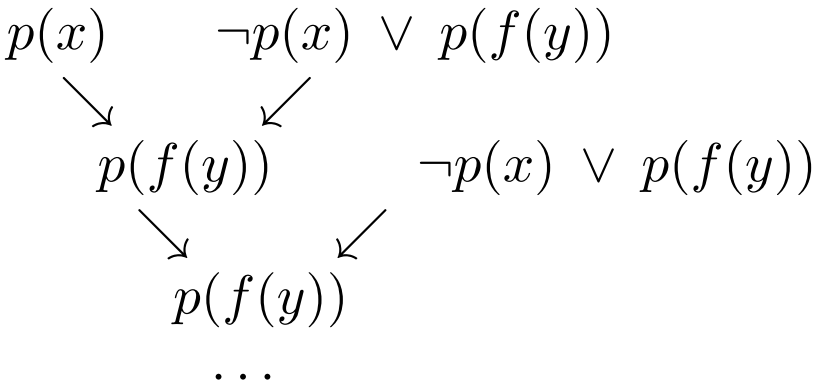
$F_3 : B$ *(résolution entre* F_2 *et* $D \vee B$*)* $\qquad F_4 : \square$ *(résolution entre* F_3 *et* F_1*)*

Théorème 4.
S'il existe une déduction de la clause vide, alors il existe une déduction linéaire de la clause vide

Remarque 7.
Une stratégie d'exploration en profondeur d'abord ne permet pas nécessairement de trouver la clause vide

Exemple 11.

$$\mathcal{C} = \{\neg p(x) \vee p(f(y)), \neg p(a), p(x)\}$$



8.4 STRATÉGIES "INPUT" ORDONNÉES

Définition 9.
On appelle déduction "input" de racine C_0 à partir de l'ensemble de clauses $\mathcal{C}$ toute déduction $\mathcal{C} F_0 F_1 \ldots F_n$ telle que :

- ◇ *F_0 est obtenue par résolution ou diminution à partir de clauses dont l'une est C_0*
- ◇ *$F_i,\ i > 0$ est obtenue par résolution ou diminution à partir de clauses* **dont l'une est dans $\mathcal{C}$**

La stratégie "input" ordonnée n'est pas complète dans le cas général

8.5 Stratégies "input" ordonnées avec des clauses de Horn (SLD)

Une **clause de Horn** est une clause comportant en ensemble de littéraux négatifs et au **plus un littéral positif**

Définition 10.
La résolution ordonnée entre deux clauses ordonnées sans variables communes

$C = \mathbf{l_1} \vee \mathbf{l_2} \vee \ldots \vee \mathbf{l_n}$ *avec* $\mathbf{l_1}$ *: littéral positif*

$C' = \mathbf{l'_1} \vee \mathbf{l'_2} \vee \ldots \vee \mathbf{l'_n}$ *avec* $\mathbf{l'_1}$ *: littéral négatif*

a pour résultat (quand elle est possible) la clause ordonnée

$\Phi(\mathbf{l_2} \vee \ldots \vee \mathbf{l_n} \vee \mathbf{l'_2} \vee \ldots \vee \mathbf{l'_n})$

où Φ *est le plus grand unificateur entre* l_1 *et* l'_1

8.5 SLD (suite)

Exemple de résolution avec des **clauses de Horn**

$$C = \mathbf{p_1(a)} \vee \mathbf{\neg q_1(y)} \vee \mathbf{\neg r_1(z)}$$

$$C' = \mathbf{\neg p_1(x)} \vee \mathbf{\neg q_2(x)} \vee \mathbf{\neg r_2(z)}$$

$$\frac{p_1(a) \vee \neg q_1(y) \vee \neg r_1(z) \qquad \neg p_1(x) \vee \neg q_2(x) \vee \neg r_2(z)}{\neg q_1(y) \vee \neg r_1(z) \vee \neg q_2(a) \vee \neg r_2(z)}$$

8.5 SLD (suite)

Si $\mathcal{C} = \mathcal{C}' \cup \{C_0\}$ est insatisfiable, que C_0 est **une clause ne comportant que des littéraux négatifs,** et que $\mathcal{C}'$ ne contient que des **clauses de Horn ordonnées dont le littéral positif** est en tête,
alors **il existe une déduction linéaire "input" ordonnée de racine C_0, n'utilisant que la résolution, conduisant à la clause vide**

Limites des stratégies "input" ordonnées avec des clauses de Horn :

- **Toute formule n'est pas transformable en clause de Horn,** e.g.,
  $\mathbf{A \Rightarrow B \vee D}$
  Exemple : ***riche(x)*** $\Rightarrow$ ***gagnant_lotto(x)*** $\vee$ ***héritier(x)***

- **Impossible de déduire des connaissances négatives,** e.g.,
  $\forall \mathbf{x}\ \neg(\mathbf{premier}(\mathbf{double}(\mathbf{succ}(\mathbf{x}))))$

8.6 SLD : EXTRACTION DE RÉSULTATS

Théorème 5.
Si *un ensemble de clauses de Horn* $\mathcal{C} = \mathcal{C}' \cup \{\mathcal{C}_\prime\}$ *est insatisfiable,*
et que C_0 *est un littéral négatif tel que* $C_0 = \neg p_1(x_1, \ldots, x_n) \vee \ldots \vee \neg p_m(x_1, \ldots, x_n)$
et que $\mathcal{C}'$ *ne contient que des clauses de Horn ordonnées en plaçant le littéral positif en tête,*
alors *chaque déduction "input" ordonnée de racine* C_0 *conduisant à la clause vide et dont les substitutions sont* $(x_1|t_1), \ldots, (x_n|t_n)$
définit *des objets* t_i *de l'univers de Herbrandt tel que*
$p_1(t_1, \ldots, t_n) \wedge \ldots \wedge p_m(t_1, \ldots, t_n)$ *est conséquence de* $\mathcal{C}'$

Exemple 12.
$\mathcal{C}' = \{p(a),\ p(b),\ r(f(x)) \vee \neg p(x),\ q(y) \vee \neg r(y),\ q(c)\}$
$C_0 = \neg q(z)$

9. SLD & PROLOG

- Langage : **clauses de Horn**
- Sémantique opérationnelle : **exploration en profondeur d'abord** avec retour arrière de l'arbre des déductions input ordonnées

  ↓

  stratégie incomplète (ordre des clauses est important)

Exemple 13.

P :
```
masculin(leon).
pere(leon,lucie).
mere(lucie,paul).
parent(P,E) :- pere(P,E ).
parent(M,E) :- mere(M,E).
grand_pere(G,E) :- pere(G,P ),parent(P,E).
```
Q :
```
:- grand_pere(G,paul)
```
$\mathcal{C}'$: $masculin(leon) \ \wedge \ pere(leon, lucie) \ \wedge \ mere(lucie, paul) \ \wedge$
$\{\forall P \ \forall E \ pere(P, E) \Rightarrow parent(P, E)\} \ \wedge$
$\{\forall M \ \forall E \ mere(M, E) \Rightarrow parent(M, E)\} \ \wedge$
$\{\forall G \ \forall P \ \forall E \ pere(G, P) \wedge parent(P, E) \Rightarrow grand\_pere(G, E))\}$

C_0 : $\neg \ (\exists G \ grand\_pere(G, paul))$