Chapitre-1
Logique propositionnelle
- Syntaxe -

Plan

1- Introduction
2- Proposition
3- Variable propositionnelle
4- Connecteur
5- Alphabet de la logique propositionnelle
6- Formule de la logique propositionnelle (forme propositionnelle
7- Sous-formule d'une forme propositionnelle
8- Analyse d'une forme propositionnelle (arbre de décomposition
9- Notation polonaise(préfixée) d'une formule
10- Longueur - profondeur - complexité d'une formule
11- Simplification syntaxique d'une forme propositionnelle
12- Substitution d'une variable propositionnelle

1-Introduction

La logique propositionnelle appelée aussi logique booléenne est la plus simple des logiques .

Elle est aussi la plus utilisée (domaine électronique, informatique, mathématique,...)

Comme toutes les logiques, la logique propositionnelle est définie par :

Une syntaxe: construction des formule

Une sémantique: signification des formule

Un système de preuve: démonstration des formules

3

Plan

4

2. Proposition

Une proposition est une assertion qui a la faculté d'être Vraie ou Fausse

C'est donc une phrase dont on peut dire sans ambiguïté si elle est Vraie ou Fausse

Exemples:

$\left.\begin{array}{l}\text{" Alger est la capitale de l'Algérie"} \\ \text{" un carré a 4 angles droits"} \\ \text{"il pleut"}\end{array}\right\}$ propositions

5

Il existe des assertions qui ne sont pas des propositions

Exemples:

$\left.\begin{array}{l}\text{" Demain il fera beau"} \\ \text{"Mohamed est assez grand",} \\ \text{"il y' a environ 80 étudiants dans l'amphi 10"} \\ \text{" Il est algérien "}\end{array}\right\}$ Non propositions

Dans la logique propositionnelle on s'intéresse à l'étude des propositions. Les autres types d'assertions sont étudiées dans d'autres logiques.

6

Un proposition peut être :

- **Atomique** : c'est une proposition indécomposable
- **Composée**

Exemples:

Atomique : "il pleut", "la route est mouillée", "il neige"

Composée : "Il pleut *et* la route est mouillée",

"il pleut *mais* il ne neige pas"

« il *ne* neige *pas*"

Plan

1- Introduction

2- Proposition

3- Variable propositionnelle

4- Connecteurs

5- Alphabet de la logique propositionnelle

6- Formule de la logique propositionnelle (forme propositionnelle

7- Sous-formule d'une forme propositionnelle

8- Analyse d'une forme propositionnelle (arbre de décomposition

9- Notation polonaise(préfixée) d'une formule

10- Longueur - profondeur - complexité d'une formule

11- Simplification syntaxique d'une forme propositionnelle

12- Substitution d'une variable propositionnelle

3- Variable propositionnelle

Une variable propositionnelle est une variable qui représente une proposition atomique

Les valeurs d'une variable propositionnelles sont des propositions atomiques

Les variables propositionnelles sont notées en général par des lettres en majuscules (ex: A, B, .. A1,A2,….A', A",.., A1')

9

Plan

1- Introduction

2- Proposition

3- Variable propositionnelle

4- Connecteurs

5- Alphabet de la logique propositionnelle

6- Formule de la logique propositionnelle (forme propositionnelle

7- Sous-formule d'une forme propositionnelle

8- Analyse d'une forme propositionnelle (arbre de décomposition

9- Notation polonaise(préfixée) d'une formule

10- Longueur - profondeur - complexité d'une formule

11- Simplification syntaxique d'une forme propositionnelle

12- Substitution d'une variable propositionnelle

10

4- Connecteurs:

A partir d'un certain nombre de propositions (vraie ou fausses) on peut construire d'autres propositions à l'aide d'opérations logiques

Une opération logique peut concerner une seule proposition (on dit qu'elle est unaire), deux assertions (binaire).

Les opérations concernant plus de deux propositions sont plutôt rares mais peuvent être construites à partir d'autres opérations (binaires et/ou unaires).

11

Les opérations logiques les plus courantes sont:

La négation : (non) opération unaire notée $\rceil$

La conjonction: (et)opération binaire notée $\wedge$

La disjonction: (ou)opération binaire notée $\vee$

L'implication: (si _alors) opération binaire notée $\rightarrow$

L'équivalence: (si et seulement si) opération binaire notée $\leftrightarrow$

12

Chacun de ces connecteurs est défini par une table de vérité

A	Non A
Vrai	Faux
Faux	Vrai

A	B	A et B	A ou B	Si A alors B	A ssi B
Faux	Faux	Faux	Faux	Vrai	Vrai
Faux	Vrai	Faux	Vrai	Vrai	Faux
Vrai	Faux	Faux	Vrai	Faux	Faux
Vrai	Vrai	Vrai	Vrai	Vrai	Vrai

Exemples:

Si le prophète Mohamed est mort alors le prophète Mohamed n’est pas mort **Faux**

Si le prophète Mohamed n’est pas mort alors le prophète Mohamed est mort **Vrai**

L’entier 5 est impair et Mohamed est mort, ou tout triangle a trois cotés. **Vrai**

Si la 100ème décimale de π est 9, donc si la 100ème décimale de π n'est pas 9 alors la 100ème décimal de π est 9. **Vrai**

Plan

5- Alphabet de la logique propositionnelle

L'*alphabet* de la logique propositionnelle est constitué de

- **un ensemble de *variables propositionnelles***
- **les *connecteurs* $\urcorner, \wedge, \vee, \rightarrow, \leftrightarrow$**
- **les séparateurs (ou parenthèses) (et)**

A partir de cet alphabet on peut construire plusieurs mots:
$A \wedge B \to C$ $\quad C \to$ $\quad \neg AB \wedge \neg C$ $\quad (A)\neg B$ $\quad \neg(A \wedge B)$ ….

Parmi tous les mots qu'on peut construire seules quelques uns vont former ce qu'on appelle les formules de la logique propositionnelle

17

Plan

1- Introduction

2- Proposition

3- Variable propositionnelle

4- Connecteurs

5- Alphabet de la logique propositionnelle

6- Formule de la logique propositionnelle (forme propositionnelle

7- Sous-formule d'une forme propositionnelle

8- Analyse d'une forme propositionnelle (arbre de décomposition

9- Notation polonaise(préfixée) d'une formule

10- Longueur - profondeur - complexité d'une formule

11- Simplification syntaxique d'une forme propositionnelle

12- Substitution d'une variable propositionnelle

18

6- Formule de la logique propositionnelle (Forme propositionnelle)

L'ensemble des *formules* de la logique propositionnelle est défini par:

- si A est une variable propositionnelle alors A est une formule
- si φ est une formule alors $(\rceil \varphi)$ est une formule
- si $\varphi 1$ et $\varphi 2$ sont des formules et $K \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ alors

  $\varphi 1 \; K \; \varphi 2$ est une formule

Exemples:

formules	`non-formules'
A	A + B
$(\rceil A)$	$A \rceil B$
$A \rightarrow B$	$\wedge$ A
$(A \wedge (B))$	$(\wedge)$
$(\rceil(A \wedge B) \vee C)$	(B
$(A \rightarrow (B \vee C))$	$(A \rightarrow (B \vee C)$

Définition:

On appelle forme propositionnelle tout assemblage φ pour lequel il existe une <u>suite finie</u> d'assemblages $\varphi1, \varphi2, .. \varphi n$ avec $\varphi n=\varphi$ et tel que pour chaque $1\leq i \leq n$ l'une des conditions suivantes est vérifiée:

- **φ_i est une variable propositionnelle**
- **il existe $j<i$ $\varphi i = \rceil\varphi_j$**
- **il existe $j1,j2<i$ et $K \in \{\wedge,\vee,\rightarrow,\leftrightarrow\}$ tel que $\varphi_i=\varphi_{j1}\, K\, \varphi_{j2}$**

21

Exemples:

- **$\varphi= ((A\wedge \rceil(B)) \rightarrow C)$ est une forme propositionnelle**

  car $\exists$ la suite finie $\varphi1,\varphi2,\varphi3,\varphi4,\varphi5 ,\varphi6/$ $\varphi6=\varphi$

  $\varphi1=A,$

  $\varphi2= B,$

  $\varphi3= C,$

  $\varphi4= \rceil(B),$

  $\varphi5= (A\wedge \rceil(B),$

  $\varphi6= ((A\wedge \rceil(B)) \rightarrow C)$

- **$\varphi'= \rceil(A)\rightarrow$ n'est pas une forme propositionnelle**

22

Plan

1- Introduction

2- Proposition

3- Variable propositionnelle

4- Connecteurs

5- Alphabet de la logique propositionnelle

6- Formule de la logique propositionnelle (forme propositionnelle

7- Sous-formule d'une forme propositionnelle

8- Analyse d'une forme propositionnelle (arbre de décomposition

9- Notation polonaise(préfixée) d'une formule

10- Longueur - profondeur - complexité d'une formule

11- Simplification syntaxique d'une forme propositionnelle

12- Substitution d'une variable propositionnelle

7- Sous-formule de la logique propositionnelle

Les *sous-formules* d'une formule φ sont définies par:

- **φ est une sous-formule de φ.**
- **Si ⌉φ' est une sous-formule de φ**

  alors φ' est une sous-formule de φ.
- **Si K∈ {∧,∨,→,↔} et φ1 K φ2 est une sous-formule de φ**

  alors φ1 et φ2 sont des sous-formules de φ .

Une sous-formule de φ est dite stricte si elle est différente de φ elle-même

Exemples:

- l'ensemble des sous-formules de $(((A \vee B) \wedge \neg A) \rightarrow C)$ est:

$\{ \; (((A \vee B) \wedge \neg A) \rightarrow C)$,
$((A \vee B) \wedge \neg A)$, C ,
$(A \vee B)$, $(\neg A)$,
A , B $\}$.

- l'ensemble des sous-formules de $\neg\neg\neg A$ est:
$\{\neg\neg\neg A, \;\; \neg\neg A, \;\; \neg A, \;\; A\}$.

25

Plan

1- Introduction
2- Proposition
3- Variable propositionnelle
4- Connecteurs
5- Alphabet de la logique propositionnelle
6- Formule de la logique propositionnelle (forme propositionnelle
7- Sous-formule d'une forme propositionnelle
8- Analyse d'une forme propositionnelle (arbre de décomposition)
9- Notation polonaise(préfixée) d'une formule
10- Longueur - profondeur - complexité d'une formule
11- Simplification syntaxique d'une forme propositionnelle
12- Substitution d'une variable propositionnelle

26

8- Analyse syntaxique d'une Formule (Arbre de décomposition)

Pour chaque formule φ, il est possible de retrouver les formules qui ont servi à la construire.

Pour cela, on construit l'arbre de décomposition de φ en se faisant guidé par les parenthèses utilisées dans φ.

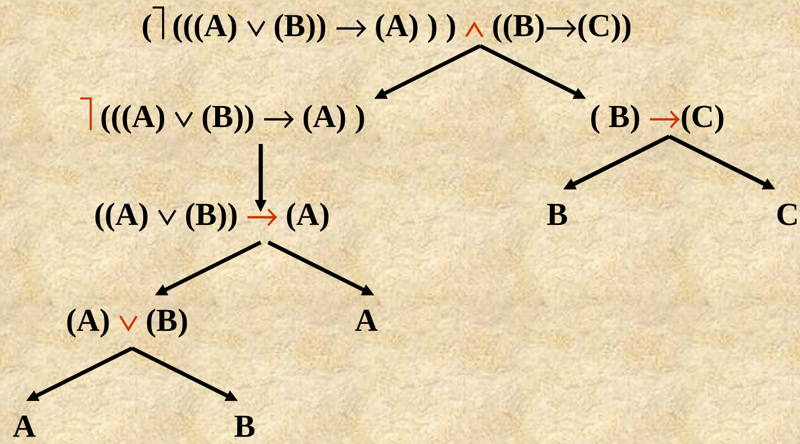
Description de la procédure de décomposition:

1. **on découpe au niveau du connecteur principal de la FP φ**
2. **on enlève les parenthèses les plus externes de chacun des arguments du connecteur principal**
3. **on répète l'opération de décomposition pour chacun des arguments jusqu'à arriver aux variables**

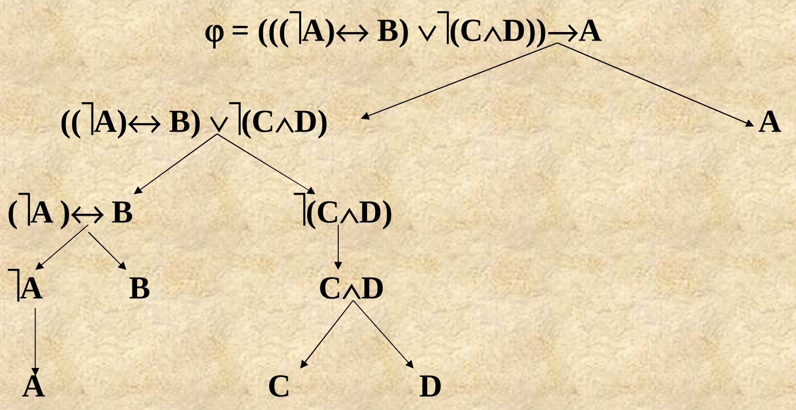
Exemples:

L'arbre de décomposition de

$$\varphi = (\neg(((A) \vee (B)) \rightarrow (A))) \wedge ((B) \rightarrow (C))$$

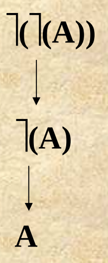


Exemples:



29

Exemples:



30

Remarques:

Aucun calcul ne doit être fait car on s'intéresse seulement à l'aspect syntaxique de la formule (comment elle a été construite et non pas à quoi elle est "sémantiquement" équivalente)

Chaque FP a un seul arbre de décomposition

Les nœud de l'arbre de décomposition d'une formules φ sont les sous-formules de φ

31

Plan

1- Introduction

2- Proposition

3- Variable propositionnelle

4- Connecteurs

5- Alphabet de la logique propositionnelle

6- Formule de la logique propositionnelle (forme propositionnelle

7- Sous-formule d'une forme propositionnelle

8- Analyse d'une forme propositionnelle (arbre de décomposition

9- Notation polonaise(préfixée) d'une formule

10- Longueur - profondeur - complexité d'une formule

11- Simplification syntaxique d'une forme propositionnelle

12- Substitution d'une variable propositionnelle

32

9- Notation polonaise d'une formule:

En Informatique, les FP sont souvent représentées en utilisant la notation dite "polonaise" qui consiste à mettre les connecteurs en tête des formes qu'il relient

La notation polonaise se déduit facilement à partir de l'arbre de décomposition

On parcourt l'arbre de haut en bas de la façon suivante:

<u>R</u>acine, Branche <u>G</u>auche, Branche <u>D</u>roite

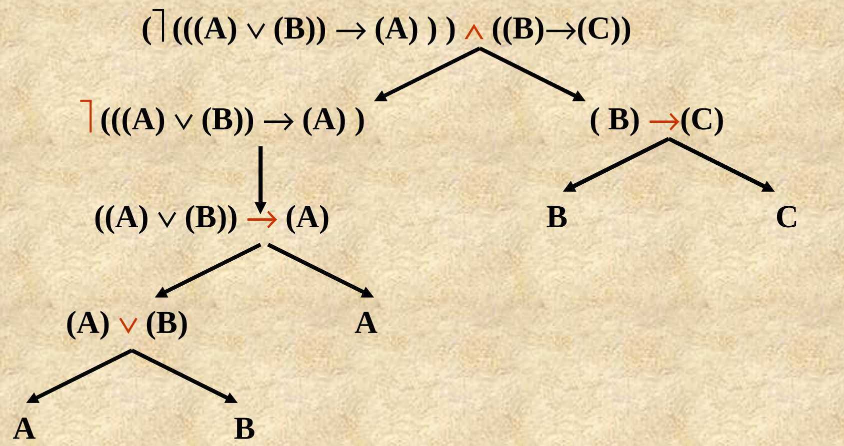
RGD

33

<u>Exemples:</u>

L'arbre de décomposition de

$$\varphi = (\neg(((A) \vee (B)) \rightarrow (A))) \wedge ((B) \rightarrow (C))$$

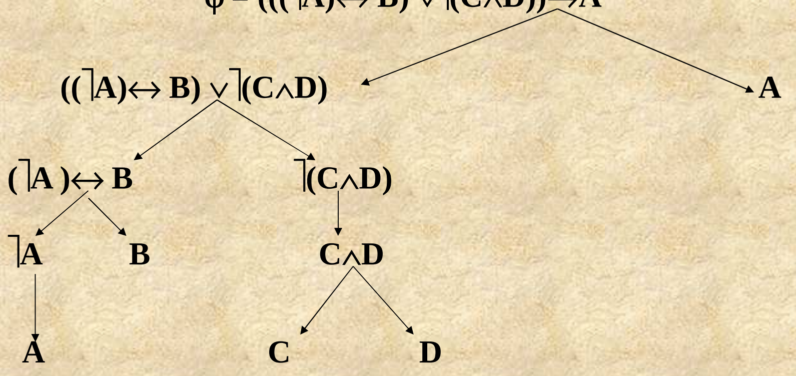


La notation polonaise: $\wedge \neg \rightarrow \vee A\, B\, A \rightarrow B\, C$

34

Exemples:

$\varphi = (((\neg A) \leftrightarrow B) \vee \neg(C \wedge D)) \rightarrow A$



La notation polonaise: $\rightarrow \vee \leftrightarrow \neg A\ B \neg \wedge C\ D\ A$

35

La notation polonaise permet de faciliter le calcul de la valeur de vérité d'une formule

La démarche à suivre consiste à:

- **parcourir l'expression de droite à gauche,**
- **si on rencontre un connecteur unaire ($\neg$),**

  on l'applique à la variable située à sa droite

  (on remplace ce connecteur et cette variable par la valeur de vérité résultat)
- **si on rencontre un connecteur binaire,**

  on l'applique aux deux formules à sa droite

  (on remplace ce connecteur et les deux formules par le résultat)

36

Exemples:

Soit A est vrai, B fausse , C est vrai

$$\varphi = (\rceil (((A) \vee (B)) \rightarrow (A))) \wedge ((B) \rightarrow (C))$$

la notation polonaise de φ est $\wedge \rceil \rightarrow V\ A\ B\ A \rightarrow B\ C$

$\wedge \rceil \rightarrow V\ A\ B\ A \rightarrow B\ C$

$\wedge \rceil \rightarrow V\ A\ B\ A$ **Vrai**

$\wedge \rceil \rightarrow$ **Vrai A Vrai**

$\wedge \rceil$ **Vrai Vrai**

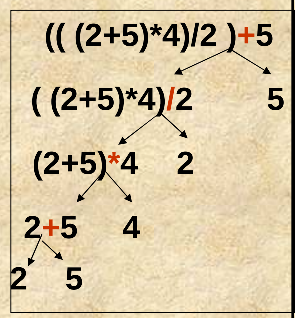
$\wedge$ **Faux Vrai**

Faux

37

Ce principe d'évaluation est utilisé aussi par exemple dans l'évaluation des expressions arithmétiques

(((2+5)*4)/2)+5



Notation polonaise: + / * + 2 5 4 2 5

+ / * + 2 5 4 2 5

+ / * 7 4 2 5

+ / 28 2 5

+ 14 5

19

38

Plan

10- Longueur - profondeur - complexité d'une formule

a) Longueur d'une formule:

La longueur d'une FP φ est le nombre de symboles (de l'alphabet) qu'elles contient.

Exemples:

Forme propositionnelle	longueur
A	1
˥(A)	4
˥((A)∧(B))	10
(˥(A))∧(B)	10
((A)∧(B))→(C))	14
((A)∧(B))→((C)∧(D))	19

a) Longueur d'une formule:

Elle est définie par:

- **Long(φ)= 1 si φ est une variable**
- **Long ((φ)) = 2 + long(φ)**
- **Long(⌉φ)= 1+ long(φ)**
- **Long (φ K φ') = 1 + long(φ)+long(φ')**

41

b) Profondeur d'une formule:

la profondeur d'une FP correspond donc à la profondeur de l'arbre de décomposition (i.e le nombre de branches qui sépare la racine de la feuille la plus éloignée)

Exemples:

Forme propositionnelle	Profondeur
A	0
⌉(A)	1
⌉((A)∧(B))	2
(⌉(A))∧(B)	2
((A)∧(B))→(C))	2
((A)∧(B))→((C)∧(D))	2

42

b) Profondeur d'une formule:

La profondeur d'une formule φ est définie de la manière suivante:

- **Si φ=A où A est une variable propositionnelle alors prof(φ)=0**
- **Si φ=⌉φ1 où où φ1 est une formule prof(φ)=1+prof(φ1)**
- **Si φ=φ1 K φ2 (K ∈{∧ , v , → ,↔})**

  prof(φ)=1+ max(prof(φ1), prof(φ2))

c) Complexité (ordre) d'une formule:

La complexité (ordre) d'une formule est donc le nombre de connecteurs qu'elle contient.

Exemples:

Forme propositionnelle	Complexité
A	0
⌉(A)	1
⌉((A)∧(B))	2
(⌉(A))∧(B)	2
((A)∧(B))→(C))	2
((A)∧(B))→((C)∧(D))	3

c) Complexité (ordre) d'une formule:

La complexité d'une formule propositionnelle φ est définie par:

- Si φ est une variable propositionnelle alors $c(\varphi)= 0$
- Si $\varphi=\rceil\varphi'$ alors $c(\varphi)=1+c(\varphi')$
- Si $\varphi=(\varphi'K\varphi'')$ et $K\in \{\wedge,\vee,\rightarrow,\leftrightarrow\}$ alors $c(\varphi)= c(\varphi')+c(\varphi'')+1$

45

Plan

46

11- Simplification syntaxique d'une formule

Le nombre de parenthèses rend parfois l'écriture des formes propositionnelles complexes

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

1- Suppression des parenthèses placées de part et d'autre d'une variable propositionnelle.

Exemple: $(A) \to ((B) \wedge (C))$ sera notée $A \to (B \wedge C)$

2- Suppression des parenthèses qui séparent des négations consécutives

Exemple: $\neg(\neg(\neg(\neg(A \wedge B))))$ sera notée $\neg\neg\neg\neg(A \wedge B)$

3- Application des Règles de précédence (priorité) entre connecteurs. L'ordre est comme suit:

1. $\neg$
2. $\wedge$, v avec associativité à gauche
3. $\to$, $\leftrightarrow$ avec associativité à gauche

Formule φ	Formule φ simplifiée
$((A \to B) \to C)$	$A \to B \to C$
$(((A \wedge B) \to ((\neg C) \vee D)) \to E)$	$A \wedge B \to \neg C \vee D \to E$
$(\neg(A)) \wedge (B)$	$\neg A \wedge B$
$((A \wedge B) \vee C) \wedge D)$	$A \wedge B \vee C \wedge D$
$((A) \to (B)) \wedge (\neg((C) \wedge (B)))$	$(A \to B) \wedge \neg(C \wedge B)$

49

Plan

1- Introduction

2- Proposition

3- Variable propositionnelle

4- Connecteurs

5- Alphabet de la logique propositionnelle

6- Formule de la logique propositionnelle (forme propositionnelle

7- Sous-formule d'une forme propositionnelle

8- Analyse d'une forme propositionnelle (arbre de décomposition

9- Notation polonaise(préfixée) d'une formule

10- Longueur - profondeur - complexité d'une formule

11- Simplification syntaxique d'une forme propositionnelle

12- **Substitution d'une variable propositionnelle**

50

12- Substitution d'une variable propositionnelle

a- Substitution simple

Une ***substitution*** **(ou substitution uniforme) associe à une variable propositionnelle A une formule α .**
Elle est notée $[\alpha/A]$.

L'application de $[\alpha/A]$ à une formule φ (notée $\varphi[\alpha/A]$) est le résultat du remplacement *simultané* de *toutes* les occurrences de A dans φ par α.

51

Exemples

$$((A \vee B) \wedge \neg A) \quad [(C \wedge D)/A]$$

$$= ((C \wedge D) \vee B) \wedge \neg (C \wedge D))$$

$(A \vee B)[r/A]$	$= r \vee B$
$(A \to B)[A/B]$	$= A \to A$
$(A \to B)[C/D]$	$= A \to B$
$((A \vee B) \wedge \neg A)[\neg A/A]$	$= (\neg A \vee B) \wedge \neg \neg A$
$(A \to C)[(B \to D)/A]$	$= (B \to D) \to C$

52

Définition: Si α est une formule, φ[α/A] est la formule obtenue en substituant toutes les occurrences de A dans φ par α

- si φ=A alors φ[α/A] = α
- si φ= B (B différent de A) alors φ[α/A]=φ
- si φ=⌉φ' alors φ[α/A]=⌉φ' [α/A]
- si φ=φ1Kφ2 alors φ[α/A]= φ1[α/A] K φ2[α/A]

où (K ∈{∧ , v , → ,↔})

b- Substitution simultanée

On peut généraliser à la substitution simultanée de n variables propositionnelles

φ [α1/A1, α2/A2, …. , αn/An] est la substitution dans φ des variables A1,..An par les formes α1,.. αn respectivement

(Les Ai sont deux à deux distincts i.e φ[α1/A, α2/A] n'est pas une substitution).

Exemples:

(A v B)[C/A,D/B]	= C v D
(A→B)[A/B,C/A]	= C→A
(A→B)[C/D,A/A]	= A→B
((A v B) ∧ ⌉A)[⌉A/A, A∧B/B]	= (⌉A v (A∧B)) ∧ ⌉⌉ A

c- Substitution composée

Plusieurs substitutions peuvent se succéder (composition de substitutions)

$((\varphi[\alpha 1/A1,\dots \alpha n/An]) [\beta 1/B1, \dots \beta p/Bp]) [\varphi 1/C1,\dots \varphi m/Cm]$

qui signifie substituer dans φ les variables Ai, par les formule αi . Dans la formule obtenue, substituer les Bi par lesβi. Ensuite substituer les Ci par les φi.

Exemples:

$((A \vee B)[B\to C/A])[\neg B/B]$	$= (\neg B\to C) \vee \neg B$
$((A\to B)[A/B])[C\wedge D/A]$	$= (C\wedge D)\to(C\wedge D)$
$((A\to B)[C/D])[A/A]$	$= A\to B$
$(((A\vee B)\wedge\neg A)[\neg A/A])[A\wedge B/B]$	$= (\neg A \vee (A\wedge B)) \wedge \neg\neg A$