

# Logique des propositions

S. Mazouz & K. Akli

USTHB

[Campusvirtuel.usthb.dz](http://Campusvirtuel.usthb.dz)

[smazouz@hotmail.com](mailto:smazouz@hotmail.com)

# Plan

1. Introduction
2. Langage
3. Système déductif
4. Etude Sémantique
5. Relation entre la syntaxe et la sémantique

# Introduction

**Proposition** : toute phrase, toute expression à laquelle on peut attribuer **une valeur de vérité** (vrai ou faux).

**Exemples** :

1. Un triangle isocèle possède deux côtés égaux.
2. La terre tourne autour de la lune
3.  $2+3=6$

- Rangez vos affaires (ordre)
- Quelle heure est-il?(Question)
- Combien avez-vous acheté ce livre?(Question)

# Introduction

- La logique des propositions permet:
  - La **représentation** des propositions d'une manière **formelle** à l'aide d'un **langage** ;
  - La **déduction de nouvelles propositions** à partir d'un ensemble de propositions en utilisant *un système déductif* ;
  - La **vérification formelle** de la véracité ou de la fausseté des propositions par l'utilisation de **tables de vérité**.

# Langage

Un langage formel est :

- Un ensemble de mots ou **symboles**. Cet ensemble est appelé **alphabet**.
- Et des règles de combinaison de ces mots. Ces règles indiquent comment construire des propositions complexes appelées **formules** (grammaire).

# Langage (Alphabet)

**Définition 1 (Alphabet) :** L'alphabet du langage  $L_p$  ( $\neg, \wedge$ ) se compose de trois classes de symboles :

1. **Symboles de variables propositionnelles** désignés par les lettres latines majuscules éventuellement indicées  $A, B, C, A_1, B_2, \dots$  ;
2. **Symboles logiques ou connecteurs** : ce sont les symboles qui permettent de lier les propositions entre elles.
  - un connecteur unaire (ou monaire)  $\neg$ , appelé « *non logique* »
  - un connecteur binaire  $\wedge$ , appelé « *et logique* » ou « *conjonction* ».

Ce sont des connecteurs primitifs ou de base.

3. **Symboles impropres** : les parenthèses ( et ).

# Langage(Formules)

**Définition 2 (Formules)** : L'ensemble des formules ou **expressions bien formées** (ebf) du langage  $L_p (\neg, \wedge)$  est défini récursivement de la manière suivante :

1. Toute variable propositionnelle de l'alphabet de  $L_p (\neg, \wedge)$  est une formule, dite **formule atomique** ;
2. Si  $\alpha$  et  $\beta$  sont des formules de  $L_p (\neg, \wedge)$  alors  $\neg \alpha$ ,  $\alpha \wedge \beta$  et  $(\alpha)$  sont des formules, dites **formules composées**.
3. Clause de clôture : Rien d'autre n'est une formule (*une expression ne peut être une formule qu'en vérifiant les points 1 ou 2*).

# Langage

## Exemples

1.  $A, B, C$  sont des formules atomiques de  $L_p(\neg, \wedge)$ .
2.  $\neg A, \neg\neg A, \neg A \wedge B, \neg(A \wedge B), \neg\neg(A \wedge \neg B)$  sont des formules de  $L_p(\neg, \wedge)$ .
3. Par contre  $A \neg \wedge B, \wedge A \wedge B, (A \wedge \neg(B \rightarrow C))$  ne sont pas des formules de  $L_p(\neg, \wedge)$ .

## Remarque

Les lettres grecques  $\alpha, \beta, \delta, \gamma, \dots$  ne sont pas des symboles du langage  $L_p(\neg, \wedge)$ . Ils désignent les **noms de formules**; ce sont des **symboles du métalangage** qui nous sert à décrire le langage  $L_p(\neg, \wedge)$ .

# Langage

## Connecteurs d'abréviations

Connecteurs	notation	définition
ou logique	$\vee$	$\alpha \vee b =_{\text{déf}} \neg(\neg \alpha \wedge \neg \beta)$
Implication	$\rightarrow$	$\alpha \rightarrow b =_{\text{déf}} \neg(\alpha \wedge \neg \beta)$
Équivalence	$\leftrightarrow$	$\alpha \leftrightarrow b =_{\text{déf}} (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$ $=_{\text{déf}} \neg(\alpha \wedge \neg \beta) \wedge \neg(\beta \wedge \neg \alpha)$

**Priorité entre les connecteurs** (par ordre décroissant) :

**Les parenthèses** (des plus internes aux plus externes) ,  $\neg$  (le plus interne d'abord) ,  
 $\wedge$  **et**  $\vee$  (de gauche à droite),  $\rightarrow$  **et**  $\leftrightarrow$  (de gauche à droite)

Formules	Formules simplifiées
$((\alpha \wedge \beta) \wedge \delta)$	$\alpha \wedge \beta \wedge \delta$
$((\alpha \rightarrow \beta) \rightarrow \delta)$	$\alpha \rightarrow \beta \rightarrow \delta$
$((\alpha \rightarrow \beta) \vee \delta)$	$(\alpha \rightarrow \beta) \vee \delta$

# Langage

## Formalisation

si le train arrive en retard et il n'y a pas de taxis à la gare  
alors l'invité arrive en retard

Considérons les variables propositionnelles suivantes :

**R : Le train arrive en Retard**

**T : il y a des taxis à la gare**

**A : l'invité arrive en retard**

On obtient :

$$(R \wedge \neg T) \rightarrow A$$

# Systeme déductif

## Terminologie ou vocabulaire

1. **Axiome** : énoncé admis comme vrai.

### Axiomes d'Euclide (IV<sup>e</sup> siècle av JC) :

1. Étant donnés deux points A et B, il existe une droite passant par A et B;
2. Tout segment [AB] est prolongeable en une droite passant par A et B;
3. Pour tout point A et tout point B distinct de A, on peut décrire un cercle de centre A passant par B;
4. Tous les angles droits sont égaux entre eux;
5. **Par un point extérieur à une droite, on peut mener une parallèle et une seule à cette droite.**



**Axiome des parallèles**

# Systeme déductif

2. **Théorème** : énoncé qui est conséquence logique des axiomes.

**Exemple** Géométrie Euclidienne

## **Théorème de Pythagore**

Dans un triangle **rectangle**, le carré de la longueur de **l'hypoténuse** est égal à la somme des carrés des longueurs des côtés de l'angle droit.

# Systeme d'eductif

- En cherchant à démontrer par l'absurde le cinquième postulat d'Euclide d'après lequel par un point extérieur à une droite, il ne peut passer qu'une seule parallèle,
- Lobatchevski postule que « par un point extérieur à une droite passe plus d'une droite parallèle ».
- On démontre qu'alors il y a une infinité de droites parallèles.
- Il se rend compte alors que cela n'a rien de contradictoire et une nouvelle géométrie est née.  
C'est la naissance des géométries non- euclidiennes.

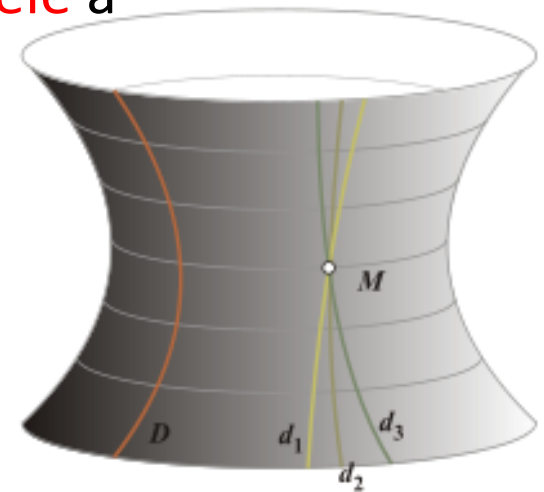
# Systeme déductif

Géométries non Euclidiennes : Le 5<sup>ème</sup> postulat d'Euclide est remplacé par un autre axiome.

**Géométrie Hyperbolique 19<sup>e</sup> siècle** (Lobatchevski, Klein et Poincaré)

Etant donné une droite et un point extérieur à cette droite, **il existe infinité de droite parallèle** à cette droite passant par ce point.

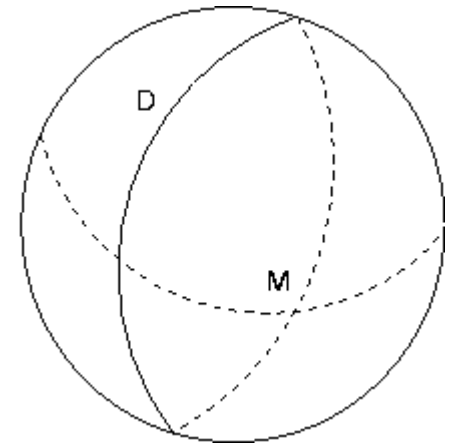
- Le [théorème de Pythagore](#) n'est plus valable
- La [somme des angles d'un triangle](#) n'est plus égale à  $180^\circ$ .



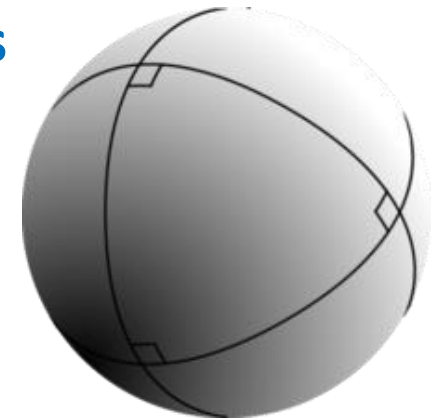
# Systeme déductif

## Géométrie elliptique (Riemann 1867):

Etant donnés une droite et un point extérieur à cette droite, **il n'existe aucune droite parallèle** à cette droite passant par ce point.



Il existe des triangles dont la **somme des angles est supérieure à 180°**.



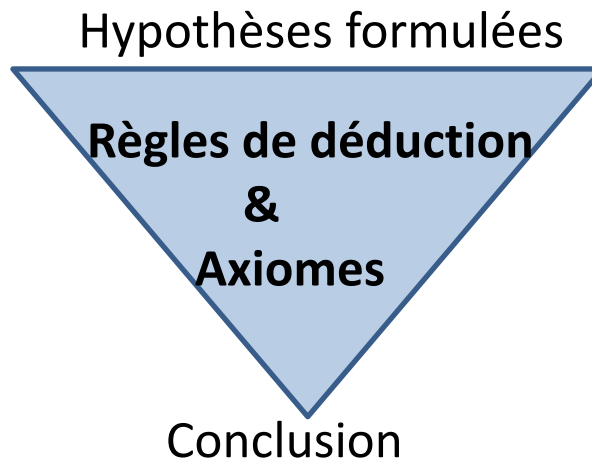
# Systeme déductif

3. **Preuve** : Argumentation permettant

- en un **nombre fini d'étapes** d'obtenir de nouvelles conséquences à partir éventuellement d'hypothèses

- en se basant sur un **ensemble fini d'axiomes** et des **règles de déduction**.

**Ce procédé est appelé déduction ou démonstration.**



# Systeme d'eduction

Il existe plusieurs systemes de deduction logique

1. Systeme de deduction axiomatique - Hilbert :

- Ensemble fini d'axiomes
- La regle de deduction « **modus ponens** »  
 $\alpha \rightarrow \beta$  est un theoreme et  $\alpha$  est un theoreme alors  $\beta$  est un theoreme
- La regle de substitution

remplacer une variable propositionnelle par une formule dans un axiome.

Demontrer des theoremes a partir d'axiomes

2. Systeme de deduction naturelle -- Gentzen :

Ensemble fini de regles de deduction proches du raisonnement naturel.

Pour chaque connecteur, on distingue deux types de regles :

- les *regles d'introduction*
- les *regles d'elimination*.



# Systeme déductif

- La déduction naturelle démontre des propositions sous des hypothèses.

Pour dire qu'une proposition  $\alpha$  est conséquence d'un ensemble  $\Gamma$  d'hypothèses, on écrit  $\Gamma \vdash \alpha$  qui se lit de  $\Gamma$  on déduit  $\alpha$ .

- Les formules de  $\Gamma$  sont dites **hypothèses de base de la déduction**
  - $\alpha$  est appelée **conclusion de la déduction**
  - Le symbole  $\vdash$  est appelé symbole de déduction
- Si  $\Gamma$  est vide,  $\alpha$  est un **théorème** et on écrit  $\vdash \alpha$ .

# Systeme d'eduction

- Le systeme d'eduction « naturel » dans le langage  $L_p(\neg, \wedge)$  est muni de quatre regles de deduction adoptees des regles naturelles de GENTZEN (1935).
- Les regles de deduction (appelees aussi regles d'inférence) s'écrivent :

$$\frac{\alpha_1 \dots \alpha_n}{\beta} \quad (R)$$

- Les formules  $\alpha_1, \dots, \alpha_n$  sont appelees premisses ou hypotheses de la regle
- La formule  $\beta$  est appelee conclusion
- R est l'abréviation du nom de la regle de deduction

# Systeme d'eduction

## Règles du connecteur $\wedge$

- Règle d'introduction ( $I \wedge$ ) :

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta} \quad I \wedge$$

Elle exprime le fait que :

« si on peut déduire  $\alpha$

et si on peut déduire  $\beta$

alors on peut déduire la formule  $\alpha \wedge \beta$  ».

# Systeme d'eduction

## Règles du connecteur $\wedge$

Règle d'élimination ( $E \wedge$ )

$$\frac{\alpha \wedge \beta}{\alpha} \quad E \wedge$$

$$\frac{\alpha \wedge \beta}{\beta} \quad E \wedge$$

La déduction de  $\alpha \wedge \beta$

permet celle de  $\alpha$  d'une part  
ou de  $\beta$  d'autre part».

# Systeme d'eduction

## Rgles du connecteur $\neg$

- Rgle d'Elimination ( $E \neg$ )

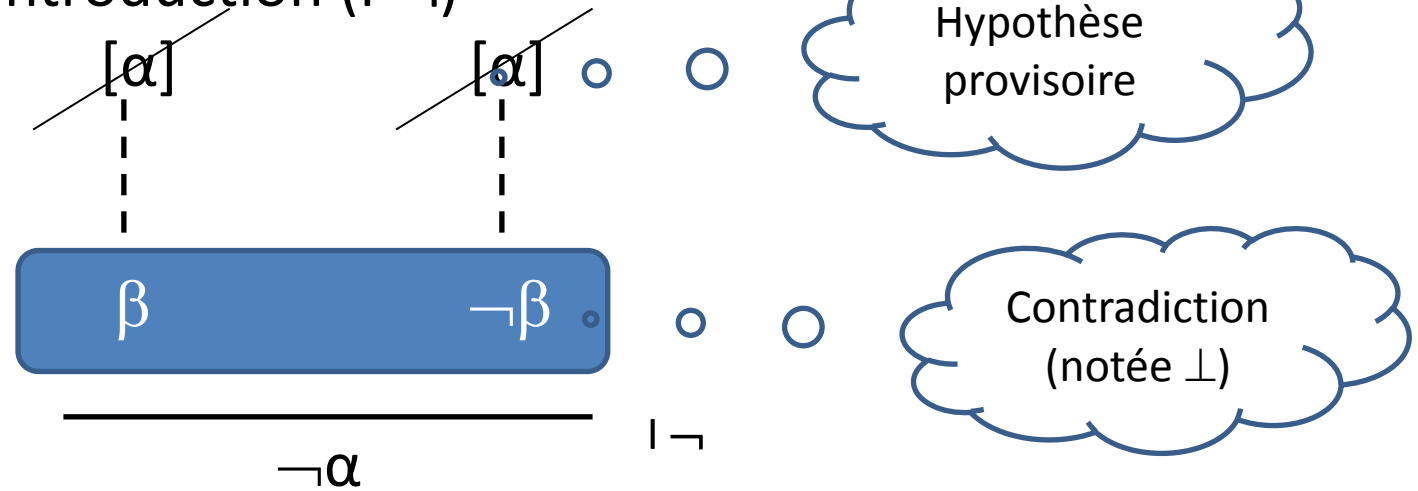
$$\frac{\neg \neg \alpha}{\alpha} \quad E \neg$$

La d'eduction de  $\neg \neg \alpha$  permet celle de  $\alpha$ .

# Systeme déductif

## Règles du connecteur $\neg$

- Règle d'Introduction ( $I \neg$ )



Si **sous l'hypothèse « provisoire »  $\alpha$** ,  
on peut déduire une formule et sa négation (contradiction),  
alors on peut **déduire la négation de  $\alpha$**  (ie  $\neg\alpha$ ) à partir de toutes les  
prémises antérieures sauf  $\alpha$  ( $\alpha$  est éliminée, on dit aussi que  $\alpha$  est  
déchargée).

# Systeme d'eductif

## Rgles du connecteur $\neg$

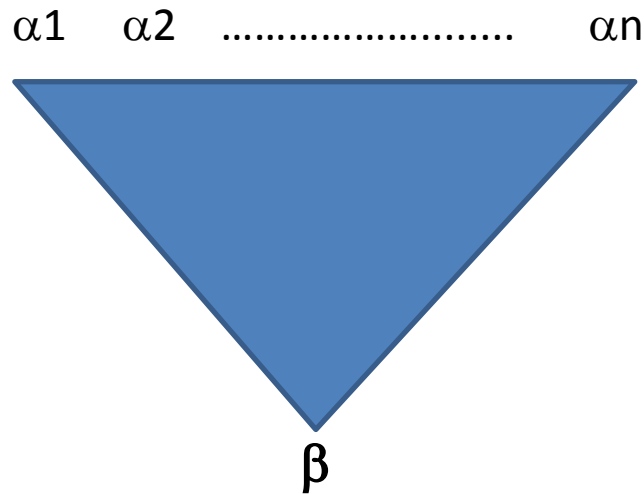
**Remarque** Les hypotheses provisoires sont mises entre crochets.

$$\frac{\begin{array}{c} \text{[}\alpha\text{]} \\ \vdots \\ \beta \wedge \neg\beta = \perp \end{array}}{\neg\alpha} \quad \neg\text{I}$$

# Systeme déductif

**Déduction** : Une déduction naturelle dans le système déductif  $L_p(\neg, \wedge)$

- est un **arbre fini** utilisant les règles naturelles de Gentzen ( $I\neg$ ), ( $E\neg$ ), ( $I\wedge$ ) et ( $E\wedge$ )
- dont les feuilles sont les prémisses (hypothèses formulées ou faits observés)  $\alpha_1, \alpha_2, \dots, \alpha_n$ .
- et la racine est la conclusion  $\beta$ .



**Hypothèses**

**Règles naturelles de Gentzen**

**Conclusion**

# Systeme d'eduction

## Notations

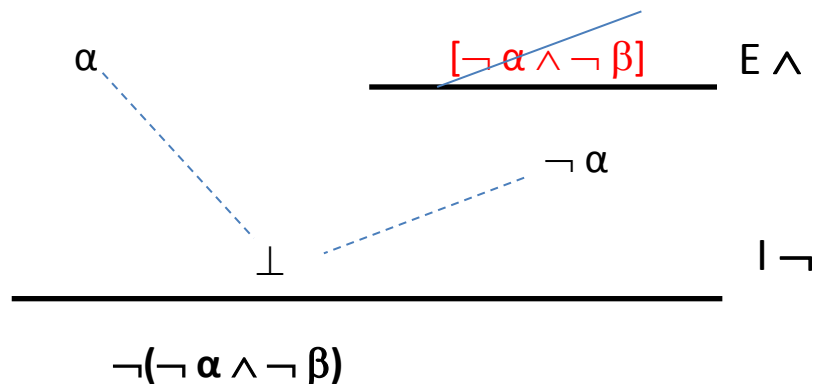
Soient  $\Gamma$  un ensemble de formules et  $\beta$  une formule, on note :

- $\Gamma \vdash \beta$  : s'il existe une deduction  $D$  ayant pour conclusion  $\beta$  et dont **toutes les premisses non eliminees sont dans  $\Gamma$** .  
On dit que  $D$  est une deduction de  $\beta$  a partir de  $\Gamma$ .
- $\vdash \beta$  : s'il existe une deduction  $D$  ayant pour conclusion  $\beta$  et **dont toutes les premisses ont ete eliminees**.  
Dans ce cas,  $\Gamma = \emptyset$ , on dit que  $\beta$  est un theoreme dans  $L_p(\neg, \wedge)$ .
- Si  $\Gamma$  est un ensemble de formules et  $\beta, \alpha, \alpha_1, \dots, \alpha_n$  sont des formules, alors on notera :
  - $\alpha_1, \dots, \alpha_n \vdash \beta$  au lieu de  $\{\alpha_1, \dots, \alpha_n\} \vdash \beta$
  - $\Gamma, \alpha \vdash \beta$  au lieu de  $\Gamma \cup \{\alpha\} \vdash \beta$

# Systeme d'eduction

**Exemple** montrer la d'eductibilit' de  $\alpha \vee \beta$  ' partir de  $\alpha$  qu'on notera :  $\alpha \vdash \alpha \vee \beta$  ?

- Etant donn' qu'on est dans le syst'eme d'eductif de  $L_p(\neg, \wedge)$ , on doit remplacer le  $\vee$  par sa d'efinition. On doit alors prouver :  $\alpha \vdash \neg(\neg \alpha \wedge \neg \beta)$
- De fa'con g'n'rale, si la conclusion est de la forme  $\neg \beta$ , on peut ajouter  $\beta$  comme hypoth'ese provisoire. Dans cet exemple, on ajoute la formule  $\neg \alpha \wedge \neg \beta$ . Les hypoth'es ajout'es sont mises entre crochets.



- De l'hypoth'ese provisoire  $\neg \alpha \wedge \neg \beta$ , on d'eduit par la r'egle ( $E \wedge$ ) la formule  $\neg \alpha$  et on obtient une contradiction avec la pr'emisse  $\alpha$ . Dans ce cas, on utilise la r'egle ( $I \neg$ ) qui permet d'eliminer la formule  $\neg \alpha \wedge \neg \beta$  et de d'eduire sa n'egation. On obtient notre conclusion et il ne reste que  $\alpha$ .

# Systeme déductif

Montrer la déduction  $\vdash \alpha \vee \neg \alpha$  qui peut se traduire aussi par montrer que  $\alpha \vee \neg \alpha$  est un théorème?

# Systeme déductif

- Montrer qu'à partir de  $\alpha \rightarrow \beta$  et  $\alpha$  on peut déduire  $\beta$  i.e.  $\alpha \rightarrow \beta, \alpha \vdash \beta$ .

# Systeme déductif

Démontrer l'équivalence entre les deux déductions :

$$\Gamma, \alpha \vdash \beta \Leftrightarrow \Gamma \vdash \alpha \rightarrow \beta ?$$

# Systeme d'eduction

Montrer la deduction  $\vdash (\alpha \rightarrow \beta) \rightarrow (\neg \beta \rightarrow \neg \alpha)$

# Systeme déductif

Montrer que le raisonnement suivant est correct :

Hypothèses :

1. si le train arrive en retard et il n'y a pas de taxis à la gare alors l'invité arrive en retard
- 2- l'invité n'est pas arrivé en retard
- 3- le train est arrivé en retard

**Conclusion : il y avait des taxis à la gare**

# Logique des propositions partie 2

S. Mazouz & K. Akli

USTHB

[Campusvirtuel.usthb.dz](http://Campusvirtuel.usthb.dz)

# Logique des propositions

1. Langage
2. Système déductif
3. Sémantique

# Sémantique

- Donner une sémantique à un langage formel :
  - définir une fonction (au sens mathématique), appelée **Interprétation**,
  - qui est capable d'associer à toute formule bien formée un « sens ».
- Le sens d'une formule sera simplement une valeur de vérité (vrai ou faux).

# Sémantique

On procède en deux étapes :

1. On donne un **sens aux symboles du langage**  
c-à-d d'une part les **variables propositionnelles**  
et d'autre part les **connecteurs**
  
2. On donne **une méthode de calcul**  
pour déterminer le **sens** d'une **formule complexe**  
à partir du **sens des constituants plus simples**.

# Sémantique

## Interprétation (Tables de vérité)

1. On interprète les variables propositionnelles (v.p) en leur associant une valeur de vérité.

Chaque v.p ne peut prendre que deux valeurs de vérité : vrai ou faux, notées V et F

2. On interprète les connecteurs en associant à chaque connecteur sa table de vérité qui définit son sens.

# Sémantique

## Interprétation (Tables de vérité)

A	$\neg A$
V	<b>F</b>
F	V

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
V	V	V	V	V	V
V	F	<b>F</b>	V	<b>F</b>	<b>F</b>
F	V	<b>F</b>	V	V	<b>F</b>
F	F	<b>F</b>	<b>F</b>	V	V

Grec Antique « Du vrai suit le vrai, du faux suit le faux, du faux suit le vrai mais du vrai le faux ne peut s'ensuire »

# Sémantique

## Interprétation (Tables de vérité)

3. On interprète une formule  $\alpha$  en lui associant une table de vérité.

La méthode de calcul repose sur la **décomposition de cette formule en sous-formules**.

**Exemple** : Etablir la table de vérité de la formule  $\alpha = \neg A \wedge B \rightarrow \neg C \vee A$

A	B	C	$\neg A$	$\neg C$	$\neg A \wedge B$	$\neg C \vee A$	$\alpha$
V	V	V	F	F	F	V	V
V	V	F	F	V	F	V	V
V	F	V	F	F	F	V	V
V	F	F	F	V	F	V	V
F	V	V	V	F	V	F	F
F	V	F	V	V	V	V	V
F	F	V	V	F	F	F	V
F	F	F	V	V	F	V	V

Instanciation  
 $A=V, B=V$  et  
 $C=F$

Une instanciation  $I$  est une application de l'ensemble des variables propositionnelles dans  $\{V, F\}$

# Sémantique

## Interprétation (Tables de vérité)

L'interprétation d'une formule  $\alpha$ , pour chaque instantiation, peut être donnée par les règles suivantes :

1.  $\neg\alpha = V$  ssi  $\alpha = F$
2.  $\alpha \wedge \beta = V$  ssi  $\alpha = V$  et  $\beta = V$
3.  $\alpha \vee \beta = V$  ssi  $\alpha = V$  ou  $\beta = V$
4.  $\alpha \rightarrow \beta = V$  ssi  $\alpha = F$  ou  $\beta = V$
5.  $\alpha \leftrightarrow \beta = V$  ssi  $(\alpha = V \text{ et } \beta = V)$  ou  $(\alpha = F \text{ et } \beta = F)$

# Sémantique

## Définition(Satisfaisable)

1. Une formule  $\alpha$  est satisfaisable s'il existe une instanciation  $I$  (une ligne du TV) donnant la valeur  $V$  à  $\alpha$ .

$\alpha$  est dite satisfaite pour cette instanciation ;

**Exemple** La formule  $P \rightarrow Q$  est satisfaisable.

En effet, il suffit de prendre l'instanciation  $I$  telle que :  $P=V$  et  $Q=V$  par exemple.

**Définition** (Modèle d'une formule).

Une instanciation  $I$  qui donne la valeur  $V$  à une formule  $\alpha$  est un **modèle** de la formule.

**Exemple** L'instanciation  $I / P=V$  et  $Q=V$  est un modèle de la formule  $P \rightarrow Q$

# Sémantique

## Définition (Ensemble satisfaisable)

Un ensemble de formules  $\Gamma$  est satisfaisable s'il existe une instantiation pour laquelle **toutes les formules de  $\Gamma$  sont vraies**.

**Exemple** L'ensemble  $\{P \rightarrow Q, P \wedge Q\}$  est satisfaisable.

En effet, l'instanciation  $I$  telle que  $P=V$  et  $Q=V$  donne la valeur  $V$  à  $P \rightarrow Q$  et la valeur  $V$  à  $P \wedge Q$

## Définition (Modèle d'un ensemble de formules)

Une instantiation  $I$  est un modèle d'un ensemble de formules  $\Gamma$  ssi  $I$  est un **modèle de chaque formule** de  $\Gamma$ .

L'interprétation  $I / P=V$  et  $Q=V$  est un modèle de  $\{P \rightarrow Q, P \wedge Q\}$

# Sémantique

Exemples

Instanciation	A	B	$\alpha=A\wedge\neg B$	$\beta=B\rightarrow A$	$\delta=A\rightarrow B$
1	V	V	F	V	V
2	V	F	V	V	F
3	F	V	F	F	V
4	F	F	F	V	V

La formule  $\alpha$  est satisfaisable car **il existe une instanciation** (ligne 2 :  $A=V$  et  $B=F$ ) où  $\alpha=V$ .

La formule  $\beta$  est aussi satisfaisable car **il existe une instanciation** (ligne 1 par exemple) où  $\beta=V$

L'instanciation 2 est un modèle de  $\alpha$  et l'instanciation 1 est un modèle de  $\beta$ .

L'ensemble des formules  $\{\alpha, \beta\}$  est satisfaisable

car **il existe une instanciation** (ligne 2) où  $\alpha=V$  et  $\beta=V$ . L'instanciation 2 est un modèle de  $\{\alpha, \beta\}$

L'ensemble des formules  $\{\alpha, \beta, \delta\}$  n'est pas satisfaisable

car il **n'existe aucune instanciation** où les **trois formules** soient **vraies en même temps**.

**Remarque** Un ensemble de formules non satisfaisable ne possède pas de modèle.

# Sémantique

## Définition (Tautologie)

Une formule  $\alpha$  est une tautologie (ou valide) ssi elle est vraie pour toute instantiation, et on note  $\models \alpha$ .

## Définition (Antilogie)

Une formule  $\alpha$  est une antilogie ssi elle est fausse pour toute instantiation.

## Remarque

La valeur de vérité d'une tautologie ou d'une antilogie est indépendante des valeurs de vérité des variables propositionnelles qui y apparaissent.

## Exemples

- $A \vee \neg A$  est une tautologie
- $A \wedge \neg A$  est une antilogie.
- $A \rightarrow (B \rightarrow A)$  est une tautologie.

Est-ce que  $A \rightarrow A$  est une tautologie?

Est-ce que  $(A \rightarrow B) \rightarrow A$  est une tautologie ?

A	B	$B \rightarrow A$	$A \rightarrow (B \rightarrow A)$
V	V	V	V
V	F	V	V
F	V	F	V
F	F	V	V

# Sémantique

## Remarques

Soient  $\alpha$  et  $\beta$  deux formules :

1.  $\models \neg\alpha$  ( $\neg\alpha$  est une tautologie)

$\Leftrightarrow \alpha$  est une antilogie

$\Leftrightarrow \alpha$  n'est pas satisfaite c à d il n'existe aucune instantiation satisfaisant  $\alpha$

2.  $\models \neq \alpha$  ( $\alpha$  n'est pas une tautologie)

$\Leftrightarrow$  il existe au moins une instantiation ne satisfaisant pas  $\alpha$

Donc non tautologie n'implique pas antilogie.

# Sémantique

3.  $\models \alpha \wedge \beta \iff \models \alpha \text{ et } \models \beta$

4.  $\models \alpha \rightarrow \beta \iff$  *pour chaque instantiation  
si  $\alpha = V$  alors  $\beta = V$*

5.  $\models \alpha \rightarrow \beta \text{ et } \models \alpha \implies \models \beta$

6.  $\models \alpha \leftrightarrow \beta \iff$  *pour chaque instantiation  
 $\alpha$  et  $\beta$  ont même valeur de vérité*

7.  $\models \alpha \vee \beta \iff$  *pour chaque instantiation  
 $\alpha = V$  ou  $\beta = V$*

# Sémantique

## Définition (Equivalence Logique)

Soient  $\alpha$  et  $\beta$  deux formules, on dit que  $\alpha$  est logiquement équivalente à  $\beta$ , notée  $\alpha \equiv \beta$ , ssi

$\alpha$  et  $\beta$  ont même valeur de vérité pour chaque instantiation.

## Exemples

$$\neg\neg\alpha \equiv \alpha$$

Double négation

$$\alpha \wedge \beta \equiv \beta \wedge \alpha$$

commutativité

$$\alpha \vee \beta \equiv \neg(\neg\beta \wedge \neg\alpha)$$

$$\alpha \rightarrow \beta \equiv \neg(\alpha \wedge \neg\beta)$$

# Sémantique

## Remarque

Ne pas confondre l'équivalence logique  $\equiv$  avec le connecteur logique  $\leftrightarrow$ :

- $\leftrightarrow$  est un symbole de l'alphabet du langage propositionnelle
- $\equiv$  est un symbole du méta langage.

## Propriétés de l'équivalence logique

1.  $\equiv$  est une relation d'équivalence sur l'ensemble des formules

$$2. \alpha \equiv \beta \leftrightarrow \models \alpha \leftrightarrow \beta$$

# Sémantique

## Interprétation (Tables de vérité)

Complexité de la méthode des TVs :

Etant donné une formule  $\alpha$  avec  $n$  v.ps,

il y a  $2^n$  interprétations (instanciations) possibles.

Le temps de calcul est exponentiel avec le nombre de variables propositionnelles

Par exemple, une formule avec 100 vps possède

1 267 650 600 228 229 401 496 703 205 376 interprétations

Si on teste 1 milliard d'interprétations par seconde,

on en aura pour 40 196 936 841 311 années !!!

# Sémantique

## Théorème de substitution

Soient  $\alpha$  une formule contenant la v.p  $A$   
et  $\alpha'$  la formule obtenue en substituant toutes les  
occurrences de  $A$  dans  $\alpha$  par une formule  $\beta$  alors

$$\text{si } \models \alpha \quad \text{alors } \models \alpha'$$

### Exemple

$$\alpha = (A \vee B) \leftrightarrow (B \vee A),$$

La substitution de toutes les occurrences de  $B$  par la formule  $A \wedge \neg B$

$$\alpha' = \alpha [A \wedge \neg B / B] = (A \vee (A \wedge \neg B)) \leftrightarrow ((A \wedge \neg B) \vee A)$$

Or  $\alpha$  est une tautologie (en effet,  $A \vee B \equiv B \vee A$ )

donc  $\alpha'$  est une tautologie

Le théorème de substitution permet de découvrir de nouvelles tautologies sans passer par les tables de vérité.

# Sémantique

## Théorème de remplacement

Soit  $\beta$  une sous formule de  $\alpha$ . Si  $\beta \equiv \beta'$  alors

1. Le remplacement de  $\beta$  par  $\beta'$  dans  $\alpha$  donne une formule  $\alpha'$  équivalente à  $\alpha$  ( $\alpha \equiv \alpha'$ ).
2. Si  $\models \alpha$  alors  $\models \alpha'$ .

Ainsi, on ne change pas la valeur de vérité d'une formule en remplaçant une sous-formule par une sous-formule équivalente.

# Sémantique

Ce théorème nous permet de produire une démonstration mathématique au sens classique en mathématique :

partant d'une formule qu'on sait vraie,

on peut produire par **remplacement de nouvelles formules** en préservant l'équivalence ;

## Exemples

Montrer l'équivalence  $\alpha \rightarrow \beta \equiv \neg \beta \rightarrow \neg \alpha$

1<sup>ère</sup> méthode : Utiliser les tables de vérité

2<sup>ème</sup> méthode : Utiliser le théorème de remplacement en se basant sur des équivalences déjà prouvées ou plus simples à faire.

$$\alpha \rightarrow \beta \equiv \neg(\alpha \wedge \neg \beta)$$

$$\equiv \neg(\neg \beta \wedge \alpha)$$

$$\equiv \neg(\neg \beta \wedge \neg \neg \alpha)$$

$$\equiv \neg \beta \rightarrow \neg \alpha$$

$$\text{car } \alpha \wedge \beta \equiv \beta \wedge \alpha$$

$$\text{car } \alpha \equiv \neg \neg \alpha$$

$$\text{car } \neg(\alpha \wedge \neg \beta) \equiv \alpha \rightarrow \beta$$

Montrer que  $P \rightarrow (Q \wedge R) \equiv (P \rightarrow Q) \wedge (P \rightarrow R)$

# Sémantique

## Corollaire

Une formule est logiquement équivalente à une formule qui ne contient que les connecteurs  $\neg$  et  $\wedge$ . On dira que  $\{\neg, \wedge\}$  est un **système complet de connecteurs**.

## Preuve

En utilisant le théorème de remplacement et les équivalences logiques suivantes :

$$\alpha \vee \beta \equiv \neg(\neg\beta \wedge \neg\alpha)$$

$$\alpha \rightarrow \beta \equiv \neg(\alpha \wedge \neg\beta)$$

$$\alpha \leftrightarrow \beta \equiv \neg(\alpha \wedge \neg\beta) \wedge \neg(\beta \wedge \neg\alpha)$$

On obtient une formule équivalente à la formule initiale et où tous les connecteurs  $\vee$ ,  $\rightarrow$  et  $\leftrightarrow$  sont éliminés.

# Sémantique

## Systeme Complet de Connecteurs

### Définition (système complet de connecteurs)

Un ensemble  $\Omega$  de connecteurs est un système complet de connecteurs (SCC) si toute formule de la forme  $\neg\alpha$  et  $\alpha\wedge\beta$  peut s'exprimer au moyen des connecteurs de  $\Omega$ .

### Explication

- $\{\neg, \wedge\}$  est un SCC donc toute formule de la logique des propositions est exprimable avec  $\{\neg, \wedge\}$
- Et si toute formule de la forme  $\neg\alpha$  et  $\alpha\wedge\beta$  peut s'exprimer au moyen des connecteurs de  $\Omega$
- Par transitivité, toute formule de la logique des propositions est exprimable avec les connecteurs de  $\Omega$ .

# Sémantique

## Systeme Complet de Connecteurs

### Exemple

$\Omega_1 = \{\neg, \vee\}$  est un SCC car :

- $\neg$  appartient à  $\Omega_1$
- $\alpha \wedge \beta$  peut être exprimé à l'aide des connecteurs de  $\Omega_1$  :  
on utilise le théorème de remplacement

$$\begin{aligned}\alpha \wedge \beta &\equiv \neg \neg \alpha \wedge \neg \neg \beta && \text{car } \alpha \equiv \neg \neg \alpha \\ &\equiv \neg \neg (\neg \neg \alpha \wedge \neg \neg \beta) && \text{car } \alpha \equiv \neg \neg \alpha \\ &\equiv \neg (\neg \alpha \vee \neg \beta) && \text{car } \neg (\neg \alpha \wedge \neg \beta) \equiv \alpha \vee \beta\end{aligned}$$

**Exercice** Montrer que  $\{\neg, \rightarrow\}$  est un SCC.

# Sémantique

## Systeme Complet de Connecteurs

### Remarques

1. Aucun ensemble de connecteurs **monaires** ne peut former un système complet de connecteurs. En effet toute formule construite à partir d'un ensemble de connecteurs monaires ne peut intégrer qu'une seule variable propositionnelle.
2. Il existe d'autres systèmes complets de connecteurs tels que  $\{\neg, \vee\}$ ,  $\{\neg, \rightarrow\}$ , etc....
3. Il existe des systèmes complets de connecteurs de plus de deux connecteurs tels que  $\{\neg, \wedge, \vee\}$ , etc....
4. Il existe des systèmes complets de connecteurs dont l'un des connecteurs est d'arité supérieure à 2.

# Sémantique

## Systeme Complet de Connecteurs

5. Il existe deux systèmes complets de connecteurs à un seul connecteur binaire : **les barres de Shaffer**.

Supposons qu'un tel connecteur existe et notons le  $|$ . Pour pouvoir exprimer le connecteur  $\neg$ , la table de vérité de  $|$  est telle que :

- à l'instanciation « V, V » elle doit donner F
- et à l'instanciation « F, F », elle doit donner V.

Ce qui nous donne 4 possibilités pour  $|$  comme illustrée ci-dessous.

A	B	? $\neg$ (A $\wedge$ B)	? $\neg$ B	? $\neg$ A	? $\neg$ (A $\vee$ B)
V	V	F	F	F	F
V	F	<b>V</b>	<b>V</b>	<b>F</b>	<b>F</b>
F	V	<b>V</b>	<b>F</b>	<b>V</b>	<b>F</b>
F	F	V	V	V	V

# Sémantique

## Systeme Complet de Connecteurs

- Les deux colonnes centrales correspondent à  $\neg B$  et  $\neg A$ , donc ne conviennent pas (le connecteur ne peut être monaire)
- Les deux autres colonnes donnent  $\neg(A \wedge B)$  et  $\neg(A \vee B)$ . On peut montrer qu'ils correspondent aux barres de Shaffer.

Notons  $A \uparrow B = \neg(A \wedge B)$  et  $A \downarrow B = \neg(A \vee B)$ .

On a d'une part  $\neg A \equiv \neg(A \wedge A) \equiv A \uparrow A$

et d'autre part

$$A \wedge B \equiv (A \wedge B) \vee (A \wedge B)$$

$$\equiv \neg \neg (A \wedge B) \vee \neg \neg (A \wedge B)$$

$$\equiv \neg (A \uparrow B) \vee \neg (A \uparrow B)$$

$$\equiv \neg ((A \uparrow B) \wedge (A \uparrow B))$$

$$\equiv (A \uparrow B) \uparrow (A \uparrow B)$$

$$\text{car } \alpha \equiv \alpha \vee \alpha$$

$$\text{car } \alpha \equiv \neg \neg \alpha$$

$$\text{car } \neg(\alpha \wedge \beta) \equiv \alpha \uparrow \beta$$

$$\text{car } \neg \alpha \vee \neg \beta \equiv \neg(\alpha \wedge \beta)$$

$$\text{car } \neg(\alpha \wedge \beta) \equiv \alpha \uparrow \beta$$

Donc  $\{\uparrow\}$  est un SCC.

**Exercice** : Montrer que  $\{\downarrow\}$  est un SCC.

# Sémantique

## Formes normales

### Définition (Littéral)

On appelle littéral associé à une variable propositionnelle A chacune des deux expressions A et  $\neg A$  (i.e la variable elle-même et sa négation).

Etant données deux variables propositionnelles A et B, on peut construire **les conjonctions et les disjonctions de littéraux** suivants :

A	B	Conjonctions	disjonctions
V	V	$\phi_1 = A \wedge B$	$\psi_1 = \neg A \vee \neg B$
V	F	$\phi_2 = A \wedge \neg B$	$\psi_2 = \neg A \vee B$
F	V	$\phi_3 = \neg A \wedge B$	$\psi_3 = A \vee \neg B$
F	F	$\phi_4 = \neg A \wedge \neg B$	$\psi_4 = A \vee B$

#### Conjonction

Si B=V, on prend B sinon  $\neg B$

#### Disjonction

Si B=V, on prend  $\neg B$  sinon B

Noter que  $\phi_i = V$  et  $\psi_i = F$  pour une unique et même instanciation ( $i=1, 4$ )

# Sémantique

## Formes normales

### Définition (Forme normale disjonctive )

On appelle **forme normale disjonctive** (FND) toute disjonction de conjonctions de littéraux.

$$D1 \vee D2 \vee \dots \vee Dn$$

où chaque  $D_i$  est une conjonction de littéraux de la forme  $L1 \wedge L2 \wedge \dots \wedge Lp$   
où chaque  $L_k$  est un littéral (une variable propositionnelle ou sa négation)

### Exemples

$\alpha = (A \wedge \neg B) \vee (\neg A \wedge \neg B)$  est une forme normale disjonctive

**Remarque** La forme normale disjonctive permet de trouver facilement les modèles d'une formule.

La formule  $\alpha$  ci-dessus possède deux modèles :

- **A=V et B=F**
- **A=F et B=F**

# Sémantique

## Formes normales

### Définition (Forme normale conjonctive)

On appelle **forme normale conjonctive** (FNC) toute conjonction de disjonctions de littéraux :

$$C_1 \wedge C_2 \wedge \dots \wedge C_n$$

où Chaque  $C_j$  est une disjonction de littéraux, appelé aussi clause, est de la forme  $L_1 \vee L_2 \vee \dots \vee L_p$

où chaque  $L_k$  est un littéral (une variable propositionnelle ou sa négation)

### Exemples

$(\neg A \vee B) \wedge (A \vee B)$  est une forme normale conjonctive

**Remarque** La forme normale conjonctive, dite aussi forme clausale, est nécessaire pour pouvoir appliquer certains algorithmes (méthode de résolution de Robinson) pour déterminer si une formule est valide ou non.

# Sémantique

## Formes normales

### *Théorème*

*Toute formule  $\alpha$  est logiquement équivalente à une forme normale disjonctive (et à une forme normale conjonctive).*

# Sémantique

## Formes normales

**Exemple** Mise en forme normale disjunctive pour la formule  $\alpha = (P \vee Q \rightarrow R) \wedge (P \leftrightarrow R)$

1. Dresser la table de vérité de la formule
2. Construire une disjonction des **conjonctions de littéraux** associées aux **instanciations rendant Vraie  $\alpha$**

	P	Q	R	$P \vee Q$	$P \vee Q \rightarrow R$	$P \leftrightarrow R$	$\alpha$	Conjonctions
1	V	V	V	V	V	V	V	$P \wedge Q \wedge R$
2	V	V	F	V	F	F	F	
3	V	F	V	V	V	V	V	$P \wedge \neg Q \wedge R$
4	V	F	F	V	F	F	F	
5	F	V	V	V	V	F	F	
6	F	V	F	V	F	V	F	
7	F	F	V	F	V	F	F	
8	F	F	F	F	V	V	V	$\neg P \wedge \neg Q \wedge \neg R$

$$\text{FND}(\alpha) = (P \wedge Q \wedge R) \vee (P \wedge \neg Q \wedge R) \vee (\neg P \wedge \neg Q \wedge \neg R)$$

# Sémantique

## Formes normales

**Exemple** Mise en forme normale conjonctive pour la formule  $\alpha = (P \vee Q \rightarrow R) \wedge (P \leftrightarrow R)$

1. Dresser la table de vérité de la formule

2. Construire une conjonction des **disjonctions de littéraux** associées **aux instanciations rendant faux  $\alpha$**

	P	Q	R	$P \vee Q$	$P \vee Q \rightarrow R$	$P \leftrightarrow R$	$\alpha$	Conjonctions
1	V	V	V	V	V	V	V	
2	V	V	F	V	F	F	F	$\neg P \vee \neg Q \vee \neg R$
3	V	F	V	V	V	V	V	
4	V	F	F	V	F	F	F	$\neg P \vee Q \vee R$
5	F	V	V	V	V	F	F	$P \vee \neg Q \vee \neg R$
6	F	V	F	V	F	V	F	$P \vee \neg Q \vee R$
7	F	F	V	F	V	F	F	$P \vee Q \vee \neg R$
8	F	F	F	F	V	V	V	

$$\text{FNC}(\alpha) = (\neg P \vee \neg Q \vee \neg R) \wedge (\neg P \vee Q \vee R) \wedge (P \vee \neg Q \vee \neg R) \wedge (P \vee \neg Q \vee R) \wedge (P \vee Q \vee \neg R)$$

# Relation entre syntaxe et Sémantique

## Consistance :

Tout ce qui est démontrable est vrai

## Complétude :

Toutes les vérités sont démontrables

## Théorème de consistance

Soient  $\alpha_1, \dots, \alpha_n, \alpha$  des formules du langage

$L_p(\neg, \wedge)$ .

$$1. \alpha_1, \dots, \alpha_n \vdash \alpha \Rightarrow \models (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha$$

$$2. \vdash \alpha \Rightarrow \models \alpha$$

# Relation entre syntaxe et Sémantique

## Preuve du théorème de consistance

On appelle **longueur de la déduction** le nombre d'applications de règles de déductions.

Une longueur de déduction peut prendre la valeur minimale zéro pour une déduction du type

$$\alpha_1, \dots, \alpha_n \vdash \alpha_i \quad (i=1, n)$$

En d'autres termes, la conclusion est l'une des prémisses de la déduction.

Le théorème sera démontré par récurrence sur la longueur  $m$  de la déduction  $D$  de  $\alpha$  à partir de  $\{\alpha_1, \dots, \alpha_n\}$ .

# Relation entre syntaxe et Sémantique

## Cas de base ( $m = 0$ )

Comme la déduction est de longueur zéro,

$\alpha$  est l'un des  $\alpha_i$  c'ad il existe  $i$  tel que  $\alpha_i = \alpha$

Montrons que  $\models (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha_i \quad (i=1,n)$ .

Deux cas se présentent :

- $\alpha_i = F \Rightarrow \alpha_1 \wedge \dots \wedge \alpha_n = F \Rightarrow (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha_i = V$
- $\alpha_i = V \Rightarrow$

quelque soit la valeur de vérité de  $\alpha_1 \wedge \dots \wedge \alpha_n$ ,

on a  $(\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha_i = V$

Donc  $\models (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha$

# Relation entre syntaxe et Sémantique

## Cas général

Supposons le résultat est vrai jusqu'à l'ordre  $m$  et montrons qu'il reste vrai à l'ordre  $m+1$ .

Soit  $D$  une déduction de longueur  $m+1$ .

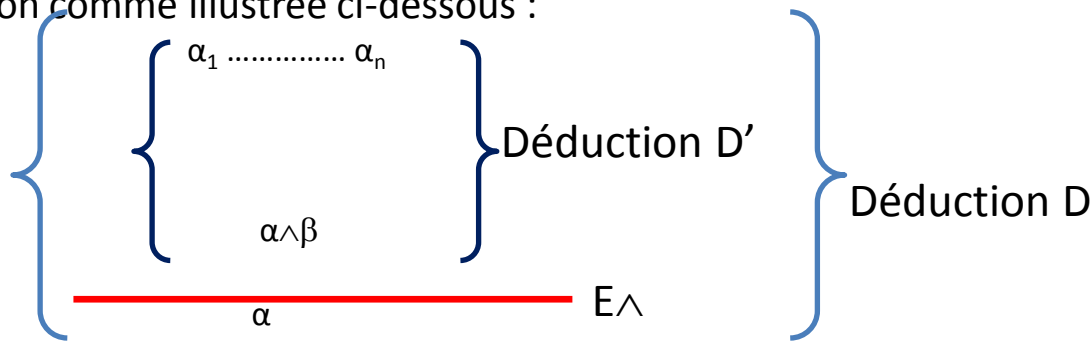
Il s'agira de décomposer la déduction  $D$  en une déduction  $D'$  de longueur  $m$  et la dernière règle utilisée.

La règle de déduction peut être l'une des 4 règles :  $E\wedge$ ,  $I\wedge$ ,  $\neg E$ ,  $\neg I$ .

# Relation entre syntaxe et Sémantique

supposons que la règle est  $E_{\wedge}$

La déduction D peut être décomposée en une déduction D' de longueur m suivi d'une règle d'élimination comme illustrée ci-dessous :



On a..  $\alpha_1, \dots, \alpha_n \vdash \alpha \wedge \beta$       Dédution D'

Donc **par hypothèse de récurrence** sur la déduction **D'** qui est de longueur m, on a alors

$$\models (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha \wedge \beta$$

$\Rightarrow$  Pour chaque instantiation,  $(\alpha_1 \wedge \dots \wedge \alpha_n) = F$  ou  $\alpha \wedge \beta = V$

$\Rightarrow$  Pour chaque instantiation,  $(\alpha_1 \wedge \dots \wedge \alpha_n) = F$  ou  $\alpha = V$

$$\Rightarrow \models (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha$$

Les 3 autres cas sont similaires

# Relation entre syntaxe et Sémantique

## Complétude :

Toutes les vérités sont démontrables

## Théorème de complétude

Soient  $\alpha_1, \dots, \alpha_n, \alpha$  des formules du langage  $L_p(\neg, \wedge)$ .

1.  $\models (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha \implies \alpha_1, \dots, \alpha_n \vdash \alpha$
2.  $\models \alpha \implies \vdash \alpha$

# Relation entre syntaxe et Sémantique

## Lemme 1

$\alpha_1, \dots, \alpha_n \vdash \alpha \iff \{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est  
inconsistant

c-à-d  $\alpha_1, \dots, \alpha_n, \neg\alpha \vdash \perp$

## Rappel

Un ensemble de formules  $\Gamma$  est inconsistant ssi  $\Gamma \vdash \perp$

# Relation entre syntaxe et Sémantique

## Lemme 2

Tout ensemble de formules **consistant** est **satisfiable**

## Remarque

Consistant  $\Rightarrow$  Satisfiable

donc Non satisfiable  $\Rightarrow$  Inconsistant (Contraposée)

# Relation entre syntaxe et Sémantique

## Preuve du théorème de complétude

Hypothèse  $\models (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha$  et on veut montrer  $\alpha_1, \dots, \alpha_n \Vdash \alpha$

### Lemme 1

$\alpha_1, \dots, \alpha_n \Vdash \alpha \Leftrightarrow \{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est **inconsistant**

Au lieu de montrer la déduction  $\alpha_1, \dots, \alpha_n \Vdash \alpha$ ,  
on cherche à montrer que  $\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est **inconsistant**

### Lemme 2 (Contraposée)

Non satisfiable  $\Rightarrow$  Inconsistant

Pour montrer qu'un ensemble de formules est **inconsistant**,  
il suffit de montrer qu'il est **non satisfaisable**

# Relation entre syntaxe et Sémantique

## Preuve du théorème de complétude

Hypothèse  $\models (\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha$

Pour montrer que  $\alpha_1, \dots, \alpha_n \Vdash \alpha$ , on tentera d'établir d'abord que  $\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est non satisfiable.

Nous avons deux cas :

**1<sup>er</sup> cas :  $\alpha_1, \dots, \alpha_n$  sont vraies en même temps**

Donc  $\alpha_1 \wedge \dots \wedge \alpha_n$  est vraie

Or  $(\alpha_1 \wedge \dots \wedge \alpha_n) \rightarrow \alpha$  est vraie par hypothèse

Donc la seule valeur de vérité possible que peut prendre  $\alpha$  est vraie

Donc  $\neg\alpha$  est à faux

Par conséquent  $\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est non satisfiable.

# Relation entre syntaxe et Sémantique

(suite preuve théorème de complétude)

2<sup>ème</sup> cas : l'un des  $\alpha_i$  ( $i=1,n$ ) est faux

Donc  $\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est non satisfiable

**Conclusion**  $\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est non satisfiable

D'après la contraposée du lemme 2, on déduit que:

$\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est **non satisfiable**  $\Rightarrow$   $\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  **est inconsistant.**

## Rappel Contraposée du lemme 2

Non satisfiable  $\Rightarrow$  Inconsistant

D'après le lemme 1, on obtient :

$\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  **est inconsistant**  $\Rightarrow$   $\alpha_1, \dots, \alpha_n \mid\!\!\!-\alpha$

## Rappel du Lemme 1

$\alpha_1, \dots, \alpha_n \mid\!\!\!-\alpha \Leftrightarrow \{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est inconsistant

# Relation entre syntaxe et Sémantique

## Preuve du lemme 1

( $\Rightarrow$ ) Hypothèse  $\alpha_1, \dots, \alpha_n \vdash \alpha$

Montrons que  $\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est inconsistant

# Relation entre syntaxe et Sémantique

(suite preuve lemme 1)

( $\Leftarrow$ ) Hypothèse  $\{\alpha_1, \dots, \alpha_n, \neg\alpha\}$  est inconsistant

Montrons que  $\alpha_1, \dots, \alpha_n \vdash \alpha$

# Relation entre syntaxe et Sémantique

## Algorithme de réfutation

L'algorithme **de réfutation** permet de vérifier

si un ensemble de formules  $\Gamma$  est **satisfiable \inconsistant** :

- Transformer  $\Gamma$  en **plusieurs ensembles de littéraux** en **préservant la propriété de  $\Gamma$  (satisfiabilité \inconsistance)**.
- Dédire la propriété de  $\Gamma$  à partir des ensembles de littéraux obtenus.

Réduire le **problème de satisfiabilité \inconsistance** de  $\Gamma$  en un **problème de satisfiabilité \inconsistance des ensembles de littéraux** dérivés à partir de  $\Gamma$ .

# Algorithme de réfutation

Soit la formule  $\alpha = A \wedge (\neg A \vee \neg B)$

Trouver une instantiation tq  $\alpha = V$

$\alpha = V$

ssi  $A = V$

et  $\neg A \vee \neg B = V$

$\neg A \vee \neg B = V$

ssi  $\neg A = V$  ou  $\neg B = V$

Donc  $\alpha = V$  ssi  $(A = V \text{ et } \neg A = V)$  ou  $(A = V \text{ et } \neg B = V)$

# Algorithme de réfutation

$\alpha = V$  ssi  $(A = V \text{ et } \neg A = V)$  ou  $(A = V \text{ et } \neg B = V)$

$\alpha$  est satisfaisable

ssi  $\{A, \neg A\}$  est satisfaisable

ou  $\{A, \neg B\}$  est satisfaisable

Nous constatons que  $\{A, \neg A\}$  est non satisfiable.

En effet, les **deux littéraux A et  $\neg A$  sont complémentaires** (une variable et sa négation).

Par contre  $\{A, \neg B\}$  est satisfiable,

il suffit de prendre  $A = V$  et  $B = F$ .

Donc  $\alpha$  est satisfiable.

# Algorithme de réfutation

Soit la formule  $\beta = (A \vee B) \wedge (\neg A \wedge \neg B)$

$\beta = V$  ssi  $A \vee B = V$  et  $\neg A \wedge \neg B = V$

ssi  $(A = V$  ou  $B = V)$  et  $(\neg A = V$  et  $\neg B = V)$

ssi  $(A = V$  et  $\neg A = V$  et  $\neg B = V)$

ou

$(B = V$  et  $\neg A = V$  et  $\neg B = V)$

$\beta$  est **satisfiable** ssi

$\{A, \neg A, \neg B\}$  est satisfiable

ou  $\{B, \neg A, \neg B\}$  est satisfiable

Les deux ensembles de littéraux **sont inconsistants**  
car ils contiennent **des littéraux complémentaires**.

Donc  $\beta$  est non satisfiable.

# Algorithme de réfutation

Soit  $\Gamma = \{\alpha_1, \dots, \alpha_n\}$

- **1<sup>er</sup> cas**  $\Gamma$  est composé de littéraux uniquement,

donc  $\Gamma = \{A_1, \dots, A_p, \neg B_1, \dots, \neg B_q\}$

Il existe deux possibilités qui s'excluent :

1.  $\exists i, \exists j, A_i = B_j$

ie que  $\Gamma$  contient une variable propositionnelle et sa négation.

Donc  $\Gamma$  est non satisfiable et  $\Gamma$  est inconsistant

2.  $\forall i, \forall j, A_i \neq B_j$  donc  $\Gamma$  est consistant

et  $\Gamma$  est satisfiable, il suffit de prendre  $A_i = V$  ( $i=1..p$ ) et  $B_j = F$  ( $j=1..q$ )

**Conclusion :** un ensemble de littéraux  $\Gamma$  est inconsistant s'il comporte un couple de littéraux de la forme  $(A, \neg A)$  sinon  $\Gamma$  est satisfiable.

# Relation entre syntaxe et Sémantique

## Algorithme de réfutation

**2<sup>ème</sup> cas**  $\Gamma$  comporte au moins une formule autre qu'un littéral.

On a trois cas possibles : cette formule est

- soit de la forme  $\alpha \wedge \beta$ ,
- soit de la forme  $\neg \neg \alpha$ ,
- **et** soit de la forme  $\neg (\alpha \wedge \beta)$

### 1. Règle R1 :

$\Gamma = \Sigma \cup \{\alpha \wedge \beta\}$  sera remplacé par  $\Gamma' = \Sigma \cup \{\alpha, \beta\}$

**On peut montrer que**

- $\Gamma'$  est satisfiable  $\Rightarrow \Gamma$  est satisfiable
- $\Gamma'$  est inconsistant  $\Rightarrow \Gamma$  est inconsistant

# Relation entre syntaxe et Sémantique

## Algorithme de réfutation

### 2. Règle R2 :

$\Gamma = \Sigma \cup \{\neg\neg\alpha\}$  sera remplacé par  $\Gamma' = \Sigma \cup \{\alpha\}$

On peut montrer aussi que

- $\Gamma'$  est satisfiable  $\Rightarrow \Gamma$  est satisfiable
- $\Gamma'$  est inconsistant  $\Rightarrow \Gamma$  est inconsistant

# Relation entre syntaxe et Sémantique

## Algorithme de réfutation

### 3. Règle R3 :

$\Gamma = \Sigma \cup \{\neg(\alpha \wedge \beta)\}$  sera remplacé par

$$\Gamma_1 = \Sigma \cup \{\neg \alpha\} \text{ et } \Gamma_2 = \Sigma \cup \{\neg \beta\}$$

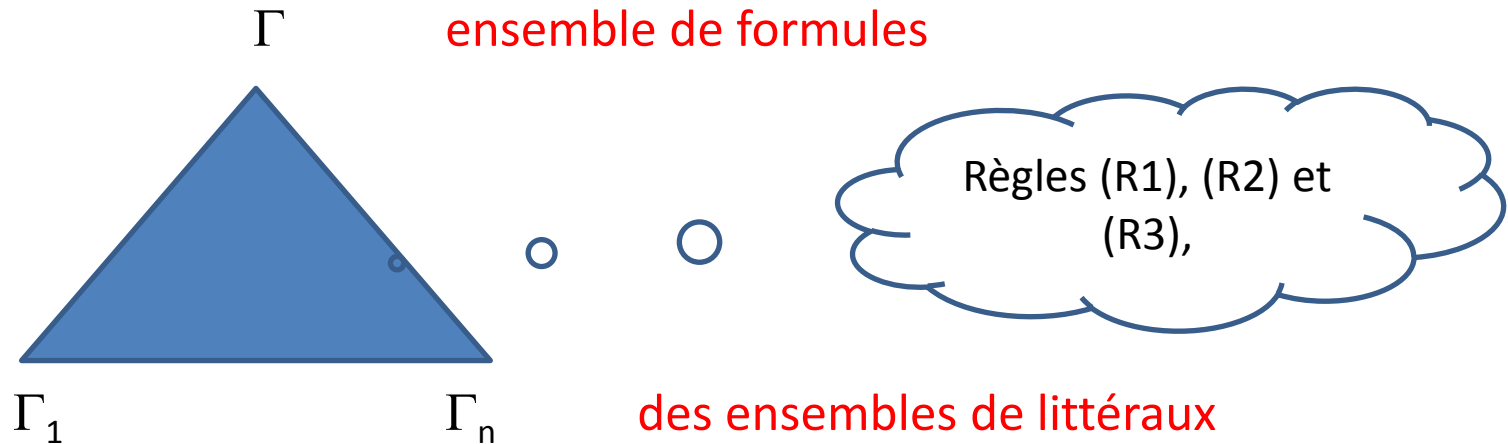
On peut montrer que

- $\Gamma_1$  est satisfiable **ou**  $\Gamma_2$  est satisfiable  
 $\Rightarrow \Gamma$  est satisfiable
- $\Gamma_1$  est inconsistant **et**  $\Gamma_2$  est inconsistant  
 $\Rightarrow \Gamma$  est inconsistant

# Relation entre syntaxe et Sémantique

## Algorithme de réfutation

En appliquant à  $\Gamma$  les règles (R1), (R2) et (R3), tant que cela possible, on obtient un arbre, appelé arbre de réfutation, dont les feuilles sont des ensembles  $\Gamma_i$  de littéraux. On revient alors au 1<sup>er</sup> cas. Ainsi on peut décider pour chaque  $\Gamma_i$  s'il est satisfiable ou inconsistant.



Pour  $\Gamma$ , on a deux possibilités :

- Il existe au moins un  $\Gamma_i$  satisfiable alors  $\Gamma$  est satisfiable
- Tous les  $\Gamma_i$  sont inconsistants alors  $\Gamma$  est inconsistant.

# Relation entre syntaxe et Sémantique

## Algorithme de réfutation

### Exemple

Appliquer l'algorithme de réfutation sur

$$\Gamma = \{\neg C \rightarrow A \wedge B, B \vee \neg C, \neg A \wedge B\}$$

1. Remplacer chaque connecteur d'abréviation par sa définition dans le langage  $L_p(\neg, \wedge)$ .

On obtient  $\Gamma = \{\neg(\neg C \wedge \neg(A \wedge B)), \neg(\neg B \wedge \neg\neg C), \neg A \wedge B\}$

2. A chaque étape, on doit choisir dans  $\Gamma$  une formule qui ne soit pas un littéral puis appliquer la règle appropriée.



# Relation entre syntaxe et Sémantique

## Algorithme de réfutation

### Exemple

Montrer la déduction suivante en utilisant l'algorithme de réfutation :

$(P \vee \neg R) \rightarrow \neg Q \wedge R, P \vdash \neg Q \wedge R$

D'après le lemme 1, la déduction est équivalente à montrer que l'ensemble  $\Gamma = (P \vee \neg R) \rightarrow \neg Q \wedge R, P, \neg(\neg Q \wedge R)$

est inconsistent.

Cela revient à **réfuter la conclusion**, dans un premier temps.

Puis, montrer que l'ensemble  $\Gamma$  est **inconsistent** :

il n'existe aucune instantiation qui rend vraie les hypothèses  
et la conclusion fausse.

Donc, on **ne peut pas réfuter la conclusion**.

L'arbre construit est appelé **arbre de réfutation**.

On remplace les connecteurs d'abréviations par leur définition

$\Gamma = \{ \neg(\neg(\neg P \wedge \neg \neg R) \wedge \neg(\neg Q \wedge R)), P, \neg(\neg Q \wedge R) \}$

# Relation entre syntaxe et Sémantique

## Algorithme de réfutation

$$\Gamma = \{\neg(\neg(\neg P \wedge \neg \neg R) \wedge \neg(\neg Q \wedge R)), P, \neg(\neg Q \wedge R)\}$$

$$\Gamma_1 = \{\neg\neg(\neg P \wedge \neg \neg R), P, \neg(\neg Q \wedge R)\}$$

$$\Gamma_2 = \{\neg\neg(\neg Q \wedge R), P, \neg(\neg Q \wedge R)\}$$

$$\Gamma_3 = \{(\neg P \wedge \neg \neg R), P, \neg(\neg Q \wedge R)\}$$

$$\Gamma_4 = \{\neg P, \neg \neg R, P, \neg(\neg Q \wedge R)\}$$

$$\Gamma_5 = \{\neg P, R, P, \neg(\neg Q \wedge R)\}$$

$$\Gamma_6 = \{\neg P, R, P, \neg \neg Q\}$$

$$\Gamma_7 = \{\neg P, R, P, \neg R\}$$

$$\Gamma_8 = \{\neg P, R, P, Q\}$$

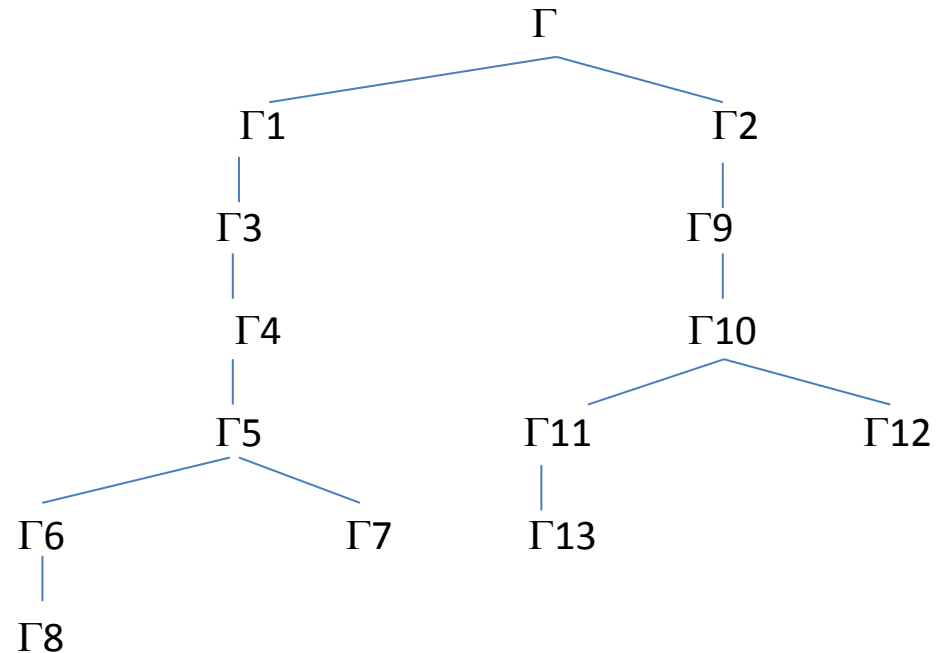
$$\Gamma_9 = \{(\neg Q \wedge R), P, \neg(\neg Q \wedge R)\}$$

$$\Gamma_{10} = \{\neg Q, R, P, \neg(\neg Q \wedge R)\}$$

$$\Gamma_{11} = \{\neg Q, R, P, \neg \neg Q\}$$

$$\Gamma_{12} = \{\neg Q, R, P, \neg R\}$$

$$\Gamma_{13} = \{\neg Q, R, P, Q\}$$



Tous les ensembles de littéraux obtenus  $\Gamma_7$ ,  $\Gamma_8$ ,  $\Gamma_{12}$  et  $\Gamma_{13}$  sont inconsistants donc  $\Gamma$  est inconsistant.

# Relation entre syntaxe et Sémantique

## Exercice

Dans un rapport d'enquête, on trouve les affirmations suivantes :

$\alpha_1$  : « Si Ali dit la vérité alors Ali n'est pas l'assassin »

$\alpha_2$  : « Si Ali dit la vérité alors le crime a eu lieu après minuit et Ali est l'assassin »

$\alpha_3$  : « Ali n'est pas l'assassin ou le crime n'a pas eu lieu après minuit »

Soient les variables propositionnelles suivantes :

A : Ali est l'assassin

B : Ali dit la vérité

C : le crime a eu lieu après minuit

Les affirmations sont représentées par les formules suivantes :

$$\alpha_1 = B \rightarrow \neg A$$

$$\alpha_2 = \neg B \rightarrow C \wedge A$$

$$\alpha_3 = \neg A \vee \neg C$$

# Relation entre syntaxe et Sémantique

On veut répondre à la question :

« Ali est-il l'assassin ? »

1. Tableau de vérité : Vérifier si  $\alpha_1, \alpha_2, \alpha_3 \models A$

Vérifier, pour toute instantiation où

les formules  $\alpha_1, \alpha_2$  et  $\alpha_3$  sont vraies,

que la variable propositionnelle  $A$  est aussi vraie.

Dans le cas positif, on dit que

$\alpha_1, \alpha_2$  et  $\alpha_3$  **impliquent sémantiquement**  $A$ .

# Relation entre syntaxe et Sémantique

A	B	C	$\neg A$	$B \rightarrow \neg A$	$\neg B$	$C \wedge A$	$\neg B \rightarrow C \wedge A$	$\neg C$	$\neg A \vee \neg C$
V	V	V							
V	V	F							
V	F	V							
V	F	F							
F	V	V							
F	V	F							
F	F	V							
F	F	F							

# Relation entre syntaxe et Sémantique

Déduction : Démontrer que  $\alpha_1, \alpha_2, \alpha_3 \vdash \neg A$

On a :

$$\alpha_1 = B \rightarrow \neg A = \neg (B \wedge \neg \neg A)$$

$$\alpha_2 = \neg B \rightarrow C \wedge A = \neg (\neg B \wedge \neg (C \wedge A))$$

$$\alpha_3 = \neg A \vee \neg C = \neg (\neg \neg A \wedge \neg \neg C)$$

Montrons que

$$\neg (B \wedge \neg \neg A), \neg (\neg B \wedge \neg (C \wedge A)), \neg (\neg \neg A \wedge \neg \neg C) \vdash \neg A$$

# Relation entre syntaxe et Sémantique

3. Utiliser l'algorithme de réfutation pour savoir si Ali est l'assassin ou non, s'il dit la vérité ou non et si le crime a eu lieu ou non après minuit.

Soit  $D$  un littéral tel que  $D \in \{A, \neg A, B, \neg B, C, \neg C\}$  (ces littéraux correspondent aux différentes situations possibles)

Appliquer l'algorithme de réfutation sur l'ensemble  $\Gamma = \{\alpha_1, \alpha_2, \alpha_3, \neg D\}$