Logique du Premier Ordre
Sémantique

1

Plan

2

1- Introduction

Considérons la formule:

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

est-elle vraie ou fausse?

La valeur de vérité de la formule dépend de comment on "lit" le symbole p et du domaine de discours considéré.

1- Introduction

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

Exemple-1

domaine = N
p(x,y) = $x \leq y$
La formule se lit:
« il existe un entier naturel inférieur ou égal à tous les entiers »
La formule est VRAIE

Exemple-2

domaine = N
p(x,y) = $x < y$
La formule se lit:
« il existe un entier naturel inférieur strictement à tous les entiers »
La formule est FAUSSE (car on n'a pas $0 < 0$)

Exemple-3

domaine = Z
p(x,y) = $x \leq y$
La formule se lit:
« il existe un entier relatif inférieur ou égal à tous les entiers relatifs»
La formule est FAUSSE

1- Introduction

- **Le but de la sémantique de la logique du 1er ordre est de:**

  - **Donner une signification aux symboles de prédicats**
  - **Donner une signification et aux symboles de fonctions**
  - **Fixer le domaine dans lequel les variables prennent valeur**

Afin d'établir la valeur de vérité (vrai ou faux) des formules.

2- Notion de Structure

Définition

On appelle structure tout quadruplet S = (D, C, F, R) où:

D: est un ensemble *non vide* appelé domaine de discours

C: un ensemble (vide ou non) de constantes

F: un ensemble (vide ou non) de fonctions sur D

R: un ensemble (vide ou non) de relations sur D

2- Notion de Structure

Exemple

(D,C,F,R) telle que
D=N
C={0}
F={+,*, succ}
R={≤}

Sont des structures

(D,C,F,R) telle que
D=Réels
C={0, 1}
F={+, - , *}
R=∅
Est une structure.

2- Notion de Structure

Exemple

(D,C,F,R) telle que
D=N
C={0}
F={+, - , succ}
R={≤}

N'est pas une structure
car la soustraction n'est pas interne dans N

(D,C,F,R) telle que
D=∅
C={0}
F={+,*}
R={≤}

N'est pas une structure
car le domaine est vide

2- Notion de Structure

Donner une signification

- **aux symboles de prédicats**
- **aux symboles de fonctions**
- **aux symboles de constantes**
- **et Fixer le domaine dans lequel les variables prennent valeur**

revient à associer au langage considéré (de la logique du 1er ordre) une structure.

9

3- Interprétation d'un langage

Définition

Soit L un langage défini par :
les constante c1,c2,..
les symboles de prédicats p1,p2,….
les symboles fonctionnels f1 f2,….

Soit S=(D,C, F, R) une structure

Une interprétation de L dans S consiste à associer :

- **A chaque constante ci de L un élément de C noté $[ci]_S$**
- **A chaque prédicat pi de L de poids n une relation n-aire de R notée $[pi]_S$.**
- **A chaque symbole fonctionnel fi de L de poids n, une fonction de F notée**

$[fi]_S : D^n \rightarrow D$.

10

3- Interprétation d'un langage

Exemple-1: L1 : - constante c
- prédicat p de poids 2

S1 = (N,{0}, ∅, {<})

$[c]_{S1}=0$,

$[p(x,y)]_{S1}=x<y$

S2= ({Mohamed, Ali, Madjid }, {Mohamed},∅,{_est-frère-de_})

$[c]_{S2}$=Mohamed,

$[p(x,y]_{S2}$= x est-frère-de y)

S3 = ({les amphis}, {A10},∅ ,{_est.dans.le.même.couloir.que.})

$[c]_{S3}$=A10,

$[p(x,y)]_{S3}$= x est-dans-le-même-couloir-que y)

11

3- Interprétation d'un langage

Exemple-2: L2: - 2 constantes c1 c2
- un prédicat p de poids 2
-un symbole fonctionnel f de poids 1
- 2 symboles fonctionnels g et h de poids 2

S1=(R, {0 ,1}, {carrée,somme, produit},{<})

[c1]S1=0
[c2] S1=1
[p(x,y)] S1= x<y
[f(x)] S1= x2
[g(x,y)] S1= x + y
[h(x,y)] S1=x*y

12

3- Interprétation d'un langage

Exemple-2: **L2:**
- **2 constantes c1 c2**
- **un prédicat p de poids 2**
- **un symbole fonctionnel f de poids 1**
- **2 symboles fonctionnels g et h de poids 2**

S2=(vecteurs, {vecteur nul, vecteur unitaire i}, {opposé d'un vecteur, somme, différence} ,{parallèle.à})

$[c1]_{S2}$ = le vecteur nul
$[c2]_{S2}$ = le vecteur unitaire i
$[p(x,y)]_{S2}$ = x parallèle-à y
$[f(x)]_{S2}$ =opposé de x = -x
$[g(x,y)]_{S2}$ = x+y
$[h(x,y)]_{S2}$ = x- y

3- Interprétation d'un langage

Remarques:

- **Tout langage a au moins une structure comme interprétation (même si cette structure est abstraite).**
- **Un langage peut avoir plusieurs structures comme interprétations.**
- **Etant donnée un langage L et une structure S , Il peut exister plusieurs interprétations de L dans S.**

<u>Exemple</u>: L= - 2 constante c1,c2

S1 (N,{0,1},∅,∅)

2 interprétations possibles de L dans S1:

- **$[c1]_S1$ =0 et $[c2]_{S1}$=1**
- **$[c1]_{S1}$ =1 et $[c2]_{S1}$=0**

4- Valuation des variables dans une structure

Définition

Soient Var: l'ensemble des variables d'un langage L

D: le domaine d'une interprétation S de L

Une valuation V pour les variables Var dans S est une fonction

$$V: \quad Var \rightarrow D$$

qui attribut à chaque variable x de Var une valeur V(x) de D.

15

4- Valuation des variables dans une structure

Exemple

D={3,8,9}

V(x1)=9, V(x2)=3 V(x3)=8 V(xi)= 9 si i>3 est une valuation

x=y vrai si x=4 y=4 mais fausse si x=4 y=3

16

5- Interprétation de Termes

Une fois qu'on a défini la notion d'interprétation, le premier problème à considérer est : comment évaluer un terme t?

Le rôle d'un terme est celui d'indiquer un individu c'est à dire un élément du domaine de discours. Quel élément? (comment savoir lequel).

Considérons par exemple le terme f(a,g(a))
où f: est un symbole de fonction de poids2
g :un symbole de fonction de poids 1
a: une constante.

- Tout d'abord ceci dépend de l'interprétation choisie.

Par exemple, soit S l'interprétation du langage de t où

D=N $[f]_S$=+ $[g]_S$ =succ $[a]_S$ =0.

Par rapport à cette interprétation, le terme f(a,g(a)) indique la somme de l'entier 0 et de 'entier successeur de 0, donc le nombre 1.

5- Interprétation de Termes

Considérons maintenant le terme f(x,g(x))
où f: est un symbole de fonction de poids2
g :un symbole de fonction de poids 1
a: une constante.
Quelle est l'évaluation du terme ?

Ceci dépend de l'entier associé à x.

Une définition précise de la valeur d'un terme t dont les variables sont x1,x2,…xn , par rapport à une interprétation S, doit donc tenir compte des éléments du domaine de discours a1,a2,..an que l'on a choisi d'associer aux variables

5- Interprétation de Termes

Définition

Soient L : un langage

S: une structure interprétation de L

t: un terme de L

V: une valuation des variables de t

L'interprétation du terme t (dite aussi valeur de t) notée $[t]_{S,v}$ est définie par:

- **Si t est une constante c alors $[t]_{S,v} = [c]_S$**
- **Si t est une variable x alors $[t]_{S,v} = v(x)$**
- **Si t est de la forme f(t1,..,tn) et $[ti]_{S,v}$ =bi alors**

$$[t]_{S,v} = [f]_S ([t1]_{S,v} , [t2]_{S,v} ,\ldots[tn]_{S,v})$$

5- Interprétation de Termes

Exemple

L: - 2 constantes c1 c2
- un symbole fonctionnel f de poids 1
- 2 symboles fonctionnels g et h de poids 2

S =(D,C,F,R) une structure pour L telle que:

D= N

$[c1]_S$=0 $[c2]_S$=1

$[f(x)]_S=x^2$

$[g(x,y)]_S$=x+y

$[h(x,y)]_S$=x*y

V une valuation telle que V(x)=3, V(y) =4, V(z)=6

t1= g(y,h(c1,x)) $[t1]_{S,v}$ = 4 + (0*3) =4

t2= f(g(c2, h(y,z))) $[t2]_{S,v} = (1 + (4*6))^2 = 25^2$ =725

6- Interprétation de Formules

Définition

Soit φ : une formule d'un langage L,

S: une interprétation du langage L ayant le domaine D.

V: une valuation des variables par rapport à S

L'interprétation de φ par rapport à S et V noté $[\varphi]_{S,v}$ est définie par:

- Si $\varphi = p(t1,t2,..tn)$ alors $[\varphi]_{S,v} = [p]_S\ ([t1]_{S,v}, [t2]_{S,v}, ..[tn]_{S,v})$
- Si $\varphi = t1 \equiv t2$ alors $[\varphi]_{S,v} = [t1]_{S,v} \equiv [t2]_{S,v}$
- Si $\varphi = \rceil \varphi1$ alors $[\varphi]_{S,v} = \rceil([\varphi1]_{S,v})$
- Si $\varphi = \varphi1\ K\ \varphi2$ $K \in \{\wedge, v, \rightarrow, \leftrightarrow\}$ alors $[\varphi]_{S,v} = [\varphi1]_{S,v}\ K\ [\varphi2]_{S,v}$
- Si $\varphi = \forall x\ \varphi1$ alors $[\varphi]_{S,v} = \wedge_{a \in D}\ ([\varphi1]_{S,v})[a/x]$
- Si $\varphi = \exists 8 \varphi1$ alors $[\varphi]_{S,v} = V_{a \in D}\ ([\varphi1]_{S,v})[a/x]$

21

7- Satisfaisabilité

Satisfaisabilité d'une formule pour une structure est une valuation

Définition

Soient L un langage du 1er ordre
φ : une formule construite sur L
S : une structure interprétation de L
V : une valuation pour l'ensemble des variables Var

φ est satisfaite dans S pour la valuation V $\Leftrightarrow$ $[\varphi]_{S,v} = 1$

φ est non satisfaite dans S pour la valuation V $\Leftrightarrow$ $[\varphi]_{S,v} = 0$

Exemple

$\forall y\ p(x,y)$ est satisfaite pour $S=(N, \varnothing, \varnothing, \{\leq\})$ et la valuation V / $V(x)=0$

$\forall y\ p(x,y)$ est non satisfaite pour $S=(N, \varnothing, \varnothing, \{\leq\})$ et la valuation V/ $V(x)=5$

22

7- Satisfaisabilité

Satisfaisabilité d'une formule pour une structure

Définition

Soient L un langage du 1er ordre
φ : une formule construite sur L
S : une structure interprétation de L

φ satisfaite pour S ⇔ ∃ V telle que $[\varphi]_{S,V} = 1$

φ non satisfaite pour S ⇔ ∀V on a $[\varphi]_{S,V} = 0$

Exemple

∀y p(x,y) est satisfaite pour S=(N, ∅, ∅ ,{≤ }) (V(x)=0)

∀y p(x,y) non satisfaite pour S=(N, ∅, ∅ ,{≥ })

7- Satisfaisabilité

Satisfaisabilité d'une formule

Définition

Soient L un langage du 1er ordre
φ : une formule construite sur L

φ satisfaite ⇔ ∃ S ∃ V telles que $[\varphi]_{S,V} = 1$

φ non satisfaite ⇔ ∀S ∀V $[\varphi]_{S,V} = 0$

Exemple

∀y p(x,y) est satisfaite (S=(N, ∅, ∅ ,{≤ }) V(x)=0)

∀y (p(x,y) ∧ ⌉p(x,y)) est non satisfaite

8- Validité

Validité d'une formule dans une structure

Définition

Soient L un langage du 1er ordre
φ : une formule construite sur L
S : une structure interprétation de L

φ valide dans une structure S ⇔ ∀V $[\varphi]_{S,V} = 1$

φ non valide dans une structure S ⇔ ∃V $[\varphi]_{S,V} = 0$

Exemple

∀y p(a,y) est valide dans S=(N, {0}, ∅ ,{≤ })

∀y p(x,f(x)) est valide dans S=(N, ∅,{succ}, {≤ })

8- Validité

Validité d'une formule dans une structure

Théorème
Si φ est une formule close (sans variables libres) alors

φ valide ⇔ φ satisfaite

Démonstration:

⇒ ? φ valide dans S ⇒ φ satisfaite (évident)

⇐? $\left\{\begin{array}{l}\varphi \text{ satisfaite dans } S \\ \varphi \text{ close}\end{array}\right\} \Rightarrow \left\{\begin{array}{l}\exists v\ [\varphi]_{S,v} = 1 \\ \forall v\ [\varphi]s,v = [\varphi]s \text{ (pas de variables libres dans } \varphi)\end{array}\right\}$

⇒ ∀v [φ]s,v =1

⇒ φ valide dans S

8- Validité
Validité (universelle) d'une formule

Définition

Soient L un langage du 1er ordre
φ : une formule construite sur L

φ est (universellement)valide $\Leftrightarrow \forall S\ \ \varphi$ est valide dans S

φ est non valide $\Leftrightarrow \exists S\ \ \varphi$ est non valide dans S

Exemple

$\forall x\ p(x) \rightarrow \exists y\ p(y)$ est valide

$\exists x \forall y\ \varphi(x,y) \rightarrow \forall y \exists x\ \varphi(x,y)$ est valide

$\forall y \exists x\ \varphi(x,y) \rightarrow \exists x \forall y\ \varphi(x,y)$ n'est pas valide

9- Thèse et tautologie
Thèse

Définition

Soient L un langage du 1er ordre
φ : une formule construite sur L

φ thèse $\Leftrightarrow\ \varphi$ valide
φ thèse $\Leftrightarrow \forall S\ \varphi$ valide dans S

Exemple

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

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

9- Thèse et tautologie

Tautologie

Définition

Soient L un langage du 1er ordre
φ : une formule construite sur L

φ est une tautologie s'il existe une tautologie φ' de la logique propositionnelle telle que φ se mette sous la forme de φ'

Exemples

$\varphi = p(x) \vee \neg p(x)$ est une tautologie
car φ a la forme de $A \vee \neg A$ qui est une tautologie en logique propositionnelle

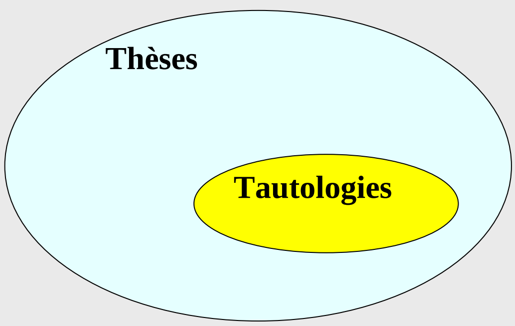
$\varphi = p(x,y) \rightarrow \forall x\, r(x) \vee \neg \forall x\, r(x)$ est une tautologie
car elle a la forme $A \rightarrow B \vee \neg B$ qui est une tautologie en logique propositionnelle

9- Thèse et tautologie

Remarque:

φ tautologie $\rightarrow$ φ est une thèse

(mais l'inverse n'est pas vrai)



10- Formules équivalentes

Définition

Soient L un langage du 1er ordre

$\varphi 1$, $\varphi 2$: des formules construites sur L

$\varphi 1$ et $\varphi 2$ équivalentes $\Leftrightarrow$ $\forall S\ \forall v$ on a $[\varphi 1]_{S,V} = [\varphi 2]_{S,V}$

$\varphi 1$ et $\varphi 2$ équivalentes $\Leftrightarrow$ la formule $\varphi 1 \leftrightarrow \varphi 2$ est valide

$\varphi 1$ et $\varphi 2$ non équivalentes $\Leftrightarrow$ $\exists S\ \exists v$ on a $\neg([\varphi 1]_{S,V} = [\varphi 2]_{S,V})$

Exemples

- **$\forall x\, F$ et $\neg\exists x \neg F$ équivalentes (propriété de définissabilité de $\forall$ par $\exists$)**
- **$\exists x\, F$ et $\neg\forall x \neg F$ équivalentes (propriété de définissabilité de $\exists$ par $\forall$)**

31

10- Formules équivalentes

Exemples

φ et $\forall x \varphi$ si x liée dans φ

φ et $\exists x \varphi$ si x liée dans φ

$\forall x(\varphi 1 \wedge \varphi 2)$ et $\forall x \varphi 1 \wedge \forall x \varphi 2$

$\exists x(\varphi 1 \vee \varphi 2)$ et $\exists x \varphi 1 \vee \exists x \varphi 2$

$\exists x(\varphi 1 \rightarrow \varphi 2)$ et $\exists x(\neg\varphi 1 \vee \varphi 2)$

$\exists x \varphi$ et $\exists y\ \varphi[y/x]$ si x libre dans φ et y n'apparaît pas dans φ

$\forall x \varphi$ et $\forall y \varphi[y/x]$ si x est libre dans φ et y n'apparaît pas dans φ

$\forall x(\varphi 1 \wedge \varphi 2)$ et $\forall x \varphi 1 \wedge \varphi 2$ si x liée dans $\varphi 2$

$\exists x(\varphi 1 \wedge \varphi 2)$ et $\exists x \varphi 1 \wedge \varphi 2$ si x liée dans $\varphi 2$

$\forall x(\varphi 1 \vee \varphi 2)$ et $\forall x \varphi 1 \vee \varphi 2$ si x liée dans $\varphi 2$

$\exists x(\varphi 1 \vee \varphi 2)$ et $\exists x \varphi 1 \vee \varphi 2$ si x liée dans $\varphi 2$

32

11- Modèles

Modèle d'une formule

Définition

Soient - L un langage du 1er ordre

- φ une formule de L.

Les modèles de φ sont les structures interprétations de L dans lesquelles φ est valide

S modèle de φ $\Leftrightarrow$ φ valides dans S

Notation **S $\models$ φ**

11- Modèle

Modèle d'un ensemble de formules

Définition

Soit Σ: un ensemble de formules d'un langage L

S: une réalisation de L

S est u. modèle Σ $\Leftrightarrow$ $\forall\varphi \in \Sigma$ S I==φ

Notation **S $\models$ Σ**

12 Compatibilité

Compatibilité d'une formule

Définition

Une formule est compatible si elle a au moins un modèle

$$\varphi \text{ compatible } \Leftrightarrow \exists S \;\; S \models \varphi$$

Compatibilité d'un ensemble de formules

Définition

Un ensemble de formules d'un langage est compatible ssi il a au moins un modèle.

$$\Sigma \text{ compatible} \Leftrightarrow \exists S \;\; S \models \Sigma$$

13 Conséquence logique

Définition

Soit Σ: un ensemble de formules.
φ: une formule.
φ conséquence de Σ ssi tout modèle de Σ est un modèle de φ

$$\varphi \text{ est conséquence de } \Sigma \Leftrightarrow \forall S\ (S \models \Sigma \rightarrow S \models \varphi)$$

$$\varphi \text{ est conséquence de } \Sigma \Leftrightarrow \forall S\ \forall v\ (\ \forall \varphi i \in \Sigma\ [\varphi i]_{S,v}=1 \rightarrow [\varphi]_{S,v}= 1)$$

Notation $\Sigma \vdash \varphi$

$$\varphi \text{ non conséquence de } \Sigma \Leftrightarrow \exists S\ (S \models \Sigma\ \wedge \neg (S \models \varphi))$$

Remarque: φ **thèse** $\Leftrightarrow \;\; \vdash \varphi$

$\varphi 1$ **et** $\varphi 2$ **équivalentes** $\Leftrightarrow (\varphi 1 \vdash \varphi 2 \;\text{ et }\; \varphi 2 \vdash \varphi 1)$

14- Formes normales des formules de la logique du 1er ordre

Parmi les formes normales les plus connues on a:

- **Forme normale Prénexe (FNP)**
- **Forme normale de Skolem (FNS)**
- **Forme normale d'Herbrand (FNH)**
- **Forme normale Clausale (FNCl)**

37

14- Formes normales des formules de la logique du 1er ordre

1- Forme Normale Prénexe (FNP)

Définition

Une formule φ est sous forme normale Prén%xe (FNP) ssi elle a la forme:

$$\boxed{\varphi = Q_1 x_1 \ldots Q_n x_n \ \varphi'}$$

où
$$\begin{cases} n \geq 0 \\ \forall i=1..n \ \ Q_i \text{ est un quantificateur et } x_i \text{ une variable} \\ \varphi' \text{ est une formule qui } \underline{\textit{ne contient aucun quantificateur}}. \end{cases}$$

- Une formule a la forme prénexe lorsque tous ses quantificateurs sont placés en avant

- La suite des quantificateurs $Q_1x_1Q_2x_2\ldots Q_nx_n$ est appelée _préfixe_

- φ' est appelée _matrice_

38

1- Forme Normale Prénexe (FNP)

Exemples

$p(x) \wedge \neg p(y)$ **est sous FNP**

$\forall x \exists y\ ((p(x) \wedge q(x,y)) \vee r(z))$ **est sous FNP**

$\exists x\ ((p(x) \to \forall x\ p(x))$ **n'est pas sous FNP**

Théorème:

Pour chaque formule φ il existe une formule φ' qui est logiquement équivalente et qui est sous formes normale prénexe .

14- Formes normales des formules de la logique du 1er ordre

1- Forme Normale Prénexe (FNP)

Méthode de Mise sous Forme normale Prénexe

1- Eliminer les connecteurs → et ↔ en appliquant les équivalences:

$\varphi 1 \to \varphi 2 = \neg \varphi 1 \vee \varphi 2$ $\qquad \varphi 1 \leftrightarrow \varphi 2 = (\neg \varphi 1 \vee \varphi 2) \wedge (\varphi 1 \vee \neg \varphi 2)$

2- Faire entrer les ¬ vers l'intérieur en appliquant les équivalences:

$\neg \forall x \varphi = \exists x \neg \varphi$ $\qquad \neg \exists x \varphi = \forall x \neg \varphi$

$\neg\neg \varphi 1 = \varphi 1$ $\qquad \neg(\varphi 1 \wedge \varphi 2) = \neg \varphi 1 \vee \neg \varphi 2$ $\qquad \neg(\varphi 1 \vee \varphi 2) = \neg \varphi 1 \wedge \neg \varphi 2$

14- Formes normales des formules de la logique du 1er ordre

1- Forme Normale Prénexe (FNP)

3- Appliquer itérativement les équivalences suivantes pour " faire sortir les quantificateurs"

Si x non libre dans $\varphi 2$

$(\forall x \varphi 1) \vee \varphi 2 = \forall x(\varphi 1 \vee \varphi 2)$	$(\exists x \varphi 1) \vee \varphi 2 = \exists x(\varphi 1 \vee \varphi 2)$
$\varphi 2 \vee (\forall x \varphi 1) = \forall x(\varphi 2 \vee \varphi 1)$	$\varphi 2 \vee (\exists x \varphi 1) = \exists x(\varphi 2 \vee \varphi 1)$
$(\forall x \varphi 1) \wedge \varphi 2 = \forall x(\varphi 1 \wedge \varphi 2)$	$(\exists x \varphi 1) \wedge \varphi 2 = \exists x(\varphi 1 \wedge \varphi 2)$
$\varphi 2 \wedge (\forall x \varphi 1) = \forall x(\varphi 2 \wedge \varphi 1)$	$\varphi 2 \wedge (\exists x \varphi 1) = \exists x(\varphi 2 \wedge \varphi 1)$

Renommage de variables pour être toujours dans les conditions ci-dessus.

14- Formes normales des formules de la logique du 1er ordre

1- Forme Normale Prénexe (FNP)

Exemple de mise sous FNP

$\varphi = \exists x\, (p(x) \rightarrow \forall x p(x))$

$= \exists x\, (\neg p(x) \vee \forall x\, p(x))$ élimination de $\rightarrow$

$= \exists x(\neg p(x) \vee \forall y\, p(y))$ Renommage de $\forall x p(x)$ par $\forall y\, p(y)$

$= \exists x\, \forall y(\neg p(x) \vee p(y))$ Sortir $\forall y$

sous FNP

N.B. Le renommage est essentiel ici :

$\exists x\, (\neg p(x) \vee \forall x\, p(x))$ et $\exists x\, \forall x(\neg p(x) \vee p(x))$

ne sont pas équivalentes

14- Formes normales des formules de la logique du 1er ordre

1- Forme Normale Prénexe (FNP)

Exemple de mise sous FNP

$\varphi = \neg (p(x) \rightarrow ((\exists y\ q(x,y)) \wedge r(y)))$

$= \neg (\neg p(x) \vee ((\exists y\ q(x,y)) \wedge r(y)))$ **éliminer →**

$= \neg \neg p(x) \wedge \neg ((\exists y\ q(x,y)) \wedge r(y)))$ **rentrer** ⌉ sortir ∀y

$= p(x) \wedge (\neg(\exists y\ q(x,y)) \vee \neg r(y)))$

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

$= p(x) \wedge ((\forall y\ \neg q(x,y)) \vee \neg r(z)))$ **renommage r(y) par r(z)**

$= p(x) \wedge \forall y(\neg q(x,y) \vee \neg r(z))$ **sortir ∀y**

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

sous FNP

14- Formes normales des formules de la logique du 1er ordre

1- Forme Normale Prénexe (FNP)

Formule prénexe sous FNC

Une formule prénexe $\varphi = Q_1 x_1 \ldots Q_n x_n\ \varphi'$ est sous FNC (forme normale conjonctive) ssi la matrice φ' est:

- **une disjonction de littéraux**

  (un littéral étant une formule atomique ou sa négation)

- **ou une conjonction de disjonction de littéraux**

Exemple

$\forall x \forall y \exists z(\neg r(x,y) \vee x=y \vee (y=g(x,h(z,z))))$ **est sous FNC**

14- Formes normales des formules de la logique du 1er ordre

1- Forme Normale Prénexe (FNP)

Formule prénexe sous FND

Une formule prénexe $\varphi = Q_1 x_1 \ldots Q_n x_n \ \varphi'$ est sous FND (forme normale disjonctive) ssi la matrice φ' est:

- **une conjonction de littéraux**
- **ou une disjonction de conjonction de littéraux**

***Théorème*:**

Toute formule φ de la logique du 1er ordre est équivalente à une formule prénexe sous FNC et une formule prénexe sous FND.

45

14- Formes normales des formules de la logique du 1er ordre

2- Forme normale de Skolem (FNS)

Définition

Une formule est sous forme normale de Skolem ssi elle est de la forme

$$\boxed{\varphi = \forall x_1 \forall x_2 \ldots \forall x_n \ \varphi'}$$

où φ' est sans quantificateur

C'est une formule prénexe où tous les quantificateurs sont universels

46

2- Forme normale de Skolem (FNS)

Théorème

Etant donnée une formule φ sous forme prénexe , il existe une formule φ' (dite forme de Skolem de φ) telle que:

- **si φ contient *n* quantificateurs existentiels, *n* nouveaux symboles de fonctions (et/ou constantes) apparaissent dans φ' (il s'agit des fonctions/constantes de Skolem).**
- **φ' ne contient pas de quantificateurs existentiels**
- **φ est satisfaite ssi φ' est satisfaite.**

14- Formes normales des formules de la logique du 1er ordre

2- Forme normale de Skolem (FNS)

Skolémisation

Soit φ une formule ayant la forme prénexe. Transformons φ ainsi :

on remplace chaque variable quantifiée existentiellement par une fonction des variables quantifiées universellement avant et on supprime les quantificateurs existentiels.

On convient de n'introduire que des fonctions nouvelles et différentes.

La formule obtenue est la skolémisée de φ . Elle est notée φ^{sko}.

14- Formes normales des formules de la logique du 1er ordre

2- Forme normale de Skolem (FNS)

Mise sous Forme normale de Skolem (Skolémisation)

1- Mettre φ sous forme prénexe pour avoir φ_1

$$\varphi_1 = \forall x_1 \forall x_2 \ldots \forall x_k \, \exists x \, \varphi''$$

telle que φ'' est sous forme prénexe

2- Si k=0 i.e. $\varphi_1 = \exists x \, \varphi''$ alors

On construit $\varphi_2 = \varphi''[c/x]$

où c: est une nouvelle constante qui ne figure pas dans φ''

(c est appelée constante de Skolem)

La formule φ_2 contient un quantificateur existentiel en moins que φ_1

49

14- Formes normales des formules de la logique du 1er ordre

2- Forme normale de Skolem (FNS)

3- Si k>0 Alors

On construit la formule $\varphi_2 = \forall x_1 \forall x_2 .. \forall_{xk} \, (\varphi''[f(x_1,x_2,\ldots,x_k)/x])$

où f: est un nouveau symbole de fonction (qui ne figure pas dans φ'')

(f est appelée fonction de Skolem)

La formule φ_2 contient un quantificateur existentiel en moins que φ_1

4- Si φ_2 contient au moins un quantificateur $\exists$ Alors

Skolémiser φ_2.

50

14- Formes normales des formules de la logique du 1er ordre

2- Forme normale de Skolem (FNS)

Exemple de Skolémisation

$\varphi = \exists x \forall y\ p(x,y)$

Remplacer x par la constante de Skolem a

$\varphi' = \forall y\ p(a,y)$ sous FNS

$\varphi = \forall x\ \exists y\ p(x,y)$

Remplacer y par la fonction f(x)

$\varphi' = \forall x\ p(x, f(x))$ sous FNS

51

14- Formes normales des formules de la logique du 1er ordre

2- Forme normale de Skolem (FNS)

Exemple de Skolémisation

$\varphi = \exists u\ \forall x \exists y \forall z \exists t\ (p(x) \wedge q(y) \wedge r(x,z,t) \wedge s(y) \wedge k(u))$

Remplacer u par la constante a

$\forall x \exists y \forall z \exists t\ (p(x) \wedge q(y) \wedge r(x,z,t) \wedge s(y) \wedge k(a))$

Remplacer y par le terme f(x)

$\forall x\ \forall z \exists t\ (p(x) \wedge q(f(x)) \wedge r(x,z,t) \wedge s(f(x)) \wedge k(a))$

Remplacer t par la fonction g(x,z)

$\forall x\ \forall z\ (p(x) \wedge q(f(x)) \wedge r(x,z,g(x,z)) \wedge s(f(x)) \wedge k(a))$

sous FNS

52

14- Formes normales des formules de la logique du 1er ordre

2- Forme normale de Skolem (FNS)

Remarque

la formule φ' obtenue par skolémisation de φ n'est pas équivalente à φ.

Justification par contre-exemple :

$\varphi=\forall x\exists y\ p(x,y)$

$\varphi'=\forall x\ p(x,f(x))$

$S=(\{1,2\},\varnothing,\ \{f\},R)$

$f(1)=2,\ f(2)=1$

$R(1,1)=R(2,2)=\text{vrai}$, $R(1,2)=R(2,1)=\text{faux}$

S I==φ mais S I=/= φ'

14- Formes normales des formules de la logique du 1er ordre

3-Forme normale d'Herbrand (FNH)

Définition

Une formule est sous forme normale d'Herbrand ssi elle est de la forme

$$\boxed{\varphi = \exists x_1 \exists x_2 \ldots \exists x_n\ \varphi'}$$

où φ' est sans quantificateur

c'est une formule prénexe où tous les quantificateurs sont existentiels.

14- Formes normales des formules de la logique du 1er ordre

3-Forme normale d'Herbrand (FNH)

Théorème

Etant donnée une formule φ sous FNP, il existe une formule φ' dite forme d'Herbrand de φ telle que:

- **si φ contient *n* quantificateurs universels, *n* nouveaux symboles de fonctions (et/ou de constantes) apparaissent dans φ' (il s'agit des fonctions/constantes d'Herbrand).**
- **φ' ne contient pas de quantificateurs universels**
- **φ est valide ssi φ' est valide.**

14- Formes normales des formules de la logique du 1er ordre

3-Forme normale d'Herbrand (FNH)

Mise sous FNH (herbrandisation)

La mise de φ sous FNH revient à appliquer la négation à la FNS de ¬φ.

1- Soit φ la formule à mettre sous FNH

Poser φ1= ¬φ

2- Skolémiser φ1 pour avoir une formule:

φ2= ∀x1∀x2 …∀xn φ' (φ' sans quantificateurs)

3- Poser φ3= ¬φ2 = ¬(∀x1∀x2 …∀xn φ')

= ∃x1∃x2…∃xn ¬φ' qui est sous FNH

14- Formes normales des formules de la logique du 1er ordre

3-Forme normale d'Herbrand (FNH)

Exemple d'Herbrandisation

$\varphi = \forall x \, \exists y \; p(x,y)$

$\varphi 1 = \rceil(\forall x \exists y p(x,y))$

Skolémisation de φ1

$\exists x \; \forall y \; \rceil p(x,y)$

$\forall y \; \rceil p(a,y)$

négation de φ1

$\rceil(\forall y \rceil p(a,y))$

$\exists y \; p(a,y)$ **sous FNH**

14- Formes normales des formules de la logique du 1er ordre

Remarque:

- **La mise de φ sous FNP donne une formule équivalente à φ**
- **La mise de φ sous FNS préserve la satisfaisabilité de φ**
- **La mise de φ sous FNH préserve la validité de φ.**

14- Formes normales des formules de la logique du 1er ordre

Forme normale Clausale (clause)FNCL

Définition

Une Clause est une formule est sous la forme

$$\varphi = \forall x_1 \forall x_2 \ldots \forall x_n (L_1 \vee L_2 \vee \ldots \vee L_k)$$

Où: Li sont des littéraux et $n \geq 0$

(Un littéral est un atome ou la négation d'un atome**)**

- **Clause est une disjonction de littéraux quantifiée universellement**
- **Une clause de Horn est une clause contenant au plus un littéral positif**
- **Une clause négative est une clause ne contenant que des littéraux négatifs**

59

15- Programmation logique (Prolog)

Une clause de programme Prolog est une clause de Horn

- **Si $\varphi = \forall x1 \forall x2 \ldots \forall xk\ L$ (où L est un littéral positif)**

  Alors la règle Prolog correspondante est un fait Prolog

  L.

- **Si $\varphi = \forall x1 \forall x2 \ldots \forall xk\ (L1 \vee \neg L2 \vee \neg L3 \vee \ldots \vee \neg Ln)$**

  (où Li des littéraux positifs)

  Alors la règle Prolog correspondante a la forme

  L1 :- L2, L3 , …, Ln

60