PLan du cours

Logique formelle

– Introduction
– Logique propositionnelle
– Logique des prédicats

Logique

- Introduction

  - La *logique* s’intéresse à la notion de *raisonnement.* Un raisonnement est un discours qui permet d’établir la *vérité* d’un fait.

  - Dans ce cours, nous nous intéressons à la *logique mathématique* qui s’appuie sur un langage *formel* et un ensemble de *règles* fixé.
  - Le raisonnement ne sera donc pas un discours en langue naturel mais une suite bien précise de constructions qui obéissent aux règles du jeu.

Logique

- Introduction

La logique, chemin vers la vérité

Logique vient du grec logos (raison, *langage*, *raisonnement*).

Enoncé

La logique manipule des énoncés (on parle de manière équivalente de propriété, ou de formule) qui sont des phrases auxquelles on peut attribuer un sens qui est une valeur de vérité, vrai ou faux.

Les phrases suivantes sont des énoncés.

> "Tous les moutons ont 5 pattes"
>
> "Aucun étudiant ne joue sur son téléphone pendant le cours de logique"
>
> "Les trains ne passent pas à un passage à niveau ouvert"

Par contre la phrase "la couleur du cheval blanc d'Henri IV" *n'est pas un énoncé.* En effet cette phrase désigne une couleur et pas une valeur de vérité.

Logique

- Introduction

Raisonnement

La logique ne s’intéresse pas seulement aux énoncés et à la notion de vérité. Elle concerne également la notion de raisonnement. Tout comme en informatique, il peut y avoir plusieurs manières d’arriver à un résultat.

Le raisonnement est un cheminement (une <u>déduction</u>) qui permet à partir d’énoncés que l’on suppose vrais (que l’on appelle des <u>hypothèses</u> ou encore des <u>axiomes</u>), de construire de nouveaux énoncés appelés <u>conséquences</u> ou <u>conclusion</u>. Un raisonnement suit un ensemble de règles logiques données.

Exemple

Tous les hommes sont mortels
or Socrate est un homme
donc Socrate est mortel.

Le raisonnement suivant est par contre beaucoup plus douteux. Il part d’hypothèses possibles pour aboutir à une conclusion qui est fausse.

Logique

- Introduction

Logique et informatique
Il existe des liens très étroits entre logique et informatique dont voici quelques illustrations :

▪ Le calcul booléen a des liens avec les circuits logiques (on peut utiliser des formules booléennes pour représenter des circuits et des circuits pour implanter des fonctions booléennes).
▪ Logique et informatique suivent une démarche analogue: machine pour calculer ou raisonner reposant *sur un nombre limité d'opérations*; liens entre syntaxe (une suite de caractères répondant à des contraintes syntaxiques) et sémantique (le ou les sens possibles attribués à ces phrases).

La logique est un outil de modélisation utilisé dans le domaine des bases de données et du développement de programmes (génie logiciel).

Logique

- Introduction
  - Les notions de logique servent à la création des expressions logiques utilisées pour spécifier par exemple la condition de continuation d'une boucle (structure répétitive) ou la condition d'exécution du bloc d'instructions associé à une structure conditionnelle dans un programme.
  - Le formalisme utilisé s'exprime sous forme de langages logiques, la logique propositionnelle.

Logique propositionnelle

- Logique propositionnelle
  - Cette logique est une logique très simple qui se trouve à la base de presque toutes les logiques qui sont étudiées aujourd'hui.
  - Les éléments de bases sont des propositions (ou variables propositionnelles) qui représentent des énoncés qui peuvent être soit **vrais** ou **faux**.
  - La proposition p: "Marc est un étudiant" ou une proposition q: "Deux est un nombre pair".
  - Des propositions complexes peuvent être construites à partir des propositions en utilisant les connecteurs logiques : $\wedge$ **("et"),** $\vee$ **("ou"), et ¬ (négation).**

Logique propositionnelle

- Logique propositionnelle (exemple de proposition)

  - "C'est nuageux." (Dans une situation donnée.)
  - "Ottawa est la capitale du Canada."
  - "1 + 2 = 3"

<u>Exemples qui ne sont pas des propositions:</u>

- "Quelle heure est-il?" (interrogation, question)
- "OH ! OH! OH!." (sans signification)
- "Fait ce devoir !" (impératif, commande)
- "Roule 4-5 minutes, tourne à gauche..." (vague)
- "1 + 2" (expression sans valeur de vérité)

Logique propositionnelle

- Logique propositionnelle (proposition)

- **<u>Définition :</u>** Une proposition est une assertion (phrase éventuellement) qui ne peut qu'être Vraie ou Fausse.
  - Donc deux valeurs de vérités Vraie (v) et Faux (f).
- **<u>Exemple :</u>**
  - 'La terre est carrée' est une assertion *Fausse.*
  - 'Trois-Rivières est au Québec' est une assertion *Vraie.*

Logique propositionnelle

- Logique propositionnelle (opérateurs)
  - Il y a plusieurs façons de combiner des propositions simples de sorte à produire des propositions plus complexes.
  - Des connecteurs (opérateurs) comme et ($\wedge$), ou ($\vee$), non($\neg$, ~), si … alors (implique, $\Rightarrow$), si et seulement si (équivalent à, $\Leftrightarrow$)
  - L’opérateur **non** opère sur une seule proposition
  - Les autres opérateurs opèrent sur plusieurs propositions (**2 ou plus**)
  - Pour connaître la valeur de vérité d’une proposition nous pouvons utilisé les tables de vérité.

Logique Propositionnelle

- **Les connecteurs logiques**

- L'opérateur de *négation* "¬" (*NOT, NON*) transforme une prop. dans sa forme logique complémentaire (*négation*).

Ex: SI p = "J'ai les cheveux blanc."

ALORS $\neg p$ = "Je n'ai pas les cheveux blanc."

- Table de vérité du NOT/NON:

p	$\neg p$
T	F
F	T
Opérande	Résultat

T :≡ True; F :≡ False

":≡" "est défini comme"

Logique Propositionnelle

- Logique propositionnelle (opérateurs populaires)

  - L'opérateur de *conjonction* "$\wedge$" (*AND, ET*) combine deux propositions pour former leur *conjonction logique*.

$\wedge$ND

Ex: SI p="J'irai à Québec." et q="J'irai au Château Frontenac.", alors $p \wedge q$="J'ira à Québec et au Château Frontenac."

Rappel: "$\wedge$" pointe vers le haut comme un "A", et correspond au "$\wedge$ND"

Logique Propositionnelle

- Logique propositionnelle (opérateurs populaires)

- Notez qu'une conjonction $p_1 \wedge p_2 \wedge \ldots \wedge p_n$ de n propositions aura 2^n rangées dans sa table de vérité.

Opérandes

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

- Aussi: Les opérations ¬ et ∧ sont suffisantes pour déduire n'importe quelles tables de vérité

Logique Propositionnelle

- Logique propositionnelle (opérateurs populaires)

- L’opérateur de *disjonction* “$\vee$” (OR/*OU*) combine deux propositions pour former une *disjonction* logique.

p=“Mon ordinateur a une bonne carte graphique.”

q=“Mon ordinateur a un CPU performant.”

$p \vee q$=“Mon ordinateur a soit une bonne carte graphique, **or (ou)** mon ordinateur a un CPU performant.”

Logique Propositionnelle

- Logique propositionnelle (opérateurs populaires)

  - Notez que $p \vee q$ signifie que p est VRAI, ou q est vrai, **ou les deux** sont vraies!
  - Cette opération est aussi appelée *ou **inclusif**,* et **inclus** la possibilité que p et q soient VRAIES.

p	q	$p \vee q$
V	V	V
V	F	**V**
F	V	**V**
F	F	F

Logique Propositionnelle

- Logique propositionnelle (opérateurs populaires)

  - Posons p = “Il a plue la nuit dernière”,
    q = “Le balai mécanique a lavée la rue cette nuit”,
    r = “La rue est mouillée ce matin.”

  Tranduisez chaque proposition:

  $\neg p$ = “Il n’a pas plue la nuit dernière.”

  $r \wedge \neg p$ = “La rue est mouillée mais il n’a pas plue”

  $\neg r \vee p \vee q$ = “Soit que la rue n’était pas mouillée, ou il a plue la nuit dernière, ou la rue a été lavée cette nuit.”

Logique Propositionnelle

- Logique propositionnelle (opérateurs populaires: **implication**)

- L' *implication* $\overbrace{p}^{\text{Hypothèse}} \rightarrow \overbrace{q}^{\text{Conclusion}}$ signifie que p implique q.

Ex:, posons p = "Vous étudiez beaucoup."
q = "Vous obtenez une bonne note."

$p \rightarrow q$ = "Si vous étudiez beaucoup, vous obtiendrai Alors une bonne note."

Logique Propositionnelle

- Logique propositionnelle (opérateurs populaires: implication)
  - $p \rightarrow q$ est **faux** seulement quand p est VRAI mais que q n'est pas VRAI (**not** true).
  - $p \rightarrow q$ ne veut pas dire que p a causé q
  - $p \rightarrow q$ ne requiert pas que p ou q **soit VRAIE**
  - *EX:* "(1=0) $\rightarrow$ Dumbo l'éléphant vole" est VRAIE

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	**V**
F	F	V

Logique Propositionnelle

- Logique propositionnelle (opérateurs populaires: implication)
  - "Si ce cours ne se termine jamais, Alors le soleil se lèvera demain." *True* or *False*?
  - "Si lundi est un jour de la semaine, Alors Je suis un singe." *True* or *False*?
  - "Si 1+1=6, Alors Trump est président." *True* or *False*?
  - "Si la Suisse est un fromage, Alors je suis plus riche que Warren Buffet." *True* or *False*?

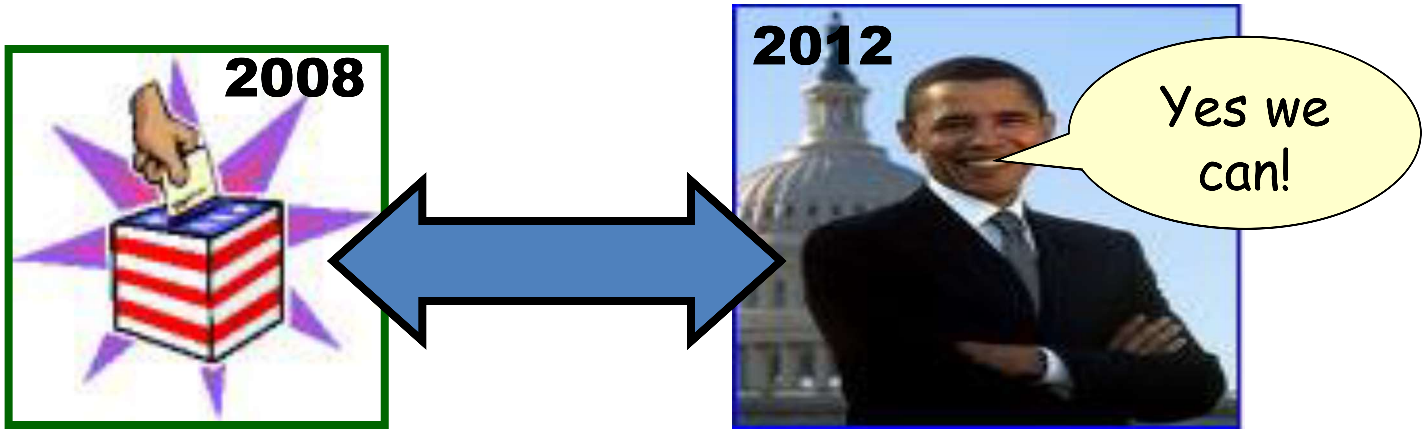
Logique Propositionnelle

- Logique propositionnelle (**biconditionnelle/équivalence** $\leftrightarrow$)
  - Une forme *biconditionelles* $p \leftrightarrow q$ est vraie si la proposition p est vraie et si *la proposition* q est vraie. Nous dirons p SSI q.

p = “Obama gagne les élections de 2008.”

q = “Obama sera président jusqu’en 2012.”

$p \leftrightarrow q$ = “Obama gagne les élections de 2008, SSI, Obama sera président jusqu’en 2012.”



Logique Propositionnelle

- Logique propositionnelle (biconditionnelle $\leftrightarrow$)
  - $p \leftrightarrow q$ signifie que p et q ont la même valeur de vérité.

p	q	$p \leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

  - $p \leftrightarrow q$ ne veut pas dire que p et q sont VRAIES, ou que chacune est la cause de l'autre, ou découlent d'une cause commune.

Logique Propositionnelle

- Logique propositionnelle (précédence des opérateurs)

  - Nous pouvons avoir des énoncés composés
    - $r \lor p \rightarrow q$
  - Quel est l'ordre d'application des opérateurs logiques?
    - Les parenthèses spécifient l'ordre
    - $r \lor (p \rightarrow q)$: Implication en premier
  - Si pas de parenthèses, la précédence des opérateurs intervient

Opérateur	*Précédence*
$\neg$	1
$\land$	2
$\lor$	3
$\rightarrow$	4
$\leftrightarrow$	5

Logique Propositionnelle

- Logique propositionnelle (table de vérité et propositions plus complexes)
  - Par exemple, considérons la proposition $(A \wedge B) \vee (\neg B \wedge C)$ où A, B et C sont des propositions.
  - Cette proposition complexe est construite à partir de propositions moins complexes.

Logique Propositionnelle

- Logique propositionnelle (table de vérité et propositions plus complexes)
  - Table de vérité correspondante:

A	B	C	$(A \wedge B)$	$\neg B$	$(\neg B \wedge C)$	$(A \wedge B) \vee (\neg B \wedge C)$
v	v	v	v	f	f	v
v	v	f	v	f	f	v
v	f	v	f	v	v	v
v	f	f	f	v	f	f
f	v	v	f	f	f	f
f	v	f	f	f	f	f
f	f	v	f	v	v	v
f	f	f	f	v	f	f

Logique Propositionnelle

- Logique propositionnelle (table de vérité et propositions plus complexes)

Remarques

- Le nombre de combinaisons dépend du nombre de propositions de départ.
- Si nous avons 2 propositions nous aurons alors une table de vérité de 4 combinaisons. Si nous avons 3 propositions alors nous aurons 8 combinaisons.
- En fait si le nombre de propositions est n alors le nombre de combinaisons est 2 * 2 * 2 … * 2 … n fois (autrement dit 2^n)

Logique Propositionnelle

- Logique propositionnelle (table de vérité et propositions plus complexes)

Définitions

- Une proposition complexe est construite à partir de propositions initiales au moyen de la conjonction, de la disjonction et de la négation.
- Si x et y sont des propositions alors (x), $\neg$x, (x) $\vee$ (y) et (x) $\wedge$ (y) sont aussi des propositions.

Logique Propositionnelle

- Logique propositionnelle (table de vérité et propositions plus complexes)

Autres remarques

- La proposition $A \wedge B \vee C$ est équivalente à $(A \wedge B) \vee C$
- La proposition $A \wedge B \wedge C$ est équivalente à $((A) \wedge (B)) \wedge C$
- La proposition $\neg A \vee \neg B$ est équivalente à $(\neg A) \vee (\neg B)$

Logique Propositionnelle

- Logique propositionnelle (table de vérité et propositions plus complexes: **Exercices**)

1. Construire les tables de vérités de :

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

2. Réécrire les propositions suivantes avec le moins de parenthèses possibles :

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

3. Dans quel ordre sont les opérations logiques dans cette expression :

$$A \vee \neg B \vee C \wedge \neg A$$

Exemples de formules équivalentes

– Les formules de chaque paire des formules ci-dessous sont équivalentes tautologiquement :

$$(P \Rightarrow Q)\ et\ (\neg P \vee Q)$$

$$((P \wedge Q) \Leftrightarrow P)\ et\ ((P \vee Q) \Leftrightarrow Q)$$

$$((P \vee Q) \Leftrightarrow Q)\ et\ (P \Rightarrow Q)$$

$$(P \Rightarrow (Q \Rightarrow R))\ et\ ((P \wedge Q) \Rightarrow R)$$

$$(P \Rightarrow (Q \vee R))\ et\ ((P \Rightarrow Q) \vee (P \Rightarrow R))$$

$$(P \Rightarrow Q)\ et\ (\neg Q \Rightarrow \neg P)$$

Élimination des connecteurs

- Le remplacement des équivalences permettant d'éliminer des connecteurs:

Élimination de l'implication :	$A \Rightarrow B$	$\neg A \vee B$
Élimination de l'équivalence :	$A \Leftrightarrow B$	$(A \Rightarrow B) \wedge (B \Rightarrow A)$
Élimination de F :	F	$p \wedge \neg p$ (pour un p qlcq)
Élimination de la négation :	$\neg A$	$A \Rightarrow F$
Éliminations de la disjonction :	$A \vee B$ $A \vee B$	$\neg(\neg A \wedge \neg B)$ $(A \Rightarrow F) \Rightarrow B$
Éliminations de la conjonction :	$A \wedge B$ $A \wedge B$	$\neg(\neg A \vee \neg B)$ $(A \Rightarrow (B \Rightarrow F)) \Rightarrow F$

Propriétés des connecteurs

– Les équivalences suivantes traduisent les propriétés algébriques des connecteurs:

Idempotence de $\wedge$ et $\vee$	$A \wedge A \Leftrightarrow A$ $A \vee A \Leftrightarrow A$
Idempotence de $\neg$	$\neg\neg(A) \Leftrightarrow A$
Associativité de de $\wedge$ et $\vee$	$A \wedge (B \wedge C) \Leftrightarrow (A \wedge B) \wedge C$ $A \vee (B \vee C) \Leftrightarrow (A \vee B) \vee C$
Commutativité de de $\wedge$ et $\vee$	$A \wedge B \Leftrightarrow B \wedge A$ $A \vee B \Leftrightarrow B \vee A$
Distributivité (lois de Morgan)	$(A \vee B) \wedge C \Leftrightarrow (A \wedge C) \vee (B \wedge C)$ $(A \wedge B) \vee C \Leftrightarrow (A \vee C) \wedge (B \vee C)$

II. Logique des propositions

Dans la logique de propositions, nous pouvons considérer trois niveaux d'analyse:

- Langage propositionnel (**syntaxique**) : définition des formules bien formées (fbf), i.e. les propositions correctes syntaxiquement;
- Théorie des modèles (**sémantique**) : définition des notions de validité des propositions et de relation de conséquence logique entre propositions;
- Théorie de la preuve (**axiomatique**) : définition des notions de prouvabilité des propositions et de déduction;

But : valide = prouvable

II. Logique des propositions

Langage propositionnel : LP

Définition 1 : l'alphabet de LP

- L'*alphabet* de la logique propositionnelle est constitué de :
  - un ensemble dénombrable de *variables propositionnelles* (ou *formules atomiques*, ou encore atomes) : par exemple, p, q, r, …., il_pleut, la_route_est_mouillée, …
  - Les constantes : *F* (faux, ie: '0' de Boole) *et V* (vrai, ie: '1' de Boole)

    (Rq1: $\neg$ *F* est équivalente à *V*, on peut s'en passer de *V* si on veut)

    (Rq2: *F* est équivalente à (p $\wedge\neg$p) on peut s'en passer de *F* si on veut)
  - les *connecteurs :* $\neg$, $\vee$, $\wedge$, $\Rightarrow$ et $\Leftrightarrow$

    (Rq: on préfère ne pas utiliser $\oplus$: 'ou exclusif ')
  - les séparateurs '(' et ')'.

II. Logique des propositions

Langage syntaxique propositionnel

Définition 2 : *Formules bien formées (fbf)*

- L'ensemble des formules (ou propositions) de la logique propositionnelle est le plus petit ensemble de mots construits sur l'alphabet tel que :
  - si A est une formule atomique alors A est une formule;
  - *F* (faux) est une formule ;
  - $\neg$ A est une formule si A est une formule;
  - $(A \vee B)$, $(A \wedge B)$, $(A \Rightarrow B)$ et $(A \Leftrightarrow B)$ sont des formules si A et B sont des formules.
- **N.B***:* A et B qui désignent ci-dessus des formules sont des ‘metavariables’ car ils ne font pas partie de l'alphabet de LP.
- Si on n’utilise pas des parenthèses, l’ordre de priorité des connecteurs est comme suit : $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, et l’associativité est à gauche pour chaque connecteur.

II. Logique des propositions

Sémantique des propositions

Théorie des modèles :

Définition 1: Interprétation

– Une *interprétation* I (ou valuation) est une application de l'ensemble des variables propositionnelles dans l'ensemble des valeurs de vérité {V,F} (ou {0,1}).

Définition 2: Interprétation des formules

– Une interprétation donnée **I** peut être *étendue* à l'ensemble des formules comme suit (A et B étant des formules) :

- I(*F*) = F (ou 0) et I(*V*) = V (ou 1)
- I(¬A) = **V** si I(A)= F et I(¬A) = F sinon (ou 1 - I(A))
- I(A $\wedge$ B) = **V** si I(A)= V et I(B) =V et I(A $\wedge$ B) = **F** sinon (ou min(I(A),I(B)))
- I(A v B) = **V** si I(A)= V ou I(B) =V et I(A $\wedge$ B) = **F** sinon (ou max(I(A),I(B)))
- I(A $\Rightarrow$ B) = **V** si I(A) = F (ou 0) ou I(B) =V (ou 1) et I(A $\Rightarrow$ B) = **F** sinon.

II. Logique des propositions

Sémantique des propositions

Théorie des modèles (suite):

Définition 3 : Modèle

- I est un modèle pour une formule A (ou I satisfait A) ssi I(A) = V .
- I est un modèle pour un ensemble de formules S ssi I est un modèle pour toute formule A de S.

→ *Exemples*

Définition 4 : Validité, Satisfaisabilité

– Soit A une formule:

- A est valide (ou tautologique) si I(A) = V pour toute interprétation I. Sinon A est invalide ou falsifiable.
- A est satisfaisable ssi il existe une interprétation I t.q. I(A) = V. Sinon A est non satisfaisable ou contradictoire.

II. Logique des propositions

Théorie des modèles (suite):

Définition 5 : consistance.

– Soit S un ensemble de formules.

- S est ***inconsistant*** s'il n'existe aucun modèle pour S, autrement dit, un modèle pour lequel toutes les formules de S ont simultanément la valeur vrai. Si un tel modèle existe S est dit ***consistant*** ou satisfaisable.

Définition 6 : Conséquence logique.

– Une formule A est *conséquence logique* de n formules $A_1, \ldots, A_n$, noté $\{A_1, \ldots, A_n\} \models A$, ssi tout modèle de $\{A_1, \ldots, A_n\}$ est un modèle de A.

Définition 7 : Équivalence tautologique → *Exemples*

– Soit A1 et A2 deux formules, elles sont dites tautologiquement équivalentes si $A_1 \models A_2$ et $A_2 \models A_1$.

II. Logique des propositions

Théorie des modèles (suite):

Théorème 1 : Conséquence logique et implication

– Soient A et B deux formules et S un ensemble de formules contenant A :

- $A \models B$ ssi $\models (A \Rightarrow B)$.
- $S \models B$ ssi $S / \{A\} \models (A \Rightarrow B)$.

II. Logique des propositions

Système de preuve en logique propositionnelle:

Introduction

- L’intérêt pratique de la théorie de la preuve est qu’elle répond à l’un des buts recherchés en intelligence artificielle, qui est la démonstration automatique de théorèmes (formules valides).
- Dans la théorie de la preuve, il existe différentes méthodes (appelées aussi systèmes formels) qui permettent de **prouver** la validité –ou seulement la satisfiabilité- d’une formule propositionnelle.
- Ces méthodes peuvent être utilisées pour **déduire** une proposition à partir d’un certain nombre de propositions (hypothèses).

II. Logique des propositions

Système de preuve en logique propositionnelle:

Introduction

- **Méthode des tables de vérité**
- Méthode des tableaux sémantiques
- **Méthode de résolution**

II. Logique des propositions

Théorie de la preuve :

Méthode des Tables de vérité

Définition 5 :

- **Elle permet de donner les valeurs de vérité possibles d'une formule composée F pour chaque combinaison possible des valeurs de vérité des propositions atomiques qui sont sous-formules de F.**

Construction :

- **On commence par définir, pour chaque connecteur *c*, une table qui associe une valeur de vérité à une formule de type 'A *c* B' pour chaque attribution de valeurs de vérité à ces sous-formules (A et B dans ce cas). C'est ce qu'on appelle les *tables de vérité des connecteurs* *tvc*.**
- **Étant donné une formule quelconque, il est alors possible de construire sa table de vérité, en calculant sa valeur de vérité pour chaque combinaison possible des valeurs de vérité de ces variables propositionnelles moyennant les tables de vérité des connecteurs.**

II. Logique des propositions

Systeme de preuve : Méthode des Tables de vérité (suite)

Validité d’une formule :

- **Chaque ligne d'une table de vérité d’une formule A correspond à une combinaison possible des valeurs de vérité de toutes les variables propositionnelles apparaissant dans A. Ceci peut être vue comme la description d’une interprétation de l'ensemble des n variables propositionnelles de la formule dans l'ensemble des valeurs de vérité {F, V}.**
- **L’ensemble des lignes (2^n) de la table représente, donc, toutes les interprétations possibles de l'ensemble des variables propositionnelles de la formule dans l'ensemble des valeurs de vérité {F, V}.**
- **Si pour chaque ligne la valeur de la formule A est V alors, chaque interprétation -de l'ensemble des variables propositionnelles de la formule dans l'ensemble des valeurs de vérité {F, V} - est un modèle pour A, et donc A est valide.**

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution

Formes normales et clausales :

Définition 6 : littéral, forme normale disjonctive

- **Un *littéral* est une formule atomique ou la négation d'une formule atomique (exps : p et ¬ q avec p et q atomes).**
- **Une formule est une *Forme Normale Disjonctive* (FND) si elle est une disjonction de conjonctions de littéraux.**

Définition 7: clause, forme normale conjonctive

- **Une *clause* est une disjonction de n ($n \geq 0$) littéraux (exp: $\neg p_1 \vee p_2 \vee p_3$).**
- **Une formule est une *forme normale conjonctive* (FNC) si elle est une conjonction de clauses .**

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Théorème 6 : Équivalence et formes normales

☞ **Toute formule propositionnelle est équivalente à une forme normale conjonctive et à une forme normale disjonctive**

Algorithme de mise en forme normale conjonctive

Entrée : une formule A1 **Sortie : une fnc A'1 équivalente à A1**

Début Eliminer $\Leftrightarrow$ et $\Rightarrow$ et *F* (si il est utilisé);

Appliquer à A1 les remplacements suivants autant de fois que nécessaire:

$$\neg\neg A \;\text{-->}\; A$$
$$\neg(A \vee B) \;\text{-->}\; \neg A \wedge \neg B\ \neg$$
$$(A \wedge B) \;\text{-->}\; \neg A \vee \neg B$$
$$A \vee (B \wedge C) \;\text{-->}\; (A \vee B) \wedge (A \vee C)$$

A et B des sous formules de A1

Fin

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Théorème 7 : Validité d’une clause

- **Une formule en fnc est valide ssi toutes ses clauses sont valides.**
- **Une clause est valide ssi elle contient deux littéraux opposés, c-à-d , de la forme $L_1 \vee \ldots \vee p \vee \ldots \vee \neg p \vee \ldots \vee L_n$.**

Définition 8 : Notation ensembliste/ Forme clausale

- **Une clause $C = L_1 \vee \ldots \vee L_i \vee \ldots \vee L_n$ peut s’écrire comme étant l’ensemble de ces littéraux : $C = \{L_1 , \ldots , L_i , \ldots , L_n \}$.**
- **Une forme normale conjonctive $A= C_1 \wedge \ldots \wedge C_i \wedge \ldots \wedge C_n$ est écrite en *forme clausale* sous la forme d’un ensemble de clauses $A = \{C_1 , \ldots , C_i , \ldots , C_n \}$ ou d’une suite de clauses $C_1 , \ldots , C_i , \ldots , C_n$.**
  - **A est donc valide ssi toutes les clauses qu’elle contient sont valides.**

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Conséquence 1: clause vide

- ☞ **La clause vide est équivalente à F, elle est contradictoire.**
- ☞ **En effet, elle ne contient pas deux littéraux opposés vu qu'elle n'en contient pas du tout.**

Conséquence 2 : ensemble vide de clauses

- ☞ **L'ensemble vide de clauses est équivalent à $\neg F$, il est donc valide.**
- ☞ **En effet, on peut dire que toutes ses clauses sont valides puisque il n'en contient pas du tout.**

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Théorème 8 : Principe de Résolution

- **Soient C1 et C2 deux clauses :**

  Si il existe un littéral L_1 tel que $L_1 \in C1$ et son opposé $L_2 \in C2$ (ie l'équivalent de $\neg L_1$) alors, la clause $C3 = (C1 - L_1) \cup (C2 - L_2)$ est une conséquence logique de C1 et C2 : $\{C1, C2\} \models C3$.

- **C3 est dite un résolvant de C1 et C2.**

Rappel : Règle de résolution

$$\frac{\neg X \vee A \qquad X \vee B}{A \vee B} \qquad\qquad \frac{\{\neg X, A\} \qquad \{X, B\}}{\{A, B\}}$$

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Conséquence

- **Les clauses C1={ p } et C2= {¬p } ont pour résolvant la clause vide { }.**
- **Or, une formule A= p ∧ ¬p est équivalente à l'ensemble de clauses S={C1,C2} qui est tel que S |= { } d'après le théorème 8.**
- **Or A est équivalente à *F* (toujours fausse) ou encore S inconsistant.**
- **D'où le théorème qui suit :**

Théorème 9 : Ensemble de clauses inconsistant

- **Soit S un ensemble de clauses, S est inconsistant ssi S |= { } .**

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Théorème 10 : Réfutation

- **Soit une formule A et S un ensemble de formules, $S \models A$ est équivalent à $S \cup \{\neg A\}$ est inconsistant (i.e. $S \cup \{\neg A\} \models F$) .**

Preuve par réfutation

- **Soit S un ensemble de clauses et G une clause à montrer comme conséquence logique de S ($S \models G$) ;**
- **On considère $S \cup \{\neg G\}$ et on montre qu'il est inconsistant, c-à-d, $S \cup \{\neg G\} \models \{\ \}$, d'après le théorème précédent ceci est équivalent à $S \models G$.**

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Algorithme de résolution par réfutation

Soit $S= \{C1, C2, \ldots, Cn\}$ un ensemble fini de clauses :

Soit G une clause à prouver:

1/ Remplacer S par $S \cup \{\neg G\}$ ($S \leftarrow S \cup \{\neg G\}$) .

2/ Si on peut choisir une nouvelle paire de clauses C1 et C2 dans S telqu'il existe p atome avec $p \in C1$ et $\neg p \in C2$ alors calculer le résolvant R de C1 et C2.

a/ Si $R = \{\ \}$, la preuve est faite.

b/ Sinon, $S \leftarrow S \cup \{R\}$ et retourner à l'étape 2

4/ Sinon (plus de choix possibles) alors S est consistant.

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Exemple de résolution par réfutation

$$\boxed{\begin{array}{l} H1 : P \Rightarrow Q \\ H2 : Q \Rightarrow R \\ B\ : P \Rightarrow R? \end{array}}$$

➔

$H1 : \neg P \vee Q \rightarrow C1 : \{\neg P , Q\}$

$H2 : \neg Q \vee R \rightarrow C2 : \{\neg Q , R\}$

$\neg B : \neg(\neg P \vee R) \rightarrow P \wedge \neg R$

$\left.\begin{array}{l} \rightarrow C3 = \{P\} \\ \rightarrow C4 = \{\neg R\} \end{array}\right\}$ 2 clauses

$C1 : \{\neg P , Q\}$

$C2 : \{\neg Q , R\}$

$R1 : \{\neg P, R\}$

$C3 : \{P\}$

$\left.\begin{array}{l} R2 : \{R\} \\ C4 : \{\neg R\} \end{array}\right\} \rightarrow R3 : \{\ \}$

C.Q.F.D

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (suite)

Théorème 11 : Conséquence logique et clauses

☞ **Soient C1 et C2 deux clauses (en notation ensembliste),**

- **Si C1 $\subset$ C2 alors C1 $\models$ C2.**

Théorème 12 : Élimination des clauses valides

☞ **Soit S un ensemble de clauses et C une clause valide ,**

$S \cup \{C\}$ est valide (resp. satisfaisable) ssi S est valide (resp. satisfaisable) .

Remarque :

☞ **Ce théorème stipule que pour montrer la validité (resp. satisfaisabilité) d'un ensemble de clauses, on peut toujours ignorer les clauses tautologiques.**

II. Logique des propositions

Théorie de la preuve : Méthode de Résolution (fin)

Théorème 13: Validité d'une preuve par résolution

☞ **Soit S un ensemble de clauses et C une clause ,**

- **Si $S \vdash^{res} C$ alors $S \models C$.**

Remarque :

☞ **La preuve par réfutation d'une clause C à partir d'un ensemble de clauses S est une déduction par résolution de la clause vide à partir de l'ensemble $S \cup \{\neg C\}$ qu'on appelle, plus brièvement, réfutation.**

Théorème 13: Complétude de la résolution par réfutation

☞ **Soit S un ensemble de clauses et C une clause ,**

- **Si $S \cup\{\neg C\} \models \{\ \}$, alors $S \cup\{\neg C\} \vdash^{res} \{\ \}$.**

Chap 2. Logique des prédicats

Sommaire

III. Logique des prédicats

Introduction

☞ Considérons l'exemple de raisonnement qui suit :

> TOUT être humain est mortel.
> OR Socrate est un être humain.
> DONC Socrate est mortel.

☞ La logique des propositions ne permet pas de 'formuler' des raisonnements comme celui-ci, vu qu'il n'existe pas de notion *d'énoncé singulier* (exp. Socrate est mortel) et *d'énoncé générique* (Tout être humain est mortel).

☞ En logique de prédicats, nous avons cette notion d'énoncé singulier et d'énoncé générique qui se base sur d'autres notions, à savoir, les objets ou les individus et leurs désignateurs : les constantes et les variables (cf. *Lois de la pensée*) , les propriétés d'objets (exp. *Mortalité*) et les relations entre les objets (exp. Ahmed *est le père* d'Ali) appelées prédicats.

III. Logique des prédicats

Introduction (fin)

- En logique de prédicats, nous pouvons écrire (ou formuler) ‘Socrate est un être humain’ sous la forme : être-humain(Socrate) où ‘Socrate’ est l‘individu ou l’objet, et ‘être-humain’ est la propriété.

- Pour formuler un énoncé générique comme ‘TOUT être humain est mortel’, la logique des prédicats emploie le quantificateur universel $\forall$.

- Le raisonnement précédent peut être formulé comme suit :

  $\forall x$ être-humain(x) $\Rightarrow$ mortel(x)
  être-humain(Socrate)
  DONC mortel(Socrate)

- La logique de prédicats a pour but de généraliser la logique propositionnelle en introduisant les variables (exp. x), les quantificateurs (exp. . $\forall$) et les prédicats (exp. mortel), c’est pour cela qu’on la nomme aussi logique de 1er ordre.

III. Logique des prédicats

Langage des prédicats : LP1

Définition 1 : l'alphabet de LP1

– L'*alphabet* de la logique des prédicats est constitué de :

- un ensemble dénombrable R de symboles de *prédicats* à n arguments ($0 \leq n$), par exemple, *p, q, r, ….,mortel, aime*, …
- un ensemble dénombrable VAR de *variables* d'objets ou d'individus, par exemple, `x,y,z,...`
- un ensemble dénombrable F de symboles de *fonctions* à à n arguments ($0 \leq n$), par exemple, `f, g, ... , âge-de, ...`
- les quantificateurs $\forall$, $\exists$
- les *connecteurs :* $\neg$, $\vee$, $\wedge$, $\Rightarrow$ et $\Leftrightarrow$
- les séparateurs '(' et ')'.
- **Rq:** les constantes : *F* (faux,) *et V* (vrai) sont généralement non considérées.

III. Logique des prédicats

Langage des prédicats : LP1

Définition 2 : Fonctions et prédicats à 0 arguments

– Les **constantes** de l'*alphabet* de la logique des prédicats sont les fonctions à **0** arguments et elles permettent de désigner des objets et des individus particuliers.

– Dans la suite, elles seront notées en minuscule, par exemple, `a,b,socrate, ali,...`

– Les prédicats à 0 arguments peuvent être considérés comme des variables propositionnelles, par exemple, il-pleut, la-route-est-mouillée, p, q, …

– **Rq:** On peut donc considérer l'alphabet propositionnel comme faisant partie de l'alphabet de la logique des prédicats.

III. Logique des prédicats

Langage des prédicats : LP1 (suite)

Définition 3 : *Termes*

– L'ensemble des termes de LP1 est le plus petit ensemble de mots construits sur l'alphabet de la logique des prédicats tel que :

- Toute variable est un terme ;
- Toute constante est un terme ;
- $f(t_1,\ldots,t_n)$ est un terme(dit fonctionnel) si f est une fonction à n arguments ($n>0$) et $t_1,\ldots,t_n$ sont des termes.

Définition 4 : *Formule atomique*

– Si p est un prédicat à n (resp 0) arguments ($n>0$) et $t_1,\ldots,t_n$ sont des termes alors $p(t_1,\ldots,t_n)$ (resp p) est une formule atomique.

III. Logique des prédicats

Langage des prédicats : LP1 (suite)

Définition 5 : *Formules bien formées (fbf ou wff)*

- L'ensemble des formules de LP1 (appelé FOR) est le plus petit ensemble de mots construits sur l'alphabet de LP1 tel que :
  - si A est une formule atomique alors A est une formule;
  - $\neg$ A est une formule si A est une formule;
  - $(A \vee B)$, $(A \wedge B)$, $(A \Rightarrow B)$ et $(A \Leftrightarrow B)$ sont des formules si A et B le sont.
  - (**Q x** A) est une formule si Q est un quantificateur, x une variable et A une formule.
- **Rq** :L'ensemble FOR est défini de la même manière que celui de la logique propositionnelle, mais en considérant en plus les quantificateurs et les variables.
- **N.B** : **tout ensemble L1 définie sur un alphabet restreint de LP1(donc, R1 $\subset$ R et F1$\subset$ F) et respectant les règles concernant les fbf est aussi un langage de prédicat avec FOR1 $\subset$ FOR**.

III. Logique des prédicats

Langage des prédicats : LP1 (suite)

Définition 6 : *Sous-formules*

– La même définition que dans la logique des propositions, avec en plus :
  - Si (Q x B) est une sous-formule de A (avec Q quantificateur et x variable) alors B est une sous-formule de A.

Définition 7 : *Occurrence d'une variable*

– Une occurrence d'une variable x dans une formule A est un endroit où x apparaît dans A sans être immédiatement précédée par ∀ ou ∃.

Définition 8 : *Portée d'un quantificateur*

– Dans une formule A=Q x B, avec Q quantificateur et x variable , B est appelée la portée du quantificateur Q.

III. Logique des prédicats

Langage des prédicats : LP1 (suite)

Définition 7 : ***occurrence libre (resp. liée) d'une variable***

- Une **occurrence libre** de x dans une formule A est définie comme suit *:*
  - Si A = $\neg$ (B), les occurrences libres de x dans A sont celles de B.
  - Si A = (B) Δ (C), les occurrences libres de x sont celles de B et celles de C (Δ est un connecteur parmi $\wedge$, $\vee$,$\Rightarrow$, $\Leftrightarrow$).
  - Si A= $\forall$y(B) ou A = $\exists$y(B) avec x distincte de y, les occurrences libres de x sont celles de B.
  - Si A = $\forall$ x (B) ou A = $\exists$ x (B), aucune occurrence de x dans A n'est libre.
  - Si A est un atome alors toute occurrence d'une variable x dans A est libre.
- Une **occurrence liée** de x dans A est toute occurrence non libre .

III. Logique des prédicats

Langage des prédicats : LP1 (suite)

Définition 8 : *variable libre*

– Une variable est dite **libre** dans une formule A si elle a au moins **une** occurrence libre, sinon on dit qu'elle est liée.

–RQ : si $\mathbf{x_1, \ldots , x_n}$ sont les variables libres dans A, pour exprimer ceci, on note A comme suit : $A(\mathbf{x_1, \ldots , x_n})$

Définition 9 : *formules closes*

– Une formule est dite **close** (ou fermée) si elle ne contient pas de variables libres. Sinon elle est dite ouverte.

Définition 10 : *clôture universelle d'une formule*

- La clôture (ou fermeture) universelle d'une formule $A(\mathbf{x_1, \ldots , x_n})$ est la formule **close** $A' = \forall \mathbf{x_1} \ldots \forall \mathbf{x_n}\, A(\mathbf{x_1, \ldots , x_n})$.

III. Logique des prédicats

Langage des prédicats : LP1 (suite)

Définition 11 : *clôture existentielle d'une formule*

- La clôture (ou fermeture) existentielle d'une formule $A(\mathbf{x_1}, \dots, \mathbf{x_n})$ est la formule close $\exists \mathbf{x_1} \dots \exists \mathbf{x_n}\, A(\mathbf{x_1}, \dots, \mathbf{x_n})$.

Définition 12 : *formules propres*

- Une formule A est dite **propre** lorsque :
  - Il n'existe pas de variable dans A qui a, à la fois, des occurrences libres et des occurrences liées.
  - Deux occurrences liées d'une même variable dans A appartiennent à la portée d'un même quantificateur.

Définition 13 : *Renommage* de *formules*

- Un renommage d'une variable consiste à changer les noms de certaines de ces occurrences et ce, en donnant le même nom pour ces occurrences liées appartenant à la même portée d'un quantificateur et le même nom pour ces occurrences libres.

III. Logique des prédicats

Langage des prédicats : LP1 (suite)

Définition 13 : *formules impropres et renommage*

- Une formule A est dite impropre si elle n'est pas propre.
- On peut rendre une formule propre par un **renommage** qui consiste à :
  - Changer les occurrences liées d'une variable libre par d'autres noms de variables, de telle sorte que toute variable libre ne puisse pas avoir d'occurrences liées.
  - Pour chaque occurrence liée d'une variable qui appartient à la portée d'un quantificateur différent donner un nom différent.
- **Exemples** :
  - A = $\forall$**x ($\exists$y p(x, y)** $\vee$ **$\exists$z r(y, z))** $\rightarrow$ impropre
  - A'= $\forall$**x ($\exists$y1 p(x, y1)** $\vee$ **$\exists$z r(y, z))** $\rightarrow$ propre.

III. Logique des prédicats

Langage des prédicats : LP1 (suite)

Définition 14 : Substitution de variables

- **Une substitution s de n variables $\{x_1 , \dots , x_n\}$ est une application de l'ensemble des variables dans l'ensemble des termes qui est telle que:**
  - **Pour tout x_i dans $\{x_1 , \dots , x_n\}$, $s(x_i) = t_i$**
  - **Pour tout x / $x \notin \{x_1 , \dots , x_n\}$, $s(x) = x$.**
  - **s est notée par :$\{x_1\backslash t_1 , \dots , x_n\backslash t_n\}$ (parfois, $(t1 /x1 ,\dots, tn /xn)$).**
- **L'application d'une substitution s à une formule (resp. un terme) E donne une formule (resp. un terme) E_s qui est le résultat du remplacement simultanée de toutes les occurrences libres des variables dans E par leur terme associé.**
  - **E_s est appelé une instance de E**
- **Si E_s ne contient pas de variables on dit que s *instancie* E.**

III. Logique des prédicats

Langage des prédicats : LP1 (fin)

Remarques 1 : Substitution et renommage

– Le résultat A_s de l'application d'une substitution $s = \{x_1 \backslash t_1 , \ldots , x_n \backslash t_n\}$ à une formule propre A n'est pas forcément une formule propre, à moins de respecter la règle suivante:

  - les variables apparaissant dans les termes t_i ne sont pas des variables liées de A.
  - On dit que **s est définie pour A** si elle respecte cette règle.

– Lorsque une substitution ne respecte pas cette règle pour une formule A on peut avoir recours au renommage,

– Si $s = \{x\backslash t\}$ est définie pour A, on dit aussi que t est substituable à x ou libre pour x.

III. Logique des prédicats

Théorie des modèles :

Définition 1: Domaine d'interprétation et conceptualisation

– Un domaine d'interprétation pour un langage de prédicat est un ensemble non vide D (appelé parfois domaine d'individus ou univers de discours) tel que il existe une application de son ensemble de constantes dans D.

Définition 2: Conceptualisation

– Une conceptualisation C est un triplet (D, Fc, Rc), telle que :

- D est un domaine d'interprétation ;
- Fc est un ensemble de fonctions f tq si n est l'arité de f;alors f est de $\mathbf{D^n}$ vers D,
- Rc est un ensemble de relations r tq si n est l'arité de r, alors r est inclue dans $\mathbf{D^n}$ (r un sous ensemble de $\mathbf{D^n}$),

III. Logique des prédicats

Théorie des modèles (suite):

Définition 3 : Interprétation

– Étant donné un langage des prédicats L1, une interprétation **I** de L1 est la donnée d'une conceptualisation C=(D,Fc,Rc) et d'une application –notée aussi **I -** qui associe à chaque élément de l'alphabet du langage L1(sauf les connecteurs, quantificateurs et les parenthèses) un élément des ensembles de la conceptualisation sous les conditions (1), (2) et (3) :

- $\mathbf{I} : L1 \rightarrow C$

  $\phi \rightarrow \mathbf{I}(\phi)$

- (1) : Si ϕ est une constante alors $\mathbf{I}(\phi) \in D$;
- (2) : Si Φ est un symbole de fonction d'arité n de F alors $\mathbf{I}(\Phi) \in Fc$ et elle est d'arité n;
- (3) : Si Φ est un un symbole de prédicat d'arité n de R alors $\mathbf{I}(\Phi) \in Rc$ et elle est d'arité n.

III. Logique des prédicats

Théorie des modèles (suite):

Définition 4: Assignation de variables, assignation de termes

– **Une assignation de variables est une fonction U qui associe à chaque variable d'un langage de prédicat L un élément de D.**

- $$U : L \rightarrow D$$
  $$v \rightarrow U(v)$$

– **Une fonction T_{IU} est dite une 'assignation de terme' correspondante à une interprétation I et à une assignation de variables U, si elle associe à un terme T du langage L un élément appartenant à D de la manière suivante :**

- **Si T est une constante alors $T_{IU}(T)=I(T)$;**
- **Si T est une variable alors $T_{IU}(T)=U(T)$;**
- **Si T est un terme fonctionnel de la forme $\Phi(T1, \ldots, Tn)$ tel que $I(\Phi) = f$ et $T_{IU}(Ti) = xi$ pour tout $0<i \leq n$, alors $T_{IU}(T) = f(x1,\ldots,xn)$.**

III. Logique des prédicats

Théorie des modèles (suite) :

Définition 5: variante d'une assignation de variable

– Une assignation de variables U' est dite une variante en x (variable) de U , si U' est identique à U sauf -peut être- en x.

Définition 6: Interprétation des formules

– Étant données une interprétation **I** et une assignation de variable **U ,** on peut calculer la valeur de vérité (dans {V, F}) de chaque formule A d'un langage de prédicat par une fonction $\mathbf{I}_{U}$, comme suit :

- Si A est atomique, càd, il existe un symbole de prédicat p tel que $A=p(t_1,\ldots,t_n)$, alors $\mathbf{I}_{U}(A)= V$ **ssi** $(T_{IU}(t_1),\ldots, T_{IU}(t_n)) \in \mathbf{I}(p)$;
- Si $A = \forall x\ B$ alors $\mathbf{I}_{U}(A)= V$ **ssi** pour tout U' variante de U en x , $\mathbf{I}_{U'}(B)= V$.
- Si $A = \exists x\ B$ alors $\mathbf{I}_{U}(A)= V$ **ssi** il existe U' variante de U en x tq $\mathbf{I}_{U'}(B)= V$.
- Plus les règles concernant les connecteurs propositionnels.

III. Logique des prédicats

Théorie des modèles (suite):

Définition 7 : Modèle

- I est un modèle pour une formule A (ou I satisfait A) ssi pour toute assignation de variable U, on a $\mathbf{I_U}(A) = V$, noté $|=_I A$ (parfois noté, I(A)= V)
- I est un modèle pour un ensemble de formules S **ssi** I est un modèle pour toute formule A de S.

Définition 8 : Validité, Satisfaisabilité

- Soit A une formule:
  - A est valide (ou tautologique ; noté $|= A$) si $|=_I A$ (ou I(A) = V) pour toute interprétation I. Sinon A est invalide ou falsifiable.
  - A est satisfaisable ssi il existe une interprétation I et une assignation de variable U, t.q. $\mathbf{I_U}(A) = V$ (ont dit que I satisfait A pour l'assignation de variable U ou (I,U) satisfait A). Sinon A est contradictoire.

III. Logique des prédicats

Théorie des modèles (suite):

Propriété 1 : Modèle d’une formule clause

- Si I satisfait une formule close A pour une assignation de variable U, càd, $\mathbf{I}_U(A)=V$, alors I est un modèle pour A : $|=_I A$.
- En effet, pour une formule close, les valeurs assignées à ses variables n’ont aucune influence sur sa valeur de vérité par une interprétation du moment qu’elles sont toutes liées.

Théorème 1 : Satisfaction d’une formule et clôture universelle

- Si une interprétation I satisfait la clôture universelle d’une formule A alors elle satisfait A.
- Si la clôture universelle d’une formule A est valide alors A est valide.

III. Logique des prédicats

Théorie des modèles (suite):

Définition 9 : Conséquence logique, consistance et équivalence.

- **Une formule B est conséquence logique d'une formule A si tout couple (I,U) satisfaisant A, satisfait également B (noté A |= B)**

- **Pour les autres, appliquer les mêmes définitions et notations que celles de la logique propositionnelle en respectant évidemment la nouvelle définition de conséquence logique, de satisfaisabilité et de modèle.**

Théorème 2 : Conséquence logique et implication

- **Soient A et B deux formules et S un ensemble de formules contenant A :**
  - **A |= B ssi |= $(A \Rightarrow B)$.**
  - **S |= B ssi S /{A} |= $(A \Rightarrow B)$.**

III. Logique des prédicats

Théorie des modèles (suite):

Théorème 3 : Remplacement des équivalences

– Soit B une sous-formule de A, et soit C une formule tautologiquement équivalente à B. Si A' est obtenue à partir de A par le remplacement d'**une** occurrence de B par C alors A et A' sont tautologiquement équivalentes.

Théorème 4 : Substitution d'une variable et quantificateurs

– Si A est une formule, x est une variable libre de A et t un terme tq la substitution s= {x\t} soit **définie pour A**, alors les formules : $((\forall\, x\, A) \Rightarrow A_s)$ et $(A_s \Rightarrow \exists\, x\, A)$ sont valides.

III. Logique des prédicats

Théorie des modèles (fin):

Théorème 5 : Renommage et équivalence

– Si A’ est une formule obtenue suite à un renommage de variables à partir d ’une formule A, alors A et A’ sont tautologiquement équivalentes.

Théorème 6 : Équivalences importantes

– Si A est une formule et x est une variable, les couples de formules suivantes sont équivalentes :

- $(\exists x \neg A$ et $\neg(\forall x\ A)\ )$ & $(\forall x \neg A$ et $\neg(\exists x\ A)\ )$
- A et $\forall x\ A$ si x n’est pas libre dans A.
- A et $\exists x\ A$ si x n’est pas libre dans A.
- $\forall x\ (A \wedge B)$ et $(\forall x\ A) \wedge (\forall x\ B)$
- $\exists x\ (A \vee B)$ et $(\exists x\ A) \vee (\exists x\ B)$

III. Logique des prédicats

Théorie de la preuve : Introduction

– Pour la logique des prédicats comme pour la logique propositionnelle , il existe au niveau de la théorie de la preuve, différentes méthodes qui permettent de **prouver** la validité –ou seulement la satisfiabilité- d’une formule ou de **déduire** une formule à partir d’un certain nombre de formules (hypothèses).

– La plupart des définitions et résultats de la logique des propositions restent plus au moins valables tant qu’on respecte les nouvelles définitions.

III. Logique des prédicats

Théorie de la preuve : Méthode de Résolution

Formes normales de prénexe :

Définition 2 : forme normale de prénexe

- Une formule A est en forme normale de prénexe si elle est sans quantificateurs ou de la forme $Q_1x_1 \ldots Q_nx_n$ B , où B est une formule sans quantificateurs et les Q_i des quantificateurs.
- La suite des quantificateurs est appelée préfixe, et B est appelée matrice.

Théorème 4 : Équivalence à une forme normale de prénexe

- Toute formule A est équivalente à une forme normale de prénexe .

III. Logique des prédicats

Théorie de la preuve : Méthode de Résolution (suite)

Algorithme de mise en forme normale de prénexe (sans $\Leftrightarrow$ et $\Rightarrow$)

Entrée : une formule A1 Sortie : une fn de prénexe A'1 équivalente à A1

Début Eliminer $\Leftrightarrow$ et $\Rightarrow$ (et *F* si il est utilisé);

Appliquer à A1 les remplacements suivants autant de fois que nécessaire:

Passage des $\neg$ devant les prédicats

$\neg\neg A$	$\rightarrow$	A
$\neg(A \vee B)$	$\rightarrow$	$\neg A \wedge \neg B$
$\neg(A \wedge B)$	$\rightarrow$	$\neg A \vee \neg B$
$\neg(\forall x\ A)$	$\rightarrow$	$\exists x\ \neg A$
$\neg(\exists x\ A)$	$\rightarrow$	$\forall x\ \neg A$

Sortie des quantificateurs devant les formules

$(\forall x\ A(x)) \wedge (\forall x\ B(x) \rightarrow \forall x\ (A(x) \wedge B(x))$

$(\exists x\ A(x)) \vee (\exists x\ B(x)) \rightarrow \exists x\ (A(x) \vee B(x))$

Rendre A1 propre par renommage.

Appliquer à A1 les remplacements suivants autant de fois que nécessaire

Idem pour B

$(Q\,x\ A(x)) \wedge B \rightarrow Q\,x\ (A(x) \wedge B)$ (x est non libre dans B)

$(Q\,x\ A(x)) \vee B \rightarrow Q\,x\ (A(x) \vee B)$ (x est non libre dans B)

Fin

III. Logique des prédicats

Théorie de la preuve : Méthode de Résolution (suite)

Définition 3 : Forme de Skolem

- Une formule est en forme normale de Skolem si elle est en forme normale prénexe et ne contient pas de quantificateur existentiel.

Définition 4 : Pas de skolémisation

- Soit une formule A en fn de prénexe et qui n'est pas en fn de skolem, donc, il existe une variable y tq :

  $A= Q_1x_1 \ldots Q_{i-1}x_{i-1}\ \exists y\ Q_{i+1}x_{i+1} \ldots Q_nx_n\ B$ (avec $Q_k = \forall$ pour tout $k<i$), càd :

  $A= \forall x_1 \ldots \forall x_{i-1}\ \exists y\ C$, avec $C= Q_{i+1}x_{i+1} \ldots Q_nx_n\ B$ (C est donc en fn de prénexe).

  Un pas de skolémisation appliqué à A et portant sur la variable y consiste à substituer y par $f(x_1, \ldots x_{i-1})$ dans C, avec f un nouveau symbole de fonction.

- N.B : si i=1 alors f est d'arité 0, càd, une constante qu'on notera avec un nouveau symbole de constante, par exemple, a, b, c…

III. Logique des prédicats

Théorie de la preuve : Méthode de Résolution (suite)

Définition 5 : Skolémisation d'une fn de prénexe

- **La skolémisation A^s d'une formule A en forme normale prénexe est la formule résultante de l'application, n fois successivement, d'un pas de skolémisation à A avec n le nombre de variables existentiellement quantifiées de A.**
- **A^s est dite aussi forme de skolem de A.**
- **Rappel: une forme de skolem est propre car une forme de prénexe est propre .**

Théorème 5 : (de Skolem)

- **Le pas de skolémisation préserve la satisfaisabilité.**
- **Soit A une formule sous forme prénexe et A^s la forme de skolem associée. La formule A est satisfaisable dans une interprétation I (pour U) ssi A^s est satisfaisable dans une interprétation étendue I' (pour U).**
- **Rq : la démonstration utilise le théorème 3 de la partie Théorie des modèles et les définitions de modèle et de satisfaisabilité.**