

# **Chapitre-4**

## **Logique du 1er ordre**

### **- Syntaxe -**

1

#### **Plan**

- 1- Introduction**
- 2- Alphabet**
- 3-Termes d'un langage**
- 4- Formule de la logique du 1er ordre**
- 5-Sous-formule d'une formule du 1er ordre**
- 6- Arbre de décomposition**
- 7- Longueur - profondeur - complexité d'une formule**
- 8- Simplification syntaxique d'une formule:**
- 9- Variables libres et variables liées**
- 10- Substitution**

2

## Plan

### 1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

3

### 1- Introduction

#### Limites de la logique propositionnelle(1)

1. La logique propositionnelle a une **expressivité limitée** par rapport au langage naturel:

On considère la phrase comme un bloc indivisible

#### Exemple

« Mohamed habite à Constantine »

On ne peut pas considérer séparément:

Mohamed,

Constantine,

et la relation « habiter à »

4

## 1- Introduction

### Limites de la logique propositionnelle(2)

2- En logique propositionnelle on **n'arrive pas à prouver** certaines propositions pourtant très intuitives

**Exemple:**

Tous les humains sont mortels.  
Socrate est un humain.  
Donc Socrate est mortel.

La formalisation de cet énoncé en logique propositionnelle donne la formule:

$$\varphi = A \wedge B \rightarrow C$$

A: Tous les humains sont mortels

B: Socrate est un humain

C: Socrate est mortel

On ne peut pas prouver que:

**$\varphi$  est toujours Vraie**

5

## 1- Introduction

Tous les humains sont mortels.

Socrate est un humain.

Donc Socrate est mortel.

6

## 1- Introduction

La logique des propositions traite les propositions comme **un tout**.

❑ Ce qui manque à la logique propositionnelle c'est de pouvoir parler **d'objets**, de **propriétés d'objets** et de mettre en **relation** des objets.

❑ La **logique du 1er ordre** nous permettra de modéliser ces situations.

7

## 1- Introduction

La célèbre phrase de Socrate

« Tous les humains sont mortels. »

peut être analysée plus finement à l'aide  
du prédicat **est.humain** ,  
du prédicat **est.mortel**,  
et de la variable d'individu **x** qui représentera  
génériquement l'homme.

Ainsi, « Tous les humains sont mortels. »

peut être traduit par la formule

$\forall x (\text{est.humain}(x) \rightarrow \text{est.mortel}(x))$  .

8

## 1- Introduction

□ La logique du premier ordre (ou logique des prédicats) est la partie de la logique qui traite des *propositions analysées*.

□ Ceci signifie que:

➤ on ne se contente plus de la notion de proposition vue comme un **bloc** (et notée, dans le cas élémentaire par des variables propositionnelles).

➤ La proposition est divisée (selon les termes anciens) en son *sujet* et son *prédicat*

□ **Platon** fut le premier, à avoir réfléchi à cette décomposition (Il a introduit en effet la division fondamentale de la phrase grecque en **composant nominal** et **composant verbal**).

9

## 1- Introduction

□ **Exemple**

'Mohamed habite Constantine',  
'Ali habite Constantine',  
'Sara habite Constantine'

sont des instances d'un même moule:

'**x** habite Constantine'.

où

□ **x** est une variable (elle marque la place du "sujet".)

□ '**habite Constantine**', est un Groupe Verbal.  
(il s'agit d'un *prédicat*.)

□ **Notation**

Habite.constantine(Mohamed)  
Habite.constantine(Ali)  
Habite.constantine(Sara)

10

## 1- Introduction

Un Prédicat peut avoir plusieurs arguments

### Exemples

Mohamed habite Constantine  
Ali habite Annaba  
Sara habite Alger

$x$  habite  $y$

La partie verbale elle-même peut être analysée. La constante 'habite' est utilisée pour dénoter une relation entre des individus et des villes

$\text{Habite}(x,y): x \text{ Habite } y$

11

## 1- Introduction

### Logique Propositionnelle & Logique du 1er ordre

- ❑ La logique propositionnelle représente le monde par des faits
- ❑ La logique du 1er ordre représente le monde par :
  - des **objets** : Mohamed, Socrate, 0,1, Constantine ...
  - des **propriétés** des objets et des relations entre ces objets: est.mortel, est.le.père.de, est.inférieur.à, habite,...
  - Des **fonctions** sur ces objets: père.de, succ, carré, +, ...

12



## 1- Introduction

### Etude de la logique du 1er ordre:

- Comment écrire une formule?

*Aspects syntaxiques*

- Comment déterminer la valeur de vérité d'une formule?

*Aspects sémantiques*

- Comment démontrer(automatiquement) de nouveaux résultats?

*Aspects Preuves*

Ce chapitre concerne les aspects syntaxiques

13

## Plan

1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

14

## 2- Alphabet

L'alphabet d'un langage L du 1er ordre comporte :

- un ensemble V de **variables d'objets**
- les connecteurs  $\neg \wedge \vee \rightarrow \leftrightarrow$
- les parenthèses ( ) et la virgule ,
- l'identité  $\equiv$
- les quantificateurs  $\forall \exists$

Les symboles  
communs  
à tous les  
langages

- les symboles de **constantes** ( individus)
- les symboles de **prédicats** (ou relations)
- les symboles de **fonctions**

Les symboles  
qui caractérisent  
le langage L

15

## 2- Alphabet

### Exemples de Constantes

- Socrate
- Mohamed
- Constantine
- 4
- a , b

16



## 2- Alphabet

### Exemples de Symboles de relations(ou de prédicats)

- $\text{Mortel}(x)$ :  $x$  est mortel
- $\text{humain}(x)$ :  $x$  est humain
- $P(x)$ :  $x$  est pair
- $S(x,y)$ :  $x$  est supérieur à  $y$ .
- $E(x,y,z)$ :  $x$  est entre  $y$  et  $z$ .
- $E(y,x,z)$ :  $y$  est entre  $x$  et  $z$ .
- $E'(x,y,z)$ :  $y$  est entre  $x$  et  $z$ .
- $x > y$  :  $x$  est supérieur strictement à  $y$

17

## 2- Alphabet

### Exemples de Symboles de fonctions

$$\text{succ}(x) = x + 1$$

$$g(x, y) = x + y$$

$$h(x, y, z) = x * y - z$$

$$\text{Père}(x)$$

18

## 2- Alphabet

A chaque prédicat  $p$  (resp. fonction  $f$ ), on associe un entier positif appelé **l'arité** de  $p$  (resp. de  $f$ ) qui désigne le nombre d'arguments de  $p$  (resp. de  $f$ )

Il existe une grande variété de langages du 1er ordre, chacun étant défini par son vocabulaire (symboles de fonctions et symboles de prédicats, constantes)

19

## 2- Alphabet

L'alphabet d'un langage  $L$  du 1er ordre comporte :

- un ensemble  $V$  de **variables d'objets**
- les connecteurs  $\neg \wedge \vee \rightarrow \leftrightarrow$
- les parenthèses  $( )$  et la virgule  $,$
- l'identité  $\equiv$
- les quantificateurs  $\forall \exists$

Les symboles  
communs  
à tous les  
langages

- les symboles de **constantes** ( individus)
- les symboles de **prédicats** (ou relations)
- les symboles de **fonctions**

Les symboles  
qui caractérisent  
le langage  $L$

20

## 2- Alphabet

### Exemple de langages du 1er ordre:

L0: Constante: **Socrate**

Prédicats :

**h(x)** : x est humain

**m(x)**: x est mortel

$( \forall x ( h(x) \rightarrow m(x) ) \wedge h(\text{Socrate}) ) \rightarrow m(\text{Socrate})$

21

## 2- Alphabet

### Exemple de langages du 1e ordre:

L 1 contient - un prédicat unaire **r**  
- et une constante **c**

L2 contient - un prédicat binaire **p**  
- une fonction unaire **f**  
- deux fonctions binaire **g h**  
- deux constantes **a b**

22

## Plan

1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

23

### 3-Terms d'un langage

- Un terme désigne un objet
- L'ensemble des termes d'un langage L est défini par:
  - Si **a** est une constante Alors **a** est un terme
  - Si **x** est une variable Alors **x** est un terme
  - Si **f** est un symbole de fonction d'arité **n** et **t1,t2,...,tn** des termes de L Alors **f(t1,t2,...,tn)** est un terme

24

### 3-Terms d'un langage

#### Exemple

L0: Constante: **Socrate**

Prédicats :

**h(x)** : x est humain

**m(x)**: x est mortel

Les termes du langage L0 sont:

la constante: **Socrate**

les variables: **x y z....**

25

### 3-Terms d'un langage

#### Exemple

L 1 contient - un prédicat unaire **r**  
- et une constante **c**

Les termes du langage L1 sont:

la constante: **c**

les variables: **x y z....**

26

### 3-Terms d'un langage

#### Exemple

- L2 contient
- un prédicat binaire **p**
  - une fonction unaire **f**
  - deux fonctions binaire **g h**
  - deux constantes **a b**

Quelques termes du langage L2 sont :

Les constantes: **a, b**

Les variables: **x, y ....**

**f(a) f(b) f(x) f(y) f(f(a)) f(f(f(x))) ...**

**g(a,a) g(a,b) g(x,a) g(f(a),b) g(x,f(f(f(a))))**

**f(g(f(b),x)) h(g(a,b),x) ...**

27

### 3-Terms d'un langage

□ L'**ordre** d'un terme **t** est le nombre d'occurrence de signes fonctionnels dans **t**.

Exemple: le terme **f( g(x,y, f(z,a))** est d'ordre 3

□ Les termes **d'ordre 0** sont les variables et les constantes

□ Un terme est **Clos** (ou fermé) ssi il ne contient aucune variable

Exemple: **f(a)** est clos

28



## Plan

1- Introduction

2- Alphabet

3- Termes d'un langage

4- Formule de la logique du 1er ordre

5- Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

29

## 4- Formule de la logique du 1er ordre

### Définition

$\phi$  est une formule de la logique du 1er ordre ssi

il existe une suite finie de formules  $\phi_1, \phi_2, \dots, \phi_n$  /  $\phi_n = \phi$  et  $\forall i=1, n$  on est dans l'un des cas suivants:

- $\phi_i = t \equiv t'$  avec  $t$  et  $t'$  des termes
- $\phi_i = p(t_1, \dots, t_n)$  avec  $p$ : un prédicat de poids  $n$  et  $t_1, t_2, \dots, t_n$  : des termes
- $\exists j < i \quad \phi_i = \neg \phi_j$
- $\exists j_1, j_2$  et  $K \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$  /  $\phi_i = \phi_{j_1} K \phi_{j_2}$
- $\exists j < i$  et  $x$  une variable et  $Q \in \{\forall, \exists\}$  tel que  $\phi_i = Qx \phi_j$

30

#### 4- Formule de la logique du 1er ordre

##### Définition

L'ensemble des formules de la logique du 1<sup>er</sup> ordre est défini par:

- Si  $t$  et  $t'$  des termes alors  $\phi_i = t \equiv t'$  est une formule
- Si  $p$ : un prédicat de poids  $n$  et  $t_1, t_2, \dots, t_n$ : des termes  
Alors  $\phi_i = p(t_1, \dots, t_n)$  est une formule
- Si  $\phi_j$  une formule alors  $\phi_i = \neg \phi_j$  est une formule
- Si  $\phi_{j1}, \phi_{j2}$  des formules et  $K \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$   
Alors  $\phi_i = \phi_{j1} K \phi_{j2}$  est une formule
- Si  $\phi_j$  une formule et  $x$  une variable et  $Q \in \{\forall, \exists\}$   
Alors  $\phi_i = Qx \phi_j$  est une formule

31

#### 4- Formule de la logique du 1er ordre

##### Exemple

L0: Constante: **Socrate**

Prédicats :

**$h(x)$**  :  $x$  est humain

**$m(x)$** :  $x$  est mortel

Quelques formules du langage L0 sont:

$h(\text{Socrate})$

$\neg h(\text{Socrate})$

$h(\text{Socrate}) \wedge \neg m(\text{Socrate})$

$\forall x m(x) \rightarrow m(\text{Socrate})$

32

#### 4- Formule de la logique du 1er ordre

##### Exemple

- L 1 contient
- un prédicat unaire **r**
  - et une constante **c**

Quelques formules du langage L1 sont:

$$x \equiv c$$

$$r(c)$$

$$(x \equiv c) \vee \neg(c \equiv c) \rightarrow r(c)$$

$$r(c) \vee \neg \exists x r(x)$$

$$\forall y \exists x (r(y) \rightarrow r(x))$$

33

#### 4- Formule de la logique du 1er ordre

##### Exemple

- L2 contient
- un prédicat binaire **p**
  - une fonction unaire **f**
  - deux fonctions binaire **g h**
  - deux constantes **a b**

Quelques formules du langage L2 sont:

$$(x \equiv a) \wedge (g(y,x) \equiv b)$$

$$\forall x \exists y (g(x,y) \equiv a \wedge g(y,x) \equiv b)$$

$$\forall x \neg (f(x) \equiv a)$$

34

#### 4- Formule de la logique du 1er ordre

□ Une formule  $\varphi$  d'un langage  $L$  est dite **atomique** ssi:

- $\varphi = t_1 \equiv t_2$  où  $t_1, t_2$  sont des termes de  $L$
- $\varphi = p(t_1, t_2, \dots, t_n)$  où  $p$  est un prédicat d'arité  $n$  et  $t_1, t_2, \dots, t_n$  sont des termes de  $L$

35

#### 4- Formule de la logique du 1er ordre

##### Définition

$\varphi$  est une formule de la logique du 1er ordre ssi

il existe une suite finie de formules  $\varphi_1, \varphi_2, \dots, \varphi_n$  /  $\varphi_n = \varphi$  et  $\forall i=1, n$  on est dans l'un des cas suivants:

- $\varphi_i = t \equiv t'$  avec  $t$  et  $t'$  des termes
- $\varphi_i = p(t_1, \dots, t_n)$  avec  $p$ : un prédicat de poids  $n$  et  $t_1, t_2, \dots, t_n$  : des termes
- $\exists j < i \quad \varphi_i = \neg \varphi_j$
- $\exists j_1, j_2$  et  $K \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$  /  $\varphi_i = \varphi_{j_1} K \varphi_{j_2}$
- $\exists j < i$  et  $x$  une variable et  $Q \in \{\forall, \exists\}$  tel que  $\varphi_i = Qx \varphi_j$

Formules  
atomiques

36

#### 4- Formule de la logique du 1er ordre

##### Exemples de traduction d'énoncés en formules

`` Tout est relatif. ``

langage

Prédicat unaire  $r$

$r(x)$ : «  $x$  est relatif »

formule

$\forall x \, r(x)$

37

#### 4- Formule de la logique du 1er ordre

##### Exemples de traduction d'énoncés en formules

`` Une porte est ouverte ou fermée. ``

langage

3 Prédicats unaires

$\text{porte}(x)$ : «  $x$  est une porte »

$\text{ouvert}(x)$ : «  $x$  est ouvert »

$\text{fermé}(x)$ : «  $x$  est fermé »

formule

$\forall x ( \text{porte}(x) \rightarrow (\text{ouvert}(x) \vee \text{fermé}(x)) )$

38

#### 4- Formule de la logique du 1er ordre

##### Exemples de traduction d'énoncés en formules

`` Tous les chemins mènent à Rome. ''

2 Prédicats unaires

langage

chemin(x): « x est un chemin »

mène-à-rome(x): « x mène à Rome »

formule

$\forall x (\text{chemin}(x) \rightarrow \text{mène-à-Rome}(x))$

39

#### 4- Formule de la logique du 1er ordre

##### Exemples de traduction d'énoncés en formules

`` Tous les chemins mènent à Rome. ''

1 Prédicat unaires ,1 prédicat binaire, 1 constante

langage

chemin(x): « x est un chemin »

mène(x,y): « x mène à y »

Constante: Rome

formule

$\forall x (\text{chemin}(x) \rightarrow \text{mène}(x, \text{Rome}))$

40



#### 4- Formule de la logique du 1er ordre

##### Exemples de traduction d'énoncés en formules

`` Pour tout entier il existe un entier plus grand. ``

1 Prédicat unaires ,1 prédicat binaire

langage

entier(x): « x est un entier »

Plus-grand(x,y): « x plus grand que y »

formule

$\forall x (\text{entier}(x) \rightarrow \exists y (\text{entier}(y) \wedge \text{plus-grand}(y,x)) )$

41

#### 4- Formule de la logique du 1er ordre

##### Exemples de traduction d'énoncés en formules

`` Pour tout entier il existe un entier plus grand. ``

1 Prédicat unaires ,1 prédicat binaire

langage

e(x): « x est un entier »

$x > y$  : « x plus grand que y »

formule

$\forall x (e(x) \rightarrow \exists y (e(y) \wedge (y > x)) )$

42

#### 4- Formule de la logique du 1er ordre

##### Exemples de traduction de formule en énoncé

$\exists x (restaurant(x) \wedge servir(x, sushi) \wedge \text{êtrePrèsDe}(x, laGare))$

langage

2 Constantes: **Sushi**, **laGare**

1 Prédicat unaire

**restaurant(x)**: « x est un restaurant »

2 prédicats binaires

**Servir(x,y)**: « x sert y »

**êtrePrèsDe(x,y)**: « x est près de y »

énoncé

« il existe au moins un restaurant qui sert du sushi et qui est près de la gare »

43

#### 4- Formule de la logique du 1er ordre

##### Exemples de traduction de formule en énoncé

$\forall x (restaurantJaponais(x) \rightarrow servir(x, sushi))$

langage

1 Constantes: **Sushi**

1 Prédicat unaire

**restaurantJaponais(x)**: « x est un restaurant japonais »

1 prédicat binaire

**Servir(x,y)**: « x sert y »

énoncé

« Tous les restaurants japonais servent du sushi »

44

## Plan

1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

45

## 5-Sous-formule d'une formule du 1er ordre

Les *sous-formules* d'une formule  $\varphi$  sont définies par:

- $\varphi$  est une sous-formule de  $\varphi$ .
- Si  $\neg \varphi'$  est une sous-formule de  $\varphi$   
alors  $\varphi'$  est une sous-formule de  $\varphi$ .
- Si  $\varphi_1 K \varphi_2$  ( $K \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ ) et  $\varphi_1$  est une sous-formule de  $\varphi$   
alors  $\varphi_1$  et  $\varphi_2$  sont des sous-formules de  $\varphi$ .
- Si  $Q x \varphi'$  ( $Q \in \{\forall, \exists\}$ ) est une sous-formule de  $\varphi$   
alors  $\varphi'$  est une sous-formule de  $\varphi$ .

46

## 5-Sous-formule d'une formule du 1er ordre

### Exemple

Soit la formule  $\varphi = ((p(x) \vee r(a)) \wedge \forall x (f(x) \equiv b))$ .

Les sous-formules de  $\varphi$  sont:

$$\begin{array}{l} (p(x) \vee r(a)) \wedge \forall x (f(x) \equiv b), \\ p(x) \vee r(a) \quad \forall x f(x) \equiv b \\ p(x) \quad r(a) \quad f(x) \equiv b \end{array}$$

47

## Plan

1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

48

## 6- Arbre de décomposition

□ Toute formule d'un langage du premier ordre se décompose de manière unique sous forme d'arbre de la manière suivante.

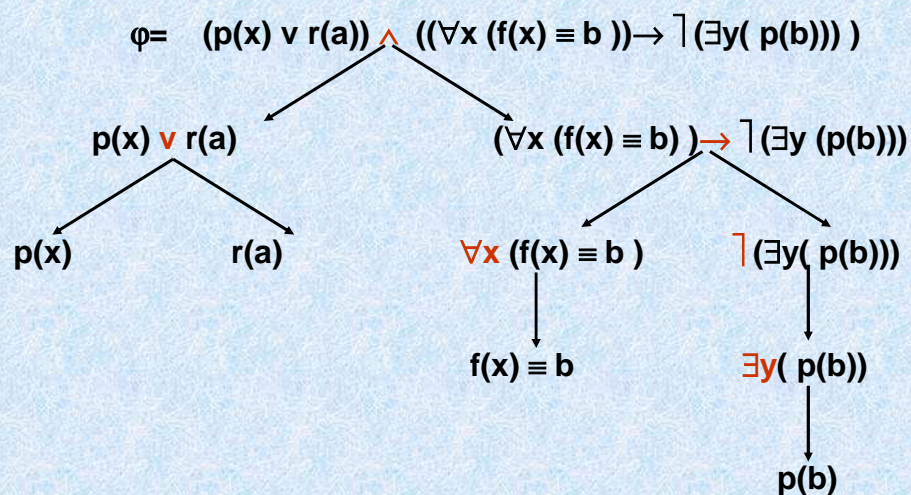
### □ Procédure de décomposition:

- on découpe au niveau du **connecteur** ou **quantificateur** principal de la formule.
- on enlève les parenthèses les plus externes de chacun des arguments du connecteurs ou quantificateur principal
- on répète l'opération de décomposition pour chacun des arguments jusqu'à arriver au **formules atomiques**.

49

## 6- Arbre de décomposition

### Exemple



50

## Plan

1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

51

## 7- Longueur - profondeur - complexité d'une formule

### a) Longueur d'une formule:

La longueur d'une formule du 1er ordre est le nombre de symboles (de l'alphabet du langage) qu'elles contient.

#### Exemple

| $\varphi$                                     | Long( $\varphi$ ) |
|---|-------------------|
| $\forall x p(x)$                              | 6                 |
| $\exists x \exists y \forall x_1 x_2 p(x, y)$ | 14                |
| $p_2(x, y, \text{const1})$                    | 8                 |

52



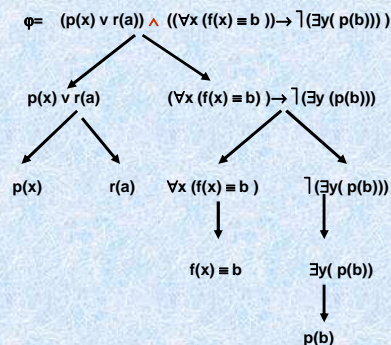
## 7- Longueur - profondeur - complexité d'une formule

### b) Profondeur d'une formule:

la profondeur d'une FP correspond à la profondeur de l'arbre de décomposition

**Exemple**  $\varphi = (p(x) \vee r(a)) \wedge ((\forall x (f(x) = b)) \rightarrow \neg (\exists y (p(b))))$

**Prof( $\varphi$ )=4**



53

## 7- Longueur - profondeur - complexité d'une formule

La profondeur d'une formule  $\varphi$  est définie de la manière suivante:

- Si  $\varphi = t \equiv t'$  où  $t$  et  $t'$  des terme alors **prof( $\varphi$ )=0**
- Si  $\varphi = p(t_1, t_2, \dots, t_n)$  où  $p$  est un prédicat  $n$ -aire alors **prof( $\varphi$ )=0**
- Si  $\varphi = \neg \varphi_1$  où  $\varphi_1$  est une formule alors **prof( $\varphi$ )=1+prof( $\varphi_1$ )**
- Si  $\varphi = Qx \varphi_1$  où  $Q$  un quantificateur alors **prof( $\varphi$ )=1+prof( $\varphi_1$ )**
- Si  $\varphi = \varphi_1 K \varphi_2$  ( $K \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ ) alors **prof( $\varphi$ )=1+ max(prof( $\varphi_1$ ), prof( $\varphi_2$ ))**

54

## 7- Longueur - profondeur - complexité d'une formule

### c) Complexité (ordre) d'une formule:

□ L'ordre d'une formule du 1er ordre est nombre d'occurrences dans  $\phi$  d'un **connecteur** ou d'un **quantificateur**.

□ Les formules d'ordre 0 sont dites atomiques (élémentaires)

#### Exemple

| $\phi$  | $c(\phi)$ |
|---|-----------|
| $\forall x p(x)$  | 1         |
| $\exists x \exists y \forall x_1 x_2 p(x, y)$               | 4         |
| $\exists x p_2(x, y, \text{const1}) \vee \neg (x \equiv y)$ | 3         |

55

## 7- Longueur - profondeur - complexité d'une formule

La complexité d'une formule  $\phi$  est noté  $c(\phi)$  et est définie par:

- Si  $\phi$  est une formule atomique alors  $c(\phi) = 0$
- Si  $\phi = \neg \phi'$  alors  $c(\phi) = 1 + c(\phi')$
- Si  $\phi = Qx \phi'$  alors  $c(\phi) = 1 + c(\phi')$
- Si  $\phi = (\phi' K \phi'')$  et  $K \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$  alors  $c(\phi) = c(\phi') + c(\phi'') + 1$

56

## Plan

1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

57

## 8- Simplification syntaxique d'une formule:

Pour alléger l'écriture des formules on supprime certaines parenthèses superflues en tenant compte de certaines conventions:

1- Elimination des parenthèses de part et d'autre d'une formule atomique

Exemple:

$\exists x(p(x))$  sera simplifiée en  $\exists x p(x)$

$\forall x \forall y (f(x) \equiv f(y))$  sera simplifiée en  $\forall x \forall y f(x) \equiv f(y)$

$\forall x (p(x)) \vee \exists y (p(y))$  sera simplifiée en  $\forall x p(x) \vee \exists y p(y)$

58

## 8- Simplification syntaxique d'une formule:

### 2- Suppression des parenthèses qui séparent des négations consécutives

#### Exemple:

$\neg(\neg(\neg(p(a) \wedge p(b))))$  sera notée  $\neg\neg\neg(p(a) \wedge p(b))$   
 $\neg(\neg p(a))$  sera noté  $\neg\neg p(a)$

### 3-Elimination des parenthèses qui séparent des quantificateurs successifs:

$Q_1x_1 (Q_2x_2( \dots(Q_nx_n( \phi))\dots))$  sera noté  $Q_1x_1 Q_2x_2 \dots Q_nx_n ( \phi )$   
 où  $Q_1, Q_2, \dots, Q_n$  sont des quantificateurs

#### Exemple

$\exists x(\exists y(\forall x (r(x,y) \wedge p(x))))$   
 est simplifié en  $\exists x \exists y \forall x (r(x,y) \wedge p(x))$

59

## 8- Simplification syntaxique d'une formule:

### 4- $\forall x_1 \forall x_2 \forall x_3 \dots \forall x_n \phi$ est remplacé par $\forall x_1 x_2 \dots x_n \phi$ $\exists x_1 \exists x_2 \exists x_3 \dots \exists x_n \phi$ est remplacé par $\exists x_1 x_2 \dots x_n \phi$

#### Exemple:

$\forall x \forall y \exists z \exists x \forall y p(x,y)$   
 est remplacée par  $\forall x y \exists z x \forall y p(x,y)$

### 5- Application des Règles de priorité entre connecteurs et quantificateurs:

$\neg, \forall, \exists$   
 $\wedge, \vee$   
 $\rightarrow, \leftrightarrow$

avec associativité à gauche  
 avec associativité à gauche

60

## 8- Simplification syntaxique d'une formule:

### Exemple

$$\forall x(\forall y(\forall z(\exists x1(\forall x2(\forall x4((x \equiv a) \vee \neg(p(x)))))) \rightarrow ((p(y)) \vee (p(a)))$$

$$\forall x(\forall y(\forall z(\exists x1(\forall x2(\forall x4((x \equiv a) \vee \neg(p(x)))))) \rightarrow ((p(y)) \vee (p(a)))$$

$$\forall x(\forall y(\forall z(\exists x1(\forall x2(\forall x4(x \equiv a \vee \neg p(x)))))) \rightarrow (p(y) \vee p(a)))$$

$$\forall x(\forall y \forall z \exists x1 \forall x2 \forall x4(x \equiv a \vee \neg p(x)) \rightarrow (p(y) \vee p(a)))$$

$$\forall x(\forall y \forall z \exists x1 \forall x2 \forall x4(x \equiv a \vee \neg p(x)) \rightarrow p(y) \vee p(a))$$

$$\forall x(\forall y \forall z \exists x1 \forall x2 \forall x4(x \equiv a \vee \neg p(x)) \rightarrow p(y) \vee p(a))$$

61

## Plan

1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

62

## 9- Variables libres et variables liées

### • Champ d'un quantificateur (portée d'un quantificateur)

Le champ d'un quantificateur **Qx** dans une formule  $\phi$  est la partie de la formule sur laquelle opère le quantificateur

Exemple  $\forall x(\exists y (p(x,y) \rightarrow p(y,x)) \wedge x=y)$

Champ de  $\forall x$

Champ de  $\exists y$

63

## 9- Variables libres et variables liées

### • Occurrence d'une variable dans une formule

On appelle occurrence d'une variable **x** dans une formule  $\phi$ , chaque endroit où **x** apparaît de façon effective dans  $\phi$  (c'est à dire qui ne suit pas un quantificateur)

Exemple:

$$p(x, z) \rightarrow \forall z (r(y, z) \vee y = z)$$

La variable **x** a 1 occurrence,  
La variable **y** a 2 occurrences  
La variable **z** a 3 occurrences

64



## 9- Variables libres et variables liées

### • Occurrence libre/liée d'une variable

Une occurrence de la variable  $x$  dans une formule  $\phi$  est une occurrence **libre** ssi elle ne se trouve dans le champs d'aucun quantificateur sur  $x$  (i.e.  $\forall x$  ou  $\exists x$ ). Dans le cas contraire, l'occurrence de  $x$  est dite **liée**.

Exemple:  $\forall x(p(x) \wedge r(x)) \vee r(y)$

la 1ère occurrence de  $x$  est liée  
la 2ème occurrence de  $x$  est liée  
L'occurrence de  $y$  est libre

65

## 9- Variables libres et variables liées

### Définition

- Si  $\phi$  est un **atome** alors toutes les occurrences de  $x$  dans  $\phi$  sont libres
- Si  $\phi = \neg \phi'$  alors les occurrences libres de  $x$  dans  $\phi$  sont celles de  $\phi'$
- Si  $\phi = \phi_1 \ K \ \phi_2$  alors les occurrences libres de  $x$  dans  $\phi$  sont l'union de celles de  $\phi_1$  et de  $\phi_2$  ( $K$ : un connecteur binaire)
- Si  $\phi = Qy \ \phi'$  alors les occurrences libres de  $x$  dans  $\phi$  sont celles de  $\phi'$  (où  $y$  est différent de  $x$ )
- Si  $\phi = Qx \ \phi'$  alors  $x$  n'a aucune occurrence libres dans  $\phi$

66

## 9- Variables libres et variables liées

Une variable peut avoir des occurrences libres et d'autres liées dans la même formule

Exemples  $\forall y ((p(x) \vee \exists x p(x)) \wedge q(y))$

la 1ère occurrence de **x** est **libre**

la 2ème occurrence de **x** est **liée**

L'occurrence de **y** est liée.

67

## 9- Variables libres et variables liées

### • Variable libre/liée

□ Une variable est **libre** dans une formule  $\phi$  ssi elle a au moins une occurrence libre dans  $\phi$

Exemple:  $\forall x \exists z p(x,z) \wedge p(y,z) \rightarrow p(x,y)$   
x , y , z sont libres

□ Une variable est **liée** dans une formule  $\phi$  ssi **toute ses occurrences sont liées** (ou elle n'a aucune occurrence libre) dans  $\phi$

Exemple:  $\forall x \exists z (p(x,z) \wedge p(z,x))$   
x et z sont liées

68

## 9- Variables libres et variables liées

### Remarque

□ Il est recommandé lorsque une même variable a des occurrences libres et d'autres liées de changer le nom soit de la variable libre, soit de la variable liée.

**Exemple**  $\forall x \exists z p(x,z) \wedge p(y, z) \rightarrow p(x,y)$   $x, y, z$  libres

est équivalente à

$\forall x \exists z p(x,z) \wedge p(y, z1) \rightarrow p(x1,y)$   $x1, y, z1$  libres

Ou bien  $\forall x2 \exists z2 p(x2,z2) \wedge p(y, z1) \rightarrow p(x1,y)$

□ Ceci est important, comme nous le verrons plus loin (chapitre sémantique)

□ Un tel changement de nom de variable a l'avantage de nous faire éviter toute confusion.

69

## 9- Variables libres et variables liées

### • Formule close/ formule ouverte

□ Une formule **close** (fermée) est une formule sans variables libres

□ Une formule **ouverte** est une formule qui a au moins une variable libre

### Exemples

$\phi1 = \forall y ((p(x) \vee \exists x p(x)) \wedge q(y))$  est ouverte  
 $\phi2 = \forall x \forall y ((p(x) \vee \exists x p(x)) \wedge q(y))$  est close

**On note**  $\phi(x1,x2,..xn)$  pour désigner une formule ouverte pour laquelle les variables libres sont  $x1,x2,..xn$ .

70

## Plan

1- Introduction

2- Alphabet

3-Terms d'un langage

4- Formule de la logique du 1er ordre

5-Sous-formule d'une formule du 1er ordre

6- Arbre de décomposition

7- Longueur - profondeur - complexité d'une formule

8- Simplification syntaxique d'une formule:

9- Variables libres et variables liées

10- Substitution

71

## 10- Substitution

□ une substitution de variables est un ensemble fini de couples associant à **une variable** un **terme**, notée:

$[t_1/x_1, \dots, t_n/x_n]$  tel que  $x_i = x_j$  implique  $i = j$ .

□ L'application de la substitution  $[t_1/x_1, \dots, t_n/x_n]$  à une formule  $\phi$  notée :  $(\phi)[t_1/x_1, \dots, t_n/x_n]$

est le résultat de la substitution *simultanée* dans  $\phi$  de chaque  $x_i$  par  $t_i$ .

72

## 10- Substitution

□ la substitution de la variable  $x$  par un terme  $t$  dans  $\phi$  revient à remplacer toutes les occurrences libres de  $x$  par  $t$  tel que la condition suivante soit vérifiée:

$\forall y$   $y$  : une variable de  $t$ ,  
 $x$  n'est pas dans le champs d'un quantificateur sur  $y$  dans  $\phi$

**Exemple**  $\phi = x \equiv y \vee \exists y p(x,y) \vee \exists x p(x,x)$

$$\phi [f(z)/x] = f(z) \equiv y \vee \exists y p(f(z),y) \vee \exists x p(x,x)$$

$$\phi [f(y)/x] = \phi$$

73

## 10- Substitution

**Exemple**

|                                      |                                |
|--------------------------------------|--------------------------------|
| $(p(x) \vee q(x,y)) [z/x]$           | $= p(z) \vee q(z,y)$           |
| $(p(x) \vee q(x,y)) [y/x]$           | $= p(y) \vee q(y,y)$           |
| $(\forall x q(x,y)) [z/x]$           | $= \forall x q(x,y)$           |
| $(p(x) \vee q(x,y)) [f(x)/x]$        | $= p(f(x)) \vee q(f(x),y)$     |
| $(\forall x q(x,y) \vee p(x)) [z/x]$ | $= \forall x q(x,y) \vee p(z)$ |

74