Chapitre 4
Partie 1
Logique des Prédicats

- Introduction
- Syntaxe
- Sémantique
- Variables
- Traduction
- Substitution
- Méthodes de preuves

Introduction (1/2)

Tout est homme mortel

Socrate est un homme

Donc Socrate est mortel

- Ce raisonnement ne peut être vérifié en Logique propositionnelle.
- On peut tout juste écrire :
  - ms : Socrate est mortel
  - hs : Socrate est un homme
  - lhm : les hommes sont mortels
  - $hs \wedge lhm \Rightarrow ms$
- Si on veut appliquer ce raisonnement à Pluton, on doit définir d'autres propositions atomiques indépendantes : Ceci est du à l'abscence de variables.

Introduction (2/2)

- On introduit alors :
  - La notion de prédicat,
  - Des quantificateurs $\forall$ et $\exists$,
  - Des variables de quantification.

quantificateur prédicats Variable de quantification

$$\begin{array}{l} \forall x: [\text{homme}(x) \Rightarrow \text{mortel}(x)] \\ \text{homme}(\text{Socrate}) \\ \hline \text{mortel}(\text{Socrate}) \end{array}$$

$$\begin{array}{l} \forall x: [\text{homme}(x) \Rightarrow \text{mortel}(x)] \\ \text{homme}(\text{Pluton}) \\ \hline \text{mortel}(\text{Pluton}) \end{array}$$

4 Syntaxe

Symboles

- Un ensemble de constantes $\mathcal{C}$={a, b, c, …}. Ex. Marie, 41, maison, rouge, vrai, faux …
- Un ensemble de variables $\mathcal{X}$={x, y, z, …}.
- Un ensemble de symboles de fonctions $\mathcal{F}$={f, g, h, …} d'arité n.
- Un ensemble de symboles de prédicats (relations) $\mathcal{P}$={P, Q, R, …} d'arité n.
- Des parenthèses (et) et des crochets [et].
- Deux quantificateurs : universel $\forall$ et existentiel $\exists$.

Termes

- Tout symbole de constante ou de variable est un terme. Ex. *Julie*, *Sociologie*, *x*.
- Si f est un symbole de fonction d'arité n et t1, …,tn sont des termes, alors f(t1,…,tn) est un terme. Ex. *père(Julie)*, *père(père(Julie))*, *père(x)*, *cours(logique)*.
- Un terme sans variables est un terme clos.

Atomes

- Une formule atomique est une formule de la forme P(t1,…,tn) où P est un symbole de prédicat et t1,…,tn sont des termes. Ex. *père(Ali,Fatma)*.

- t1=t2 est une formule atomique.

Formules bien formées (FBF)

- Toute formule atomique est une FBF.
- Si F est une FBF alors $\neg F$ est une FBF.
- Si F et G sont des FBF alors :
  - $F \wedge G$ est une FBF.
  - $F \vee G$ est une FBF.
  - $F \Rightarrow G$ est une FBF.
- Si F est une FBF alors (F) est une FBF.
- Si F est une FBF alors
  - $\forall x$: [F] est une FBF,
  - $\exists x$: [F] est une FBF.

9

Sémantique (Interprétation)

Interprétation

- L'interprétation de la quantification universelle est souvent comprise comme une conjonction :
  - $\forall x$:[$étudiant(x) \Rightarrow intelligent(x)$] et est interprété par $intelligent(et_1) \wedge intelligent(et_2) \wedge \ldots \wedge intelligent(et_n)$.

- L'interprétation de la quantification existentielle est souvent comprise comme une disjonction :
  - $\exists x$:[$étudiant(x) \wedge intelligent(x)$] est interprété par $intelligent(et_1) \vee intelligent(et_2) \vee \ldots \vee intelligent(et_n)$.

Variables

Variable libre et variable liée

- Une occurence d'une variable x dans une FBF est dite liée ssi elle se trouve dans la portée d'une variable de quantification x, sinon elle est dite libre.
- Une variable x est dite libre dans une FBF F ssi x a au moins une occurrence libre dans F.
- Une variable x est dite liée dans une FBF F ssi x a au moins une occurrence liée dans F.

Exemple

Soit la formule suivante :

$$\forall x: \forall y: [enfant(x) \wedge bonbon(y) \Rightarrow aprécie(x,y)] \wedge père(x,z)$$

Identifiez en donnant les explications nécessaires :

1. Les occurrences libres des variables.
2. Les occurrences liées des variables.
3. Les variables libres.
4. Les variables liées.

Traduction

Traduction

- Tous les oiseaux volent
- Quelqu'un aime Valentine
- Personne n'aime Quentin
- Quelqu'un aime tout le monde
- Franck est brillant dans toutes les matières
- Tout le monde déteste les voleurs
- Tout le monde déteste quelqu'un

Substitution

Substitution (1/2)

- Une substitution est un ensemble d'associations entre des variables et des termes : x_i/t_i (x_i: variable, t_i: terme).
- Φ: formule et $s=\{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$: substitution.
  - $\Phi s=\Phi'$:application de s à Φ (instance de Φ pour s),
  - donc $\Phi' = \Phi$ dans laquelle toutes les occurrences de x_i sont remplacées par t_i dans Φ.
- Exemples :
  - $\Phi=P(x,x,y,v)$ et $s=\{x/a,y/F(b),z/w\}$, $\Phi s = P(a,a,F(b),v)$.
  - $\Phi=aime(x,y)$ $s=\{y/père(z)\}$, $\Phi s= aime(x,père(z))$.

Substitution (2/2)

- Une constante ne peut être substituée (remplacée) : ~~c/a~~ ou ~~c/x~~
- Une fonction ne peut être substituée (remplacée) : ~~f(x)/a~~ ou ~~f(x)/y~~
- Si C est une clause $\{\Phi_1, \Phi_2, \ldots, \Phi_n\}$, C$s$ est la clause $\{\Phi_1 s, \Phi_2 s, \ldots, \Phi_n s\}$.

Remarque

- Une substitution d'un terme t pour une variable x : s={x/t}, dans une formule F est correcte ssi t est libre pour x dans F.
- Le terme t est dit libre pour la variable x dans F si toutes les occurrences des variables dans t ont des occurrences libres dans F*s*.
- Exemples :
  - Φ = ∃y:[aime(x,y)] et *s*={x/père(y)}.
  - Le terme t = père(y) n'est pas libre pour x dans Φ*s*.
  - ~~Φ*s* = ∃y:[aime(père(y),y)]~~: y n'est pas libre pour x dans Φ*s*.
  - Φ = ∃x:[aime(y,x)] et *s*={y/père(z)}.
  - Φ*s*= ∃x:[aime(père(z),x)]
  - Car libre père(z) pour y dans Φ.

Preuve

Principe de résolution

Définitions

- Terme :
  - Les constantes et les variables.
  - Les applications des fonctions à des termes : si $t_1, \ldots, t_n$ sont des termes et f une fonction à n arguments, alors $f(t_1, \ldots, t_n)$ est un terme. ex. $pere(Hela)$, $pere(x)$, $pere(pere(x))$.
- Littéral :
  - littéral positif : un terme atomique, ex. $méchant(Fadhel)$,
  - littéral négatif : la négation d'un terme, ex. $\neg grand(Yahia)$.

Définitions (Suite)

- Clause : une disjonction de littéraux :

  $père(Ali,Hela) \vee \neg père(Ali,Fatma) \vee \neg père(Ali,Faten).$

- Clause de Horn : Clause avec au plus un littéral positif.

- Forme clausale : un ensemble formé par les clauses d'une formule sous forme de conjonction de clauses :

  - $\Phi=(\neg P(x) \vee Q(x,y) \vee \neg r(v)) \wedge (P(z) \vee Q(x,z)) \wedge (\neg P(y) \vee R(y))$
  - Forme clausale de Φ : $\{\neg P(x) \vee Q(x,y) \vee \neg r(v), P(z) \vee Q(x,z), \neg P(y) \vee R(y)\}$

Forme clausale

Théorème :Toute formule de la logique des prédicats admet une formule en forme clausale qui lui est équivalente.

Algo de mise sous forme clausale

- Elimination des connecteurs $\Rightarrow$ et $\equiv$:
  - $X \equiv Y \equiv \left(X \Rightarrow Y\right) \wedge (Y \Rightarrow X)$
  - $X \Rightarrow Y \equiv \neg X \vee Y$

- Réduction de la portée de $\neg$:
  - $\neg\neg X \equiv X$
  - $\neg\left(X \vee Y\right) \equiv \neg X \wedge \neg Y$
  - $\neg\left(X \wedge Y\right) \equiv \neg X \vee \neg Y$
  - $\neg\forall x: [X] \equiv \exists x: [\neg X]$
  - $\neg \exists x:[X] \equiv \forall x: [\neg X]$

Algo de mise sous forme clausale (suite)

- Renommage des variables liées :

  $\forall x:[P(x,x)] \wedge \ \exists x:[Q(x)] \vee R(x,y,z)$

  devient

  $\forall w:[P(w,w)] \wedge \ \exists t:[Q(t)] \vee R(x,y,z)$

- Préfixage (forme prénexe) :

  $\forall x:[\exists y:[P(x,y)] \wedge \forall z:[Q(x,z)]]$

  devient

  $\forall x: \exists y: \forall z: [P(x,y) \wedge Q(x,z)]$

Algo de mise sous forme clausale (suite)

- Elimination du quantificateur existentiel
  - $\exists x{:}\forall y{:}[P(x,y) \wedge Q(x)]$

    devient

    $\forall y{:}[P(a,y) \wedge Q(a)]$ *avec a constante de skolem*
  - $\forall x{:}\ \exists y{:}[P(x,y) \wedge Q(x)]$

    devient

    $\forall x{:}[P(\mathrm{x},f(x)) \wedge Q(x))]$ *avec f fonction de skolem*
- Elimination du quantificateur universel
  - $\forall x{:}\forall y{:}\ [P(x,y)]$

    devient

    $P(x,y)$

Algo de mise sous forme clausale (suite)

- Mettre sous forme conjonctive :
  - $X \vee (Y \wedge Z) \equiv (X \vee Y) \wedge (X \vee Z)$
  - $X \wedge (Y \vee Z) \equiv (X \wedge Y) \vee (X \wedge Z)$
- Mettre sous forme de clause :

  $X \wedge (Y \vee Z)$ *devient* $\{X, Y \vee Z\}$
- Renommer les variables des clauses :

  $\{P(x,y) \vee \neg Q(y), R(x,y)\}$

  Devient

  $\{P(x_1,y_1) \vee \neg Q(y_1), R(x_2,y_2)\}$

Exercice

- Quiconque est romain et connait Marcus déteste César ou bien croit fou tout individu qui déteste au moins une personne.

Correction (1/2)

$\forall x{:}[(romain(x) \wedge connait(x,Marcus)) \Rightarrow$
$\quad (déteste(x,César) \vee \forall y{:}[\exists z{:}[déteste(y,z)] \Rightarrow croitfou(x,y)]])]$

- Eliminer les connecteurs ⇒ :

$\forall x{:}[\neg(romain(x) \wedge connait(x,Marcus)) \vee$
$\quad (déteste(x,César) \vee \forall y{:}[\neg\exists z{:}[déteste(y,z)] \vee croitfou(x,y)]])]$

- Distribuer les ¬ :

$\forall x{:}[\neg romain(x) \vee \neg connait(x,Marcus) \vee$
$\quad (déteste(x,César) \vee \forall y{:}[\forall z{:}[\neg déteste(y,z)] \vee croitfou(x,y)]])]$

- Renommer les variables liées : rien à faire.
- Obtenir la forme prénexe :

$\forall x{:}\forall y{:}\forall z{:}\ [\neg romain(x) \vee \neg connait(x,Marcus) \vee$
$\quad déteste(x,César) \vee \neg déteste(y,z) \vee croitfou(x,y)]$

Correction (2/2)

- Eliminer les quantificateurs existentiels : rien à faire.
- Eliminer les quantificateurs universels :

¬romain(x) ∨ ¬connait(x,Marcus) ∨
déteste(x,César)∨ ¬déteste(y,z)∨croitfou(x,y)

- Mettre sous forme conjonctive : rien à faire.
- Transformer chaque facteur en clause (ici une seule clause) :

¬romain(x) ∨ ¬connait(x,Marcus) ∨
déteste(x,César)∨¬déteste(y,z)∨croitfou(x,y)

Exercice (mise sous forme clausale)

$$\forall x:[\forall y:[P(x,y)] \Rightarrow \neg \forall y:[Q(x,y) \Rightarrow R(x,y)]]$$

Unification de littéraux (1/2)

- Un ensemble de littéraux $\Gamma=\{\Phi_1, \Phi_2,\ldots, \Phi_n\}$ est dit unifiable ssi il existe une substitution s telle que :
  - $\Phi_1 s = \Phi_2 s = \ldots = \Phi_n s$.
  - s est appelé *unificateur* de Γ.
- $\Phi_1=P(a,y,z)$, $\Phi_2=P(x,b,z)$ sont unifiables :
  - $s=\{x/a, y/b\}$ une substitution unifiant Φ_1 et Φ_2.
  - $\Phi_1 s= \Phi_2 s = P(a,b,z)$.
- $\Phi_1=Pere(x,Hela)$, $\Phi_2=Pere(Ali,x)$ ne sont pas unifiables.

Unification de littéraux (2/2)

- Un unificateur s_1 de Γ est dit unificateur le plus général (UPG) de Γ ssi pour tout unificateur s_2 de Γ, il existe une substitution s tel que $s_2 = s_1\, s$.
- Exemple :
  - $\Gamma=\{\Phi_1, \Phi_2\}$, $\Phi_1=P(a,y,z)$ et $\Phi_2=P(x,b,z)$.
  - $s_1 =\{x/a,y/b\}$ est un unificateur de Γ.
  - $s_2 =\{x/a,y/b,z/c\}$ est aussi un unificateur de Γ.
  - On dit que s_1 est plus général car $s_2 =s_1 s$ ($s=\{z/c\}$).

Principe de résolution

- Déduction en forme clausale.
- Règle d'inférence qui produit une clause qui est une conséquence logique des clauses dont elle est issue.
- Règle de résolution :

$$\frac{X \vee A \quad \neg X \vee B}{A \vee B}$$

- C_1 et C_2 clauses, si un littéral L_1de C_1 est complémentaire de L_2 ($\neg L_1$) de C_2 (après unification),C est la clause résolvante : contient tous les littéraux de C_1 et C_2 sauf L_1 et L_2.

Exemple

- $\Gamma=\{$Père(Saleh,Olfa),Parent(Kamel,Rim),$\neg$Père(x,y)$\vee$Parent(x,y)$\}$
- Parent(Saleh,Olfa) ?
- Preuve :

1. $\neg$Père(x,y)$\vee$Parent(x,y) Φ_3
2. $\neg$Père(Saleh,Olfa)$\vee$Parent(saleh,Olfa) 1,x/Saleh,y/Olfa
3. Père(Saleh,Olfa) Φ_1
4. Parent(saleh,Olfa) 2,3

Principe de résolution

Principe de résolution pour la déduction

Exemple 1

Formules en logique des prédicats

1. Marcus est une personne.
2. Marcus est un pompéien.
3. Tous les pompéiens sont des romains.
4. César est un dirigeant.
5. Tout le monde est loyal à quelqu'un.
6. Tous les romains sont loyaux à César ou le haïssent.
7. Les seuls dirigeants qu'une personne essaie d'assassiner sont ceux auxquels elle n'est pas loyale
8. Marcus a essayé d'assassiner César.

Prouvez que Marcus hait César

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x[pompeien(x) \Rightarrow romain(x)]$
4. $dirigeant(Cesar)$
5. $\forall x:\exists y : [loyal(x,y)]$
6. $\forall x:[romain(x) \Rightarrow loyal(x,Cesar) \vee hait(x,Cesar]))$
7. $\forall x \forall y:[(personne(x) \wedge dirigeant(y) \wedge assassiner(x,y)) \Rightarrow \neg loyal(x,y)]$
8. $assassiner(Marcus,Cesar)$

Prouvez : hait(Marcus,Cesar)

Etape 1: Elimination de l'implication

1. personne(Marcus)
2. pompeien(Marcus)
3. $\forall x:[(pompeien(x) \Rightarrow romain(x))]$
4. $dirigeant(Cesar)$
5. $\forall x \exists y: [loyal(x,y)]$
6. $\forall x:[romain(x) \Rightarrow loyal(x,Cesar) \vee hait(x,Cesar)]$
7. $\forall x\ \forall y:[personne(x) \wedge dirigeant(y) \wedge assassiner(x,y)) \Rightarrow \neg loyal(x,y)]$
8. $assassiner(Marcus,Cesar)$

1. personne(Marcus)
2. pompeien(Marcus)
3. $\forall x:[\neg pompeien(x) \vee romain(x)]$
4. $dirigeant(Cesar)$
5. $\forall x \exists y : [loyal(x,y)]$
6. $\forall x:[\neg romain(x) \vee loyal(x,Cesar) \vee hait(x,Cesar)]$
7. $\forall x\ \forall y:[\neg (personne(x) \wedge dirigeant(y) \wedge assassiner(x,y)) \vee \neg loyal(x,y)]$
8. $assassiner(Marcus,Cesar)$

Etape 2: Réduire la portée de la ¬

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x{:}[\neg pompeien(x) \vee romain(x)]$
4. $dirigeant(Cesar)$
5. $\forall x \exists y : [loyal(x,y)]$
6. $\forall x{:}[\neg romain(x) \vee loyal(x,Cesar) \vee hait(x,Cesar)]$
7. $\forall x \ \forall y{:}[\neg (personne(x) \wedge dirigeant(y) \wedge assassiner(x,y)) \vee \neg loyal(x,y)]$
8. assassiner(Marcus,Cesar)

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x{:}[\neg pompeien(x) \vee romain(x)]$
4. $dirigeant(Cesar)$
5. $\forall x \exists y : [loyal(x,y)]$
6. $\forall x{:}[\neg romain(x) \vee loyal(x,Cesar) \vee hait(x,Cesar)]$
7. $\forall x \ \forall y{:}[\neg personne(x) \vee \neg dirigeant(y) \vee \neg assassiner(x,y) \vee \neg loyal(x,y)]$
8. $assassiner(Marcus,Cesar)$

Etape 3: Renommer les variables

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x : [\neg pompeien(x) \vee romain(x)]$
4. $dirigeant(Cesar)$
5. $\forall x \exists y : [loyal(x,y)]$
6. $\forall x : [\neg romain(x) \vee loyal(x, Cesar) \vee hait(x, Cesar)]$
7. $\forall x \ \forall y : [\neg personne(x) \vee \neg dirigeant(y) \vee \neg assassiner(x,y) \vee \neg loyal(x,y)]$
8. $assassiner(Marcus, Cesar)$

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x_1 : [\neg pompeien(x_1) \vee romain(x_1)]$
4. $dirigeant(Cesar)$
5. $\forall x_2 \exists x_3 : [loyal(x_2,x_3)]$
6. $\forall x_4 : [\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)]$
7. $\forall x_5 \ \forall x_6 : [\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)]$
8. $assassiner(Marcus, Cesar)$

Etape 4: Mettre sous forme Prenex (Préfixage) : Rien à faire

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x_1 : [\neg pompeien(x_1) \vee romain(x_1)]$
4. $dirigeant(Cesar)$
5. $\forall x \exists y : [loyal(x,y)]$
6. $\forall x_4 : [\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)]$
7. $\forall x_5 : \forall x_6 : [\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)]$
8. $assassiner(Marcus,Cesar)$

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x_1 : [\neg pompeien(x_1) \vee romain(x_1)]$
4. $dirigeant(Cesar)$
5. $\forall x_2 : [loyal(x_2, f(x_2))]$
6. $\forall x_4 : [\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)]$
7. $\forall x_5 : \forall x_6 : [\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)]$
8. $assassiner(Marcus,Cesar)$

Etape 5: Elimination de $\exists$

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x_1:[\neg pompeien(x_1) \vee romain(x_1)]$
4. $dirigeant(Cesar)$
5. $\forall x_2 \exists x_3 : [loyal(x_2,x_3)]$
6. $\forall x_4:[\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)]$
7. $\forall x_5 \forall x_6:[\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)]$
8. $assassiner(Marcus, Cesar)$

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x_1:[\neg pompeien(x_1) \vee romain(x_1)]$
4. $dirigeant(Cesar)$
5. $\forall x_2 : [loyal(x_2,f(x_2))]$
6. $\forall x_4:[\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)]$
7. $\forall x_5 \forall x_6:[\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)]$
8. $assassiner(Marcus, Cesar)$

Etape 6: Elimination de ∀ : Rien à faire

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\forall x_1 : [\neg pompeien(x_1) \vee romain(x_1)]$
4. $dirigeant(Cesar)$
5. $\forall x_2 : [loyal(x_2, f(x_2))]$
6. $\forall x_4 : [\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)]$
7. $\forall x_5 \, \forall x_6 : [\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5, x_6) \vee \neg loyal(x_5, x_6)]$
8. $assassiner(Marcus, Cesar)$

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\neg pompeien(x_1) \vee romain(x_1)$
4. $dirigeant(Cesar)$
5. $loyal(x_2, f(x_2))$
6. $\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)$
7. $\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5, x_6) \vee \neg loyal(x_5, x_6)$
8. $assassiner(Marcus, Cesar)$

Etape 7: Forme normale conjonctive (Rien à faire)

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\neg pompeien(x_1) \vee romain(x_1)$
4. $dirigeant(Cesar)$
5. $loyal(x_2,f(x_2))$
6. $\neg romain(x_4) \vee loyal(x_4,Cesar) \vee hait(x_4,Cesar)$
7. $\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)$
8. $assassiner(Marcus,Cesar)$

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\neg pompeien(x_1) \vee romain(x_1)$
4. $dirigeant(Cesar)$
5. $loyal(x_2,f(x_2))$
6. $\neg romain(x_4) \vee loyal(x_4,Cesar) \vee hait(x_4,Cesar)$
7. $\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)$
8. $assassiner(Marcus,Cesar)$

Etape 8: Renommage (Rien à faire)

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\neg pompeien(x_1) \vee romain(x_1)$
4. $dirigeant(Cesar)$
5. $loyal(x_2, f(x_2))$
6. $\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)$
7. $\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5, x_6) \vee \neg loyal(x_5, x_6)$
8. $assassiner(Marcus, Cesar)$

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\neg pompeien(x_1) \vee romain(x_1)$
4. $dirigeant(Cesar)$
5. $loyal(x_2, f(x_2))$
6. $\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)$
7. $\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5, x_6) \vee \neg loyal(x_5, x_6)$
8. $assassiner(Marcus, Cesar)$

Etape 9: Forme clausale (Rien à faire)

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\neg pompeien(x_1) \vee romain(x_1)$
4. $dirigeant(Cesar)$
5. $loyal(x_2,f(x_2))$
6. $\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)$
7. $\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)$
8. $assassiner(Marcus, Cesar)$

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\neg pompeien(x_1) \vee romain(x_1)$
4. $dirigeant(Cesar)$
5. $loyal(x_2,f(x_2))$
6. $\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)$
7. $\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)$
8. $assassiner(Marcus, Cesar)$

Etape 10: Résolution pour la déduction

1. $personne(Marcus)$
2. $pompeien(Marcus)$
3. $\neg pompeien(x_1) \vee romain(x_1)$
4. $dirigeant(Cesar)$
5. $loyal(x_2,f(x_2))$
6. $\neg romain(x_4) \vee loyal(x_4, Cesar) \vee hait(x_4, Cesar)$
7. $\neg personne(x_5) \vee \neg dirigeant(x_6) \vee \neg assassiner(x_5,x_6) \vee \neg loyal(x_5,x_6)$
8. $assassiner(Marcus, Cesar)$

9. $romain(Marcus)$

2, 3, {x1/Marcus}

10. $loyal(Marcus, Cesar) \vee hait(Marcus, Cesar)$

6, 9, {x4/Marcus}

11. $\neg personne(Marcus) \vee \neg dirigeant(Cesar) \vee \neg assassiner(Marcus, Cesar) \vee hait(Marcus, Cesar)$

7, 10, {x5/Marcus, x6/Cesar}

13. $\neg personne(Marcus) \vee \neg dirigeant(Cesar) \vee hait(Marcus, Cesar)$

8,11

14. $\neg personne(Marcus) \vee hait(Marcus, Cesar)$

4,13

16. $hait(Marcus, Cesar)$

1, 14 (but prouvé)