**Logique du Premier Ordre**

**Sémantique**

1- Introduction

2- Notion de Structure

3- Interprétation d'un langage

4- Valuation des variables dans une structure

5- Interprétation de Termes

6- Interprétation de Formules

7- Satisfaisabilité

8- Validité

9- Thèse et tautologie

10- Formules équivalentes

11- Modèles

12 Compatibilité

13 Conséquence logique

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

15- Programmation logique (Prolog)

**1- Introduction**

**Considérons la formule:** y 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.

**y** **x p(y,x)**

**Exemple-1** domaine = N

p(x,y) = x 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 y

La formule se lit:

« il existe un entier relatif inférieur ou égal à tous les entiers relatifs»

La formule est FAUSSE

**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

**Exemple (D,C,F,R) telle que**

**D=N**

**C={0}**

**F={+,\*, succ}**

**R={****}**

**(D, C, F, R) telle que Sont des structures**

**D=Réels**

**C={0, 1}**

**F={+, - , \*}**

**R=**

**Est une 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.**

**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 D.

**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 )

**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

**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. A})

[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

**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.

**Exemple**: 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**

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: Var D

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

**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

**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.

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

**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 ,…[tn] S,v)

**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=x2

[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

t2= f(g(c2, h(y,z))) [t2] S,v = (1 + (4\*6) )2 = 252 =725

t1= g(y,h(c1,x)) [t1]S,v = 4 + (0\*3) =4

**6- Interprétation de Formules**

**Définition**

Soit : 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 par rapport à S et V noté []S,v est définie par:

Si = p(t1,t2,..tn) alors []S,v = [p] S ( [t1] S,v , [t2] S,v ,..[tn] S,v )

Si = t1t2 alors []S,v = [t1] S,v [t2] S,v

Si =1 alors []S,v =([1] S,v )

Si= 1 K 2 K{,v, , } alors []S,v = [1] S,v K [2] S,v

Si = x 1 alors []S,v = aD ([1] S,v )[a /x]

Si =81 alors []S,v = V aD ([1] S,v )[a /x]

**7- Satisfiabilité**

Satisfiabilité d'une formule pour une structure est une valuation

**Définition** Soient L un langage du 1er ordre

: une formule construite sur L

S : une structure interprétation de L

V : une valuation pour l'ensemble des variables Var

est satisfaite dans S pour la valuation V []S,v =1

est non satisfaite dans S pour la valuation V []S,v =0

Exemple

y p(x,y) est satisfaite pour S=( N, , ,{}) et la valuation V / V(x)=0

y p(x,y) est non satisfaite pour S=( N, , ,{}) et la valuation V/ V(x)=5

**Satisfiabilité d'une formule pour une structure**

**Définition**

Soient L un langage du 1er ordre

: une formule construite sur L

S : une structure interprétation de L

satisfaite pour S V telle que []S,V =1

non satisfaite pour S V on a []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, , ,{})

**Satisfiabilité d'une formule**

**Définition**

Soient L un langage du 1er ordre

: une formule construite sur L

satisfaite S V telles que []S,V =1

non satisfaite S V []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

: une formule construite sur L

S : une structure interprétation de L

valide dans une structure S V []S,V =1

non valide dans une structure S V []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}, {})

**Validité d'une formule dans une structure**

*Théorème*

Si est une formule close (sans variables libres) alors

valide satisfaite

***Démonstration:***

? valide dans S ⇒satisfaite (évident)

? satisfaite dans S v []S,v =1

close v [] s,v =[]s ( pas de variables libres dans )

v []s,v =1

valide dans S

**9- Thèse et tautologie**

**Définition**

Soient L un langage du 1er ordre

: une formule construite sur L

thèse valide

thèse S valide dans S

**Exemple**

**x p(x)****p(y)**

**x p(x,x)** **y p(y,x)**

**Tautologie**

**Définition**

Soient L un langage du 1er ordre

: une formule construite sur L

est une tautologie s'il existe une tautologie ' de la logique propositionnelle telle

que se mette sous la forme de '

**Exemples**

= p(x)vp(x) est une tautologie

Car a la forme de AvA qui est une tautologique en logique propositionnelle

= p(x,y)x r(x) v x r(x) est une tautologie

Car elle a la forme ABvB qui est une tautologie en logique propositionnelle

3**Remarque:**

tautologie est une thèse

( mais l'inverse n'est pas vrai)

**10- Formules équivalente**

**Définition :**

Soient L un langage du 1er ordre

1, 2 : des formules construites sur L

1 et 2 équivalentes S v on a [1]S,V =[2]S,V

1 et 2 équivalentes la formule 1 2 est valide

1 et 2 non équivalentes S v on a ([1]S,V =[2]S,V))

**Exemples :**

- x F et xF équivalentes ( propriété de définissabilité de par )

- x F et x F équivalentes ( propriété de définissabilité de par )

**Exemples :**

et xsi x liée dans 

et xsi x liée dans 

x(12) et x1x2

x(1v2) et x1v x2

x(12) et x(1v2)

xet y [y/x] si x libre dans et y n'apparaît pas dans 

xet y[y/x] si x est libre dans et y n'apparaît pas dans 

x(12) et x1 2 si x liée dans 2

x(12) et x1 2 si x liée dans 2

x(1v2) et x1 v 2 si x liée dans 2

x(1v2) et x1 v2 si x liée dans 2

**11- Modèles**

**Modèle d'une formule**

**Définition :**

Soient - L un langage du 1er ordre

- une formule de L.

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

S modèle de valides dans S

*Notation* S 

**Modèle d'un ensemble de formules**

**Définition :**

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

S: une réalisation de L

S est u. modèle S I==

*Notation* S 

**12 Compatibilité**

**Compatibilité d'une formule**

**Définition :**

Une formule est compatible si elle a au moins un modèle

compatible S S I==

**Compatibilité d'un ensemble de formules**

**Définition :**

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

modèle.

compatible S SI== 

**13 Conséquence logique**

**Définition :**

Soit : un ensemble de formules.

: une formule.

conséquence de ssi tout modèle de est un modèle de 

est conséquence de S (S I==S I==)

est conséquence de S v ( i [i]S,v=1[]S,v= 1)

*Notation* I

non conséquence de S (S I==(S I==))

***Remarque*:** **thèse** **I**

**1 et** **2 équivalentes** **(****1I****2 et** **2 I****-****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)

1. **Forme Normale Prénexe (FNP)**

**Définition :**

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

= Q1 x1…Qn xn '

où n0

i=1..n Qi est un quantificateur et xi une variable

' est une formule qui *ne contient aucun quantificateur*.

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

- La suite des quantificateurs Q1x1Q2x2…Qnxn est appelée *préfixe*

- ' est appelée *matrice.*

**Exemples :**

p(x) p(y) est sous FNP

x y ( (p(x)q(x,y)) v r(z)) est sous FNP

x ((p(x)x p(x)) n'est pas sous FNP

***Théorème:***

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

**Méthode de Mise sous Forme normale Prénexe**

**1- Eliminer les connecteurs** et en appliquant les équivalences:

12= 1v2 12 = (1v2)(1v2)

**2- Faire entrer les** vers l'intérieur en appliquant les équivalences:

x= xx= x

1=1 (12)=1v2 (1v2)=12

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

**quantificateurs"**

**Si x non libre dans** **2**

(x1) v2 =x(1v2) (x1) v2 =x(1v2)

2 v(x1) =x(2v1) 2 v(x1) =x(2v1)

(x1) 2 =x(12) (x1) 2 =x(12)

2(x1) =x(21) 2 (x1) =x(21)

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

***Exemple de mise sous FNP***

= x (p(x)xp(x))

= x (p(x)v x p(x) ) élimination de 

= x(p(x) vy p(y)) Renommage de xp(x) par y p(y)

= x y(p(x)v p(y)) Sortir y

sous FNP

**N.B.** Le renommage est essentiel ici :

x (p(x)v x p(x)) et x x(p(x)v p(x))

ne sont pas équivalentes

***Exemple de mise sous FNP***

= ( p(x)( (y q(x,y)) r(y) ) )

= ( p(x) v ( (y q(x,y)) r(y) ) ) éliminer 

= p(x) ( (y q(x,y)) r(y) ) ) rentrer  sortir y

= p(x) ( (y q(x,y)) v r(y) ) )

= p(x) ( (y q(x,y)) v r(y) ) )

= p(x) ( (y q(x,y)) v r(z) ) ) renommage r(y) par r(z)

= p(x) y( q(x,y) v r(z) ) sortir y

= y (p(x) ( q(x,y) v r(z) )

sous FNP

**Formule prénexe sous FNC**

Une formule prénexe = Q1 x1…Qn xn ' est sous FNC (forme normale

conjonctive) ssi la matrice ' est:

- une disjonction de littéraux

(un littéral étant une formule atomique ou sa négation)

- ou une conjonction de disjonction de littéraux

***Exemple***

xyz( r(x,y) v x=y v (y=g(x,h(z,z))) ) est sous FNC

**Formule prénexe sous FND**

Une formule prénexe = Q1 x1…Qn xn ' est sous FND (forme normale

disjonctive) ssi la matrice ' est:

- une conjonction de littéraux

- ou une disjonction de conjonction de littéraux

***Théorème*:**

Toute formule de la logique du 1er ordre est équivalente à une formule

prénexe sous FNC et une formule prénexe sous FND.

1. **Forme normale de Skolem (FNS)**

**Définition :**

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

= x1x2…xn '

où ' est sans quantificateur

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

***Théorème***

Etant donnée une formule sous forme prénexe , il existe une formule ' (dite

forme de Skolem de ) telle que:

• si contient *n* quantificateurs existentiels, *n* nouveaux symboles de

fonctions (et/ou constantes) apparaissent dans ' (il s'agit des

fonctions/constantes de Skolem).

• ' ne contient pas de quantificateurs existentiels

• est satisfaite ssi ' est satisfaite.

**Skolémisation :**

Soit une formule ayant la forme prénexe. Transformons 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 . Elle est notée sko.

**Mise sous Forme normale de Skolem (Skolémisation)**

**1- Mettre** sous forme prénexe pour avoir 1

1 =x1x2…xk x "

telle que " est sous forme prénexe

**2-** Si k=0 i.e. 1=x " alors

On construit 2="[c/x]

où c: est une nouvelle constante qui ne figure pas dans "

(c est appelée constante de Skolem)

La formule 2 contient un quantificateur existentiel en moins que 1

**3-** Si k>0 Alors

On construit la formule 2= x1x2..xk ( "[f(x1,x2,…,xk)/x] )

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

(f est appelée fonction de Skolem)

La formule 2 contient un quantificateur existentiel en moins que 1

**4-** Si 2 contient au moins un quantificateur Alors

**Skolémiser** **2.**

***Exemple de Skolémisation***

= xy p(x,y)

Remplacer x par la constante de Skolem a

’= y p(a,y) sous FNS

=x y p(x,y)

Remplacer y par la fonction f(x)

’= x p(x, f(x) ) sous FNS

***Exemple de Skolémisation :***

= u xyzt ( p(x) q(y) r(x,z,t) s(y) k(u))

Remplacer u par la constante a

xyzt ( p(x) q(y) r(x,z,t) s(y) k(a))

Remplacer y par le terme f(x)

x zt ( p(x) q( f(x) ) r(x,z,t) s(f(x) ) k(a))

Remplacer t par la fonction g(x,z)

x z ( p(x) q(f(x)) r(x,z,g(x,z)) s(f(x) ) k(a))

sous FNS.

***Remarque***

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

Justification par contre-exemple :

=xy p(x,y)

'= x p(x,f(x))

S=({1,2},, {f},R)

f(1)=2, f(2)=1

R(1,1)=R(2,2)=vrai , R(1,2)=R(2,1)=faux

S I==mais S I=/= '.

**3-Forme normale d'Herbrand (FNH)**

**Définition :**

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

= x1 x2…xn '

où ' est sans quantificateur

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

***Théorème***

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

• si contient *n* quantificateurs universels, *n* nouveaux symboles de fonctions (et/ou de constantes) apparaissent dans ' (il s'agit des fonctions/constantes d'Herbrand).

• ' ne contient pas de quantificateurs universels

• est valide ssi ' est valide.

**Mise sous FNH (herbrandisation)**

***La mise de*** ***sous FNH revient à appliquer la négation à la FNS de*** ***.***

1- Soit la formule à mettre sous FNH

Poser 1=

2- Skolémiser 1 pour avoir une formule:

2= x1x2 …xn ' (' sans quantificateurs)

3- Poser 3= 2 = ( x1x2 …xn ' )

= x1x2…xn ' qui est sous FNH.

***Exemple d'Herbrandisation***

=x y p(x,y)

1= (xyp(x,y))

Skolémisation de 1

x y p(x,y)

y p(a,y)

négation de 1

(yp(a,y))

y p(a,y) sous FNH.

***Remarque:***

La mise de sous FNP donne une formule équivalente à 

La mise de sous FNS préserve la satisfaisabilité de 

La mise de sous FNH préserve la validité de .

**Forme normale Clausale (clause)FNCL**

**Définition :**

Une Clause est une formule est sous la forme

= x1x2…xn (L1 v L2 v…v Lk)

Où: Li sont des littéraux et n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.

**15- Programmation logique (Prolog)**

Une clause de programme Prolog est une clause de Horn

Si =x1x2…xk L (où L est un littéral positif)

Alors la règle Prolog correspondante est un fait Prolog

L.

Si =x1x2…xk ( L1v L2v L3 v …vLn)

(où Li des littéraux positifs)

Alors la règle Prolog correspondante a la forme

L1 :- L2, L3 , …, Ln