



Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

Université Dr. Tahar Moulay de Saida

Faculté de Technologie

Département d'Informatique

# Cours-Logique mathématique

Présenté par :

**Dr. FELLAH Aissa**

**Maître de conférences « B » en Informatique**

**Juillet 2021**

## *Table des matières*

Avant-propos.....	5
1. Introduction et motivations.....	6
1.1 Définition (Logique).....	6
1.2 Quelques motivations.....	6
1.3 Langage formel et langage naturel.....	7
1.3.1 Définition.....	7
1.4 Bref historique.....	8
1.5 Synthèse.....	10
2. La logique des propositions.....	11
2.1 Introduction.....	11
2.1.1 Définition d'une proposition (Assertion).....	11
2.1.2 Proposition atomique et proposition complexe.....	12
2.2 Syntaxe de la logique des propositions.....	12
2.2.1 L'alphabet.....	12
2.2.2 Les formules bien formées.....	13
2.2.3 Système de connecteurs.....	14
2.2.4 Notions utiles.....	14
2.3 La Sémantique de la logique des propositions.....	16
2.3.1 Introduction.....	16
2.3.2 Sémantique.....	16
2.3.3 La sémantique des connecteurs.....	17
2.3.4 Valeur de vérité d'une proposition.....	17
2.3.5 Table de vérité d'une proposition.....	19
2.3.6 Table de vérité d'un ensemble de formules propositionnelles.....	20
2.4 Quelques notions classiques.....	20
2.4.1 Modèle.....	20
2.4.2 Contre modèle.....	20
2.4.3 Satisfaisabilité.....	21
2.4.4 Contingence.....	21
2.4.5 Validité.....	21
2.4.6 Insatisfaisabilité.....	21
2.4.7 Consistance et Inconsistance.....	21
2.4.8 Complétude.....	21

# Cours-Logique Mathématique

2.4.9 Équivalence logique.....	22
2.4.10 Équivalences de base (Axiomes).....	22
2.5 Modélisation (Représentation des connaissances).....	25
2.5.1 Introduction.....	25
2.5.2 Quelques orientations.....	25
2.5.3 Algorithme de Modélisation.....	26
2.5.4 Conclusion.....	27
2.6 Démonstrations.....	28
2.6.1 Introduction.....	28
2.6.2 Preuve par les tables de vérité (méthode basée sur les modèles).....	29
2.6.3 Preuve par les lois d'équivalences.....	30
2.6.4 La méthode de résolution en logique des propositions.....	31
2.7 Les limites de la logique des propositions.....	38
2.8 Exercices du chapitre 2 : logique des propositions.....	40
2.8.1 Exercices : Syntaxe de la logique des propositions.....	40
2.8.2 Exercices : Sémantique de la logique des propositions.....	42
2.8.3 Exercices : Modélisation en logique des propositions.....	45
2.8.4 Exercices : Démonstration en logique des propositions.....	46
3. La logique des prédicats.....	48
3.1 Introduction.....	48
3.2 Syntaxe de la logique des prédicats.....	49
3.2.1 L'alphabet.....	49
3.2.2 Les termes.....	50
3.2.3 Les atomes (Formule atomique).....	50
3.2.4 Les formules bien formées.....	50
3.2.5 Variable libre et liée.....	51
3.3 Sémantique de la logique des prédicats.....	53
3.3.1 Introduction.....	53
3.3.2 Quelques notions et définitions.....	53
3.3.3 Théorèmes d'équivalence.....	55
3.3.4 Quelques notions classiques (Validité, Insatisfiabilité, Conséquence, Complétude).....	55
3.3.5 Conséquence logique.....	56
3.3.6 Indécidabilité et semi-décidabilité de la logique des prédicats.....	56
3.4 Représentation des connaissances(Modélisation).....	56
3.4.1 Introduction.....	56

# Cours-Logique Mathématique

3.4.2 Quelques orientations.....	57
3.4.3 Exemple.....	58
3.5 Démonstrations.....	59
3.5.1 Introduction.....	59
3.5.2 Forme normale prénexe.....	59
3.5.3 Méthode de Skolémisation.....	61
3.5.4 Forme clausale.....	62
3.6 Unification.....	64
3.6.1 Introduction.....	64
3.6.2 Les substitutions.....	65
3.6.3 Algorithme.....	66
3.7 La méthode de résolution dans le calcul des prédicats.....	67
3.8 Exercices du chapitre 3 : logique des prédicats.....	70
3.8.1 Exercices : Syntaxe de la logique des prédicats.....	70
3.8.2 Exercices : Sémantique de la logique des prédicats.....	71
3.8.3 Exercices : Modélisation en logique des prédicats.....	72
3.8.4 Exercices : démonstration en logique des prédicats.....	75
Références bibliographiques.....	79
Annexe A : Solutions des exercices.....	80
1. Solution des exercices du chapitre 2 : logique des propositions.....	80
1.1 Solutions des exercices : Syntaxe de la logique des propositions.....	80
1.2 Solutions des exercices : Sémantique de la logique des propositions.....	86
1.3 Solutions des exercices : Modélisation en logique des propositions.....	91
1.4 Solutions des exercices : Démonstration en logique des propositions.....	93
2. Solution des exercices du chapitre 3 : logique des prédicats.....	97
2.1 Solutions des exercices : Syntaxe de la logique des prédicats.....	97
2.2 Solutions des exercices : Sémantique de la logique des prédicats.....	99
2.3 Solutions des exercices : Modélisation en logique des prédicats.....	100
2.4 Solutions des exercices : démonstration en logique des prédicats.....	103

## Avant-propos

### **Objectifs du manuscrit**

Le but de ce manuscrit est de donner aux étudiants une connaissance des parties pertinentes de la logique, principalement la logique pour informatique. Il s'agit d'un premier cours de logique mathématique pour les étudiants en informatique.

En s'appuyant sur des sources (examens et des travaux dirigés) électroniques de plusieurs universités, chaque chapitre est enrichi d'exercices, ainsi que leurs corrigés.

Il n'est pas aisé de présenter dans un seul document toutes les facettes du monde de la logique mathématique.

### **Organisation du Manuscrit**

Le chapitre 1 est une introduction à la logique mathématique avec quelques motivations sur l'utilité de la logique pour l'informatique. Il propose quelques définitions, un bref historique et une synthèse de la logique mathématique.

Le chapitre 2, est consacré à la première partie de la logique : logique des propositions. Les propositions sont des formules exprimant des faits mathématiques, Le but du calcul propositionnel est de donner un fondement formel à un ensemble restreint d'énoncés du langage. Nous utiliserons comme élément de base des propositions élémentaires (i.e. des énoncés déclaratifs). Le chapitre est scindé en quatre parties : la syntaxe, la sémantique, la modélisation et enfin le raisonnement.

Les chapitres 3, présente la logique des prédicats qui étend le calcul propositionnel en permettant d'écrire des formules qui dépendent de paramètres; pour cela le calcul des prédicats introduit les notions de variables, de symboles de fonctions et de relations, de termes et de quantificateurs. La démarche du chapitre 2 est adoptée aussi pour les parties du chapitre 3.

Chaque fin de chapitre inclue des séries d'exercices organisées par parties. Le corrigé de chaque exercice est joint en annexe du document.

*"Computer programming is an exact science, in that all the properties of a program and all the consequences of executing it can, in principle, be found out from the text of the program itself by means of purely deductive reasoning".* C.A.R. Hoare (1969)

*"Computers are mathematical machines and computer programs are mathematical expressions".* C.A.R. Hoare (1986)

## 1. Introduction et motivations

### 1.1 Définition (Logique)

La logique vient du grecque « logos » qui signifie « parole, discours », et par extension « rationalité », la logique est donc la science de la raison. Plus précisément, c'est la science qui étudie les règles qui permettent de distinguer un raisonnement valide d'un raisonnement qui ne l'est pas.

La logique est en effet une composante essentielle aussi bien de l'informatique et de la linguistique que des mathématiques et de la philosophie.

La logique mathématique ou métamathématique est une discipline des mathématiques introduite à la fin du xix<sup>e</sup> siècle, qui s'est donné comme objet l'étude des mathématiques en tant que langage.

Les objets fondamentaux de la logique mathématique sont les formules représentant les énoncés mathématiques, les dérivations ou démonstrations formelles représentant les raisonnements mathématiques et les sémantiques ou modèles ou interprétations dans des structures qui donnent un « sens » mathématique générique aux formules.

### 1.2 Quelques motivations

Il est crucial de disposer d'un langage rigoureux. Une langue naturelle (français, arabe, anglais,..) est souvent ambiguë. Prenons l'exemple de la disjonction « ou » ; un invité à la maison « thé ou café » signifie l'un ou l'autre mais pas les deux. Par contre si dans un lycée on cherche à récompenser les meilleurs élèves des mathématiques ou les meilleurs élèves des physiques alors il ne faut pas exclure l'élève qui est meilleur en math et en physiques. Un autre exemple : que répondre à la question « As-tu 100 Dinars en poche ? » si l'on dispose de 1000 Dinars?

- Il y a des notions difficiles à expliquer avec des mots : par exemple la continuité d'une fonction est souvent expliquée par « on trace le graphe sans lever le crayon ». Il est clair que c'est une définition peu satisfaisante. En mathématique la continuité d'une fonction  $f$  définie dans le domaine  $D_f$  en un point  $x_0 \in D_f$ .

$$\forall \varepsilon > 0, \exists \delta > 0 \text{ tel que } \forall x \in D_f \ |x - x_0| < \delta \rightarrow |f(x) - f(x_0)| < \varepsilon$$

C'est le but de rendre cette ligne plus claire ! C'est la logique.

- Enfin les mathématiques tentent de distinguer le vrai du faux. Par exemple « Est-ce qu'une augmentation de 20%, puis de 30% est plus intéressante qu'une augmentation de 50%? ». Vous pouvez penser « oui » ou « non », mais pour en être sûr il faut suivre une démarche logique qui mène à la conclusion. Cette démarche doit être convaincante pour vous mais aussi pour les autres. On parle de raisonnement. Les mathématiques sont un langage pour s'exprimer rigoureusement, adapté aux phénomènes complexes, qui rend les calculs exacts et vérifiables. Le raisonnement est le moyen de valider — ou d'infirmer — une hypothèse et de l'expliquer à autrui.
- L'idée d'utiliser un langage symbolique (formel) permet de simplifier beaucoup de choses en mathématique.

### 1.3 Langage formel et langage naturel

On va donc adapter cette idée à la logique mais avant cela, il faut préciser ce que l'on entend par « langage ». Il faut d'abord distinguer le langage naturel du langage formel.

Le langage naturel est le langage que nous utilisons dans la vie de tous les jours, qui a deux inconvénients majeurs quand on l'utilise en mathématique :

1. la complexité des phrases qui rend les choses plus compliquées, il faut parfois plusieurs lignes et une phrase complètement incompréhensible, pour dire quelque chose qui peut se résumer par une simple équation,
2. le fait que les ambiguïtés du langage courant peuvent conduire à des erreurs, et surtout une preuve se doit indiscutable par définition, ce qui est impossible lorsqu'il y a ambiguïtés.

#### 1.3.1 Définition

Lorsque l'on définit un langage formel, on doit définir deux choses qui caractérisent ce langage:

1. un alphabet c.-à-d. un ensemble de symboles (comme dans le cas des langages naturels),
2. une syntaxe c.-à-d. un ensemble de règles qui définit quels mots appartiennent au langage formel.

Il reste à préciser ce qu'en un mot, c'est très simple : un mot (on dit aussi chaîne de caractères) est une suite ordonnée de symboles, ces symboles appartenant à un alphabet.

Nous pouvons donc définir ce qu'est un langage formel : Un langage formel est un ensemble de mots de longueur finie défini par un alphabet et une syntaxe.

## Cours-Logique Mathématique

---

On peut donner un exemple pour y voir plus clair et définir un langage formel simple (sous ensemble du langage des expressions arithmétiques) : on va prendre un alphabet et une syntaxe.

L'alphabet du langage est l'ensemble contenant les éléments suivants : 1,2,3,4, +,-.

La syntaxe sera composée des règles suivantes :

1. aucun mot ne peut commencer ou terminer par +,
2. aucun mot ne peut commencer ou terminer par -,
3. si on choisit deux symboles consécutifs dans un mot, l'un des deux est un chiffre et pas l'autre.

Par exemple,  $p + q$  est un mot possible, par contre,  $- +p + +pqq$  ne l'est pas.

Nous allons appliquer tout cela à la logique, en faisant attention au fait que l'on dira des « formules » au lieu de dire des « mots ».

### 1.4 Bref historique

- Dans les premières années de ce siècle, la logique symbolique ou formelle est devenue très populaire auprès des philosophes et des mathématiciens parce qu'ils étaient intéressés par le concept de ce qui constitue une preuve correcte en mathématiques. Au cours des siècles, les mathématiciens ont déclaré que différentes preuves mathématiques étaient correctes, ce qui a été ensuite réfuté par d'autres mathématiciens. Tout le concept de logique reposait alors sur l'étude des méthodes et des principes utilisés pour distinguer le raisonnement valide de l'invalid. Cela a été amplement illustré par le nombre de preuves qui ont été formulées pour le postulat parallèle d'Euclide et pour le dernier théorème de Fermat. Il y a toujours eu des « bugs » (terme vulgarisé par les informaticiens pour les défauts d'un programme) qui étaient souvent très difficiles à détecter et il fallait donc trouver des méthodes de preuve infaillibles. Pendant des siècles (remontant au moins à Platon et à Aristote) aucune formulation rigoureuse n'a été tenté pour capturer la notion d'un argument correct qui guiderait le développement de toutes les mathématiques.
- Les premiers logiciens des XIXe et XXe siècles espéraient établir une logique formelle comme fondement des mathématiques, bien que cela ne se soit jamais vraiment produit. Mais les mathématiques reposent sur une base solide, à savoir la théorie des ensembles. Mais la théorie des ensembles elle-même a été exprimée dans une logique du premier ordre. Il fallait vraiment répondre aux questions relatives à l'automatisation ou à la mécanisation des preuves. Ces questions sont très pertinentes et importantes pour le développement de l'informatique actuelle et forment la base de nombreux développements dans la démonstration automatique des théorèmes. David Hilbert a posé la question



importante de savoir si toutes les mathématiques, si elles étaient réduites à des déclarations de logique symbolique, peuvent être dérivées par une machine. L'acte de construire une preuve peut-il être réduit à la manipulation d'énoncés en logique symbolique? La logique a permis aux mathématiciens de montrer pourquoi une preuve alléguée est fautive ou où, dans la preuve, le raisonnement a été erroné. Une grande partie du mérite de cette réalisation doit aller au fait qu'en symbolisant les arguments plutôt qu'en les écrivant dans un langage naturel (qui est plein d'ambiguïté), vérifier l'exactitude d'une preuve devient une tâche beaucoup plus viable. Bien sûr, essayer de symboliser l'ensemble des mathématiques pourrait être désastreux car il deviendrait alors tout à fait impossible de même lire et comprendre les mathématiques.

- Depuis la seconde moitié du XXe siècle, la logique a été utilisée en informatique à des fins diverses, allant de la spécification et de la vérification de programmes à la démonstration de théorèmes. Initialement, son utilisation était limitée à la simple spécification des programmes et au raisonnement sur leurs implémentations. Ceci est illustré dans certaines recherches assez élégantes sur le développement de programmes corrects utilisant une logique de premier ordre dans de tels calculs comme le calcul des plus faibles préconditions de Dijkstra. Une méthode appelée Logique de Hoare qui combine des phrases logiques de premier ordre et des phrases de programme en un mécanisme de spécification et de raisonnement est également très utile dans le développement de petits programmes. La logique sous cette forme a été également utilisée pour spécifier la signification de certains langages de programmation, notamment Pascal.
- Le lien étroit entre la logique en tant que système formel et la démonstration de théorèmes informatisés s'avère très utile, en particulier là où il y a un grand nombre de cas (suivant certains modèles) à analyser et où il existe très souvent des techniques de preuve de routine disponibles qui sont effectuées plus facilement et avec plus de précision par les prouveurs(démonstrateurs) de théorèmes que par les humains. Le cas du théorème à quatre couleurs qui, jusqu'à une date relativement récente, est resté une conjecture non prouvée est un exemple de la façon dont l'ingéniosité et les créativité humaines peuvent être utilisées pour diviser la preuve en quelques milliers de cas et où des machines peuvent être utilisées pour effectuer des vérifications de routine sur l'individu. Une autre utilisation des ordinateurs pour la démonstration de théorèmes ou la vérification de modèles est la vérification de la conception de grands circuits avant la fabrication d'une puce. Analyser des circuits contenant un milliard de transistors est au mieux sujet à des erreurs et au pire une corvée que peu d'humains aimeraient faire. Ces analyses et résultats sont mieux

## Cours-Logique Mathématique

---

réalisés par des machines utilisant des techniques de démonstration de théorèmes ou des techniques de vérification de modèle.

- Un puissant paradigme de programmation appelé programmation déclarative a évolué depuis la fin des années 70 et a trouvé plusieurs applications en informatique et en intelligence artificielle. La plupart des programmeurs utilisant ce paradigme logique utilisent un langage appelé **Prolog** qui est une forme d'implémentation de la logique. Plus récemment, les informaticiens travaillent sur une forme de logique appelée programmation logique par contraintes.

### 1.5 Synthèse

- La logique mathématique a commencé vers la fin du siècle dernier quand Frege a développé le calcul des prédicats.
- La logique mathématique implique l'application des méthodes mathématiques standards pour l'étude des systèmes qui peuvent eux-mêmes être utilisés pour formaliser les mathématiques. La circularité apparente est surmontée en distinguant le langage objet, en quel le système formel est exprimé, et le langage de l'observateur dans lequel les propriétés du système formel sont exprimées et raisonnées.
- Les éléments de base que la logique traite sont des propositions. Les propositions sont utilisées pour exprimer des croyances. En langage naturel, ils sont représentés par phrases déclaratives.
- La notion de croyance est très générale. Néanmoins il y a quelques restrictions sur la façon dont les croyances peuvent être manipulées dans le raisonnement mental. Par exemple vous ne pouvez pas simultanément avoir des croyances contradictoires (au moins sans être conscient que quelque chose ne va pas).
- L'importance de la formalisation est qu'une fois un domaine particulier de mathématiques ou d'informatique a été formalisé, le raisonnement peut être réalisé purement par la manipulation de symbole, sans référence à la signification ou la compréhension, et les propriétés mathématiques du processus de raisonnement peuvent être clairement énoncées et prouvées.

## 2. La logique des propositions

### 2.1 Introduction

La logique propositionnelle permet essentiellement de discuter des connecteurs grammaticaux comme la négation, la conjonction et la disjonction, en composant des propositions à partir de propositions atomiques. Ces connecteurs sont parfois appelés aristotéliens, car ils ont été mis en évidence par Aristote.

Le calcul propositionnel permet essentiellement de parler de fonctions booléennes, c'est-à-dire de fonctions de  $\{0,1\}^n \rightarrow \{0,1\}$ . En effet, les variables, c'est-à-dire les propositions, ne peuvent prendre que deux valeurs, vrai ou faux.

Le calcul propositionnel tient une grande place en informatique : ne serait-ce parce que nos ordinateurs actuels sont digitaux, et travaillent en binaire. Ce qui fait que nos processeurs sont essentiellement constitués de portes binaires du type de celles que l'on va étudier dans ce chapitre.

D'un point de vue expressivité logique, le calcul propositionnel reste très limité : par exemple, on ne peut pas écrire en calcul propositionnel l'existence d'un objet ayant une propriété donnée. Le calcul des prédicats, plus général, que nous étudierons dans le chapitre 3, permet lui d'exprimer des propriétés d'objets et des relations entre objets, et plus généralement de formaliser le raisonnement mathématique.

Puisque le calcul propositionnel forme toutefois la base commune de nombreux systèmes logiques, et nous allons-nous-y attarder dans ce chapitre.

#### 2.1.1 Définition d'une proposition (Assertion)

La notion de « proposition » est fondamentale en logique et déjà présente dans la logique d'Aristote. On peut définir cette notion très facilement : une proposition est tout simplement une affirmation. Il faut faire attention que cette affirmation puisse être vrai ou fausse.

Aristote a défini ainsi le concept de proposition : "... tout discours n'est pas une proposition, mais seulement le discours dans lequel réside le vrai ou le faux, ce qui n'arrive pas dans tous les cas : ainsi la prière est un discours, mais elle n'est ni vraie ni fausse"

Tout simplement, une proposition ou une assertion est une phrase soit vraie, soit fausse, pas les deux en même temps.

#### Exemples :

- « Il pleut. »
- « Je suis plus grand que toi. »
- " $2 + 2 = 4$ "
- " $2 - 3 = 7$ "

- "Pour tout  $x \in \mathcal{R}$ , on a  $x^2 \geq 0$ "

Sont des propositions par contre

- Quel âge a tu ?
- Fermez la porte S.V.P

Ne le sont pas.

Si  $p$  est une proposition et  $q$  est une autre proposition, nous allons définir de nouvelles assertions construites à partir de  $p$  et de  $q$ .

### 2.1.2 Proposition atomique et proposition complexe

#### ➤ Proposition atomique

Une proposition atomique ne comporte pas de parties. Elle ne contient pas d'autres propositions qu'elle-même.

#### ➤ Proposition complexe

Une proposition complexe est composée de plusieurs propositions atomiques ou complexes. Les propositions complexes relient entre elles des propositions atomiques ou complexes par des connecteurs logiques.

### Exemple

« Il fait beau » : est une proposition atomique.

« Le soleil brille et il y a du vent » est une proposition complexe constituée de 2 propositions atomiques : « le soleil brille » et « il y a du vent ».

#### ➤ En logique on distingue deux aspects :

-l'aspect syntaxique

-l'aspect sémantique

## 2.2 Syntaxe de la logique des propositions

Pour étudier la syntaxe d'un langage il faut donner un alphabet (un ensemble de symboles) et des règles de constructions syntaxiques d'expressions à partir de ces symboles.

### 2.2.1 L'alphabet

L'alphabet est constitué :

- de connecteurs :  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$  qui se lisent respectivement non, et, ou, implique et équivalent.
- de délimiteurs : les parenthèses (,)

## Cours-Logique Mathématique

Les parenthèses sont utiles dans les formules logiques car il faut se rendre compte par exemple que la formule  $\neg F \wedge G$  est différente de la formule  $\neg(F \wedge G)$ . Les parenthèses permettent de lever l'ambiguïté.

- d'un ensemble  $S$  dénombrable de symboles propositionnels appelés aussi propositions ou variables propositionnelles (**atomes**)
- des deux constantes propositionnelles  $V$  (vrai)( $\top$ ) et  $F$  (faux)( $\perp$ )

**NB** : Par convention pour ce cours on notera les symboles propositionnels (atomes) avec les minuscules de l'alphabet latin, et les concaténations de telles lettres  $\{a, b, \dots, p, q, \dots, z, aa, ab\}$

### 2.2.2 Les formules bien formées

Le **langage** est constitué de l'ensemble des Formules Bien Formées (appelées aussi : FBFs ou Well Formed-Formula WFF) ou expressions bien formées défini comme suit:

- Base : tout atome est une fbf, de même les constantes propositionnelles sont des fbfs
- Induction : si  $F$  et  $G$  sont des fbfs alors  $(\neg G)$ ,  $F \wedge G$ ,  $F \vee G$ ,  $F \rightarrow G$  et  $F \leftrightarrow G$  sont des fbfs
- Clôture : toutes les fbfs sont obtenues par application des 2 règles ci-dessus.

#### *Formellement*

Soit  $S$  un ensemble dénombrable de symboles propositionnels, l'ensemble des deux constantes  $\{\top, \perp\}$ , un ensemble de connecteurs  $C = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ , et un jeu de parenthèses  $D = \{(, )\}$

On définit par induction **PROP(S)**, l'ensemble des fbfs de la logique des propositions (ou **propositions**) construites sur  $S$  (donc sur l'univers des mots de  $(S \cup C \cup D)^*$ ):

- (Base) :  $PROP(S)$  contient  $S \cup \{\top, \perp\}$
- (Construction) : Si  $P$  et  $Q \in PROP(S)$  (i.e. sont des fbf)

Règle 1 :  $\neg P \in PROP(S)$

Règle 2 :  $(P C_B Q) \in PROP(S)$

( $C_B = \{\wedge, \vee, \rightarrow, \leftrightarrow\}$  Connecteurs binaires)

**NB** : On omet toujours les parenthèses les plus à l'extérieur.  $(p \vee q)$  devient  $p \vee q$

En général, l'omission de parenthèses peut être source d'ambiguïtés : ainsi,  $(\neg p \vee q)$  peut correspondre à  $(\neg p) \vee q$  et à  $\neg(p \vee q)$ . Cependant, on peut omettre la plupart des parenthèses en donnant des priorités aux connecteurs. Par exemple (le plus utilisé généralement) :

Ordre de priorité des connecteurs : (Le plus prioritaire)  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

## Cours-Logique Mathématique

$a \wedge \neg b \vee c \rightarrow d \vee e$  doit se lire  $(a \wedge (\neg b)) \vee c \rightarrow (d \vee e)$

Quand il y a un seul connecteur, l'association se fait de gauche à droite.

$a \rightarrow b \rightarrow c$  Correspond à  $((a \rightarrow b) \rightarrow c)$

✚ L'écriture préfixée (polonaise) est autre manière d'exprimer les formules sans utilisation des parenthèses

La formule infixée  $(a \vee b)$  devient  $\vee ab$  en préfixée.

✚ Différentes fbf's peuvent représenter les « mêmes conditions de vérité » (avoir la même sémantique)

$a \leftrightarrow b$  et  $(a \rightarrow b) \wedge (b \rightarrow a)$

$a$  et  $(a \wedge a \wedge a)$

### 2.2.3 Système de connecteurs

L'ensemble des connecteurs logiques, appelé système de connecteurs, que nous choisissons pour écrire nos formules. On en veut le moins de symboles possibles pour décrire notre logique.

#### Définition

On dit qu'un système de connecteurs logiques est complet si toute formule logique peut s'exprimer comme formule logique utilisant ces connecteurs et seulement ces connecteurs. Si, de plus, tous sous-ensembles de connecteurs issus de ce système ne forment pas un système de connecteur complet, on parle de système complet minimal.

#### Exemples

$\{\neg, \wedge\}$ ,  $\{\neg, \vee\}$ ,  $\{\text{Nand}\}$  sont des systèmes complets minimaux. ( $p \text{ Nand } q = \neg(p \wedge q)$ )

$\{\neg, \wedge, \vee\}$  est un système complet mais pas minimal.

### 2.2.4 Notions utiles

• On peut définir différentes fonctions en s'appuyant sur la définition par induction de  $\text{PROP}(S)$  :

#### 2.2.4.1 Ensemble des symboles propositionnels d'une fbf

$1\text{-SP}$  (ensemble des symboles propositionnels d'une fbf) :  $\text{PROP}(S) \rightarrow 2^S$

(Base) Si  $F = p$  avec  $(p \in S)$ ,  $\text{SP}(F) = \{p\}$

(Construction)

(Règle 1) Si  $F = \neg Q$ ,  $\text{SP}(F) = \text{SP}(Q)$

(Règle 2) Soient  $Q, R \in \text{PROP}(S)$ : Si  $F = (Q \text{ C}_B R)$ ,  $\text{SP}(F) = \text{SP}(Q) \cup \text{SP}(R)$

#### 2.2.4.2 Nombre de connecteurs d'une fbf

$2\text{-nbc}$  (nombre de connecteurs d'une fbf) :  $\text{PROP}(S) \rightarrow \mathbb{N}$

(Base) Si  $F = p$  avec  $(p \in S)$ ,  $\text{nbc}(F) = 0$

(Construction)

(Règle 1) Si  $F = \neg Q$ ,  $\text{NBC}(F) = 1 + \text{NBC}(Q)$

(Règle 2) Soient  $Q, R \in \text{PROP}(S)$ : Si  $F = (Q C_B R)$ ,  $\text{NBC}(F) = \text{NBC}(Q) + \text{NBC}(R)$

### 2.2.4.3 Sous-formules d'une fbf

**SF** (Sous-formules d'une fbf) :  $\text{PROP}(S) \rightarrow \wp(\text{PROP}(S))$

(Base) Si  $F = p$  avec  $(p \in S)$ ,  $\text{SF}(F) = \{F\}$

(Règle 1) Si  $F = \neg Q$ ,  $\text{SF}(F) = \{F\} \cup \text{SF}(Q)$

(Règle 2) Soient  $Q, R \in \text{PROP}(S)$ : Si  $F = (Q C_B R)$ ,  $\text{SF}(F) = \{F\} \cup \text{SF}(Q) \cup \text{SF}(R)$

- L'endroit où une sous-formule apparaît est son *occurrence*.

### Exemples de sous-formules

- Considérons la formule  $((p \vee q) \wedge \neg p)$ .  
 $p$  en est une sous-formule, ainsi que  
 $(p \vee q)$  et  $(\neg p)$ , tandis que  $(q \wedge \neg p)$  ne l'est pas.  
 $p$  a deux occurrences dans  $((p \vee q) \wedge \neg p)$ , et  $(p \vee q)$  une.
- l'ensemble des sous-formules de  $((p \vee q) \wedge \neg p) \rightarrow \perp$  est  
 $\{(((p \vee q) \wedge \neg p) \rightarrow \perp), ((p \vee q) \wedge \neg p), \perp, (p \vee q), (\neg p), p, q\}$ .

### 2.2.4.4 Arborecence associée à une fbf

**arb** (l'arbor. associée à une fbf) :  $\text{PROP}(S) \rightarrow \text{ARBO}(S)$

(Base) Si  $F = p$  avec  $(p \in S)$ ,  $\text{arb}(F)$ =arbor. réduite à un sommet étiqueté par  $p$

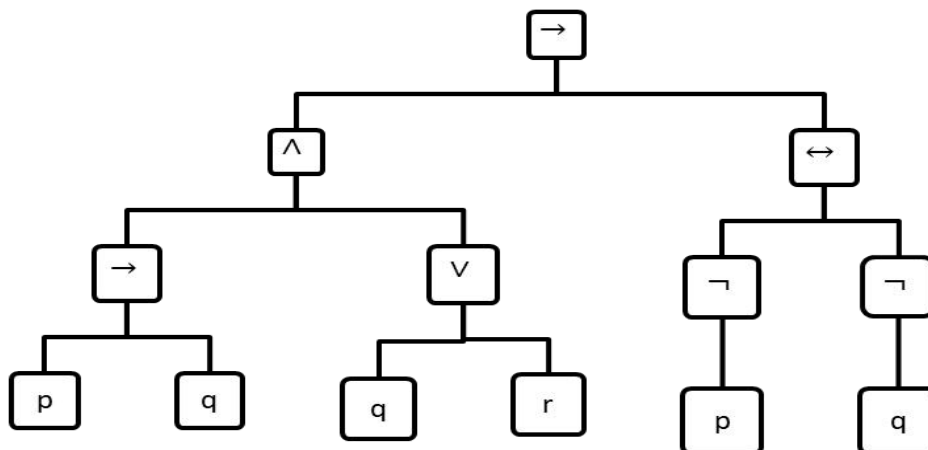
(Règle 1) Si  $F = \neg Q$ ,  $\text{arb}(F)$ =arbor. de racine étiquetée par  $\neg$  ayant comme unique fils la racine de  $\text{arb}(Q)$

(Règle 2) Si  $F = (Q C_B R)$ ,  $\text{arb}(F)$ =arbor. de racine étiquetée par  $C_B$  ayant comme fils gauche la racine de  $\text{arb}(Q)$  et comme fils droit la racine de  $\text{arb}(R)$

### Exemple

$((p \rightarrow q) \wedge (q \vee r)) \rightarrow (\neg p \leftrightarrow \neg r)$

L'arbre de la formule est :



❖ **Propriété** : l'application arb est une bijection

Si les arbres montrent bien la structure d'une fbf, ils ne sont pas « pratique » pour l'écriture

– La notation infixée est la manière classique d'écrire les fbf mais elle nécessite l'utilisation de parenthèses. Elle est obtenue à partir de l'arbre d'une fbf avec un parcours GRD(Gauche-Racine-Droit) de l'arbre.

– On peut par des conventions de priorité de connecteurs en éliminer quelques parenthèses (par exemple  $\neg$  est *prioritaire* sur  $\wedge$  et  $\vee$  qui sont prioritaires sur  $\rightarrow$  et  $\leftrightarrow$ )

$((p \vee (p \rightarrow q)) \rightarrow (\neg p))$  Devient par convention  $p \vee (p \rightarrow q) \rightarrow \neg p$

– La notation préfixée a l'avantage de ne pas nécessiter de parenthèses

$((p \vee (p \rightarrow q)) \rightarrow (\neg p))$  devient  $\rightarrow \vee p \rightarrow pq \neg p$

## 2.3 La Sémantique de la logique des propositions

### 2.3.1 Introduction

La sémantique attribue une signification à la formule. Elle est compositionnelle : la signification d'une formule est fonction de celle de ses constituants.

### 2.3.2 Sémantique

La sémantique du calcul propositionnel consiste à donner :

1. Un domaine sémantique non vide  $B$
2. Une interprétation des atomes (interprétation des variables propositionnelles) dans  $B$
3. Une définition des connecteurs par des applications de  $B$  dans  $B$  pour  $\neg$  et de  $B^2$  dans  $B$  pour  $\wedge, \vee, \rightarrow, \leftrightarrow$ .

L'interprétation classique de la logique des propositions pour lequel  $B = \{V, F\}$  ou  $B = \{0, 1\}$



Tout atome est vrai ou faux mais pas les deux à la fois

**Définition (Interprétation)** Une interprétation est une fonction qui associe une valeur de vérité à chaque variable propositionnelle. C'est donc une fonction  $I: S \rightarrow B$ .

### 2.3.3 La sémantique des connecteurs

La valeur de vérité d'une fbf est calculée à partir de l'interprétation de ces composants : on parle de sémantique compositionnelle

- On interprète les connecteurs et symboles constants toujours de la même façon (sinon on change de logique!)
- Donner une **interprétation**  $I$  consiste donc uniquement à spécifier **l'application de S (l'ensemble des symboles propositionnels) dans B**  
(Qui d'un point de vue sémantique intuitive modélise un monde possible)

#### Interprétation des symboles

- Soit  $S = \{p, q, r\}$  on peut définir
  - $I1: S \rightarrow B$
  - $I1(p) = 0, I1(q) = 0, I1(r) = 0$
  - $I2: S \rightarrow B$
  - $I2(p) = 0, I2(q) = 1, I2(r) = 1$

...

- Si  $card(S) = n$ , il y a  $2^n$  interprétations différentes

– L'interprétation des connecteurs et des symboles constants étant la même dans toutes les interprétations. Les constantes sont interprétées par chacune des 2 valeurs de B :

$$I(\perp) = 0$$

$$I(\top) = 1$$

- Les connecteurs sont interprétés par des fonctions de  $B^n \rightarrow B$  (où n est l'arité du connecteur)

$$I(\neg) = NON : B \rightarrow B \text{ t. q. } NON(0) = 1 \text{ et } NON(1) = 0$$

$$I(\vee) = ET : B^2 \rightarrow B \text{ t. q. } ET(a, b) = 1 \text{ ssi } a = b = 1$$

$$I(\vee) = OU : B^2 \rightarrow B \text{ t. q. } OU(a, b) = 0 \text{ ssi } a = b = 0$$

$$I(\rightarrow) = SIALORS : B^2 \rightarrow B \text{ t. q. } SiALORS(a, b) = 0 \text{ ssi } a = 1 \text{ et } b = 0$$

$$I(\leftrightarrow) = SSI : B^2 \rightarrow B \text{ t. q. } SSI(a, b) = 1 \text{ ssi } a = b$$

### 2.3.4 Valeur de vérité d'une proposition

- Soit  $I$  une interprétation des symboles propositionnels d'une fbf  $F$  on définit la valeur de vérité de  $F$  dans l'interprétation  $I$  que l'on note  $V(F, I)$  par :

## Cours-Logique Mathématique

---

(Base) Si  $F = p$  ( $p \in S$ ),  $V(F, I) = I(p)$

(Construction)

(Règle 1) Si  $F = \neg Q$ ,  $V(F, I) = NON(V(Q, I))$

(Règle 2) Si  $F = (Q \wedge R)$ ,  $V(F, I) = ET(V(Q, I), V(R, I))$

(Règle 3) Si  $F = (Q \vee R)$ ,  $V(F, I) = OU(V(Q, I), V(R, I))$

(Règle 4) Si  $F = (Q \rightarrow R)$ ,  $V(F, I) = SIALORS(V(Q, I), V(R, I))$

(Règle 5) Si  $F = (Q \leftrightarrow R)$ ,  $V(F, I) = SSI(V(Q, I), V(R, I))$

### Définition

Une fbf  $F$  est vraie (respectivement fausse) dans une interprétation  $I$  si la valeur de  $F$  est vraie (respectivement fausse) On écrit  $V(F, I) = 1$  (respectivement  $V(F, I) = 0$ )

### Exemple

Soit la formule  $F$  définie par :  $F = ((x \rightarrow y) \rightarrow x) \rightarrow x$  et  $I$  l'interprétation  $I(x) = 0, I(y) = 1$ .

$$V(F, I) = V(((x \rightarrow y) \rightarrow x) \rightarrow x, I)$$

$$V(((x \rightarrow y) \rightarrow x) \rightarrow x, I) = SiAlors(V((x \rightarrow y) \rightarrow x), I), V(x, I))$$

$$= SiAlors(SiAlors(V(x \rightarrow y), I), I(x)), I(x))$$

$$= SiAlors(SiAlors(SiAlors(I(x), I(y)), I(x)), I(x))$$

$$= SiAlors(SiAlors(SiAlors(0,1), 0), 0)$$

$$= SiAlors(SiAlors(1,0), 0)$$

$$= SiAlors(0,0)$$

$$= 1$$

○ donc  $V(((x \rightarrow y) \rightarrow x) \rightarrow x, I) = 1$ .

Cette formule est en fait une tautologie (vraie pour toute interprétation).

### Exemple

**G:**  $(p \rightarrow (q \vee (\neg r)))$

La notation  $I_1(p)$  se lit valeur de  $p$  selon l'interprétation  $I_1$

Soit  $I_1$  telle que:  $I_1(p) = I_1(r) = 1$  et  $I_1(q) = 0$  (la formule  $G$  est fausse dans  $I_1$ )

Soit  $I_2$  telle que:  $I_2(p) = I_2(r) = 0$  et  $I_2(q) = 1$  (la formule  $G$  est vraie dans  $I_2$ )

$$V(G, I) = V(p \rightarrow (q \vee (\neg r)), I) = Sialors(V(p, I), V(q \vee \neg r, I))$$

$$= Sialors(I(p), OU(V(q, I), V(\neg r, I)))$$

$$= Sialors(I(p), OU(I(q), Non(V(r, I))))$$

=

$$= \text{Sialors}(I(p), \text{OU}(I(q), \text{Non}(I(r))))$$

- Pour  $I = I_1$  on a :

$$\text{Non}(I_1(r)) = 0$$

$$\text{OU}(I_1(q), \text{Non}(I_1(r))) = 0 \text{ et } \text{Sialors}(I_1(p), \text{OU}(I_1(q), \text{Non}(I_1(r)))) = 0$$

G est fausse dans  $I_1$

- Pour  $I = I_2$  on a :

$$\text{Non}(I_2(r)) = 1$$

$$\text{OU}(I_2(q), \text{Non}(I_2(r))) = 1 \text{ et } \text{Sialors}(I_2(p), \text{OU}(I_2(q), \text{Non}(I_2(r)))) = 1$$

G est vraie dans  $I_2$

La valeur d'une formule ne dépend que de la valeur de l'interprétation pour les variables qui apparaissent dans la formule. Si deux interprétations  $I_1$  et  $I_2$  coïncident (ont la même valeur) sur l'ensemble des variables propositionnelles d'une formule  $F$  alors  $V(F, I_1) = V(F, I_2)$ .

### 2.3.5 Table de vérité d'une proposition

Une table de vérité d'une fbf  $F$  est un tableau ayant :

- pour indice de lignes les  $2^n$  interprétations possibles des  $n$  symboles propositionnels de  $F$ ,
- pour indices de colonnes les sous-fbf de  $F$  (la dernière colonne étant  $F$ ),
- dans une case de ligne  $I$  et de colonne  $Q$  la valeur  $V(Q, I)$  (la valeur de vérité de la fbf  $Q$  dans l'interprétation  $I$ )

- Ainsi la dernière colonne d'une table de vérité d'une fbf contient les valeurs de  $F$  pour toutes les interprétations possibles.

**Définition des connecteurs par des tables de vérité :**

$a$	$b$	$\neg a$	$a \wedge b$	$a \vee b$	$a \rightarrow b$	$a \leftrightarrow b$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

- Pour une fbf  $F$  composée de différents atomes (variables propositionnels) :  $a_1, \dots, a_n$ , une interprétation de  $F$  est une assignation des valeurs de vérité à  $a_1, \dots, a_n$ .
- $F$  Comporte  $n$  atomes  $\implies 2^n$  interprétations possibles

**Exemple**

## Cours-Logique Mathématique

La table vérité de :  $F = (p \vee \neg q) \wedge (\neg p \vee q)$

	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_5$	$Q_6$	$Q_7(F)$
	$p$	$q$	$\neg p$	$\neg q$	$p \vee \neg q$	$\neg p \vee q$	$(p \vee \neg q) \wedge (\neg p \vee q)$
$I_1$	0	0	1	1	1	1	1
$I_2$	0	1	1	0	0	1	0
$I_3$	1	0	0	1	1	0	0
$I_4$	1	1	0	0	1	1	1

### 2.3.6 Table de vérité d'un ensemble de formules propositionnelles

Soit  $E$  un ensemble de fbfs, on construit le tableau dont :

-les lignes correspondent aux  $2^n$  interprétations possibles des  $n$  symboles propositionnels apparaissant dans les fbfs de  $E$

-les colonnes sont les fbfs (et les sous fbf) de  $E$

➤ L'intérêt :

-la comparaison de fbfs

-égalité sémantique

-déduction

### Exemple

Soit  $E = \{\neg \perp, p \vee \neg p, \neg(p \rightarrow p)\}$

$\perp$	$\neg \perp$	$p$	$\neg p$	$p \vee \neg p$	$p \rightarrow p$	$\neg(p \rightarrow p)$
0	1	0	1	1	1	0
0	1	1	0	1	1	0

## 2.4 Quelques notions classiques

### 2.4.1 Modèle

On dit qu'une interprétation  $I$  est un modèle d'une formule  $F$  si la valeur de  $F$  selon l'interprétation  $I$  est vraie  $V(F, I) = 1$

On dit que  $I$  est un modèle d'un ensemble de formules  $E$  si  $I$  est modèle de tout élément de  $E$

### 2.4.2 Contre modèle

On dit qu'une interprétation  $I$  est un contre modèle d'une formule  $F$  si la valeur de  $F$  selon l'interprétation  $I$  est fausse  $V(F, I) = 0$

## 2.4.3 Satisfaisabilité

Une fbf  $A$  est consistante ou satisfiable

- si et seulement si elle n'est pas inconsistante
- si il existe une interprétation  $I$  telle que  $V(F, I) = 1$
- si elle admet au moins un modèle

$A \wedge B$  et  $A \vee B$  sont deux formules consistantes il suffit que  $A$  et  $B$  soient vraies

- $F$  est satisfiable si seulement si  $\neg F$  n'est pas valide

## 2.4.4 Contingence

Une formule est contingente si elle possède au moins un modèle et un contre modèle

- $F$  est contingente si seulement si  $\neg F$  contingente

## 2.4.5 Validité

Une fbf  $A$  est une **valide (tautologie)** si et seulement si elle est vraie dans toute interprétation;  
(une formule est valide si toute interprétation est un modèle)

-Une formule toujours vraie est une tautologie,

( $F$  est une tautologie SSI  $V(F, I) = 1 (\forall I)$  On écrit alors :  $\models A$ )

-Une tautologie est appelée aussi théorème en mathématique

- $\neg A \vee A$  est une formule valide
- $A \wedge B$  et  $A \vee B$  sont deux formules invalides (il suffit que  $A$  et  $B$  soient fausses)

## 2.4.6 Insatisfaisabilité

Une fbf est inconsistante ou insatisfiable si et seulement si elle est fausse dans toute interprétation

( $F$  est insatisfiable SSI  $V(F, I) = 0 (\forall I)$ )

(Une formule est insatisfiable si elle ne possède pas de modèle)

- Exemple :  $\neg A \wedge A$  est une formule inconsistante
- $F$  est insatisfiable si seulement si  $\neg F$  est valide

## 2.4.7 Consistance et Inconsistance

-On étend la satisfaisabilité à un ensemble de fbfs :

-  $\{F_1, F_2, \dots, F_n\}$  est dit consistant si et seulement si  $F_1 \wedge F_2 \wedge \dots \wedge F_n$  est satisfiable

-  $\{F_1, F_2, \dots, F_n\}$  est dit contradictoire (inconsistant) si et seulement si  $F_1 \wedge F_2 \wedge \dots \wedge F_n$  est insatisfiable

## 2.4.8 Complétude

**Théorème de complétude du calcul propositionnel :**

Pour toute formule  $A$  si  $\models A$  alors  $\vdash A$  **Autrement dit** : toutes les tautologies sont des théorèmes.

### 2.4.9 Équivalence logique

#### Définition :

Deux fbfs  $F$  et  $G$  sont (sémantiquement) **équivalentes** si et seulement si les valeurs de vérité de  $F$  et de  $G$  sont les mêmes dans toute interprétation. Plus formellement :

$(F \equiv G) \text{ SSI } \forall I \ V(F, I) = V(G, I) \ (\forall I)$  Le symbole " $\equiv$ " se lit "est (sémantiquement) équivalent à".

➤ Si  $F \models G$  et  $G \models F$ , on écrit alors  $F \equiv G$ ,

➤ Théorème :  $(F \equiv G)$  si et seulement si  $(F \leftrightarrow G)$  est valide

(À ne pas confondre entre :

- $(F \equiv G)$  (Équivalence logique, notion sémantique)
- $(F \leftrightarrow G)$  (Connecteur équivalent)
- $(F = G)$  (Égalité syntaxique, notion syntaxique))

### 2.4.10 Équivalences de base (Axiomes)

Soient  $A$ ,  $B$  et  $C$  des formules bien formées.

#### 1. Implication matérielle (implication-comme-disjonction)

$$A \rightarrow B \equiv \neg A \vee B$$

#### 2. Équivalence matérielle (double-implication)

$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$$

#### 3. Commutativité

a)  $A \vee B \equiv B \vee A$  **commutativité-ou**

b)  $A \wedge B \equiv B \wedge A$  **commutativité-et**

#### 4. Associativité

a)  $(A \vee B) \vee C \equiv A \vee (B \vee C)$  **associativité-ou**

b)  $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$  **associativité-et**

#### 5. Distributivité

a)  $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$  **distributivité-ou-sur-et**

b)  $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$  **distributivité-et-sur-ou**

#### 6. True

a)  $A \vee \top \equiv \top$  **vrai est absorbant pour ou**

b)  $A \wedge \top \equiv A$  **vrai est neutre pour et**

#### 7. Absurde

- a)  $A \vee \perp \equiv A$  faux est neutre pour ou
- b)  $A \wedge \perp \equiv \perp$  faux est absorbant pour et

### 8. Complémentarité

- a)  $A \vee \neg A \equiv \top$  tiers-exclus
- b)  $A \wedge \neg A \equiv \perp$  contradiction

### 9. Involution (double négation)

$$(\neg(\neg A)) \equiv A$$

### 10. Lois de De Morgan

- a)  $\neg(A \vee B) \equiv (\neg A) \wedge (\neg B)$  De-Morgan-négation-ou
- b)  $\neg(A \wedge B) \equiv (\neg A) \vee (\neg B)$  De-Morgan-négation-et

### 11. Absorption

- a)  $A \vee (\neg A \wedge B) \equiv A \vee B$  absorption-ou-sur-et avec  $\neg$
- b)  $A \wedge (\neg A \vee B) \equiv A \wedge B$  absorption-et-sur-ou avec  $\neg$
- c)  $A \wedge (A \vee B) \equiv A$  absorption-et-sur-ou
- d)  $A \vee (A \wedge B) \equiv A$  absorption-ou-sur-et

### 12. Idempotence

- a)  $A \wedge A \equiv A$  idempotence- et
- b)  $A \vee A \equiv A$  idempotence- ou

On peut démontrer que ces formules sont équivalentes en montrant qu'elles ont les mêmes valeurs dans toutes les interprétations. Un moyen est donc de construire leurs tables de vérité.

On peut utiliser ces théorèmes d'équivalence pour transformer une formule bien formée en une autre formule bien formée qui lui est équivalente. Cela va permettre de simplifier l'écriture de formules bien formées.

#### 2.4.11 Substitution

Etant donnée une fbf  $F$ , la substitution consiste remplacer une occurrence (plusieurs ou toutes) d'une sous-fbf  $Q$  de  $F$  par une autre fbf  $R$ .

**Exemple :**

$$F = ((p \rightarrow q) \vee \neg r) \quad Q = p \rightarrow q \quad R = \neg p \vee q \text{ on substitue } R \text{ par } Q \text{ dans } F :$$

$$F' = ((\neg p \vee q) \vee \neg r)$$

#### 2.4.12 Théorème de substitution

Soit  $F$  une fbf,  $Q$  une sous-fbf de  $F$  et  $R$  une fbf équivalence  $Q$  ( $Q \equiv R$ ) ;

Si on note par  $F'$  la fbf obtenue à partir de  $F$  en remplaçant une occurrence de  $Q$  par  $R$  alors

$$F \equiv F'$$

### 2.4.13 Conséquence logique

Une fbf  $C$  est une conséquence logique d'un ensemble de fbfs  $E = \{H_1, \dots, H_n\}$  si et seulement si toutes les interprétations qui rendent vraies toutes les formules de  $E$  rendent vraie la formule  $C$ .

On note alors  $E \models C$

- La notion de conséquence logique peut être considérée comme une modélisation d'un raisonnement.

On dit qu'une formule  $C$  est une conséquence logique de  $H_1 \dots H_n$

- si et seulement si tout modèle de  $\{H_1, \dots, H_n\}$  est un modèle de  $C$
- si et seulement si  $H_1 \wedge H_2 \wedge \dots \wedge H_n \rightarrow C$  est valide

Dans ce contexte les formules  $H_i$  sont les hypothèses et  $C$  est la conclusion.

#### Exemple : Règles d'inférence pour la logique propositionnelle

- $\{p, p \rightarrow q\} \models q$  (Modus ponens)
- $\{\neg q, p \rightarrow q\} \models \neg p$  (Modus tolens ou preuve par l'absurde)
- $\{p \rightarrow q, q \rightarrow r\} \models p \rightarrow r$  (transitivité de l'implication)
- $\{p \vee q, \neg p \vee r\} \models q \vee r$  (résolution de Robinson)
- $\{p \wedge q\} \models p$  (élimination selon  $\wedge$ )
- $\{p, q\} \models p \wedge q$  (introduction selon  $\wedge$ )
- $\{p\} \models p \vee q$  (introduction selon  $\vee$ )
- $\{p \leftrightarrow q\} \models p \rightarrow q$  (élimination selon  $\leftrightarrow$ )
- $\{p \rightarrow q, q \rightarrow p\} \models p \leftrightarrow q$  (introduction selon  $\leftrightarrow$ )
- $\{p, p\} \models \perp$  (contradiction)
- $\{p \vee q, p \rightarrow r, q \rightarrow r\} \models r$  (preuve par cas)
- $\{\neg p \rightarrow \perp\} \models p$  (preuve par l'absurde)
- Si  $\{p, Q\} \models q$  alors  $\{Q\} \models p \rightarrow q$  (théorème d'Herbrand)

#### Propriétés

- si  $F$  est valide alors  $E \models F$  (pour un  $E$  quelconque y compris  $\emptyset$ , on la note  $\models F$ )
- si  $E$  est contradictoire alors  $E \models F$  (pour un  $F$  quelconque)
- $F \equiv Q$  si et seulement  $\{F\} \models Q$  et  $\{Q\} \models F$

#### Propriété importante

$\{h_1, h_2, \dots, h_k\} \models C$  Si et seulement si

- $(h_1 \wedge h_2 \wedge \dots \wedge h_k) \rightarrow C$  est valide



➤  $h_1 \wedge h_2 \wedge \dots \wedge h_k \wedge \neg C$  est insatisfiable

Ainsi résoudre le problème de la validité d'un raisonnement revient à celui de la validité ou la satisfiabilité d'un raisonnement

### 2.5 Modélisation (Représentation des connaissances)

#### 2.5.1 Introduction

La représentation des connaissances en Intelligence Artificielle consiste à faire une correspondance entre le monde réel et un système symbolique manipulable par un ordinateur. La représentation des connaissances comporte un aspect passif : il faut mémoriser. Par exemple, un livre ne connaît pas l'information qu'il contient. Mais aussi un côté actif : il faut inférer, manipuler ces connaissances, effectuer un raisonnement.

**CONNAITRE = MEMORISER + INFERER**

**REPRESENTER = FORMALISER + RAISONNER**

Les sens des connecteurs ne veulent pas dire exactement la même chose que ceux du langage naturel. La capacité d'expression dans la représentation de la connaissance en logique des propositions est beaucoup moins riche qu'en langage naturel. Mais toutefois les connecteurs logique ont des correspondances ou "équivalents" dans la langue naturelle.

#### 2.5.2 Quelques orientations

##### ❖ La conjonction

**$P \wedge Q$**  Peut se traduire :

- P et Q
- Q et P
- à la fois P et Q
- P, Q
- P bien que Q
- P quoique Q
- P mais Q (sous-entendu mais aussi)
- Non seulement P mais Q
- P et pourtant Q
- P tandis que Q

##### ❖ La disjonction

**$P \vee Q$**  Peut se traduire :

- P ou Q

- ou P ou Q
- ou bien P ou bien Q
- soit P soit Q
- P à moins que Q
- P sauf si Q
- P ou Q ou les deux (OU inclusif)

### ❖ Le conditionnel

$P \rightarrow Q$  Peut se traduire :

- si P alors Q
- P condition suffisante de Q
- Q condition nécessaire de P
- P alors Q
- Q si P
- Q lorsque P
- P seulement si Q
- Q pourvu que P
- ...

### ❖ L'équivalence

$P \leftrightarrow Q$  Peut se traduire :

- P si et seulement si Q
- P si Q et Q si P
- P condition nécessaire et suffisante de Q
- ...

### 2.5.3 Algorithme de Modélisation

1-Choix de propositions atomiques et attribution de symboles propositionnels,  
2-Représentation des données du problème par des fbfs  
3-identification du problème à résoudre pour répondre au problème réel posé  
(satisfiabilité, équivalence, validité, etc.)

#### Exemple :

Par exemple soit à traduire le groupe de phrases suivantes :

Ahmed et Ali prirent le café et Mustapha fit de même.

## Cours-Logique Mathématique

---

Ahmed prit le café, et Ali ou Mustapha aussi

Ahmed et Ali ont dîné tous les deux, ou bien Ahmed et Mustapha prirent le café

Ahmed a dîné, ainsi que Mustapha ou Ali

Mustapha étudie bien à moins qu'il ne soit fatigué, auquel cas non

1. les propositions atomiques (Univers du discours) :

$a$  : Ahmed prend le café

$l$  : Ali prend le café

$m$  : Mustapha prend le café

$d$  : Ahmed a dîné

$e$  : Ali a dîné

$f$  : Mustapha a dîné

Etudie : Mustapha étudie bien

Fatigue : Mustapha est fatigué

2. Ensuite, pour chacune des phrases je vais écrire une formule bien formée à l'aide des propositions définies ci-dessus, des connecteurs et des parenthèses.

Ahmed et Ali prirent le café et Mustapha fit de même

**$a \wedge l \wedge m$**

Ahmed prit le café, et Ali ou Mustapha aussi

**$a \wedge (a \vee m)$**

Ahmed et Ali ont dîné tous les deux, ou bien Ahmed et Mustapha prirent le café

**$(d \wedge e) \vee (a \wedge m)$**

Ahmed a dîné, ainsi que Mustapha ou Ali

**$d \wedge (e \vee f)$**

Mustapha étudie bien à moins qu'il ne soit fatigué, auquel cas non

**$\neg ETUDE \leftrightarrow FATIGUE$**

### 2.5.4 Conclusion

Les ressemblances entre les connecteurs logiques et ceux de la langue naturelle sont limitées.

Il mange ou il dort et il dort ou il mange semblent synonymes

Par contre, il a faim et il mange n'est pas semblable à il mange et il a faim car le "et" a une connotation de causalité et de temps.

La porte est ouverte ou la porte est fermée. Le "ou" du français est parfois exclusif

La traduction est en général liée au contexte. Il peut aussi exister plusieurs traductions possibles.

### 2.6 Démonstrations

#### 2.6.1 Introduction

Comme nous l'avons dit au chapitre 1, la logique consiste à étudier la validité des raisonnements. Un **raisonnement** (ou inférence) est le passage d'une proposition ou d'un groupe de propositions (**prémisses**) à une nouvelle proposition (**conclusion**). *On va s'intéresser à montrer que les raisonnements sont valides.*

L'objectif de cette section est de commencer à aborder la question fondamentale suivante : qu'est-ce qu'une démonstration ?

Pour cela, plus précisément, on va se focaliser sur le problème suivant : on se donne une formule propositionnelle  $F$ , et on veut déterminer si  $F$  est une tautologie. Une tautologie est aussi appelée un théorème. On dit encore que  $F$  est valide. Cela va nous amener à décrire des algorithmes particuliers.

Une première méthode pour résoudre ce problème, qui est celle que nous avons utilisée dans la section 2.3, est la suivante : si  $F$  contient les variables propositionnelles  $p_1, p_2, \dots, p_n$ , on teste pour chacune des  $2^n$  interprétations de  $F$ .

**Remarque :** Cette observation peut sembler étrange et quelque part se contenter de peu, mais nous verrons que lorsque l'on considère des logiques plus générales, même très simples, cela devient problématique : il n'existe pas toujours d'algorithme pour déterminer si une formule  $F$  est une tautologie. Cependant, cette méthode est particulièrement inefficace. Elle a l'inconvénient majeur de garantir que lorsque  $F$  est une tautologie, on fera  $2^n$  fois un test du type "l'interprétation  $I$  est-elle un modèle de  $F$  ?" Lorsque  $n$  est grand,  $2^n$  explose très vite : si l'on peut effectivement programmer la méthode, le programme obtenu sera en pratique inutilisable, car prenant un temps énorme, dès que l'on considérera des formules  $F$  avec un grand nombre de variables.

Revenons alors à notre problème : on peut se dire que dans le raisonnement classique en mathématique, la méthode usuelle pour prouver qu'une assertion est un théorème est de la démontrer. Si l'on veut faire mieux que la méthode exhaustive précédente, il y a essentiellement deux angles d'attaque. Le premier angle d'attaque est de chercher à s'approcher de la notion de démonstration dans le raisonnement usuel : des méthodes de preuve dans l'esprit de la section suivante apparaissent. Le deuxième angle d'attaque est de chercher à produire des algorithmes plus efficace possible : des méthodes comme les preuves par résolution ou par tableaux

## Cours-Logique Mathématique

sémantiques. En général, on s'attend à ce qu'une méthode de preuve soit toujours valide : elle ne produit que des déductions correctes. Dans tous les cas se pose la question de la complétude : est-ce que tous les théorèmes (tautologies) sont prouvables ?

Dans ce qui suit, on présentera seulement trois systèmes de déduction valides et complets :

### 2.6.2 Preuve par les tables de vérité (méthode basée sur les modèles)

La preuve par la méthode de la table de vérité ou des tableaux matriciels permet de manière mécanique de dire si une formule est valide, consistante, inconsistante ou invalide .

1. Une fbf peut être à la fois invalide et consistante
2.  $G$  est valide si et seulement si  $(\neg G)$  est inconsistante
3.  $G$  est valide  $\rightarrow$  que  $G$  est consistante  
 $G$  est inconsistante  $\rightarrow$  que  $G$  est invalide
4. Il existe une procédure effective pour déterminer si une fbf est valide: la table de vérité

#### Exemple

Montrer que :  $(p \rightarrow (q \rightarrow p))$  est valide

$p$	$q$	$(q \rightarrow p)$	$(p \rightarrow (q \rightarrow p))$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

La fbf est vraie selon toutes les interprétations donc elle est valide

Montrer que :  $(p \vee \neg q) \wedge (\neg p \vee q)$  est satisfiable

$p$	$q$	$p \vee \neg q$	$\neg p \vee q$	$(p \vee \neg q) \wedge (\neg p \vee q)$
0	0	1	1	1
0	1	0	1	0
1	0	1	0	0
1	1	1	1	1

Il existe une interprétation dans laquelle la formule est vraie elle est donc **consistante**

**(satisfiable)**

Il existe une interprétation dans laquelle la formule est fausse elle est donc **invalid**

Remarquez qu'il n'est pas nécessaire de calculer la valeur de vérité pour toutes les interprétations (de remplir toutes les lignes de la table de vérité). Tant qu'on obtient que des vrais (respectivement que des faux), il faut calculer l'interprétation suivante, mais dès qu'on obtient un faux (respectivement un vrai) on s'arrête. En effet un vrai suffit pour prouver qu'une formule est consistante et un faux suffit pour prouver qu'elle est invalide.

- cette méthode a un gros inconvénient : elle demande d'examiner exhaustivement toutes les interprétations possibles et ce nombre d'interprétations croît très rapidement. En effet, pour une formule bien formée de 5 variables propositionnelles on aura 32 interprétations et donc 32 lignes à remplir. Son encombrement croît exponentiellement avec le nombre d'atomes.

De plus, elle est à peu près limitée au calcul des propositions, il est difficile de l'étendre aux autres logiques.

### 2.6.3 Preuve par les lois d'équivalences

Chacun des théorèmes d'équivalence (ou lois d'équivalence (*voir section 2.4.2*)) permet un acte d'inférence. Une autre manière de prouver qu'une formule bien formée est valide (respectivement inconsistante) consiste à raisonner en utilisant les lois d'équivalence. Un objectif est alors démontrer par équivalence que la formule bien formée de départ est équivalente à  $\top$  (respectivement à  $\perp$ ).

**Exemple :**

$$\begin{aligned}(p \rightarrow (q \rightarrow p)) &\equiv (p \rightarrow (\neg q \vee p)) && \mathbf{loi\ 1(voir\ 2.4.10)} \\ &\equiv (\neg p \vee (\neg q \vee p)) && \mathbf{loi\ 1} \\ &\equiv (\neg p \vee (p \vee \neg q)) && \mathbf{loi\ 3} \\ &\equiv ((\neg p \vee p) \vee \neg q) && \mathbf{loi\ 4} \\ &\equiv (\top \vee \neg q) && \mathbf{loi\ 8} \\ &\equiv \top && \mathbf{loi\ 6}\end{aligned}$$

## 2.6.4 La méthode de résolution en logique des propositions

### 2.6.4.1 Présentation

- Méthode simple constituée d'une règle (dite règle de résolution) permettant de produire une clause à partir de clauses existantes. Cette règle est appliquée itérativement jusqu'à aboutir à la clause vide
- Cette méthode permet de déterminer si une forme clausale est insatisfiable, elle permet donc de savoir si
  - Une fbf est insatisfiable (en passant à sa forme clausale)
  - Une fbf  $C$  est la conséquence logique d'un ensemble de fbf  $\{H_1, H_2, \dots, H_k\}$  (en passant à la forme clausale de  $(H_1 \wedge H_2 \wedge \dots \wedge H_k \wedge \neg C)$ )
- Cette méthode généralisée à la logique des prédicats et restreinte aux clauses de Horn est à la base du langage Prolog

### 2.6.4.2 Formes normales en logique propositionnelle

Les formes normales d'une formule bien formée permettent d'écrire la formule de départ sous une forme donnée avec les connecteurs logiques sont restreints et ordonnés selon un certain ordre.

- **Un littéral** est un atome (symbole propositionnel) ou sa négation. On parle de littéral positif (ex :  $p$ ) ou de littéral négatif (ex :  $\neg p$ ). On parle aussi de littéral opposé d'un littéral donné (ex :  $p$  et  $\neg p$  sont opposés).
- **Formes conjonctive et disjonctives :**  
Une fbf est dite sous **forme conjonctive** lorsqu'elle est composée d'une conjonction de disjonctions de littéraux.  
**Exemple :**  $(p \vee q) \wedge (\neg p \vee q \vee r)$
- Une fbf est dite sous **forme disjonctive** lorsqu'elle est composée d'une disjonction de conjonctions de littéraux  
**Exemple :**  $(\neg p \wedge \neg q) \vee (\neg q \wedge q \wedge \neg r) \vee (p \wedge \neg q)$
- Une fbf est mise sous forme normale conjonctive (FNC) si et seulement si elle est de la forme :  $F_1 \wedge F_2 \wedge \dots \wedge F_n$  où chaque  $F_i$  est une disjonction de littéraux
- Une fbf  $F$  est dite sous forme normale conjonctive (respectivement disjonctive) lorsqu'elle est sous forme conjonctive (respectivement disjonctive) telle que tous les symboles propositionnels (Ceux de  $F$  ou ceux de l'ensemble  $S$  considéré) apparaissent dans chaque disjonction (respectivement conjonction)

## Cours-Logique Mathématique

---

**Exemple** : FNC :  $(p \vee q \vee \neg r) \wedge (p \vee q \vee r) \wedge (\neg p \vee q \vee r)$

**Exemple** : FND :  $(\neg p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge \neg r)$

- Pour passer d'une forme conjonctive (respectivement d'une forme disjonctive) à une formule normale conjonctive (respectivement disjonctive) on introduit les symboles manquants à l'aide des équivalences suivantes :

$$D \equiv (D \vee p) \wedge (D \vee \neg p)$$

$$C \equiv (C \wedge p) \vee (C \wedge \neg p)$$

**Exemple** : FC =  $(p \vee q) \wedge (\neg p \vee q \vee r)$

$$(p \vee q \vee r) \wedge (\neg p \vee q \vee r)$$

$$\text{FNC} = (p \vee q \vee r) \wedge (p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r)$$

- Puisque le passage d'une forme conjonctive (respectivement disjonctive) à une forme normale conjonctive (respectivement forme normale disjonctive) est mécanique : plusieurs ouvrages utilisent les deux concepts (c.-à-d. FC et FNC) (respectivement disjonctive FD et FND) d'une manière interchangeable pour définir la même chose. Donc ils proposent tout simplement :

-Une fbf est mise sous forme normale disjonctive (FND) si et seulement si elle est de la forme :  $F_1 \vee F_2 \vee \dots \vee F_n$  où chaque  $F_i$  est une conjonction de littéraux

-Une fbf est mise sous forme normale conjonctive (FNC) si et seulement si elle est de la forme :  $F_1 \wedge F_2 \wedge \dots \wedge F_n$  où chaque  $F_i$  est une disjonction de littéraux

- Il faut noter que si la formule est un littéral il est à la fois sous forme normale disjonctive et conjonctive
- Pour toute formule  $F$  du calcul des propositions il existe une forme normale conjonctive FNC et une forme normale disjonctive FND telles que :

$$\vdash F \leftrightarrow \text{FNC} \quad \text{et} \quad \vdash F \leftrightarrow \text{FND}$$

Comment faire pour transformer une fbf en forme normale conjonctive ou disjonctive ?

Pour **mettre** une fbf sous forme normale conjonctive (respectivement disjonctive)

1. On élimine les connecteurs  $\rightarrow$  et  $\leftrightarrow$  en utilisant les théorèmes sur l'implication matérielle et l'équivalence matérielle
2. On développe le  $\neg$  en utilisant les lois de De Morgan et on élimine les  $\neg\neg$  par l'involution
3. On regroupe les  $\vee$  (respectivement les  $\wedge$ ) par distributivité

**Exemple** :



## Cours-Logique Mathématique

---

$$(p \wedge \neg (q \vee \neg r) \rightarrow (\neg q \rightarrow (p \wedge q)))$$

On élimine les  $\rightarrow$  et  $\leftrightarrow$  :  $\neg(p \wedge \neg (q \vee \neg r)) \vee (\neg \neg q \vee (p \wedge q))$

On développe le  $\neg$  et on élimine les  $\neg \neg$  :  $(\neg p \vee \neg \neg (q \vee \neg r) \vee (\neg \neg q \vee (p \wedge q)))$

$$(\neg p \vee (q \vee \neg r) \vee (q \vee (p \wedge q)))$$

On regroupe les  $\vee$  :  $\neg p \vee q \vee \neg r \vee q \vee (p \wedge q)$

$$\equiv (\neg p \vee q \vee \neg r \vee p) \wedge (\neg p \vee q \vee \neg r)$$

On obtient une forme normale conjonctive équivalente à la formule bien formée de départ.

**Remarque** : il est possible d'obtenir FND et FNC à partir d'une table de vérité :

– FND : on fait la disjonction des conjonctions des littéraux associés aux interprétations donnant la valeur 1 à la fbf

– FNC : on fait la conjonction des disjonctions des opposés des littéraux associés aux interprétations donnant la valeur 0 à la fbf

**Exemple :**

<b>p</b>	<b>q</b>	<b>F</b>
0	0	1
0	1	1
1	0	0
1	1	0

$$\mathbf{FND(F)} = (\neg p \wedge \neg q) \vee (\neg p \wedge q)$$

$$\mathbf{FNC(F)} = (\neg p \vee q) \wedge (\neg p \vee \neg q)$$

Nous allons voir les notions de clauses et de résolvant avant d'aborder le principe de résolution.

### 2.6.4.3 Clause

**Définition** : Une clause est une fbf qui a la forme d'une disjonction de littéraux

**Cas particulier** : un littéral isolé est une clause.

**Exemple de clauses :**

$$A \vee B \vee C \vee \neg D \quad \text{CLAUSE}$$

$$\neg D \quad \text{CLAUSE}$$

Une **clause** est une représentation ensembliste de disjonctions de littéraux

– Exemple :  $\{\neg p, q, \neg r, s\}$

## Cours-Logique Mathématique

- La sémantique d'une clause est complètement définie par l'ensemble des littéraux qui la composent
- Une infinité de disjonctions de littéraux peut être associée à une clause mais elles sont toutes sémantiquement équivalentes (idempotence, associativité et commutativité de la disjonction)
- On associe la proposition  $\perp$  à la clause vide (notée  $\emptyset$ )
- toute disjonction associée à une clause  $C = \{L_1, L_2, \dots, L_k\}$  est logiquement équivalente à  $((L_1 \vee L_2 \vee \dots \vee L_k) \vee \perp)$
- La sémantique de  $\emptyset$  est donc 0

### Propriétés des clauses

- Dans la suite, on emploie indifféremment le terme de clause pour parler de l'ensemble de littéraux ou d'une des fbf disjonctives que l'on peut lui associer
- Une clause est valide si elle contient un littéral et son opposé
- Une clause non vide ne contenant pas de littéraux opposés est contingente
- Seule la clause vide est insatisfiable
- Définition
- Une clause  $C$  subsume une clause  $C'$  si et seulement si  $C \sqsubseteq C'$
- Si  $C$  subsume  $C'$  alors  $C \models C'$
- Si  $C'$  non valide et  $C \models C'$  alors  $C$  subsume  $C'$

### 2.6.4.4 Forme clausale

**Définition :** Une forme clausale est une représentation ensembliste d'une forme conjonctive

-Exemple:  $\{\{p, q, \neg r\}, \{p, \neg s\}, \{\neg r, \neg s\}, \{q\}\}$

- La sémantique d'une forme clausale est complètement définie par l'ensemble des clauses qui la composent

- Une infinité de formes conjonctives peut être associée à une clause mais elles sont toutes sémantiquement équivalentes

• idempotence, associativité et commutativité de la conjonction

- On associe la proposition  $\top$  à l'ensemble vide de clause

- toute forme disjonctive associée à une forme clausale

$F = \{C_1, C_2, \dots, C_k\}$  est logiquement équivalente à  $((C_1 \wedge C_2 \wedge \dots \wedge C_k) \wedge \top)$

- La sémantique de l'ensemble vide de clause est donc 1

### Propriétés des formes clausales

• Soit  $F$  une forme clausale et soit  $C$  une clause valide de  $F$ , on a  $F \equiv F - \{C\}$

On peut donc éliminer les clauses tautologiques des formes clauseuses en conservant leur sémantique

• Soit  $F$  une forme clauseuse et soit  $C, C'$  (avec  $C \neq C'$ ) deux clauses de  $F$  tel que  $C$  subsume  $C'$ , on a  $F$  insatisfiable si et seulement si  $F - \{C'\}$  insatisfiable

On peut donc éliminer les clauses subsumées des formes clauseuses lorsque qu'on cherche à démontrer leur insatisfiabilité

### 2.6.4.5 Clause de Horn

• Une **clause de Horn** est une clause ayant au plus un littéral positif

– Exemple :  $\{\neg p, q, \neg r, \neg s\}$  ou  $\{\neg r, \neg s\}$

• On appelle *règle de Horn*, une fbf composée d'une implication entre deux conjonctions de littéraux positifs

– Exemple :  $(p_1 \wedge p_2 \dots \wedge p_n) \rightarrow (c_1 \wedge c_2 \dots \wedge c_n)$

#### Propriété

« La forme clauseuse d'un ensemble de règles de Horn ne contient que des clauses de Horn »

### Comment obtenir à partir d'une formule bien formée un ensemble de clauses ?

Il faut d'abord transformer la formule en sa forme normale conjonctive et ensuite éliminer les connecteurs  $\wedge$ . On obtient ainsi un ensemble  $S$  de clauses.

### 2.6.4.6 Clause résolvente ou règle de résolution

Si  $C1 = \{p, l_1, l_2, \dots, l_k\}$  et  $C2 = \{\neg p, m_1, m_2, \dots, m_n\}$  sont deux clauses ayant des littéraux opposés (au moins  $p$  et  $\neg p$ ), la résolvente de  $C1$  et  $C2$  selon  $p$  est la clause

$C = Res(C1, C2, p) = \{l_1, l_2, \dots, l_k, m_1, m_2, \dots, m_n\}$  Obtenue par union des clauses restantes

#### Exemple

Soient  $C1$  et  $C2$  les deux clauses suivantes :

**C1** :  $e1 \vee e2$  ou **C1** =  $\{e1, e2\}$

**C2** :  $\neg e2 \vee e3$  ou **C2** =  $\{\neg e2, e3\}$

La résolvente de **C1** et **C2** est  $Res(\mathbf{C1}, \mathbf{C2}) = \mathbf{C} = \{e1, e3\}$

Soient **C1** et **C2** les deux clauses suivantes :

**C1** :  $p$       **C2** :  $\neg p$

Le résolvant de **C1** et de **C2** est  $\mathbf{C} = \phi$  (**la clause vide**)

La clause Faux est notée  $\phi$  c'est la **clause vide**

## Propriétés

-Si plusieurs résolvantes peuvent être calculées à partir de deux clauses  $C_1$  et  $C_2$  alors ces résolvantes sont logiquement équivalentes et valides

-La résolvante  $C$  de deux clauses  $C_1$  et  $C_2$  est une conséquence logique de  $C_1$  et de  $C_2$  (c.-à-d.  $\{C_1, C_2\} \models C$ ). La règle de résolution est une règle d'inférence.

### 2.6.4.7 Séquence de résolution

Soit  $F_c$  une forme clausale et  $C$  une clause, On appelle **déduction** (ou résolution) d'une clause  $C$  à partir de  $F_c$  ( $F_c \vdash_{Res} C$ ) si et seulement s'il existe une séquence finie de clauses  $(C_1, C_2, \dots, C_r)$

telle que :

- $C_r = C$  et
- $C_i$  soit une clause de  $F_c$  soit une résolvante des clauses précédentes  
Pour tout  $i = 1, \dots, r$  :
  - $C_i$  est dans  $F$  ou
  - $C_i$  est une résolvante de deux clauses avant  $C_i$  dans la séquence ( $\exists j < i$  et  $\exists k < i$ ,  $C_i$  étant une résolvante de  $C_j$  et  $C_k$ )

### 2.6.4.8 Synthèse et Algorithme de résolution

Soit  $F_c$  une forme clausale, s'il existe une déduction de la clause vide à partir de  $F_c$  alors  $F_c$  est insatisfiable ( **$F_c$  est insatisfiable si et seulement si  $F_c \vdash_{Res} \emptyset$** )

On appelle **réfutation** la déduction de la clause vide  $\emptyset$  à partir de  $F_c$

Donc, montrer qu'une formule bien formée  $F$  est valide :

- C'est équivalent à montrer que  $\neg F$  est insatisfiable
- C'est aussi équivalent à montrer que  $F_c \vdash_{Res} \emptyset$  (montrer qu'il existe une déduction de la clause vide  $\emptyset$ )

Ce qui est résumé par l'algorithme suivant :

#### Algorithme résolution

##### Entrées

$F$  : fbf ;

##### Sortie

Valide ou non ;

**Début**

Obtenir  $\neg F$  ;

Mettre  $\neg F$  sous forme clausale  $F_C$  ;

**Tant que** la clause vide n'est pas rencontrée et qu'il existe des paires de clauses réductibles **faire**

**Début**

Chercher des clauses résolvantes ;

Ajouter la clause résultat à la forme causale et éliminer les deux clauses réductibles ;

**Fintantque ;**

**Si** on trouve la clause vide **alors** F est valide

**sinon** F est invalide

**Finsi ;**

**Fin ;**

**Exemple**

Soit  $F_C = \{\neg p \vee \neg q \vee r, \neg r, p, \neg t \vee q, \neg t\}$

$F_C$  est la forme clausale d'une fbf  $\neg F$  .

Montrons que cet ensemble est insatisfiable

$C1 = \neg p \vee \neg q \vee r$     $C2 = \neg r$     $C3 = p$     $C4 = \neg t \vee q$     $C5 = t$

$Res(C1, C2) = \neg p \vee \neg q = C6$

$Res(C6, C3) = \neg q = C7$

$Res(C7, C4) = \neg t = C8$

$Res(C8, C5) = \emptyset = C9$

A partir de la forme clausale  $F_C$  de la fbf  $\neg F$  on a déduit la clause vide donc on peut conclure que  $\neg F$  est insatisfiable et donc que  $F$  est valide.

**Exemple**

Soit  $H_C = \{C1, C2, C3, C4, C5, C6\}$  Avec  $H_C$  est la forme clausale d'une fbf  $H$  et

$C1 = \{d\}$     $C2 = \{c, \neg d\}$     $C3 = \{\neg d\}$     $C4 = \{a, e\}$     $C5 = \{\neg c, \neg e\}$     $C6 = \{\neg a, d\}$

Montrons que H est insatisfiable

$Res(C1, C2) = \{c\} = C7$

$$Res(C7, C5) = \{\neg e\} = C8$$

$$Res(C8, C4) = \{a\} = C9$$

$$Res(C9, C6) = \{d\} = C10$$

$$Res(C10, C3) = \emptyset$$

A partir de la forme clausale  $H_C$  de la fbf  $H$  on a déduit la clause vide donc on peut conclure que  $H$  est insatisfiable.

### 2.7 Les limites de la logique des propositions

*La logique des propositions est limitée* ; en effet dès que l'on veut manipuler des propriétés générales un peu complexes, des relations entre des objets, on peut aussi avoir besoin de quantifier en exprimant "Tous les hommes sont mortels". Rien ne nous permet de faire cela en logique des propositions.

#### Exemple

Si l'on souhaite exprimer que "*Ahmed est un Musulman*", on a la possibilité de donner la proposition suivante : *Ahmed\_Musulman*.

Si maintenant on souhaite exprimer que "*Ali est un Musulman*", on a la proposition *Ali\_Musulman*.

On se trouve avec deux assertions distinctes et aucune similitude entre *Ahmed* et *Ali*.

On souhaiterait une meilleure représentation qui nous permettrait de dire que *Ahmed* et *Ali* sont tous les deux des *Musulmans* dans la religion :

*Musulman (Ahmed), Musulman (Ali)*

Le prédicat est le concept qui pallie ce problème. Il exprime une relation dans un contexte.

#### Exemple

"Quelqu'un a demandé quelque chose "

On peut définir le prédicat : *DEMANDE* avec deux arguments ou termes : celui qui demande et ce qu'il demande.

*DEMANDE(ALI, LOGEMENT)*

Quand on veut traduire la phrase "La voiture est verte", on a plusieurs possibilités quant au choix du prédicat et de son arité :

*EST\_VERTE(voiture)*

ou *COULEUR(voiture,verte)*

ou *VALEUR(couleur, voiture,verte)*

Si on a besoin d'exprimer des relations ou propriétés, des fonctions, la logique des propositions ne le permettent pas.

## Cours-Logique Mathématique

---

Aussi, le calcul des prédicats possède la notion de fonction.

**Exemple :**

"Le frère de ALI travaille avec le frère de MOHMED" peut se traduire par le prédicat TRAVAILLER avec deux arguments et par le symbole de fonction frère :

TRAVAILLER(frère(ALI),frère(MOHAMED))

## 2.8 Exercices du chapitre 2 : logique des propositions

### 2.8.1 Exercices : Syntaxe de la logique des propositions

#### *Exercice 2.8.1.1*

Décidez si les phrases suivantes sont des propositions ou non:

- (a) Quel est votre nom?
- (b) Fermez la porte!
- (c) L'herbe est verte.
- (d) L'herbe est rouge.
- (e) C'est faux.
- (f) Je suis honnête.
- (g) Vous ne devez pas tricher.

#### *Exercice 2.8.1.2*

Pour s'exercer essayez de déterminer si les formules suivantes appartiennent à la logique des propositions.

Soient A, B, C, D des fbfs

- a)  $((A \vee (\neg B)) \wedge (C \vee D))$
- b)  $(A \vee B)(\wedge \vee C)$

#### *Exercice 2.8.1.3*

Combien de propositions différentes peut-on écrire en ajoutant convenablement des parenthèses à

$$p \wedge q \rightarrow r \wedge p$$

#### *Exercice 2.8.1.4*

Dessiner  $arb((a \wedge \neg b) \wedge a)$

#### *Exercice 2.8.1.5*

Définir  $arb^{-1}$  la fonction qui a une arbor. associe une fbf

#### *Exercice 2.8.1.6*

Considérez les formules du calcul propositionnel suivantes :

$$F1 = \neg((q \rightarrow r) \vee p)$$

$$F2 = (((p \wedge q) \neg r) \rightarrow p)$$

$$F3 = ((\neg p \rightarrow (\neg r \vee q)) \rightarrow p)$$

$$F4 = (\neg \neg(q \rightarrow r) \vee (\neg q \rightarrow \neg r))$$

Pour chaque formule



## Cours-Logique Mathématique

---

1. dessinez son arbre syntaxique (si la formule n'est pas bien formée vous proposerez une correction);
2. une expression fonctionnelle préfixée de la formule
3. une expression infixée débarrassée des parenthèses inutiles si on considère un ordre de priorité pour les connecteurs ( $\neg$  : plus prioritaire, puis  $\wedge$  et  $\vee$  enfin  $\rightarrow$  et  $\leftrightarrow$  moins prioritaires).
4. énumérez ses sous-formules;
5. énumérez les symboles propositionnels avec leur occurrence.

### Exercice 2.8.1.7

Donner tous les arbres syntaxiques possibles des formules non parenthésées :

$$p \wedge q \vee r$$

$$\neg p \wedge q$$

$$\neg p \rightarrow \neg q \wedge r$$

### Exercice 2.8.1.8

Dites si les formules suivantes sont des sous-formules de la formule :

$$((\neg p \rightarrow (\neg q \wedge r)) \leftrightarrow ((s \rightarrow (\neg p \wedge r) \leftrightarrow (\neg q \wedge s)))$$

$$\neg p ; p \rightarrow q ; p \rightarrow q \wedge r ; p \rightarrow (q \wedge r) ; \neg p \wedge r ; (s \rightarrow (\neg p \wedge r)) \leftrightarrow (\neg q \wedge s)$$

### Exercice 2.8.1.9

Donner une définition par induction du langage des formules bien formées de la logique des propositions étendu au symbole  $\top$  vrai et au connecteur « ou exclusif »  $\otimes$  ;

### Exercice 2.8.1.10

Soit  $\Sigma$  l'ensemble des symboles propositionnels et  $PROP$  l'ensemble des fbfs construites à partir de  $\Sigma$  et soit A et B les fbf suivantes :

$$A = (\neg p \rightarrow (p \rightarrow (r \wedge p)))$$

$$B = (((p \wedge \perp) \vee (\neg r \rightarrow p)) \leftrightarrow r)$$

a) Dessiner l'arbre syntaxique associé à A et b

b) On définit par induction structurale l'application SymbProp de PROP dans  $P(\Sigma)$  par :

Si p est un symbole propositionnel alors  $SymbProp(p) = \{p\}$

$$SymbProp(\perp) = \emptyset$$

$$SymbProp(\neg p) = SymbProp(p)$$

Pour tout connecteur binaire  $C_B \in \{ \wedge, \vee, \rightarrow, \leftrightarrow \}$ ,

$$SymbProp(F1 C_B F2) = SymbProp(F1) \cup SymbProp(F2)$$

## Cours-Logique Mathématique

Calculer les ensembles  $SymbProp(A)$  et  $SymbProp(B)$

- c) Soit l'application  $prof$  de PROP dans  $\mathbb{N}$  qui à toute fbf  $F$ , associe la profondeur de l'arbre syntaxique correspondant à  $F$  (c.-à-d. le nombre maximum d'arcs d'un chemin dans cette arborescence). Donner une définition par induction structurale de l'application  $Prof$ .
- d) Calculer  $Prof(A)$  et  $Prof(B)$ .

### Exercice 2.8.1.11

Soit l'application  $Sub$  de PROP dans  $P(PROP)$  qui à toute fbf  $F$  associe l'ensemble des sous formules de  $F$ , soit l'application  $nbc$  de PROP dans  $\mathbb{N}$  qui associe le nombre d'occurrences de connecteur dans  $F$  et soit la fbf

$$A = ((p \wedge q) \leftrightarrow (\neg r \vee p))$$

- a) Donner  $Sub(A)$  et  $nbc(A)$ , puis vérifier que  $|Sub(A)| \leq 2 * nbc(A) + 1$
- b) Donner une définition par induction de  $Sub$  et  $nbc$
- c) Montrer que pour tout entier naturel  $n$  une fbf  $F$  ayant  $n$  occurrences de connecteurs a au plus  $2n + 1$  sous fbf (autrement dit pour toute fbf  $F$ :  $|Sub(F)| \leq 2 * nbc(F) + 1$ )

## 2.8.2 Exercices : Sémantique de la logique des propositions

### Exercice 2.8.2.1

Soit  $I(p)=0$  et  $I(q)=1$ , calculer la valeur de vérité de  $\neg((p \vee (q \leftrightarrow p)) \rightarrow q)$

### Exercice 2.8.2.2

Entraînez-vous à développer la négation en appliquant les lois de De Morgan

- a)  $\neg(a \wedge (b \vee c))$
- b)  $\neg[(\neg(a \wedge b) \vee (\neg d)) \wedge e \vee f]$
- c)  $\neg((\neg a) \wedge b \wedge ((\neg c) \vee d) \vee (\neg e) \wedge f \wedge (\neg g))$
- d)  $\neg(a \vee (\neg b) \vee c) \wedge ([\neg(\neg d)] \vee (\neg e))$

### Exercice 2.8.2.3

1. Quelles sont les interprétations qui donnent même la valeur à  $p \wedge q$  et  $p \rightarrow q$  ?
2. Énumérez les modèles de la formule  $(p \wedge q) \leftrightarrow (p \rightarrow q)$
3. Est-ce que cette formule est (in)satisfaisable, valide ?

### Exercice 2.8.2.4

On considère les formules

$$A = p \wedge (\neg q \rightarrow (q \rightarrow p))$$

## Cours-Logique Mathématique

---

$$B = (p \vee q) \leftrightarrow (\neg p \vee \neg q)$$

1. Soit I une interprétation. Déterminer, si possible,  $V(A,I)$  et  $V(B,I)$  dans chacun des quatre cas suivants :
  - a) on sait que  $I(p) = 0$  et  $I(q) = 1$ ;
  - b) on sait que  $I(p) = 0$ ;
  - c) on sait que  $I(q) = 1$ ;
  - d) on ne sait rien sur  $I(p)$  et  $I(q)$ .
2. Ces deux formules sont-elles satisfaisables ? Des tautologies ?
3. L'ensemble  $\{A,B\}$  est-il consistant ? C'est-à-dire, existe-t-il une interprétation I telle que  $V(A,I)=V(B,I) = 1$  ?

### *Exercice 2.8.2.5*

Proposez une formule F ayant la table de vérité suivante :

p	q	r	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

### *Exercice 2.8.2.6*

La sémantique d'une forme est déterminée par celle des symboles propositionnels qui la composent.

-Combien d'interprétations différentes peut-on donner aux symboles propositionnels de chaque formule suivante :

-Pour chaque formule donner la valeur de vérité de la formule

$$p \vee \neg p \quad ; \quad p \wedge p \quad ; \quad p \wedge \neg q \quad ; \quad p \vee q \quad ; \quad p \vee (q \vee \neg q) \quad ;$$

$$(p \rightarrow \neg\neg q)(\neg\neg p \rightarrow q) \quad ; \quad (\neg p \vee q) \leftrightarrow (p \rightarrow q)$$

## Cours-Logique Mathématique

---

### Exercice 2.8.2.7

Démontrer pour chaque couple de formules suivantes qu'elles sont "sémantiquement équivalentes"

(C'est-à-dire que leur valeur de vérité est identique quelque soit l'interprétation des symboles propositionnels) :

$$\begin{aligned} p \text{ et } \neg\neg p & ; \quad p \vee \neg \perp \text{ et } \neg \perp & ; \quad p \rightarrow q \text{ et } \neg p \vee q & ; \\ \neg(p \wedge q) \text{ et } \neg p \vee \neg q & ; (p \rightarrow q) \wedge (q \rightarrow p) \text{ et } p \leftrightarrow q & ; \\ p \wedge (q \vee r) \text{ et } (p \wedge q) \vee (p \wedge r) & ; p \wedge \neg p \text{ et } \perp \end{aligned}$$

### Exercice 2.8.2.8

Dite parmi les formules suivantes lesquelles sont équivalentes en vous aidant des équivalences

de base données en cours :  $F1 = (p \wedge q) \rightarrow r$  ;  $F2 = (p \rightarrow r) \wedge (q \rightarrow r)$  ;  
 $F3 = (p \rightarrow r) \vee (q \rightarrow r)$  ;  $F4 = (p \vee q) \rightarrow r$  ;  $F5 = p \rightarrow (q \wedge r)$  ;  
 $F6 = p \rightarrow (q \vee r)$  ;  $F7 = (p \rightarrow q) \wedge (p \rightarrow r)$  ;  $F8 = (p \rightarrow q) \vee (p \rightarrow r)$

### Exercice 2.8.2.9

Dite si les formules suivantes sont valides,(in)satisfiable , ou contingentes :

$$F1 = ((p \rightarrow q) \rightarrow p) \quad ; \quad F2 = (p \rightarrow (q \rightarrow p)) \quad ; \quad F3 = (p \wedge q) \leftrightarrow (p \rightarrow \neg q)$$

### Exercice 2.8.2.10

Que pensez-vous sur les affirmations suivantes :

- Si une formule est contingente, sa négation l'est également
- Si  $G$  et  $H$  sont deux formules contingentes alors  $G \wedge H$  et  $G \vee H$  sont deux formules contingentes
- Si  $G \vee H$  est insatisfiable alors  $G$  et  $H$  sont deux formules insatisfiables
- Si  $G \vee H$  est valide alors  $G$  et  $H$  sont deux formules valides

### Exercice 2.8.2.11

-Soit le connecteur *nand* défini par  $p \text{ nand } q \equiv \neg(p \wedge q)$  . Montrer que toute formule est équivalente à une formule n'ayant que le *nand* comme seul connecteur.

-Montrer que toute formule est équivalente à une formule n'ayant que la négation et l'implication comme connecteurs logiques.

### Exercice 2.8.2.12

Soient  $F$  et  $G$  deux formules sans variable propositionnelle commune.

-Montrer que si  $F \rightarrow G$  est une tautologie, alors l'une au moins des formules :  $F$  et  $G$  est une tautologie.

### 2.8.3 Exercices : Modélisation en logique des propositions

#### *Exercice 2.8.3.1*

Traduire les groupes de phrases suivantes en logique des propositions :

A : Je vous paierai votre installation de T.V. seulement si elle marche,

B : Or votre installation ne marche pas.

C : Donc je ne vous paierai pas.

#### *Exercice 2.8.3.2*

On considère les propositions suivantes :

- Si Ahmed est rentré chez lui, alors Ali est allé à la mosquée.
- Aicha est à l'école ou Ahmed est rentré chez lui.
- Si Ali est allé à la mosquée, alors Aicha est à l'école ou Ahmed est rentré chez lui.
- Aicha n'est pas à l'école et Ali est allé à la mosquée.
- Ahmed est rentré chez lui.

Formaliser ces propositions en logique des propositions. On notera A, B, C, D, E les cinq formules obtenues (en suivant l'ordre de l'énoncé)

#### *Exercice 2.3.8.3*

Le lieutenant Colombo enquête sur le crime dans une usine ayant eu lieu à l'une usine dans la nuit du 13 au 14 juillet. il dispose des informations suivantes :

A : le Chauffeur ou le Gardien est coupable

B : si le Gardien est coupable alors le crime a eu lieu avant minuit

C : le crime n'a pas eu lieu avant minuit

-Que peut-il en déduire sur l'identité du (ou des) coupable(s) ?

#### *Exercice 2.3.8.4*

Modéliser les énoncés suivants en logique des propositions :

- Il suffit à Ali d'assister aux cours et aux TD pour qu'il ait la moyenne
- R est une relation d'équivalence si et seulement si R est réflexive, symétrique et transitive
- Si Ahmed est chez lui, il lit ou il écoute de la musique
- Le sida ne sera éradiqué à moins qu'un nouveau vaccin ne soit découvert
- Il est nécessaire d'avoir du courage et de l'habilité pour escalader cette paroi.

### Exercice 2.3.8.5

On cherche à deviner la position d'un certain nombre de bateaux sur une grille de bataille navale possédant 2 lignes, appelées a et b et trois colonnes 1,2,3 pour cela on dispose des informations suivantes :

- A. Il y a au moins un bateau sur la ligne b
- B. Il y a au moins un bateau sur la ligne a
- C. Il n'y a pas 2 bateaux sur une même colonne
- D. Il n'y a pas de bateau en (b,1)
- E. S'il y a un bateau sur la ligne a alors il n'y a pas de bateau en (b,3)

En notant  $x_i$  (pour  $x=a$  ou  $b$  et  $i=1$  ou  $2$  ou  $3$ ) l'affirmation « il y a un bateau à la position  $x_i$ , modélisez par une formule de la logique des propositions les 5 affirmations ci-dessus. Puis traduisez sous forme logique le problème posé.

### 2.8.4 Exercices : Démonstration en logique des propositions

#### Exercice 2.8.4.1

Que pouvez-vous conclure sur les formules bien formées suivantes en utilisant les tables de vérité ?

- a)  $(p \rightarrow q) \rightarrow ((p \rightarrow \neg q) \rightarrow \neg p)$
- b)  $(p \rightarrow (q \rightarrow (p \wedge q)))$
- c)  $((p \wedge q) \rightarrow r) \wedge (p \rightarrow q) \rightarrow (p \rightarrow r)$
- d)  $\neg p \rightarrow (\neg(p \rightarrow q))$
- e)  $((p \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P))$
- f)  $p \wedge \neg(P \wedge \neg Q \wedge (Q \vee R))$

#### Exercice 2.8.4.2

Que pouvez-vous conclure sur la formule bien formée en utilisant les théorèmes d'équivalence vus à la section 2.4.10 ?

- a)  $p \rightarrow (q \rightarrow (p \wedge q))$
- b)  $((p \wedge q) \rightarrow r) \wedge (p \rightarrow q) \rightarrow (p \rightarrow r)$
- c)  $\neg p \wedge (\neg(p \rightarrow q))$
- d)  $(p \vee \neg q) \wedge (\neg p \vee q)$
- e)  $(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$
- f)  $p \wedge \neg(p \wedge \neg q) \wedge (q \vee r)$

#### Exercice 2.8.4.3

1) Mettre sous forme normale disjonctive

a)  $p \rightarrow ((q \wedge r) \rightarrow s)$

b)  $(p \wedge q) \rightarrow r$

2) Mettre sous forme normale conjonctive

a)  $(p \wedge \neg q) \vee (\neg p \wedge q)$

b)  $\neg(p \rightarrow q) \vee (p \wedge q)$

### **Exercice 2.8.4.4**

Trouver la clause résolvante dans les cas suivants :

a)  $C1 = \neg q \vee p$      $C2 = r \vee \neg p \vee s$

b)  $C1 = \neg q \vee p$      $C2 = q$

c)  $C1 = \neg p \vee \neg q$      $C2 = p \vee s \vee \neg r$

d)  $C1 = p \vee q$      $C2 = r \vee p$

### **Exercice 2.8.4.5**

Montrer dans les cas suivants que F est valide :

a)  $((p \rightarrow q) \wedge (p \vee r)) \rightarrow (q \vee r)$

b)  $((p \rightarrow q) \wedge (p \rightarrow r)) \rightarrow (q \wedge r)$

c)  $((p \rightarrow q) \rightarrow (r \rightarrow ((q \rightarrow r) \rightarrow p)))$

## 3. La logique des prédicats

### 3.1 Introduction

Dans la logique des propositions on considère une proposition comme un tout, représenté par une variable, dont on ne détaille pas le contenu. On ne s'intéresse qu'à la vérité ou à la fausseté d'une proposition. La logique des prédicats, également appelée logique du premier ordre, regarde les propositions de plus près. Dans l'analyse des propositions élémentaires en langue naturelle on peut distinguer

- un sujet (ce dont on parle) et d'éventuels compléments
- un prédicat (ce qu'on dit à propos du sujet et de ses compléments)

Par exemple, dans la proposition « Ahmed est grand » on a

- un sujet : Ahmed
- le prédicat : est grand

Dans la proposition « Aicha mange une pomme » on a

- un sujet Aicha et un complément pomme
- le prédicat : mange

On pourrait réécrire les propositions précédentes sous une forme qui met en évidence le prédicat, soit :

Est\_grand(Ahmed)

mange(Aicha, pomme)

Suivant ce modèle, la logique\* des prédicats représente les propositions élémentaires (atomiques) sous la forme :

nom-prédicat(objet1, objet2, . . .)

où objet1, objet2, . . . sont les objets sur lesquels porte le prédicat (le sujet et ses éventuels compléments)

En logique des prédicats les variables représentent non pas des propositions mes des objets sur lesquels portent les prédicats. L'introduction de variables permet de formuler deux types d'énoncés.

**Les énoncés universels.** Dans ce cas les variables représentent tous les objets d'un domaine. Comme dans Si X est une girafe alors X est un animal.

**Les énoncés existentiels.** Ici une variable représente un objet particulier du domaine, qu'on le connaisse précisément ou non. Comme dans

Il y a au moins un nombre entier Z tel que  $Z > 3$  et  $Z^2 < 30$ .

Contrairement aux variables, les constantes représentent des objets particulier et connus : 1642 (un nombre entier), Ahmed (une personne particulière), etc. Enfin, la notion de fonction



## Cours-Logique Mathématique

---

correspond à la notion habituelle de fonction qui associe à un ou plusieurs objets (les paramètres) une valeur (le résultat de la fonction). Par exemple  $age(x)$  associe à une personne  $x$  un nombre entier (son âge) ou  $distance(x, y)$  qui fournit la distance entre deux villes  $x$  et  $y$ .

### 3.2 Syntaxe de la logique des prédicats

Par rapport au calcul propositionnel, on rajoute des variables, des quantificateurs, des relations (Prédicats) et des fonctions.

#### 3.2.1 L'alphabet

L'alphabet est constitué :

- de **connecteurs** :  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$  qui se lisent respectivement non, et, ou, implique et équivalent.
- de **délimiteurs** : les parenthèses ( )
- des deux constantes propositionnelles  $V$  (vrai) et  $F$  (faux)
- de constantes (minuscules de l'alphabet latin et les concaténations de telles lettres)  
 $C = \{a, b, c, \dots, z, aa, \dots\}$
- de **variables** (minuscules de l'alphabet latin, et les concaténations de telles lettres)  
 $V = \{a, b, c, \dots, z, aa, \dots\}$
- de **prédicats** (majuscules P)

**Définition Prédicat** : Un prédicat est une propriété ou relation qui porte sur un ou plusieurs éléments d'un domaine D. C'est une fonction de D dans  $\{V, F\}$  (ou  $\{0, 1\}$ ).

- L'**arité** d'un prédicat est le nombre d'arguments du prédicat. C'est un nombre positif
- Si le **prédicat** est d'arité 0, il correspond à la notion de **proposition** de la logique des propositions
- de **fonctions** (en minuscules: f, g, successeur). Chaque symbole de fonction a une arité fixée.
- L'**arité** d'une fonction est le nombre d'argument de la fonction. C'est un nombre positif. Si la fonction est d'arité 0, elle correspond à la notion de constante.
- **quantificateurs**  
 $\exists$  prononcé "il existe" est le quantificateur existentiel  
 $\forall$  prononcé "quel que soit" est le quantificateur universel

### 3.2.2 Les termes

Par définition, tout terme est engendré par application des deux lois suivantes

- constantes et variables sont des termes
- si  $f$  est un symbole de fonction d'arité  $n$  ( $n \geq 1$ ) et si  $t_1, t_2, \dots, t_n$  sont des termes alors  $f(t_1, t_2, \dots, t_n)$  est un terme

**Exemple :**

Soient  $x$  une variable,  $b$  une constante et successeur et poids des fonctions :

$x, b$  sont des termes

**successeur**( $x$ ) est un terme

**poids**( $b$ ) est un terme

**successeur**(**poids**( $b$ )) est un terme

### 3.2.3 Les atomes (Formule atomique)

Par définition, tout atome est engendré par application de la loi suivante :

- si  $P$  est un prédicat d'arité  $n$  ( $n \geq 1$ ) et si  $t_1, t_2, \dots, t_n$  sont des termes alors  $P(t_1, t_2, \dots, t_n)$  est un atome

**Exemple :**

Soient  $(x, y, z)$  une variable, ( $bleu, table, fenetre$ ) des constantes et (successeur et appui) des fonctions et ( $P(x, y), ENTRE(x, y, z), VIDE$ ) des prédicats:

**P**( $x, bleu$ ) est un atome

**VIDE** est un atome

**ENTRE**( $table, x, appui(fenetre)$ ) est un atome

**successeur**( $x$ ) n'est pas un atome

**appui**( $fenetre$ ) n'est pas un atome

### 3.2.4 Les formules bien formées

Le langage est constitué de l'ensemble des Formules Bien Formées (appelées aussi : FBFs ou Well Formed-Formula WFF) ou expressions bien formées défini comme suit :

- Les atomes sont des fbfs
- si  $F$  et  $G$  sont des fbfs alors  $(\neg G), (F \wedge G), (F \vee G), (F \rightarrow G)$  et  $(F \leftrightarrow G)$  sont des fbfs
- si  $G$  est une fbfs et  $x$  une variable alors  $(\exists x)G$  et  $(\forall x)G$  sont des fbfs.
- toutes les fbfs sont obtenues par application des 3 règles ci-dessus.

Plus formellement :

- Soit  $L$  un langage,  $V$  un ensemble infini de variables,  
 $C = \{ \top, \perp, \neg, \wedge, \vee, \rightarrow, \leftrightarrow \}$  les connecteurs,  $Q = \{ \exists, \forall \}$  les quantificateurs et  $D = \{ (, ) \}$  un jeu de parenthèses

- On définit par induction  $F(L)$ , l'ensemble des formules bien formées, construites sur  $L$  :

**(base)**  $F(L)$  contient  $\{P(t_1, \dots, t_n) \mid (P \in \text{Pred}) \text{ est un prédicat } n\text{-aire et } t_1, \dots, t_n \text{ sont des termes}\}$

$P(t_1, \dots, t_n)$  est une formule atomique ou atome

$[F(L) \text{ contient } \{ \top, \perp \}]$

**(Construction)** si  $A$  et  $B \in F(L)$  et si  $x \in V$  :

**Règle 1** :  $\neg A \in F(L)$

**Règle 2** :  $(A C_B B) \in F(L)$

**Règle 3** :  $\forall x A \in F(L), \exists x A \in F(L)$

$C_B = \{ \wedge, \vee, \rightarrow, \leftrightarrow \}$

## Exemple

- $(\exists x)(\forall y)(P(x, y) \vee Q(x, y) \rightarrow R(x))$
- et  $(\emptyset, (P(a) \rightarrow P(b))) \rightarrow \emptyset, P(b)$  sont des fbfs
- $(\emptyset, (f(a)))$  et  $(f(P(a)))$  ne sont pas des fbfs

Il est possible d'instaurer un ordre de priorité des connecteurs

(Le plus souvent : Le plus prioritaire  $\neg, \exists, \forall, \wedge, \vee, \rightarrow, \leftrightarrow$ )

### 3.2.5 Variable libre et liée

Champ ou portée d'un quantificateur = la fbf sur laquelle il s'applique

$$\forall x \exists y (P(x, y) \wedge \exists z Q(x, y))$$

Le champ de  $\forall x$  est  $\exists y (P(x, y) \wedge \exists z Q(x, y))$

Le champ de  $\exists y$  est  $(P(x, y) \wedge \exists z Q(x, y))$

Le champ de  $\exists z$  est  $Q(x, y)$

Une occurrence de  $x$  est liée dans une fbf si elle est dans le champ d'un quantificateur  $\forall$  ou  $\exists$  qui l'utilise ou si elle le suit. Sinon cette occurrence est dite libre.

Inductivement :

- Ensemble **VarLib** des variables libres (les variables non quantifiées) d'une fbf  $F$ , se définit de façon inductive par :

## Cours-Logique Mathématique

---

(base) Si  $F$  est un atome,  $VarLib(F) = Var(F)$ .

(cons) Règle 1 : Si  $F = \neg A$ ,  $VarLib(F) = VarLib(A)$ .

Règle 2 : Si  $F = (A C_B B)$ ,  $VarLib(F) = VarLib(A) \cup VarLib(B)$

$\{C_B = \neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$

Règle 3 : Si  $F = \forall x A$  (ou  $F = \exists x A$ ),  $VarLib(F) = VarLib(A) - \{x\}$

- Ensemble **VarLiées** des variables liées (les variables quantifiées) d'une fbf  $F$ , se définit de façon inductive par :

(base) Si  $F$  est un atome,  $VarLiées(F) = \emptyset$

(cons)

Règle 1 : Si  $F = \neg A$ ,  $VarLiées(F) = VarLiées(A)$ .

Règle 2 : Si  $F = (A C_B B)$ ,  $VarLiées(F) = VarLiées(A) \cup arLiées(B)$

$\{C_B = \neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$

Règle 3 : Si  $F = \forall x A$  (ou  $F = \exists x A$ ),  $VarLiées(F) = VarLiées(A) \cup \{x\}$

### Exemple

$A = \forall x (\exists y P(x, y) \wedge Q(x, z)) \wedge R(x)$   
liée liée liée liée liée libre libre

$B = \forall x ((\exists y Q(x, y)) \wedge P(x, y, z))$   
liée liée liée liée liée libre libre

- Une variable est libre si au moins une de ses occurrences est libre
- Une variable est liée si toutes ses occurrences sont liées

Variables libres de  $A = \{z, x\}$

Variables liées de  $A = \{y\}$

Variables libres de  $B = \{z, y\}$

Variables liées de  $B = \{x\}$

- ❖ Une fbf sans variable libre est dite close ou fermée

### Exemple :

$\forall x \exists y (P(x, y) \wedge \forall z R(x, y, z))$   
liée liée liée liée liée

## 3.3 Sémantique de la logique des prédicats

### 3.3.1 Introduction

La sémantique attribue une signification aux expressions. Elle est **compositionnelle** : la signification d'une formule est fonction de celle de ses constituants. Comme en calcul propositionnel, nous allons nous efforcer de construire un modèle permettant de dégager une interprétation sémantique de nos formules.

### 3.3.2 Quelques notions et définitions

#### ❖ Interprétation

Une interprétation  $I$  est définie par :

1. Un ensemble non vide  $D_I$ , appelé domaine de  $I$ ,
2. Pour chaque constante  $c$ , un élément de  $D_I$  noté  $I(c)$ , appelé interprétation de  $c$ ,
3. Pour chaque symbole de fonction  $f$  d'arité  $n$ , une application de  $D_I^n$  dans  $D_I$  notée  $I(f)$  appelée interprétation de  $f$ ,
4. Pour chaque symbole de prédicat  $P$  d'arité  $n$ , une application de  $D_I^n$  dans  $\{0,1\}$  notée  $I(P)$  appelée interprétation de  $P$ ,

#### ❖ Assignment

une valuation (assignment)  $\sigma$  est une application qui associe à chaque variable un élément de  $D_I$ .

Donc une application d'un sous ensemble de  $V$  dans  $D_I$

❖ **Interprétation d'un terme  $t$**  pour une assignment (valuation)  $\sigma$  notée  $I_\sigma(t)$  est définie comme suit :

- si  $t$  est une constante  $c$ ,  $I_\sigma(c) = I(c)$
- si  $t$  est une variable  $x$ ,  $I_\sigma(x) = \sigma(x)$
- si  $t = f(t_1, t_2, \dots, t_n)$ ,  $I_\sigma(t) = I(f)(I_\sigma(t_1), I_\sigma(t_2), \dots, I_\sigma(t_n))$

❖ **Interprétation d'une formule  $F$**

- si  $F$  est une formule atomique,  $P(t_1, t_2, \dots, t_n)$ ,  $I_\sigma(F) = I(P)(I_\sigma(t_1), I_\sigma(t_2), \dots, I_\sigma(t_n))$
- si  $F = \neg G$ ,  $I_\sigma(F) = f_{non}(I_\sigma(G))$
- si  $F = G C_B H$ ,  $I_\sigma(F) = f_{C_B}(I_\sigma(G), I_\sigma(H))$ , (avec  $C_B = \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ )
- si  $F = \exists x G(x)$ ,  $I_\sigma(F) = 1$  s'il existe un élément  $a \in D_I$   $I_{\sigma_a}(F) = 1$   
 $\sigma_a = \sigma$  et  $\sigma_a(x) = a$
- si  $F = \forall x G(x)$ ,  $I_\sigma(F) = 1$ , si pour tout élément  $a \in D_I$   $I_{\sigma_a}(F) = 1$

## Cours-Logique Mathématique

En calcul des prédicats, il n'est pas possible d'appliquer une méthode des tables de vérité directement dérivée du calcul propositionnel, en raison des domaines de valeur des variables de chacun des prédicats. En fait, pour une formule atomique  $p(x_1 \dots, x_n)$ , nous allons avoir besoin d'une fonction (appelée fonction d'interprétation) chargée de donner un sens au symbole  $p$ , et donc de calculer sa valeur de vérité selon la valeur des  $x_1 \dots, x_n$ .

Comme dans la logique des propositions, les formules ont une valeur de vérité qui dépend de l'interprétation choisie. Cette valeur de vérité est établie comme suit :

- La valeur de vérité d'une formule atomique est la valeur de vérité du prédicat selon l'interprétation.
- La valeur de vérité d'une formule contenant des variables quantifiées est :
  - si  $F = \exists x G$ , la valeur de  $F$  sera vraie si la valeur de  $G$  selon  $I$  est vraie pour au moins une valeur de  $x \in D_I$  sinon la valeur de  $F$  est fausse.
  - si  $F = \forall x G$ , la valeur de  $F$  sera vraie si la valeur de  $G$  selon  $I$  est vraie pour toutes les valeurs de  $x \in D_I$ , sinon la valeur de  $F$  est fausse.
- La valeur de vérité d'une formule non atomique construite à partir de formules atomiques, est calculée au moyen des tables de vérité des connecteurs.

### Exemple

Soit  $F$  la formule suivante

$$F = \forall y (R(y, c) \rightarrow \exists x (R(t, c) \wedge R(c, t))) \quad \text{Avec } t = g(y, f(h(x, x)))$$

Soit  $I$  l'interprétation de:

-Domaine l'ensemble  $\mathfrak{R}$

$$-I(c) = 0$$

$$-I(f(r)) = -r \quad I(h(x, x)) = x * x \quad I(g(y, z)) = y + z$$

$$I(R(y, z)) = 1 \text{ si } y \geq z$$

$$I_\sigma(t) = I(g)(I_\sigma(y), I(f(h)(I_\sigma(x), I_\sigma(x))) = \sigma(y) - \sigma(x)^2$$

$$I_\sigma(R(t, c)) = \sigma(y) - \sigma(x)^2 \geq 0$$

$$I_\sigma(R(t, c) \wedge R(c, t)) = (\sigma(y) - \sigma(x)^2 \geq 0) \wedge (\sigma(x) - \sigma(y)^2 \geq 0) = \sigma(x) - \sigma(y)^2 = 0$$

$$I_\sigma(R(y, c)) = \sigma(y) \geq 0$$

$$I_\sigma(R(y, c) \rightarrow \exists x (R(t, c) \wedge R(c, t))) = 1 \text{ si } y \text{ est un carré parfait sinon } 0$$

$$\text{donc } I_\sigma(\forall y (R(y, c) \rightarrow \exists x (R(t, c) \wedge R(c, t)))) = 0$$

### 3.3.3 Théorèmes d'équivalence

- $(\forall x)G(x) \equiv (\forall y)G(y)$
- $(\exists x)G(x) \equiv (\exists y)G(y)$
- $((\neg(\exists x)G(x))) \equiv (\forall x)(\neg G(x))$
- $(\neg(\forall x)G(x)) \equiv (\exists x)(\neg G(x))$
- $(\forall x)(G(x) \wedge H(x)) \equiv (\forall x)G(x) \wedge (\forall x)H(x)$
- $(\exists x)(G(x) \vee H(x)) \equiv (\exists x)G(x) \vee (\exists x)H(x)$

**ATTENTION :**

- $(\forall x)(G(x) \vee H(x))$  *non équivalent* à  $(\forall x)G(x) \vee (\forall x)H(x)$
- $(\exists x)(G(x) \wedge H(x))$  *non équivalent* à  $(\exists x)G(x) \wedge (\exists x)H(x)$
- $(\forall x)G(x) \vee (\forall x)H(x) \equiv (\forall x)G(x) \vee (\forall y)H(y)$
- $(\exists x)G(x) \vee (\exists x)H(x) \equiv (\exists x)G(x) \vee (\exists y)H(y)$
- $(\forall x)G(x) \wedge (\forall x)H(x) \equiv (\forall x)G(x) \wedge (\forall y)H(y)$
- $(\exists x)G(x) \wedge (\exists x)H(x) \equiv (\exists x)G(x) \wedge (\exists y)H(y)$

### 3.3.4 Quelques notions classiques (Validité, Insatisfiabilité, Conséquence, Complétude)

❖ **Validité**

Une fbf  $A$  est une **tautologie (valide)** si et seulement si elle est vraie dans toute interprétation ; on écrit alors :  $\models A$

**Exemple**

$A \vee \neg A$  est une formule valide

Une fbf est **invalide** si et seulement si **elle n'est pas valide**

**Exemple**

$A \wedge B$  et  $A \vee B$  sont deux formules invalides il suffit que  $A$  et  $B$  soient fausses

**Insatisfiabilité**

Une fbf est **inconsistante** ou insatisfiable si et seulement si **elle est fausse dans toute interprétation**

**Exemple :**

$\neg A \wedge A$  est une formule inconsistante

Une fbf  $A$  est **consistante** ou **satisfiable**

- si et seulement si elle n'est pas inconsistante

- s'il existe une interprétation  $I$  telle que  $V(A, I) = 1$
- si elle admet un modèle

### 3.3.5 Conséquence logique

$A$  est une **conséquence logique** de  $E$  si et seulement si **toutes les interprétations qui rendent vraies toutes les formules de  $E$  rendent vraie la formule  $A$** . On écrit alors  $E \models A$

On dit qu'une formule  $C$  est une conséquence logique de  $H_1.. H_n$


- si et seulement si tout modèle de  $H_1...H_n$  est un modèle de  $C$
- si et seulement si  $H_1 \wedge H_2 \wedge \dots \wedge H_n \rightarrow C$  est valide

Dans ce contexte les formules  $H_i$  sont les hypothèses et  $C$  est la conclusion.

### 3.3.6 Indécidabilité et semi-décidabilité de la logique des prédicats

Lorsqu'une formule ne contient pas de variable, on peut, comme en calcul propositionnel, en utilisant les tables de vérité, déterminer en un nombre fini d'opérations si cette formule est valide ou non inconsistante ou non. La situation est plus complexe en présence de variables et donc de quantificateurs car il y a une infinité d'interprétations. On montre qu'il est impossible de proposer un algorithme général capable de décider en un nombre fini d'opérations de la validité ou du non validité de n'importe quelle formule de la logique des prédicats du premier ordre. On dit que la logique des prédicats est indécidable. (Théorème d'indécidabilité de Church) Cependant, on peut proposer des algorithmes généraux pour décider de la validité de certaines familles de fbfs :

- si la fbf est valide ils s'arrêteront
- si la fbf est non valide ils risquent de ne pas s'arrêter

 *La logique des prédicats est semi-décidable*

## 3.4 Représentation des connaissances(Modélisation)

### 3.4.1 Introduction

La représentation des connaissances en Intelligence Artificielle consiste à faire une correspondance entre le monde réel et un système symbolique manipulable par un ordinateur. La représentation des connaissances comporte un aspect passif : il faut mémoriser. Par exemple, un livre ne connaît pas l'information qu'il contient. Mais aussi un côté actif : il faut inférer, manipuler ces connaissances, effectuer un raisonnement.



## 3.4.2 Quelques orientations

### ❖ L'universelle affirmative

- Tous les F sont des G
- $\forall x(F(x) \rightarrow G(x))$
- Tout F est G
- Tout ce qui est F est G
- N'importe lequel F est G
- Les F sont tous G
- Si un être quelconque est F, il est G
- Chaque F est G
- Seuls les G sont F

### ❖ L'universelle négative

- Aucun F n'est G
- $\forall x(F(x) \rightarrow \neg G(x))$
- Il n'y a aucun F et G
- Rien n'est à la fois F et G
- Les F et G n'existent pas

### ❖ La particularité affirmative

- Quelques F sont G
- $\exists x(F(x) \wedge G(x))$
- Quelque F est G
- Il y a des F et G
- Quelque chose est à la fois F et G
- Il y a un F et G
- Des F et G existent

### ❖ La particularité négative

- Quelques F ne sont pas G
- $\exists x(F(x) \wedge \neg G(x))$
- Quelque F n'est pas G
- Il y a des F et non G
- Quelque chose est à la fois F et non G
- Il y a un F et non G
- Des F et non G existent

### 3.4.3 Exemple

Soit à traduire le groupe de phrases suivantes :

- a) Omar ibn al-Khattâb était un homme
- b) Omar ibn al-Khattâb était un *QURAYSHITES*
- c) Tous les *QURAYSHITES* étaient des Musulmans
- d) Omar ibn al-Khattâb était souverain
- e) Tous les Musulmans étaient fidèles à Omar ibn al-Khattâb, soit le haïssaient
- f) Chacun est fidèle à quelqu'un
- g) Les gens n'essayent d'assassiner que les souverains auxquels ils ne sont pas fidèles
- h) Aboû Louloua était un esclave persan
- i) Aboû Louloua a essayé d'assassiner Omar ibn al-Khattâb

-l'univers du discours suivant :

D = ensemble des êtres humains

**Prédicats :**

- **HOMME(x)** :  $x$  est un homme
- **QURAYSHITE(x)** :  $x$  est *QURAYSHITE*
- **SOUVERAIN(x)** :  $x$  est souverain
- **MUSULMAN(x)** :  $x$  est Musulman
- **ESCLAVE\_PERSAN(x)** :  $x$  est un esclave persan
- **PERSONNE(x)** :  $x$  est une personne
- **FIDELE(x, y)** :  $x$  est fidèle à  $y$
- **HAIR(x, y)** :  $x$  hait  $y$
- **ESSAYER\_ASSASSINER(x, y)** :  $x$  essaye d'assassiner  $y$

**Constantes :**

- **Omar ibn al-Khattâb**
- **Aboû Louloua**

-Pour chacune des phrases : une formule bien formée à l'aide des propositions définies ci-dessus, des connecteurs et des parenthèses.

- a) Omar ibn al-Khattâb était un homme     $HOMME(Omar\ ibn\ al-Khattâb)$

b) Omar ibn al-Khattâb était un QURAYSHITE  $QURAYSHITE(Omar\ ibn\ al-Khattâb)$

c) Tous les QURAYSHITES étaient des MUSULMANS

$$\forall x(QURAYSHITE(x) \rightarrow MUSULMAN(x))$$

d) Omar ibn al-Khattâb était souverain  $SOUVERAIN(Omar\ ibn\ al-Khattâb)$

e) Tous les MUSULMAN étaient fidèles à Omar ibn al-Khattâb, soit le haïssaient

$$\forall x(MUSULMAN(x) \rightarrow FIDELE(x, Omar\ ibn\ al - Khattâb) \vee HAIR(x, Omar\ ibn\ al - Khattâb))$$

ou

$$\begin{aligned} \forall x (MUSULMAN(x) \\ \rightarrow (FIDELE(x, Omar\ ibn\ al - Khattâb) \vee HAIR(x, Omar\ ibn\ al \\ - Khattâb)) \wedge \neg(FIDELE(x, Omar\ ibn\ al - Khattâb) \\ \wedge HAIR(x, Omar\ ibn\ al - Khattâb))) \end{aligned}$$

f) Chacun est fidèle à quelqu'un

$$\forall x \exists y FIDELE(x, y)$$

g) Les gens n'essayent d'assassiner que les souverains auxquels ils ne sont pas fidèles

$$\begin{aligned} \forall x \forall y (PERSONNE(x) \wedge SOUVERAIN(y) \\ \wedge ESSAYER\_ASSASSINER(x, y) \rightarrow \neg FIDELE(x, y)) \end{aligned}$$

h) Aboû Louloua était un esclave persan  $ESCLAVE\_PERSAN(Aboû\ Louloua)$

i) Aboû Louloua a essayé d'assassiner Omar ibn al-khattâb

$$ESSAYER\_ASSASSINER(Aboû\ Louloua, Omar\ ibn\ al - Khattâb)$$

### 3.5 Démonstrations

#### 3.5.1 Introduction

Objectif : Transformer une fbf sous la forme "ensemble de clauses" pour appliquer le principe de résolution. Cette méthode généralisée à la logique des prédicats et restreinte aux clauses de Horn est à la base du langage Prolog

Pour cela, plus précisément, on va se focaliser sur le problème suivant : on se donne une formule F, et on veut déterminer si F est une tautologie. On dit encore que F est valide.

#### 3.5.2 Forme normale prénexe

L'idée à la base de cette section est de construire une formule équivalente dans laquelle les quantificateurs seront rejetés en tête de la formule.

**Mise sous forme Prénexe**

## Cours-Logique Mathématique

**Définition :** Une fbf en logique des prédicats est dite **en forme prénexe** (fnp) si et seulement si elle est de la forme :

$$(Q_1 x_1)(Q_2 x_2)\dots(Q_n x_n) \quad M(x_1, x_2, \dots, x_n)$$

**Préfixe**

**Matrice**

Où chaque  $(Q_i x_i)$  est soit  $\forall x_i$  soit  $\exists x_i$  et où  $M$  est une fbf ne contenant aucun quantificateur.

**Exemple**

$$\forall x \exists y (P(x, y) \rightarrow P(y, x))$$

$$\forall x \exists y \forall z (P(x) \wedge \neg Q(y, z) \wedge P(f(y)))$$

Sont en fnp par contre

$$\forall x (P(x) \wedge \exists y Q(y, x)) \quad \text{n'est pas en fnp}$$

Par application successives des théorèmes sur les paires (fbfs) équivalentes (lois d'équivalence) on peut trouver une fbf  $G'$  en fnp équivalente à une fbf  $G$  donnée.

**Méthode de transformation d'une fbf en fnp**

1. **Éliminer les connecteurs  $\rightarrow$  et  $\leftrightarrow$**

$$(G \leftrightarrow F) \equiv (G \rightarrow F) \wedge (F \rightarrow G)$$

$$(G \rightarrow F) \equiv (\neg G \vee F)$$

2. **Accoler les connecteurs  $\neg$  aux atomes concernés**

$$\neg(\neg G) \equiv G$$

$$\neg(F \vee G) \equiv \neg F \wedge \neg G$$

$$\neg(F \wedge G) \equiv \neg F \vee \neg G$$

$$\neg((\forall x) P(x)) \equiv (\exists x) \neg P(x)$$

$$\neg((\exists x) P(x)) \equiv (\forall x) \neg P(x)$$

3. **Rebaptiser les variables liées si nécessaire** de sorte que chaque quantificateur gouverne une variable originale

$$(\forall x) P(x) \equiv (\forall y) P(y)$$

$$(\exists x) P(x) \equiv (\exists y) P(y)$$

4. **Déplacer tous les quantificateurs à gauche de la formule** (sans changer l'ordre relatif)

$$((Q1 x) F(x)) \vee ((Q2 y) H(y)) \equiv (Q1 x) (Q2 y) (F(x) \vee (H(y)))$$

$$((Q1 x) F(x)) \wedge ((Q2 y) H(y)) \equiv (Q1 x) (Q2 y) (F(x) \wedge (H(y)))$$

Au terme de ces 4 étapes on obtient une fnp de la fbf initiale qui lui est équivalente. Il n'y a pas d'unicité (On peut avoir diverses fnp pour une même fbf)

**Exemple**

- $(\forall x) P(x) \rightarrow (\exists x) Q(x)$ 
  - $\equiv (\neg((\forall x)P(x))) \vee (\exists x) Q(x)$
  - $\equiv ((\exists x)\neg P(x)) \vee (\exists x) Q(x)$
  - $\equiv (\exists x)(\neg P(x)) \vee Q(x)$
  
- $(\exists x (P(x) \rightarrow Q(x))) \rightarrow (\forall x P(x) \rightarrow \exists x Q(x))$ 
  - $\equiv (\neg(\exists x (P(x) \rightarrow Q(x)))) \vee ((\forall x) P(x) \rightarrow \exists x Q(x))$
  - $\equiv (\neg(\exists x (\neg P(x) \vee Q(x)))) \vee ((\neg(\forall x)P(x)) \vee \exists x Q(x))$
  - $\equiv (\forall x (P(x) \wedge \neg Q(x))) \vee (\exists x(\neg P(x)) \vee \exists x Q(x))$
  - $\equiv (\forall x (P(x) \wedge \neg Q(x))) \vee (\exists x(\neg P(x) \vee Q(x)))$
  - $\equiv (\forall x (P(x) \wedge \neg Q(x))) \vee (\exists y(\neg P(y) \vee Q(y)))$
  - $\equiv (\forall x \exists y ((P(x) \wedge \neg Q(x))) \vee (\neg P(y) \vee Q(y)))$

### 3.5.3 Méthode de Skolemisation

#### Forme de Skolem

**Définition** (forme de Skolem).

Une formule est en forme normale de Skolem (ou forme standard de Skolem(fss)) ou tout simplement forme de Skolem si elle est en forme normale prénexe et ne contient pas de quantificateur existentiel (elle ne contient que des quantifications universelles).

#### Exemple

Soit la formule  $\exists x \forall y p(x,y)$  :

remplacer x par la constante a :  $\forall y p(a,y)$

A partir d'une fnp  $G'$  d'une fbf  $G$  on peut produire une forme standard de Skolem par les transformations suivantes :  $G' = (Q_1 x_1)(Q_2 x_2) \dots (Q_n x_n) \quad M(x_1, x_2, \dots, x_n)$  fnp de  $G$

1. Eliminer les quantificateurs existentiels

Soit  $Q_r$  un quantificateur existentiel dans le préfixe de  $G'$

(On opère habituellement de la gauche vers la droite mais l'ordre importe peu)

- a) Si aucun quantificateur universel n'apparaît avant  $Q_r$  c'est-à-dire dans

$$G' = (Q_1 x_1) \dots (Q_{r-1} x_{r-1})$$

On choisit un symbole de constante a différent de toute constante apparaissant dans la matrice M

On supprime  $(Q_r x_r)$  du préfixe

On remplace  $x_r$  par a dans la matrice M

#### Exemple

$(\exists x)(\forall y)(P(x) \vee Q(y))$  Choix de la constante a/x ce qui donne

$$(\forall y)(P(a) \vee Q(y)) \text{ fss}$$

b) Si  $Q_{s1} Q_{s2} \dots Q_{sm}$  sont  $m$  quantificateurs universels apparaissant avant  $Q_r$  dans le préfixe on choisit un symbole de fonction  $f$  d'arité  $m$  différent de toute fonction apparaissant dans la matrice  $M$

On supprime  $(Q_r x_r)$  du préfixe

On remplace tout  $x_r$  dans  $M$  par  $f((Q_{s1})(Q_{s2}) \dots (Q_{sm}))$

2. On itère le processus jusqu'à ce qu'il n'y ait plus de quantificateur existentiel dans le préfixe

Les fonctions et les constantes introduites sont appelées : constantes et fonctions de Skolem.

Nous obtenons une fbf dite : sous forme standard de Skolem

### Exemple

$\exists x \exists y \forall z \forall t \exists v P(x, y, z, t, v)$

1. étape a|x

$\exists y \forall z \forall t \exists v P(a, y, z, t, v)$

2. étape b|y

$\forall z \forall t \exists v P(a, b, z, t, v)$

3. étape f(z, t)|v

4.  $\forall z \forall t P(a, b, z, t, f(z, t))$

### Théorème

Soit  $G_S$  la forme standard de Skolem d'une fbf  $G$  :

$G$  Inconsistante si et seulement  $G_S$  inconsistante

### 3.5.4 Forme clausale

**Définition** (Forme clausale). Une formule close est sous forme clausale si elle est

1. en forme prenexé,
2. elle est universelle (tous ses quantificateurs sont universels), et
3. sa matrice est sous forme normale conjonctive.

**Passage à la forme clausale** (Obtention d'un ensemble de clauses)

Soit  $G_S$  une forme standard de Skolem d'une fbf  $G$

1. Éliminer tous les quantificateurs

Il ne reste que des quantificateurs universels. On allège la notation en les supprimant.

On suppose donc désormais que toutes les variables sont quantifiées universellement.

2. Passer sous forme normale conjonctive

## Cours-Logique Mathématique

---

- Éliminer les connecteurs  $\wedge$  de la conjonction de clauses obtenues au 2. est considérée comme un ensemble de clauses  $S$

$S$  est dite insatisfiable ou insatisfaisable pour dire  $S$  inconsistante

- Distinguer les variables des clauses distinctes si c'est nécessaire

### Exemple

$$(\forall x (P(x) \vee Q(x) \rightarrow H(x))) \\ \forall x P(x) \wedge \forall y (Q(y) \wedge \exists z R(y, z))$$

### Conclusion

- Il existe en général, plusieurs formes standards d'une même formule. On a intérêt à introduire des fonctions de Skolem aussi simples que possible  $\rightarrow$  essayer de repousser le plus à gauche le quantificateur existentiel.
- Si  $F$  peut s'écrire  $F_1 \wedge F_2 \wedge \dots \wedge F_n$  un ensemble de clauses pour  $F$  peut être obtenu comme union des ensembles de clauses  $S_i$  de chaque  $F_i$
- $F_S = f_{S_1} \wedge f_{S_2} \wedge \dots \wedge f_{S_n}$  SKOLEM  
 $S = S_1 \wedge S_2 \wedge \dots \wedge S_n$  CLAUSES  
 C'est-à-dire on fait clauses puis Skolem

### Exemple

$$\forall x P(x) \wedge \exists y Q(y)$$

Skolem :  $\forall x (P(x) \wedge Q(f(x)))$

Clauses :  $\forall x (P(x) \wedge Q(f(x)))$

Ens. Clauses :  $S = \{P(x), Q(f(x))\}$

$$\forall x P(x) \exists y Q(y)$$

Skolem :  $\forall x P(x) Q(a)$

Ens. Clauses :  $S = \{P(x), Q(a)\}$

$$(\neg[\forall x (R(x) \rightarrow M(x) \wedge \forall x (R(x) \rightarrow M(x)) \rightarrow \forall x \neg R(x)]) \\ \text{Fbf } G \text{ équivalente à } G' \text{ fnp de } G$$

Fbf  $G$  non équivalente (si  $G$  est consistante) à  $G''$  (fss) ou forme clausale

On a uniquement si  $G$  est inconsistante alors  $G''$  est forme clausale inconsistante. Donc pour étudier la validité d'une fbf  $G$  on étudie l'inconsistance de  $\neg G$  et donc de la fss de  $\neg G$ . On dit que l'on procède par **réfutation**

$$G \text{ valide} \iff \neg G \text{ inconsistante} \\ \iff \text{fnp } (\neg G) \text{ inconsistante}$$

## Cours-Logique Mathématique

---

$\Leftrightarrow$  fss  $(\neg G)$  inconsistante

$\Leftrightarrow$  S de  $(\neg G)$  insatisfiable

Dans les applications de la logique des prédicats, en général on veut montrer qu'une fbf H est conséquence logique de fbf  $G_1, \dots, G_n$  c'est-à-dire :

$G_1 \wedge G_2 \wedge \dots \wedge G_n \rightarrow H$  valide

$\neg(G_1 \wedge G_2 \wedge \dots \wedge G_n \rightarrow H)$

$\Leftrightarrow$   
inconsistant

$\Leftrightarrow G_1 \wedge G_2 \wedge \dots \wedge G_n \rightarrow \neg H$  inconsistant

$\Leftrightarrow G_1 \wedge G_2 \wedge \dots \wedge G_n \wedge \neg H$  inconsistant

### 3.6 Unification

#### 3.6.1 Introduction

Généralisation du principe de résolution à la logique des prédicats : Etendre les définitions de "littéraux complémentaires" et de "résolvante"

*L'unification : C'est le processus qui rend 2 expressions identiques.*

En logique propositionnelle, 2 expressions sont identiques si et seulement elles sont syntaxiquement identiques.

La présence de variables en logique des prédicats complique ce fait :

$Etudiant(x) = Etudiant(Mustapha)$  SSI  $x = Mustapha$

Parfois, il faut substituer une fonction à une variable :

$Etudiant(x) \rightarrow Avoir\_bac(x)$

$Etudiant(Frère\_de(Mohamed))$

Instanciation de  $x$  par  $Frère\_de(Mohamed)$  :

$Etudiant(Frère\_de(Mohamed)) \rightarrow Avoir\_bac(Frère\_de(Mohamed))$  a

Soient  $C1 = P(x) \vee Q(a, x)$  et  $C2 = \neg P(g(y)) \vee R(y, b)$  Peut-on dire des littéraux  $P$  et  $\neg P$  qu'ils sont complémentaires ? Peut-on définir une résolvante ?

Si  $g(a)|x$  et  $a|y$  on obtient :

$C1' = P(g(a)) \vee Q(a, g(a))$

$C2' = \neg P(g(a)) \vee R(a, b)$

$C1'$  et  $C2'$  ont une résolvante  $C = Q(a, g(a)) \vee R(a, b)$

$C$  est conséquence logique de  $C1'$  et de  $C2'$  et donc de  $C1$  et  $C2$ .

On peut trouver d'autres instances de  $C1$  et  $C2$  [ $g(g(b))|x$ ] et [ $g(b)|y$ ]...

On peut continuer sur d'autres instances  $C1$  et  $C2$  dont les littéraux en  $P$  sont complémentaires

Tous les résolvants obtenus sont des instances de  $C = Q(a, x) R(y, b)$



## Objectifs :

L'unification est donc le procédé qui consiste à trouver des affectations de variables de façon à rendre des expressions identiques symboliquement.

### 3.6.2 Les substitutions

Une **substitution**  $\theta$  est une application de l'ensemble des variables dans l'ensemble des termes telle que  $\theta$  est égale à l'identité sauf pour un nombre fini de variables  $\{x_1, x_2, \dots, x_n\}$  et sera notée

$$\{x_1 / t_1, x_2 / t_2, \dots, x_n / t_n\}$$

**Exemple :**  $\theta = \{x / a, y / f(x)\}$

$$\theta(R(f(x), f(y))) = R(f(a), f(f(x)))$$

#### Composition de substitutions

Le produit  $\sigma \circ \theta$  de deux substitutions  $\theta$  et  $\sigma$  est la substitution obtenue en appliquant  $\theta$  ensuite  $\sigma$

**Exemple :**  $\sigma = \{y / f(x)\}$  et  $\theta = \{x / a\}$

$$\sigma \circ \theta(R(f(x), f(y))) = R(f(a), f(f(a)))$$

$$\theta \circ \sigma(R(f(x), f(y))) = R(f(a), f(f(x)))$$

#### Définition

Un unificateur de deux expressions  $F$  et  $G$  est une substitution  $\theta$  qui rend  $\theta(F) = \theta(G)$

Un couple de formules peut ne pas en avoir d'unificateur ou peut en avoir plusieurs. Par exemple  $f(x, y)$  et  $f(x, a)$  ont  $\{y/a\}$  comme unificateur, mais aussi  $\{x/a, y/a\}$ .

#### Définition

La substitution  $\sigma$  est une instance de la substitution  $\tau$  (ou  $\tau$  est plus générale que  $\sigma$ ), ce que l'on écrit  $\sigma \leq \tau$ , si et seulement s'il existe une substitution  $\rho$  t.q. pour toute variable  $x \in X$ ,  $\sigma(x) = (\rho \circ \tau)(x)$ .

**Exemple :**  $\{x / f(y), y / z\}$  est plus générale que  $\{x / f(b), y / h(c), z / h(c)\}$

#### Définition

Un unificateur  $\sigma$  de  $W = \{E_1, \dots, E_k\}$  est dit upg (unificateur le plus général) si et seulement si pour tout unificateur  $\theta$  de  $W$ , il existe  $\delta$  tel que :  $\theta = \delta \circ \sigma$

#### Définition

## Cours-Logique Mathématique

Soit  $S$  un ensemble de substitutions et  $\tau \in S$ . On dit que  $\tau$  est principale ssi toute substitution  $\sigma \in S$  est une instance de  $\tau$ .

**Exemple.** Soit  $\tau = [f(w, x)/x, z/y]$ ,  $\sigma = [f(g(y), x)/x, c/y]$ . On a alors  $\sigma$  est instance de  $\tau$ , à cause de :  $\delta = [g(y)/w, c/z]$

### 3.6.3 Algorithme

L'algorithme d'unification consiste en la recherche d'un upg d'un ensemble d'expressions.

Voyons quelques notions qui vont être utilisées au cours de l'algorithme d'unification, tout d'abord l'ensemble de discordance. L'ensemble de discordance d'un ensemble non vide d'expressions est obtenu en localisant la première position à partir de la gauche pour laquelle toutes les expressions n'ont pas le même symbole et en extrayant dans chaque expression, la sous-expression qui commence en cette position

$W1 = \{P(X, g(X), f(X, Y)), P(X, g(X), f(g(t, Y), Z))\}$

Ensemble de discordance :  $D1 = \{X, g(t, Y)\}$

$W2 = \{P(Y, X, Z), P(Y, f(t), h(Y)), P(Y, b, U)\}$

Ensemble de discordance :  $D2 = \{X, f(t), b\}$

$W3 = \{Q(Y), P(Y)\}$

Voici maintenant l'algorithme d'unification :

Soit  $W$  l'ensemble fini à unifier

- **Etape 1 :**  $k = 0$  ;  $W_k = W$  ;  $\sigma_k = \varepsilon$
- **Etape 2 :**  
SI  $W_k$  est un singleton **ALORS**  $\sigma_k$  upg de  $W$   
**SINON** trouver  $D_k$  l'ensemble de discordance de  $W_k$  ;
- **Etape 3 :**  
SI il existe des éléments  $V_k$  et  $t_k$  de  $D_k$  tels que :
  - $V_k$  soit une variable
  - $t_k$  soit un terme ne contenant pas  $V_k$**ALORS** aller à l'étape 4  
**SINON**  $W$  non unifiable
- **Etape 4 :**  $\sigma_{k+1} = \{t_k/V_k\} \circ \sigma_k$  ;  
 $W_{k+1} = W_k.\{t_k/V_k\}$
- **Etape 5 :**  $k = k + 1$  ;

Aller à l'étape 2

SI  $W$  est ensemble fini non vide unifiable alors l'algorithme s'arrête toujours à l'étape 2 et  $\sigma_k$  est upg.

**Exemple :**

$$W = \{P(a, X, f(g(Y))), P(Z, f(Z), f(U))\}$$

$$1. k = 0 ; W_0 = W ; \sigma_0 = \varepsilon$$

$$2. D_0 = \{a, Z\} \text{ avec } Z \text{ variable et } a \text{ un terme ne contenant pas } Z ;$$

$$3. \sigma_1 = \{a, Z\} \circ \sigma_0 = \{a/Z\}$$

$$W_1 = W_0 \cdot \{a/Z\} = \{P(a, X, f(g(Y))), P(a, f(a), f(U))\}$$

$$4. D_1 = \{X, f(a)\}, V_1 = X, t_1 = f(a) \text{ terme ne contenant pas } X ;$$

$$5. \sigma_2 = \{f(a)/X\} \circ \{a/Z\} = \{f(a)/X, a/Z\}$$

$$W_2 = W_1 \cdot \{f(a)/X\} = \{P(a, f(a), f(g(Y))), P(a, f(a), f(U))\}$$

$$6. D_2 = \{g(Y)/U\}, V_2 = U, t_2 = g(Y) \text{ terme ne contenant pas } U ;$$

$$7. \sigma_3 = \{g(Y)/U\} \circ \sigma_2 = \{g(Y)/U, f(a)/X, a/Z\}$$

$$W_3 = W_2 \cdot \{g(Y)/U\} = \{P(a, f(a), f(g(Y)))\}$$

$$8. W_3 \text{ est un singleton} \rightarrow \sigma_3 \text{ est l'upg de } W \text{ donc } W \text{ est unifiable.}$$

### 3.7 La méthode de résolution dans le calcul des prédicats

La règle de résolution en logique des prédicats est similaire à la règle de résolution en logique propositionnelle mais les formules atomiques partagées par deux clauses ne doivent pas être identiques mais unifiables. Deux formules atomiques sont unifiables s'il existe une substitution des variables par des termes qui rend les deux formules identiques.

**Exemple :**

$$C1 = P(x) \vee Q(x) \quad \text{et} \quad C2 = P(a)$$

$$\text{Résolvante } (C1, C2) = Q(a)$$

$$\frac{P(a) \quad (P(x) \vee Q(x))}{Q(a)}$$

Où  $a$  est une constante

**Exemple**

$$C1 = \neg P(x) \vee \neg Q(y) \vee R(x, y)$$

$$C2 = Q(a)$$

$$C3 = P(b)$$

## Cours-Logique Mathématique

---

La substitution  $\sigma = \{y/a\}$  permet l'unification suivante en C1 et C2

$\sigma(C1) = \neg P(x) \vee \neg Q(a) \vee R(x, a)$  et l'application de la règle de résolution

$$C'1 = \text{Res}(C1, C2) = \neg P(x) \vee R(x, a)$$

La substitution  $\sigma = \{x/b\}$  permet d'unification suivante en C'1 et C3

$\sigma(C'1) = \neg P(b) \vee R(b, a)$  et l'application de la règle de résolution

$$C'1 = \text{Res}(C'1, C3) = R(b, a)$$

En général on utilise le principe de résolution pour effectuer des preuves par réfutation. Pour prouver que la formule C est une conséquence logique des formules  $H_1 \dots H_n$  on démontre que  $((H_1 \dots H_n) \rightarrow C)$  est valide ou encore l'ensemble  $\{H_1 \dots H_n \neg C\}$  est inconsistant.

Pratiquement, il faut commencer par mettre toutes les formules sous forme clausale, pour cela on doit les mettre sous forme prénexe (tous les quantificateurs au début) puis les skolémiser.

Pour montrer qu'un ensemble de clauses est inconsistant, il faut réussir à engendrer la clause vide en appliquant la règle de résolution autant de fois que nécessaire.

Par exemple, On veut montrer que les trois formules

$$H1: \forall x((S(x) \vee T(x)) \rightarrow P(x))$$

$$H2: \forall x(\neg S(x) \vee R(x))$$

$$H3: ((P(x) \rightarrow \neg R(a)) \wedge T(x))$$

Ont pour conséquence logique la formule.

$$C: P(a)$$

$$H1 \equiv \forall x((\neg S(x) \vee P(x)) \wedge (\neg T(x) \vee P(x)))$$

Donc produit deux clauses

$$C1: \neg S(x) \vee P(x) \quad \text{et} \quad C2: \neg T(x) \vee P(x)$$

De même à partir de H2

$$C3: \neg S(x) \vee R(x)$$

$H3 \equiv (\neg R(a) \vee (\neg P(x) \wedge T(x)))$  produit deux clauses

$$C4: \neg P(x) \vee \neg R(a) \quad \text{et} \quad C5: T(x)$$

et de C C6:  $\neg P(a)$

Donc le problème se réduit à démontrer que  $F_C = \{C1, C2, C3, C4, C5, C6\}$  est insatisfiable

## Cours-Logique Mathématique

---

$$\text{Res}(C1, C3) = P(x) \vee R(x) = C7$$

La substitution  $\sigma = \{x/a\}$  permet d'unification suivante :

$$\text{Res}(C7, C4) = P(a) \vee \neg P(a) = C8 \equiv \top$$

$$\text{Res}(C2, C6) = \neg T(a) = C9$$

$$\text{Res}(C5, C9) = \emptyset ,$$

Puisque  $F_C \vdash_{\text{Res}} \emptyset$  on peut conclure que  $F_C$  est insatisfiable donc le raisonnement  $\{H2, H2, H2\} \models C$  est valide

## 3.8 Exercices du chapitre 3 : logique des prédicats

### 3.8.1 Exercices : Syntaxe de la logique des prédicats

#### Exercice 3.8.1.1

On posera que  $x, y, z$  sont des variables, que  $a, b, c$  sont des constantes, que  $f, g, h$  sont des fonctions et que  $P, Q, R$  sont des prédicats.

-Dites si les « écritures » suivantes correspondent à des termes :

1.  $y$
2.  $c$
3.  $x \vee a$
4.  $f(x)$
5.  $f(x, a)$
6.  $(\wedge)$
7.  $f(y(a), b)$
8.  $g(h(x), f(x, y), c)$
9.  $f(f(f(f(x))))$

#### Exercice 3.8.1.2

On posera que  $x, y, z$  sont des variables, que  $a, b, c$  sont des constantes, que  $f, g, h$  sont des fonctions et que  $P, Q, R$  sont des prédicats.

-Dites si les « écritures » suivantes correspondent à des formules atomiques :

1.  $P$
2.  $P(a)$
3.  $P(x)$
4.  $P(a \vee b)$
5.  $Q(x, y, b)$
6.  $R(f(x))$
7.  $Q(g(f(a)), b)$
8.  $P(b) \vee Q(a)$

#### Exercice 3.8.1.3

On posera que  $x, y, z$  sont des variables, que  $a, b, c$  sont des constantes, que  $f, g, h$  sont des fonctions et que  $P, Q, R$  sont des prédicats.

-Dites si les « formules » suivantes sont des formules bien formées :

1.  $a \vee b$
2.  $P$
3.  $P \vee R$
4.  $\forall x P(x)$
5.  $\forall x \vee Q(y)$
6.  $\exists x P(x) \vee Q(y)$
7.  $Q(x, y, b) \rightarrow P(x) \vee P(a) \wedge R$
8.  $\forall b Q(x, y, b) \rightarrow P(x) \vee P(a) \wedge R$
9.  $\exists y Q(x, y, b) \rightarrow P(x) \vee P(a) \wedge R$

10.  $\forall z \forall x (\exists y Q(x, y, b) \rightarrow P(z) \vee P(a) \wedge R)$

### Exercice 3.8.1.4

1. Quand dit-on qu'une variable est libre dans une formule ?

Dans la suite de l'exercice, nous considérons le langage du premier ordre  $L = \{R, S, f, b\}$  où  $R$  et  $S$  désignent deux symboles de relation respectivement unaire et binaire,  $f$  désigne un symbole de fonction unaire et  $b$  désigne un symbole de constante.

Soit  $F$  la formule suivante :  $(\forall x \exists y R(f(x), f(y))) \wedge ((\forall z R(x, z)) \Rightarrow S(x))$

2. Les variables  $x$  et  $y$  sont-elles libres dans la formule  $F$  ? Justifiez votre réponse en quelques mots.
3. Transformez la formule  $F$  précédente de manière à ce que les variables liées et les variables libres (éventuelles) ne portent pas le même nom.
4. Donnez la formule  $F[x | t]$  (c'est-à-dire la substitution de  $t$  à la variable  $x$  dans  $F$ ) quand  $t$  est le terme  $f(z)$ .

### Exercice 3.8.1.5

Dans les formules suivantes, dites si les variables utilisées sont des variables libres ou des variables liées.

- $p(x) \vee q(y)$
- $\forall x p(x) \vee q(y)$
- $\forall x p(x) \vee q(y, z)$
- $\exists z (\forall x p(x, z) \vee q(y, z))$
- $\exists z (p(x, z) \vee \exists y q(y, z))$
- $\forall x p(x) \vee q(x)$
- $p(x, y) \vee p(y, x) \wedge p(x, x) \rightarrow p(x, y)$
- $\forall x \forall y (p(x, y) \vee p(y, x)) \wedge p(x, x) \rightarrow p(x, y)$
- $\forall x \forall y (p(x, y) \vee p(y, x)) \wedge p(x, x) \rightarrow \exists x \forall y p(x, y)$
- $\forall x \forall y (p(x, y) \vee p(y, x)) \wedge \forall x p(x, x) \rightarrow \exists x \forall y p(x, y)$

## 3.8.2 Exercices : Sémantique de la logique des prédicats

### Exercice 3.8.2.1

Soient les énoncés suivants :

E1 : pour tout entier il existe un entier plus grand

E2 : il existe un entier plus grand que tous les entiers

1. Traduire E1 et E2 en formules dans le langage des prédicats.
2. Vérifier la validité des deux énoncés E1 et E2.

### Exercice 3.8.2.2

On donne trois formules :

Totale  $\forall x \exists y R(x, y)$  la relation  $r$  est totale.

## Cours-Logique Mathématique

---

Irréflexive  $\forall x \neg R(x, x)$  la relation  $r$  est irréflexive

Symétrique  $\forall x \forall y (R(x, y) \rightarrow R(y, x))$  la relation  $r$  est symétrique

On donne trois interprétations :

1.  $I$  de domaine  $\{0\}$  avec  $rI = \{(0,0)\}$
2.  $J$  de domaine  $\{0,1\}$  avec  $rJ = \{(0,1), (1,0)\}$
3.  $K$  de domaine  $\{0,1,2\}$  avec  $rK = \{(0,1), (1,2), (2,0)\}$

Donner les valeurs des trois formules dans chacune de ces trois interprétations.

Pour trouver ces valeurs, on suggère de dessiner les interprétations.

On devra expliquer (brièvement) comment ces valeurs ont été obtenues.

### **Exercice 3.8.2.3**

On considère le langage du premier ordre composé d'un symbole de fonction  $f$  d'arité 2, du symbole binaire de l'égalité  $=$  (on l'utilisera avec la notation infix habituelle) et d'un symbole de relation  $R$  d'arité 2.

Les variables sont notées  $x, y, z \dots$ . Soit l'interprétation suivante :

- le domaine est  $\mathbb{Z}$  (ensemble des entiers relatifs),
- l'interprétation de  $f$ , soit  $f_I$ , est l'addition sur  $\mathbb{Z}$ ,
- l'interprétation de  $R$ , soit  $R_I$ , est la relation  $<$ ,
- l'interprétation de  $=$ , soit  $=_I$ , est l'égalité sur  $\mathbb{Z}$

Quelle est la valeur de vérité de chacune des 3 formules ci-dessous dans cette interprétation ? Vous justifierez votre réponse en quelques lignes.

- $F1 : \forall x \exists z (f(z, y) = x)$
- $F2 : \exists x (R(x, y) \wedge R(y, f(x, x)))$
- $F3 : \forall x (R(x, y) \Rightarrow R(f(x, x), y))$

### **3.8.3 Exercices : Modélisation en logique des prédicats**

#### **Exercice 3.8.3.1**

Soit à traduire en logique des prédicats le groupe de phrases suivantes :

1. Quiconque sait lire est instruit
2. Les dauphins ne sont pas instruits
3. Certains dauphins sont intelligents
4. Certains êtres intelligents ne savent pas lire
5. Flipper est un dauphin
6. Le frère de Flipper est intelligent



### *Exercice 3.8.3.2*

Il s'agit de construire un modèle partiel du fonctionnement d'une banque. Considérons les règles informelles suivantes. :

- Une banque gère pour ses clients deux types de comptes : les comptes courants et les comptes épargnes.
- Chaque compte appartient à un unique client.
- Un client peut posséder plusieurs comptes courants mais un seul compte épargne. Formaliser les règles précédentes en Calcul des Prédicats Il s'agit donc de se donner des symboles de prédicats et d'énoncer les règles au moyen de ceux-ci. L'utilisation du connecteur  $\exists!$  est autorisé.

### *Exercice 3.8.3.3*

Considérons deux personnes appelées Ahmed et Zoubir représentées par des constantes  $a$  et  $z$ , et le film Carnaval Fi Dachra représenté par la constante  $c$ . En utilisant les prédicats

- $F(y)$  :  $y$  est un film

- $H(x)$  :  $x$  est un être humain

- $A(x, y)$  :  $x$  est acteur dans  $y$

- $V(x, y)$  :  $x$  a vu  $y$

et le prédicat  $=$ ,

Formaliser les énoncés suivants :

1. Ahmed a vu Carnaval Fi Dachra
2. Quelqu'un a vu Carnaval Fi Dachra
3. Tout le monde a vu Carnaval Fi Dachra
4. Tout le monde n'a pas vu Carnaval Fi Dachra
5. Quelqu'un n'a pas vu Carnaval Fi Dachra
6. Ahmed a vu un film
7. Ahmed a vu exactement deux films
8. Zoubir est l'acteur de Carnaval Fi Dachra
9. Ahmed a vu tous les films de Zoubir
10. Ahmed n'a pas vu tous les films
11. Tout le monde a vu un film de Zoubir
12. Quelqu'un a vu tous les romans de Zoubir
13. Tous ceux qui sont acteurs d'un film ont vu Carnaval Fi Dachra
14. Tous les films n'ont pas un seul acteur
15. Parmi les films de Zoubir, Ahmed n'a vu que Carnaval Fi Dachra

### Exercice 3.8.3.4

Modéliser en logique de prédicats les phrases suivantes avec l'utilisation d'un seul prédicat :

$A(x, y)$  :  $x$  aime  $y$

- Tout le monde aime quelqu'un
- Quelqu'un est aimé de tout le monde
- Chaque personne aime quelqu'un et personne n'aime tout le monde, ou bien quelqu'un aime tout le monde et quelqu'un n'aime personne.

### Exercice 3.8.3.5

On se place dans un langage du premier ordre modélisant les entiers qui utilisent les symboles suivants :

- les constantes  $0, 1$  ;
- les symboles de fonction binaires  $+$  et  $*$  qui représentent l'addition et la multiplication et seront notés de manière usuelle  $x + y$  et  $x * y$  ;
- les symboles de prédicats unaires  $Pair(x)$  et  $Prem(x)$  représentant respectivement le fait que  $x$  est un nombre pair et  $x$  est un nombre premier (on rappelle qu'un nombre est premier s'il est strictement plus grand que 1 et n'est divisible que par 1 et par lui-même).
- les symboles de prédicats binaires  $Div(y, x)$  qui représente le fait que  $y$  divise  $x$ ,  $x = y$  qui représente que  $x$  est égal à  $y$  et  $x \leq y$  qui représente que  $x$  est inférieur ou égal à  $y$ .

1. Formaliser les énoncés suivants :

- Il existe un entier plus petit ou égal à tous les autres.
- Il n'existe pas d'entier plus grand ou égal à tous les autres, mais pour tout entier il en existe un qui est strictement plus grand.
- Tout nombre entier pair est égal à la somme de deux nombres entiers premiers.
- L'ensemble des entiers premiers est non borné.

2. Expliquer par des phrases le sens de chacune des formules suivantes et dire si elles sont vérifiées dans le modèle des entiers :

- $\forall x \forall y (Pair(x) \wedge Pair(y)) \rightarrow Pair(x + y)$
- $\forall x \forall y \exists z (Div(x; z) \wedge Div(z, y))$

3. Pour chacun des prédicats suivants, donner une formule équivalente qui n'utilise que les symboles de constantes  $0$  et  $1$ , les fonctions  $+$  et  $*$  et la relation d'égalité.

- $Pair(x)$
- $Div(y; x)$

(c)  $Prem(x)$  (on pourra utiliser le prédicat  $Div$ ).

### Exercice 3.8.3.6

Soit l'énoncé suivant :

$H_1$  : Les personnes qui ont la grippe ne doivent pas aller au travail.

$H_2$  : Les personnes qui ont de la fièvre et qui toussent ont la grippe.

$H_3$  : Ceux qui ont une température supérieure à  $38^\circ$  ont de la fièvre.

$H_4$  : Ahmed tousse et a une température supérieure à  $38^\circ$ .

Et  $C$  : Ahmed ne doit pas aller au travail.

On veut montrer en utilisant la méthode de la résolution que  $\{H_1, H_2, H_3, H_4\} \models C$

Modéliser en logique des prédicats l'énoncé ci-dessus en utilisant les prédicats suivants :

- $grippe(x)$  : x a la grippe

- $travail(x)$  : x doit aller au travail

- $fièvre(x)$  : x a de la fièvre

- $tousse(x)$  : x tousse

- $temp(x, t)$  : x a la température t

- $sup(x; y)$  : x est supérieur à y

On utilisera également les constantes suivantes :

-38

-Ahmed

### 3.8.4 Exercices : démonstration en logique des prédicats

#### Exercice 3.8.4.1

Mettre sous forme normale prénexe les fbf's suivantes :

a)  $\forall x P(x) \rightarrow (\exists t Q(t) \vee \exists t C(t))$

b)  $\forall x (\forall y P(x, y) \rightarrow \exists z R(x, z))$

c)  $\forall x \forall y \exists z (P(x, y, z) \wedge (\exists u Q(x, u) \rightarrow \exists v Q(y, v)))$

d)  $((\exists x P(x) \rightarrow \exists x R(x) \vee \forall y P(y)) \wedge \forall x \exists y (R(y) \rightarrow P(x)))$

#### Exercice 3.8.4.2

Prouvez la validité de la formule suivante :

## Cours-Logique Mathématique

$$\forall x(P(x) \rightarrow R(x)) \wedge \forall(Q(x) \rightarrow R(x)) \wedge \exists x(P(x)Q(x)) \rightarrow \exists xR(x)$$

Par résolution au premier ordre. On donnera la forme clausale de la négation de la formule et la preuve par résolution de la clause vide.

### Exercice 3.8.4.3

Soient les deux expressions  $g(h(x), y)$  et  $g(h(f(y)), z)$ .

L'unificateur  $x := f(k(a)), y := k(a), z := k(a)$  est-il un unificateur le plus général de ces deux expressions ?

Si vous répondez non, indiquez deux unificateurs plus généraux Calcul des unificateurs

1. Indiquer si les expressions  $f(x, g(h(z), x), y)$  et  $f(k(z), y, g(u, z))$  sont unifiables (justifier votre réponse) et si elles le sont, donnez-leur unificateur le plus général.

2. Indiquer si les expressions  $f(x, g(h(z), x), y)$  et  $f(k(z), y, g(h(j(u)), x))$  sont unifiables (justifier votre réponse) et si elles le sont, donner leur unificateur le plus général.

### Exercice 3.8.4.4

Considérons les 3 formules suivantes :

$$(H1) \forall x(U(x) \wedge \neg V(x) \rightarrow \exists y(S(x, y) \wedge Q(y)))$$

$$(H2) \exists x(P(x) \wedge U(x) \wedge \forall y(S(x, y) \rightarrow P(y)))$$

$$(H3) \forall x(P(x) \rightarrow \neg V(x))$$

Montrez, par résolution au premier ordre, que de ces trois formules, on peut déduire :

$$\exists x(P(x) \wedge Q(x)).$$

### Exercice 3.8.4.5

Pour chacun des ensembles suivants, déterminer s'il est unifiable, et si oui, donner un unificateur principal :

1.  $W = \{Q(f(a)), g(a), R(y, y)\}$
2.  $W = \{Q(a), Q(b)\}$
3.  $W = \{Q(a, x), Q(a, a)\}$
4.  $W = \{Q(f(a), g(a)), Q(y, y)\}$
5.  $W = \{Q(a, x, f(x)), Q(a, y, y)\}$
6.  $W = \{Q(x, y, z), Q(u, h(v, v), u)\}$

### Exercice 3.8.4.6

Considérons le langage du premier ordre  $L = \{R, S, =, f, g, a\}$  où

- R, S et = désignent des symboles prédicats binaires
- f et g désignent deux symboles de fonctions, f d'arité 1 et g d'arité 2
- a désigne un symbole de constante

Soit  $F$  la formule

$$\exists y(\forall z(R(x, w) \rightarrow f(x) = y)) \wedge (z = g(x, y)) \wedge \forall z\exists x(f(z) = g(x, x))$$

Pour chacun des termes suivants  $t$ , écrire  $\sigma[x|t]$  (c'est-à-dire la substitution de  $t$  à la variable  $x$  dans  $\sigma$ ).

- $t = g(a, a)$
- $t = g(f(y), w)$
- $t = g(x, y)$
- $t = g(x, z)$

### Exercice 3.8.4.7

Soit l'énoncé suivant :

$H_1$  : Les personnes qui ont la grippe ne doivent pas aller au travail.

$H_2$  : Les personnes qui ont de la fièvre et qui toussent ont la grippe.

$H_3$  : Ceux qui ont une température supérieure à  $38^\circ$  ont de la fièvre.

$H_4$  : Ahmed tousse et a une température supérieure à  $38^\circ$ .

Et  $C$  : Ahmed ne doit pas aller au travail.

On veut montrer en utilisant la méthode de la résolution que  $\{H_1, H_2, H_3, H_4\} \models C$

1. Modéliser en logique des prédicats l'énoncé ci-dessus en utilisant les prédicats suivants :

-*grippe*( $x$ ) :  $x$  a la grippe

-*travail*( $x$ ) :  $x$  doit aller au travail

-*fièvre*( $x$ ) :  $x$  a de la fièvre

-*tousse*( $x$ ) :  $x$  tousse

-*temp*( $x, t$ ) :  $x$  a la température  $t$

-*sup*( $x; y$ ) :  $x$  est supérieur à  $y$

On utilisera également les constantes suivantes :

-38

-Ahmed

2. Mettre sous forme prénexe les énoncés ainsi que la négation de la proposition à prouver.
3. Éliminer les quantificateurs.
4. Mettre le problème sous forme de clauses.
5. Montrer que "Ahmed ne doit pas aller au travail" en utilisant la méthode de résolution avec variables en précisant les différentes étapes.

### *Exercice 3.8.4.8*

Prouver la validité de la formule suivante :

$$\exists x P(x) \forall x Q(x) \rightarrow \exists x (P(x) \wedge Q(x))$$

Par résolution au premier ordre. On donnera la forme clausale de la négation de la formule et la preuve par résolution de la clause vide.

### Références bibliographiques

- [1] Devismes, S., Lafourcade, P., & Lévy, M. (2012). *Logique et démonstration automatique: introduction à la logique propositionnelle et à la logique du premier ordre*. Ellipses.
- [2] Bournez, O. (2011). *Fondements de l'Informatique: Logique, Modèles, Calculs*. École polytechnique.
- [3] Arun-Kumar, S. (2002). *Introduction to Logic for Computer Science*.
- [4] Reeves, S., & Clarke, M. (1990). *Logic for computer science*. Wokingham: Addison-Wesley.
- [5] Pelletier, F. J. (2000). *A history of natural deduction and elementary logic textbooks. Logical consequence: Rival approaches, 1*, 105-138.
- [6] Kleene, S. C. (2002). *Mathematical logic*. Courier Corporation.
- [7] Kreisel, G., & Krivine, J. L. (1969). *Elements of mathematical logic*. North-Holland Publishing Company Amsterdam.
- [8] [https://fr.wikipedia.org/wiki/Logique\\_mathématique](https://fr.wikipedia.org/wiki/Logique_mathématique)

## Annexe A : Solutions des exercices

### 1. Solution des exercices du chapitre 2 : logique des propositions

#### 1.1 Solutions des exercices : Syntaxe de la logique des propositions

##### *Exercice 2.8.1.1\_Solution*

- (a) Quel est votre nom? N'est pas une fbf
- (b) Fermez la porte! N'est pas une fbf
- (c) L'herbe est verte. une fbf
- (d) L'herbe est rouge. une fbf
- (e) C'est faux. N'est pas une fbf
- (f) Je suis honnête. une fbf
- (g) Vous ne devez pas tricher. N'est pas une fbf

##### *Exercice 2.8.1.2\_Solution*

- a) Pour la formule  $((A \vee (\neg B)) \wedge (C \vee D))$  on peut construire l'arbre syntaxique, donc une fbf
- c) Pour la formule  $(A \vee B)(\wedge \vee C)$  il n'est pas possible de construire l'arbre syntaxique (deux connecteurs binaire  $(\wedge \vee C)$  pour un seul opérande !! donc n'est pas une fbf.

##### *Exercice 2.8.1.3\_Solution*

Exemple de propositions différentes en ajoutant convenablement des parenthèses à  $p \wedge q \rightarrow r \wedge p$

$$(p \wedge q) \rightarrow (r \wedge p)$$

$$p \wedge (q \rightarrow r) \wedge p$$

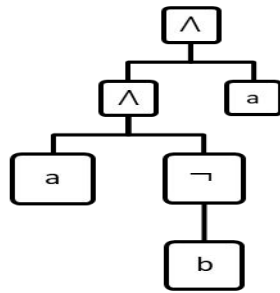
$$((p \wedge q) \rightarrow r) \wedge p$$

$$p \wedge ((q \rightarrow r) \wedge p)$$

##### *Exercice 2.8.1.4\_Solution*

$$a \wedge b((a \wedge \neg b) \wedge a)$$



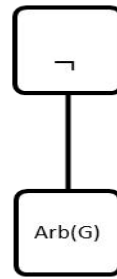


*Exercice 2.8.1.5\_Solution*

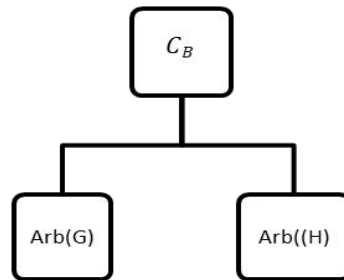
Définir  $arb^{-1}$  la fonction qui à une arbor. associe une fbf

$arb^{-1} : ARBOS : \rightarrow Prop(S)$

- Si arbre est réduit à un point étiqueté par  $p$  alors la fbf correspondante est :  $F = p$



- Si arbre est de la forme  
alors la fbf correspondante est :  $F = \neg (Arb^{-1}(G))$

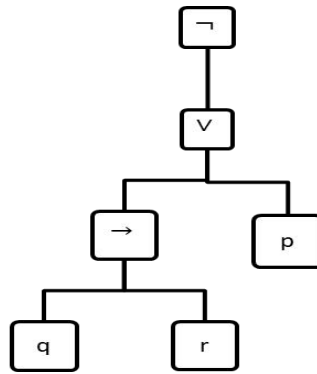


- Si arbre est de la forme  
alors la fbf correspondante est :  $F = ((arb^{-1}(G)) C_B (Arb^{-1}(H)))$

*Exercice 2.8.1.6\_Solution*

On considère  $F1 = \neg((q \rightarrow r) \vee p)$

1. Arbre syntaxique :

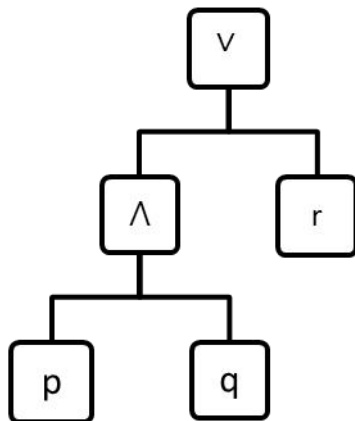
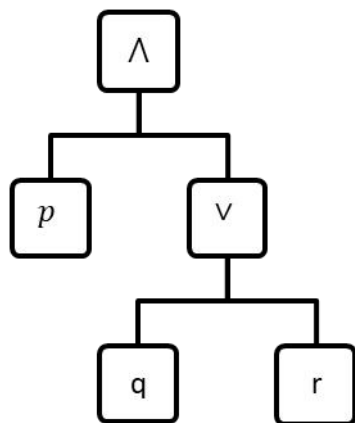


2. Expression fonctionnelle préfixée de la formule :  $\neg \vee \rightarrow q r p$
  3. une expression infixée débarrassée des parenthèses inutiles  $F1 = \neg((q \rightarrow r) \vee p)$
  4. sous-formules :  $\{ q, r, p, q \rightarrow r, (q \rightarrow r) \vee p, \neg((q \rightarrow r) \vee p) \}$
  5. les symboles propositionnels avec leur occurrence :  $q(1), r(1), p(1)$
- Idem pour F3 et F4.

*pour*  $F2 = (((p \wedge q) \neg r) \rightarrow p)$  on peut pas construire l'arbre syntaxique car  $F2$  n'est une fbf, une correction possible :  $F2 = (((p \wedge q) \wedge \neg r) \rightarrow p)$

**Exercice 2.8.1.7\_Solution**

$$p \wedge q \vee r \quad \left\{ \begin{array}{l} ? p \wedge (q \vee r) \\ (p \wedge q) \vee r \end{array} \right\}$$



*Exercice 2.8.1.8\_Solution*

$\neg p$  ;  $\neg p \wedge r$  ;  $\neg p \wedge r$  ;  $(s \rightarrow (\neg p \wedge r)) \leftrightarrow (\neg q \wedge s)$  : sous formules

*Exercice 2.8.1.9\_Solution*

Soit  $S$  un ensemble dénombrable de symboles propositionnels, l'ensemble des deux constantes  $\{\top, \perp\}$ , un ensemble de connecteurs  $C = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ , et un jeu de parenthèses  $D = \{(, )\}$

On définit par induction **PROP(S)**, l'ensemble des fbfs de la logique des propositions (ou **propositions**) construites sur  $S$  (donc sur l'univers des mots de  $(S \cup C \cup D)^*$ ):

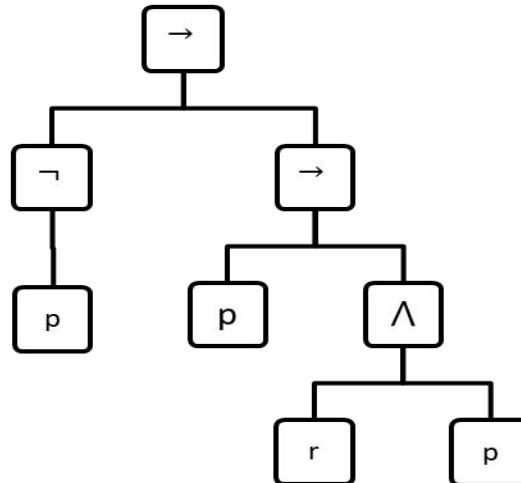
- **(Base)** **PROP(S)** contient  $S \cup \{\top, \perp\}$
  - **(Construction)** Si  $P$  et  $Q \in \mathbf{PROP}(S)$  (i.e. sont des fbf)
    - Règle 1 :  $\neg P \in \mathbf{PROP}(S)$
    - Règle 2 :  $(P C_B Q) \in \mathbf{PROP}(S)$
- $(C_B = \{\wedge, \vee, \rightarrow, \leftrightarrow, \otimes\}$  **Connecteurs binaires**)

**Exercice 2.8.1.10\_Solution**

Soit  $\Sigma$  l'ensemble des symboles propositionnels et  $PROP$  l'ensemble des fbf construites à partir de  $\Sigma$  et soit A et B les fbf suivantes :

$$A = (\neg p \rightarrow (p \rightarrow (r \wedge p)))$$

a) l'arbre syntaxique associé à A :



b) Calcul les ensembles  $SymbProp(A)$  :

$$\begin{aligned}
 SymbProp(A) &= SymbProp((\neg p \rightarrow (p \rightarrow (r \wedge p)))) \\
 &= SymbProp(\neg p) \cup SymbProp(p \rightarrow (r \wedge p)) \\
 &= SymbProp(\neg p) \cup SymbProp(p) \cup SymbProp(r \wedge p) \\
 &= SymbProp(p) \cup SymbProp(p) \cup SymbProp(r) \cup SymbProp(p) \\
 &= \{p, p, r, p\} = \{p, r\}
 \end{aligned}$$

c) Définition par induction de prof :

**Prof**:  $Prop \rightarrow \mathbb{N}$

**Base** : Si  $F = p, (p \in S), Prof(F) = 0$

**Construction** :

**Règle 1** : Si  $F = \neg G, Prof(F) = 1 + Prof(G)$

**Règle 2** : Si  $F = G C_B H, Prof(F) = 1 + \text{Max}(Prof(G), Prof(H))$

$$C_B \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$$

d) Calculer  $\text{Prof}(A)$

$$\begin{aligned}
 \text{Prof}(A) &= \text{Prof}((\neg p \rightarrow (p \rightarrow (r \wedge p)))) \\
 &= 1 + \max(\text{Prof}(\neg p), \text{Prof}(p \rightarrow (r \wedge p))) \\
 &= 1 + \max(\text{Prof}(p), 1 + \max(\text{Prof}(p), \text{Prof}(r \wedge p))) \\
 &= 1 + \max(\text{Prof}(p), 1 + \max(\text{Prof}(p), 1 + \\
 &\quad \max(\text{Prof}(r), \text{Prof}(p)))) \\
 &= 1 + \max(0, 1 + \max(0, 1 + \max(0, 0))) = 3
 \end{aligned}$$

**Exercice 2.8.1.11\_Solution**

a)  $\text{Sub}(A) = \{p, q, r, \neg r, p \wedge q, \neg r \vee p, ((p \wedge q) \leftrightarrow (\neg r \vee p))\}$  donc

$$|\text{Sub}(A)| = 7$$

$$\text{NBC}(A) = 4, \quad 2 * \text{NBC}(A) + 1 = 2 * 4 + 1 = 9$$

Puisque  $7 < 9$  On a bien  $|\text{Sub}(A)| \leq 2 * \text{NBC}(A) + 1$

b) une définition par induction de  $\text{Sub}$  et  $\text{NBC}$

**Sub: Prop**  $\rightarrow$   $\wp(\mathbf{PROP})$

**Base :** Si  $F = p, (p \in S), \text{Sub}(F) = \{p\}$

**Construction :**

**Règle 1 :** Si  $F = \neg G, \text{Sub}(F) = \{F\} \cup \text{Sub}(G)$

**Règle 2 :** Si  $F = G C_B H, \text{Sub}(F) = \{F\} \cup \text{Sub}(G) \cup \text{Sub}(H)$

$$C_B \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$$

**NBC: Prop**  $\rightarrow$   $\mathbb{N}$

**Base :** Si  $F = p, (p \in S), \text{NBC}(F) = 0$

**Construction :**

**Règle 1 :** Si  $F = \neg G, \text{NBC}(F) = 1 + \text{NBC}(G)$

**Règle 2 :** Si  $F = G C_B H, \text{NBC}(F) = 1 + \text{NBC}(G) + \text{NBC}(H)$

$$C_B \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$$

c) Montrer que pour toute fbf  $F: |\text{Sub}(F)| \leq 2 * \text{NBC}(F) + 1$

-la propriété est vérifiée pour A

-On suppose qu'elle est vraie pour  $F$  avec  $n$  connecteurs ( $\text{NBC}(F) = n$ ) et

$$|\text{Sub}(F)| = k$$

$$|\text{Sub}(F)| \leq 2 * \text{NBC}(F) + 1 \quad \text{et donc } k \leq 2 * n + 1$$

Si On ajoute un connecteur binaire à  $F: F' = F C_B p$

Alors on aura :  $|\text{Sub}(F')| = |\text{Sub}(F)| + 2 = k + 2 = k'$  et

$$\text{NBC}(F') = \text{NBC}(F) + 1 = n + 1 \quad ??? \quad k' \leq 2 * n' + 1 \quad \text{avec } n' = n +$$

$$1 \quad \text{et } k' = k + 2$$

On a par supposition :  $k \leq 2 * n + 1$

$$k + 2 \leq 2 * n + 1 + 2 \rightarrow k + 2 \leq 2 * (n + 1) + 1 \rightarrow k + 2 \leq 2 * n' + 1 \text{ donc } k' \leq 2 * n' + 1$$

Si On ajoute un connecteur unaire à F

$F' = F$  + un connecteur unaire pour un symbol positif:

Alors on aura :  $|Sub(F')| = |Sub(F)| + 1 = k + 1 = k'$  et

$nbv(F') = nbv(F) + 1 = n + 1$  ???  $k' \leq 2 * n' + 1$  avec  $n' = n + 1$  et  $k' = k + 1$

On a par supposition :  $k \leq 2 * n + 1$

$$k + 1 \leq 2 * n + 1 + 1 \leq 2 * n + 1 + 2 \rightarrow k + 2 \leq 2 * (n + 1) + 1 \rightarrow k + 2 \leq 2 * n' + 1$$

donc  $k' \leq 2 * n' + 1$

On bien

$$|Sub(F)| \leq 2 * nbv(F) + 1$$

### 1.2 Solutions des exercices : Sémantique de la logique des propositions

#### Exercice 2.8.2.1 \_Solution

$$V(F, I) = V(\neg((p \vee (q \leftrightarrow p)) \rightarrow q), I)$$

$$Non(V((p \vee (q \leftrightarrow p)) \rightarrow q, I))$$

$$Non(SiAlors(ET(V(p, I), V(A2, I)), V(q, I)))$$

$$Non(SiAlors(ET(V(p, I), SSI(V(q, I), V(p, I))), V(q, I)))$$

$$Non(SiAlors(ET(I(p), SSI(I(q), I(p))), I(q)))$$

$$Non(SiAlors(ET(0, SSI(1, 0)), 1))$$

$$Non(SiAlors(ET(0, 0), 1))$$

$$Non(SiAlors(0, 1))$$

$$Non(1) = 0$$

#### Exercice 2.8.2.2 \_Solution

a)  $(\neg(a \wedge (b \vee c))) = (\neg a \vee (\neg b \wedge \neg c))$

b)  $(\neg[(\neg(a \wedge b) \vee (\neg d)) \wedge e \vee f]) = ((a \wedge b) \wedge d) \vee \neg e \wedge \neg f$

c) **Idem pour la formule**

d) **Idem pour la formule**

#### Exercice 2.8.2.3 \_Solution

1. les interprétations qui donnent même la valeur à  $p \wedge q$  et  $p \rightarrow q$  :

$$I_1(p) = 1; I_1(q) = 0 \text{ et } I_2(p) = 1; I_2(q) = 1$$

$p$	$q$	$p \wedge q$	$p \rightarrow q$
0	0	0	1
0	1	0	1
1	0	0	0
1	1	1	1

2. les modèles de la formule  $(p \wedge q) \leftrightarrow (p \rightarrow q)$  :

$$I_1(p) = 1; I_1(q) = 0 \text{ et } I_2(p) = 1; I_2(q) = 1$$

3. la formule n'est pas valide et n'est pas insatisfiable ; elle est satisfiable

**Exercice 2.8.2.4\_Solution**

Soit la formule  $A = p \wedge (\neg q \rightarrow (q \rightarrow p))$

1.  $V(A, I)$  ?

$$\begin{aligned} V(A, I) &= V(p \wedge (\neg q \rightarrow (q \rightarrow p)), I) \\ &= Et(V(p, I), V((\neg q \rightarrow (q \rightarrow p)), I)) \\ &= Et(I(p), SiAlors(V(\neg q, I), V(q \rightarrow p, I))) \\ &= Et(I(p), SiAlors(Non(V(q, I), SiAlors(V(q, I), V(p, I)))) \\ &= Et(I(p), SiAlors(Non(I(q), SiAlors(I(q), I(p)))) \end{aligned}$$

a)  $I(p) = 0$  et  $I(q) = 1$ ;

$$\begin{aligned} V(A, I) &= Et(0, SiAlors(Non(1), (SiAlors(1, 0))) \\ &= Et(0, SiAlors(0, 0)) \\ &= Et(0, 1) \\ &= 0 \end{aligned}$$

b)  $I(p) = 0$ ;

$$V(A, I) = Et(0, SiAlors(Non(I(q), SiAlors(I(q), 0))) = 0$$

c)  $I(q) = 1$ ;

$$\begin{aligned} V(A, I) &= Et(I(p), SiAlors(Non(1), (SiAlors(1, I(p)))) \\ &= Et(I(p), 1) \\ &= I(p) \end{aligned}$$

d) on ne sait rien sur  $I(p)$  et  $I(q)$ .

D'après a) et b) et c)

## Cours-Logique Mathématique

---

### Exercice 2.8.2.5\_Solution

$$F = (\neg p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge r) \vee (p \wedge q \wedge r)$$

$$F = (p \vee \neg q \vee r) \wedge (\neg p \vee q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee \neg q \vee r)$$

### Exercice 2.8.2.6\_Solution

$$p \vee \neg p$$

- 01 symboles propositionnels donc  $2^1$  d'interprétations différentes

$$I_1(p) = 0 \quad \text{et} \quad I_2(p) = 1$$

- Les valeurs de vérité

$p$	$p \vee \neg p$
0	1
1	1

$$p \wedge \neg p$$

- 02 symboles propositionnels donc  $2^2$  d'interprétations différentes

$$I_1(p) = 0 \quad \text{et} \quad I_1(q) = 0$$

$$I_2(p) = 0 \quad \text{et} \quad I_2(q) = 1$$

$$I_3(p) = 1 \quad \text{et} \quad I_3(q) = 0$$

$$I_4(p) = 1 \quad \text{et} \quad I_4(q) = 1$$

- Les valeurs de vérité

$p$	$q$	$\neg q$	$p \wedge \neg q$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

### Exercice 2.8.2.7\_Solution

- o  $F1 = p$  et  $F2 = \neg\neg p$

- Les valeurs de vérité

$p$	$\neg p$	$\neg\neg p$
0	1	0
1	0	1



## Cours-Logique Mathématique

---

$\forall I, V(F1, I) = V(F2, I)$  On peut conclure que  $F1 \equiv F2$

○  $F1 = p \rightarrow q$  et  $F2 = \neg p \vee q$

Les valeurs de vérité

$p$	$q$	$\neg p$	$p \rightarrow q$	$\neg p \vee q$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	0
1	1	0	1	1

$\forall I, V(F1, I) = V(F2, I)$  On peut conclure que  $F1 \equiv F2$

### *Exercice 2.8.2.8\_Solution*

$$\begin{aligned}
 F1 &= (p \wedge q) \rightarrow r \equiv \neg(p \wedge q) \vee r \equiv (\neg p \vee \neg q) \vee r \equiv (\neg p \vee \neg q) \vee (r \vee r) \\
 &\equiv (\neg p \vee r) \vee (\neg q \vee r) \equiv (p \rightarrow r) \vee (q \rightarrow r) = F3
 \end{aligned}$$

Donc  $F1 \equiv F3$

De la même manière :

$F2 \equiv F4$  et  $F5 \equiv F7$   $F6 \equiv F8$

### *Exercice 2.8.2.9\_Solution*

$F1 = ((p \rightarrow q) \rightarrow p)$

$p$	$q$	$\neg q$	$p \rightarrow q$	$q \rightarrow p$	$p \wedge q$	$p \rightarrow \neg q$	$F1$	$F2$	$F3$
0	0	1	1	1	0	1	0	1	0
0	1	0	1	0	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0
1	1	0	1	1	1	0	1	1	0

- F1 possède deux modèles et deux contre modèles : donc elle est contingente
- Toutes les interprétations sont des modèles pour F2 : donc elle est valide
- Toutes les interprétations sont des contre modèles pour F3 : donc elle est insatisfiable

## Cours-Logique Mathématique

---

### Exercice 2.8.2.10\_Solution

- a) Si  $F$  est contingente  $\exists I_1, V(F, I_1) = 0$  et  $\exists I_2, V(F, I_2) = 1$  et on a :
- $$V(\neg F, I_1) = \text{Non}(V(F, I_1)) = \text{Non}(0) = 1 \quad \text{et}$$
- $$V(\neg F, I_2) = \text{Non}(V(F, I_2)) = \text{Non}(1) = 0 \quad \text{donc}$$
- $$\exists I_1, V(\neg F, I_1) = 1 \quad \text{et} \quad \exists I_2, V(\neg F, I_2) = 0 \quad \text{alors } \neg F \text{ est contingente et a) est vraie}$$
- b) Par exemple :  $G = p$  et  $H = \neg p$ , on a  $G$  et  $H$  sont contingente et  $G \wedge H$  est instatisfiable et  $G \vee H$  est valide donc b) est fausse
- c)  $G \vee H$  est instatisfiable,  $\forall I, V(G \vee H, I) = 0$ ,  
 $\forall I, OU(V(G, I), V(H, I)) = 0$   
 $OU(V(G, I), V(H, I)) = 0 \text{ SSI } V(G, I) = 0 \text{ et } V(H, I) = 0$   
 Donc c) est vraie
- d) Est fausse (voir le contre-exemple de b))

### Exercice 2.8.2.11\_Solution

Pour montrer que toutes formule est équivalente à une formule n'ayant que le *nand* comme seul connecteur, suffit de montrer que tous connecteur peut être exprimé comme macro du connecteur *nand*

- ( $\neg$  :)

$$\neg p \equiv \neg p \vee \neg p \equiv \neg(p \wedge p) \equiv p \text{ nand } p$$

- ( $\vee$  :)

$$p \vee q \equiv \neg\neg(p \vee q) \equiv \neg(\neg p \wedge \neg q) \equiv (\neg p \text{ nand } \neg q) \\ \equiv (p \text{ nand } p) \text{ nand } (q \text{ nand } q)$$

- ( $\wedge$  :)

$$(p \wedge q) \equiv \neg\neg(p \wedge q) \equiv \neg(p \text{ nand } q) \equiv (p \text{ nand } q) \text{ nand } (p \text{ nand } q)$$

- ( $\rightarrow$  :)

$$(p \rightarrow q) \equiv \neg p \vee q \equiv \neg(p \wedge \neg q) \equiv p \text{ nand } \neg q \equiv p \text{ nand } (q \text{ nand } q)$$

- ( $\leftrightarrow$  :)

$$(p \leftrightarrow q) \equiv (p \rightarrow q) \wedge (q \rightarrow p) \equiv (p \text{ nand } (q \text{ nand } q)) \wedge (q \text{ nand } (p \text{ nand } p)) \\ \equiv ((p \text{ nand } (q \text{ nand } q)) \text{ nand } (q \text{ nand } (p \text{ nand } p))) \text{ nand } ((p \text{ nand } (q \text{ nand } q)) \text{ nand } (q \text{ nand } (p \text{ nand } p)))$$

- La même démarche pour le système  $\{\neg, \rightarrow\}$

### Exercice 2.8.2.12\_Solution

La démonstration se fait par l'absurde.

## Cours-Logique Mathématique

Supposons  $F \rightarrow G$  est une tautologie (i.e. vraie pour toute interprétation), et que  $\neg F$  et  $G$  ne sont pas des tautologies.

$\neg F$  n'est pas une tautologie donc il existe une interprétation  $I_1$  telle  $V(\neg F, I_1) = 0$  donc  $V(F, I_1) = 1$ .

Cette interprétation est définie sur les variables de  $F$ .

$G$  n'est pas une tautologie donc il existe une interprétation  $I_2$  telle  $V(G, I_2) = 0$ . Cette interprétation est définie sur les variables de  $G$ .

Définissons l'interprétation  $I_3$  de  $F \rightarrow G$  par :

-  $I_3(p) = I_1(p)$  si  $p$  est une variable de  $F$

-  $I_3(p) = I_2(p)$  si  $p$  est une variable de  $G$ .

Cette interprétation est parfaitement définie puisque les variables de  $F$  et de  $G$  sont distinctes (Hypothèse de l'énoncé).

On a  $V(F \rightarrow G, I_3) = 0$  car  $V(F, I_3) = V(F, I_1) = 1$  et

$V(G, I_3) = V(G, I_2) = 0$ , ce qui est contradictoire avec l'hypothèse,  $F \rightarrow G$  est une tautologie.

### 1.3 Solutions des exercices : Modélisation en logique des propositions

#### *Exercice 2.8.3.1\_Solution*

-Je vous paierai votre installation de T.V.---- $p$

-installation de T.V marche----- $q$

A :  $q \rightarrow p$

B :  $\neg q$

C :  $\neg p$

#### *Exercice 2.8.3.2\_Solution*

Notons : (avec  $p, q, r$  : variables propositionnelles)

Aïcha est à l'école ----- $p$

Ali est allé à la mosquée----- $q$

Ahmed est rentré chez lui----- $r$

A :  $r \rightarrow q$ .

B :  $p \vee r$ .

C :  $q \rightarrow (p \vee r)$ .

D :  $p \wedge q$ .

E :  $r$ .

### *Exercice 2.3.8.3\_Solution*

Modélisation :

Le chauffeur est coupable ----- $p$

Le gardien est coupable----- $q$

Le crime a eu lieu avant minuit----- $r$

$A : p \vee q \equiv p \vee q$

$B : q \rightarrow r \equiv \neg q \vee r$

$C : \neg r$

Connaitre l'identité du coupable : revient à vérifier la validité des raisonnements :

$\{A, B, C\} \models p$  ? Pour Le chauffeur

$\{A, B, C\} \models q$  ? Pour Le gardien

*Pour Le chauffeur*

$\{A, B, C\} \models p$  SSI  $(A \wedge B \wedge C) \rightarrow p$  est valide SSI  $A \wedge B \wedge C \wedge \neg p$  est insatisfiable

La forme clausale à vérifier l'(in)satisfiabilité est :  $\{C1 = p \vee q, C2 = \neg q \vee r, C3 = \neg r, C4 = \neg p\}$

Appliquons la résolution on obtient :

$Res(C1, C2) = C5 = p \vee r ; Res(C3, C5) = C6 = p ; Res(C4, C6) = \emptyset$

$A \wedge B \wedge C \wedge \neg p$  Est insatisfiable

Donc le chauffeur est coupable.

### *Exercice 2.3.8.4\_Solution*

1.  $(p \wedge q) \rightarrow r$

Ali d'assister aux cours— $p$       Ali d'assister aux TD--- $q$       Ali ait la moyenne--- $r$

2.  $p \leftrightarrow (q \wedge r \wedge s)$

R est une relation d'équivalence--- $p$       R est une relation réflexive--- $q$       R est une relation--  
- $r$

R est une relation transitive--- $s$

3.  $p \rightarrow (q \vee r)$

Ahmed est chez lui-- $p$ , Ahmed lit--- $q$ , Ahmed écoute de la musique--- $r$

4.  $q \rightarrow p$  ou  $\neg q \vee p$

Le sida sera éradiqué--- $p$ , un nouveau vaccin soit découvert— $q$

5.  $(p \wedge q) \rightarrow p$

Il est nécessaire d'avoir du courage— $p$ , Il est nécessaire d'avoir de l'habilité— $q$

Escalader cette paroi--- $r$

### Exercice 2.3.8.5\_Solution

Modélisation :

A.  $b_1 \vee b_2 \vee b_3$

B.  $a_1 \vee a_2 \vee a_3$

C.  $(\neg a_1 \vee \neg b_1) \wedge (\neg a_2 \vee \neg b_2) \wedge (\neg a_3 \vee \neg b_3)$

D.  $\neg b_1$

E.  $(a_1 \vee a_2 \vee a_3) \rightarrow b_3 \equiv (\neg a_1 \vee b_1) \wedge (\neg a_2 \vee b_1) \wedge (\neg a_3 \vee b_1)$

$\{H1, H2, H3, H4, H5\} \rightarrow C$

Posons :

$C1 = b_1 \vee b_2 \vee b_3$       $C2 = a_1 \vee a_2 \vee a_3$       $C3 = a_1 \vee a_2 \vee a_3$       $C4 = (\neg a_1 \vee \neg b_1)$

$C5 = (\neg a_2 \vee \neg b_2)$       $C6 = (\neg a_3 \vee \neg b_3)$       $C7 = \neg b_1$       $C8 = (\neg a_1 \vee b_1)$       $C9 =$

$(\neg a_2 \vee b_1)$       $C10 = (\neg a_3 \vee b_1)$

Pour savoir par exemple s'il y a un bateau dans la case  $a_2$ , on cherche à étudier la validité du raisonnement :

$\{C1, C2, C3, C4, C5, C6, C7, C8, C9, C10\} \rightarrow a_2$

### 1.4 Solutions des exercices : Démonstration en logique des propositions

#### Exercice 2.8.4.1\_Solution

Que pouvez-vous conclure sur les formules bien formées suivantes en utilisant les tables de vérité ?

a)  $(p \rightarrow q) \rightarrow ((p \rightarrow \neg q) \rightarrow \neg p)$

$p$	$q$	$\neg p$	$\neg q$	$p \rightarrow q$	$p \rightarrow \neg q$	$((p \rightarrow \neg q) \rightarrow \neg p)$	$(p \rightarrow q) \rightarrow ((p \rightarrow \neg q) \rightarrow \neg p)$
0	0	1	1	1	1	1	1
0	1	1	0	1	1	1	1
1	0	0	1	0	1	0	1
1	1	0	0	1	0	1	1

On peut conclure que La formule est valide

b)  $(p \rightarrow (q \rightarrow (p \wedge q)))$

## Cours-Logique Mathématique

$p$	$q$	$p \wedge q$	$q \rightarrow (p \wedge q)$	$p \rightarrow (q \rightarrow (p \wedge q))$
0	0	0	1	1
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

On peut conclure que La formule est valide

c)  $((p \wedge q) \rightarrow r) \wedge (p \rightarrow q) \rightarrow (p \rightarrow r)$

Idem table de vérité

d)  $\neg p \rightarrow (\neg(p \rightarrow q))$

Idem table de vérité

e)  $((p \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P))$

f) Idem table de vérité

g)  $p \wedge \neg(P \wedge \neg Q \wedge (Q \vee R))$

Idem table de vérité

### ***Exercice 2.8.4.2\_Solution***

a)  $p \rightarrow (q \rightarrow (p \wedge q))$

$$\equiv \neg p \vee (\neg q \vee (p \wedge q))$$

$$\equiv (\neg p \vee \neg q) \vee (p \wedge q)$$

$$\equiv (\neg p \vee \neg q \vee p) \wedge (\neg p \vee \neg q \vee q)$$

$$\equiv (\neg q \vee \top) \wedge (\neg p \vee \top)$$

$$\equiv \top$$

Valide

b) De la même manière que a)

c)  $\neg p \wedge (\neg(p \rightarrow q))$

$$\equiv \neg p \wedge (\neg(\neg p \vee q))$$

$$\equiv \neg p \wedge (p \wedge \neg q)$$

$$\equiv (\neg p \wedge p) \wedge \neg q$$

$$\equiv \perp \wedge \neg q$$

$$\equiv \perp$$

Insatisfiable

d) De la même manière que a)

- e) De la même manière que a)  
 f) De la même manière que a)

### **Exercice 2.8.4.3\_Solution**

1) Mise sous forme normale disjonctive

$$\begin{aligned} \text{a) } p \rightarrow ((q \wedge r) \rightarrow s) &= \neg p \vee (\neg(q \wedge r) \vee s) \\ &= \neg p \vee ((\neg q \vee \neg r) \vee s) \\ &= \neg p \vee \neg q \vee \neg r \vee s \end{aligned}$$

$$\text{b) } (p \wedge q) \rightarrow r = \neg(p \wedge q) \vee r = \neg p \vee \neg q \vee r$$

2) Mettre sous forme normale conjonctive

$$\begin{aligned} \text{a) } (p \wedge \neg q) \vee (\neg p \wedge q) &= (p \vee (\neg p \wedge q)) \wedge (\neg q \vee (\neg p \wedge q)) \\ &= (p \vee \neg p) \wedge (p \vee q) \wedge (\neg q \vee \neg p) \wedge (\neg q \vee q) \\ &= (p \vee q) \wedge (\neg q \vee \neg p) \end{aligned}$$

$$\begin{aligned} \text{b) } \neg(p \rightarrow q) \vee (p \wedge q) &= \neg(\neg p \vee q) \vee (p \wedge q) \\ &= (p \wedge \neg q) \vee (p \wedge q) \\ &= (p \vee (p \wedge q)) \wedge (\neg q \vee (p \wedge q)) \\ &= (p \vee p) \wedge (p \vee q) \wedge (\neg q \vee p) \wedge (\neg q \vee q) \\ &= p \wedge (p \vee q) \wedge (\neg q \vee p) \end{aligned}$$

### **Exercice 2.8.4.4\_Solution**

Trouver la clause résolvente dans les cas suivants :

$$\text{a) } C1 = \neg q \vee p \quad C2 = r \vee \neg p \vee s$$

$$\text{Res}(C1, C2) = \text{Res}(\neg q \vee \mathbf{p}, r \vee \neg \mathbf{p} \vee s) = \neg q \vee r \vee s$$

$$\text{b) } C1 = \neg q \vee p \quad C2 = q$$

$$\text{Res}(C1, C2) = \text{Res}(\neg \mathbf{q} \vee p, \mathbf{q}) = \neg p$$

$$\text{c) } C1 = \neg p \vee \neg q \quad C2 = p \vee s \vee \neg r$$

$$\text{Res}(C1, C2) = \text{Res}(\neg \mathbf{p} \vee \neg q, \mathbf{p} \vee s \vee \neg r) = \neg q \vee s \vee \neg r$$

$$\text{d) } C1 = p \vee q \quad C2 = r \vee p$$

$$\text{Res}(C1, C2) = \text{Res}(p \vee q, r \vee p) : \text{pas de résolvente}$$

### **Exercice 2.8.4.5\_Solution**

$$\text{a) } ((p \rightarrow q) \wedge (p \vee r)) \rightarrow (q \vee r)$$

$$F = ((p \rightarrow q) \wedge (p \vee r)) \rightarrow (q \vee r)$$

$$\equiv \neg((\neg p \vee q) \wedge (p \vee r)) \vee (q \vee r)$$

$$\equiv (\neg(\neg p \vee q) \vee \neg(p \vee r)) \vee (q \vee r)$$

## Cours-Logique Mathématique

$$\begin{aligned}
 &\equiv ((p \wedge \neg q) \vee (\neg p \wedge \neg r)) \vee (q \vee r) \\
 &\equiv ((p \vee \neg p) \wedge (p \vee \neg r) \wedge (\neg q \vee \neg p) \wedge (\neg q \vee \neg r)) \vee (q \vee r) \\
 &\equiv ((p \vee \neg r) \wedge (\neg q \vee \neg p) \wedge (\neg q \vee \neg r)) \vee (q \vee r) \\
 &\equiv (p \vee \neg r \vee q \vee r) \wedge (\neg q \vee \neg p \vee q \vee r) \wedge (\neg q \vee \neg r \vee q \vee r) \\
 &\equiv (p \vee \neg r \vee q \vee r) \wedge (\neg q \vee \neg p \vee q \vee r) \wedge (\neg q \vee \neg r \vee q \vee r) \\
 &\equiv \top
 \end{aligned}$$

$F \equiv \top$  Donc  $F$  est valide

Ou bien avec la méthode de résolution :

$$\neg F = \neg(((p \rightarrow q) \wedge (p \vee r)) \rightarrow (q \vee r))$$

Mise sous forme clausale de  $\neg F$

$$\begin{aligned}
 &\equiv \neg(\neg((\neg p \vee q) \wedge (p \vee r)) \vee (q \vee r)) \\
 &\equiv (((\neg p \vee q) \wedge (p \vee r)) \wedge (\neg q \wedge \neg r)) \\
 &\equiv (\neg p \vee q) \wedge (p \vee r) \wedge \neg q \wedge \neg r
 \end{aligned}$$

$$C1 = (\neg p \vee q) ; C2 = (p \vee r) ; C3 = \neg q ; C4 = \neg r$$

$$F_C = \{C1, C2, C3, C4\}$$

$$Res(C1, C2) = \{q, r\} = C5 ; Res(C5, C3) = \{r\} = C6 ; Res(C6, C4) = \emptyset$$

$\neg F \vdash_{Res} \emptyset$  alors  $\neg F$  est (in)satisfiable donc  $F$  est valide

Ou encore avec la table de vérité :

$$((p \rightarrow q) \wedge (p \vee r)) \rightarrow (q \vee r)$$

$p$	$q$	$r$	$p \rightarrow q$	$p \vee r$	$q \vee r$	$(p \rightarrow q) \wedge (p \vee r)$	Formule
0	0	0	1	0	0	0	1
0	0	1	1	1	1	1	1
0	1	0	1	0	1	0	1
0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	1	0	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

$\forall I, V(F, I) = 1$  donc  $F$  est valide

- b) de la même manière que a)
- c) de la même manière que a)



## 2. Solution des exercices du chapitre 3 : logique des prédicats

### 2.1 Solutions des exercices : Syntaxe de la logique des prédicats

#### *Exercice 3.8.1.1\_Solution*

-Dites si les « écritures » suivantes correspondent à des termes :

1.  $y$  Terme
2.  $c$  Terme
3.  $x \vee a$
4.  $f(x)$  Terme
5.  $f(x, a)$  Terme
6.  $f(a \wedge b)$
7.  $f(y(a), b)$
8.  $g(h(x), f(x, y), c)$  Terme
9.  $f(f(f(f(x))))$  Terme

#### *Exercice 3.8.1.2\_Solution*

-Dites si les « écritures » suivantes correspondent à des formules atomiques :

1.  $P$  formule atomique
2.  $P(a)$  formule atomique
3.  $P(x)$  formule atomique
4.  $P(a \vee b)$
5.  $Q(x, y, b)$  formule atomique
6.  $R(f(x))$  formule atomique
7.  $Q(g(f(a)), b)$  formule atomique
8.  $P(b) \vee Q(a)$

#### *Exercice 3.8.1.3\_Solution*

-Dites si les « formules » suivantes sont des formules bien formées :

1.  $a \vee b$  formule bien formée
2.  $P$  formule bien formée
3.  $P \vee R$
4.  $\forall x P(x)$  formule bien formée
5.  $\forall x \vee Q(y)$
6.  $\exists x P(x) \vee Q(y)$  formule bien formée
7.  $Q(x, y, b) \rightarrow P(x) \vee P(a) \wedge R$  formule bien formée
8.  $\forall b Q(x, y, b) \rightarrow P(x) \vee P(a) \wedge R$

## Cours-Logique Mathématique

---

9.  $\exists y Q(x, y, b) \rightarrow P(x) \vee P(a) \wedge R$  formule bien formée

10.  $\forall z \forall x (\exists y Q(x, y, b) \rightarrow P(z) \vee P(a) \wedge R)$  formule bien formée

### Exercice 3.8.1.4\_Solution

1. Une variable est dite libre dans une formule si elle possède au moins une occurrence libre.

Nous considérons le langage du premier ordre  $L = \{R, S, f, b\}$  où  $R$  et  $S$  désignent deux symboles de relation respectivement unaire et binaire,  $f$  désigne un symbole de fonction unaire et  $b$  désigne un symbole de constante.

Soit  $F$  la formule suivante :  $(\forall x \exists y R(f(x), f(y))) \wedge ((\forall z R(x, z)) \Rightarrow S(x))$

2.  $x$  est libre dans la formule car il existe deux occurrences libres (la deuxième et la troisième occurrence quand on lit la formule de gauche à droite. En revanche  $y$  est une variable liée car toutes ses occurrences (une seule) sont liées.
3. On renomme les variables liées qui ont aussi une occurrence libre. Soit ici le  $x$  quantifié universellement. On ne renomme pas les variables libres, on changerait le sens de la formule.

$(\forall u \exists y R(f(u), f(y))) \wedge ((\forall z R(x, z)) \Rightarrow S(x))$

4. Cette substitution ne concerne que l'occurrence libre de  $x$ . Pour éviter la capture de la variable  $z$  (dans le terme  $t = f(z)$ ) Par le quantificateur, on commence par renommer dans la formule  $F$  la variable liée  $z$  en  $v$  puis on fait le remplacement. D'où la formule :  $(\forall x \exists y R(f(x), f(y))) \wedge ((\forall v R(f(z), v)) \Rightarrow S(f(z)))$

### Exercice 3.8.1.5\_Solution

Dans les formules suivantes, dites si les variables utilisées sont des variables libres ou des variables liées.

- $p(x) \vee q(y)$  Libre: x,y
- $\forall x p(x) \vee q(y)$  Libre: y ; liée : x
- $\forall x p(x) \vee q(y, z)$  Libre: x ; liée : y,z
- $\exists z (\forall x p(x, z) \vee q(y, z))$  Libre: y ; liée : x,z
- $\exists z (p(x, z) \vee \exists y q(y, z))$  Libre: x ; liée : y,z
- $\forall x p(x) \vee q(x)$  Libre: x
- $p(x, y) \vee p(y, x) \wedge p(x, x) \rightarrow p(x, y)$
- $\forall x \forall y (p(x, y) \vee p(y, x)) \wedge p(x, x) \rightarrow p(x, y)$
- $\forall x \forall y (p(x, y) \vee p(y, x)) \wedge p(x, x) \rightarrow \exists x \forall y p(x, y)$
- $\forall x \forall y (p(x, y) \vee p(y, x)) \wedge \forall x p(x, x) \rightarrow \exists x \forall y p(x, y)$

## 2.2 Solutions des exercices : Sémantique de la logique des prédicats

### Exercice 3.8.2.1\_Solution

1. Traduire E1 et E2 en formules dans le langage des prédicats

$$E1: \forall x \in \mathbb{N}, \exists y \in \mathbb{N} \ y, (y > x)$$

$$E2: \exists y \in \mathbb{N}, \forall x \in \mathbb{N}, (y > x)$$

2. Validité :

E1 : Pour tout  $x \in \mathbb{N}$ , on prend  $y \in \mathbb{N}$  tq :  $y = x + 4$  alors  $y > x$  donc valide

E2 : Supposons E2 valide donc  $\exists y = y_0$  tq pour tout  $x: y_0 > x$ , puisque  $y_0$  est un entier alors  $y_0 + 1$  est aussi or  $x = y_0 + 1 > y_0$  ce qui est contradictoire avec l'hypothèse initiale, donc E2 n'est pas valide

### Exercice 3.8.2.2\_Solution

On donne tout d'abord les réponses, puis la façon de les calculer :

Interprétation	Totale	Irréflexive	Symétrique
I	1	0	1
J	1	1	1
K	1	1	0

– Evaluons dans I les trois formules

$$\forall x \exists y R(x, y) = R(0,0) = 1$$

$$\forall x \neg R(x, x) = \neg R(0,0) = 0$$

$$\forall x \forall y (R(x, y) \rightarrow R(y, x)) = R(0,0) \rightarrow R(0,0) = 1$$

– Evaluons dans J les trois formules

$$\forall x \exists y R(x, y) = (R(0,0) + R(0,1)). (R(1,0) + R(1,1)) \text{ car } R(0,1) = R(1,0) = 1$$

$$\forall x \neg R(x, x) = \neg R(0,0). \neg R(1,1) = 1, \text{ car } R(0,0) = R(1,1) = 0$$

$$\forall x \forall y (R(x, y) \rightarrow R(y, x)) = (R(0,0) \rightarrow R(0,0)). (R(0,1) \rightarrow R(1,0)). (R(1,0) \rightarrow R(0,1)). (R(1,1) \rightarrow R(1,1)) = 1$$

$$\text{car } R(0,1) = R(1,0) = 1$$

– Evaluons dans K les trois formules :

$$\forall x \exists y R(x, y) = (R(0,0) + R(0,1) + R(0,2)). (R(1,0) + R(1,1) + R(1,2)). (R(2,0) + R(2,1) + R(2,2)) = 1$$

$$\text{car } R(0,1) = R(1,2) = R(2,0) = 1$$

$$\forall x \neg R(x, x) = \neg R(0,0). \neg R(1,1). \neg R(2,2) = 1, \text{ car } R(0,0) = R(1,1) = R(2,2) = 0$$

$$x \forall y (R(x, y) \rightarrow R(y, x)) = (r(0,0) \Rightarrow r(0,0)). (r(0,1) \Rightarrow r(1,0)) \dots = 0$$

$$\text{car } r(0,1) = 1 \text{ et } r(1,0) = 0$$

### Exercice 3.8.2.3\_Solution

- La valeur de vérité de la formule  $F_1$  dépend a priori de la valeur associée à la variable libre  $y$ . Soit  $a$  la valeur de  $y$  (c'est un entier). La valeur de vérité de  $F_1$  sera 1 si pour tout entier  $n$  il existe un entier  $m$  tel que  $m + a = n$ . Pour  $n$  et  $a$  donnés, il suffit de prendre  $m = n - a$ . Donc la valeur de vérité de  $F_1$  est 1, quelle que soit la valeur de  $y$ .
- De même, ici  $y$  est une variable libre. Soit  $a$  sa valeur. On se pose la question de savoir s'il existe un entier  $n$  tel que  $n < a$  et  $a < 2n$ . La valeur de vérité de  $F_2$  est si  $a > 2$ , 0 sinon.
- De même, ici,  $y$  est une variable libre. Soit  $a$  sa valeur. On se pose la question de savoir si pour tout entier  $n$  tel que  $n < a$  et  $2n < a$ . Ceci est vrai si  $a < 2$ . Donc la valeur de vérité de  $F_3$  est 1 si  $a < 2$ , 0 sinon.

## 2.3 Solutions des exercices : Modélisation en logique des prédicats

### Exercice 3.8.3.1\_Solution

Soit à traduire en logique des prédicats le groupe de phrases suivantes :

1. Quiconque sait lire est instruit  
 $\forall x (Lire(x) \rightarrow Instruit(x))$
2. Les dauphins ne sont pas instruits  
 $Dauphin(x) \rightarrow \neg Instruit(x)$
3. Certains dauphins sont intelligents  
 $\exists x (Dauphin(x) \wedge Intelligent(x))$
4. Certains êtres intelligents ne savent pas lire  
 $\exists x Etre\_vivant(x) \wedge \neg Lire(x)$
5. Flipper est un dauphin  
 $Dauphin(Flipper)$
6. Le frère de Flipper est intelligent  
 $Intelligent( (Frere( de Flipper)) )$
7. Rien n'est identique à rien.  
 $\nexists x \nexists y (x = y)$
8. Les baleines sont des mammifères.  
 $\forall x (Baleine(x) \rightarrow Mamm(x))$
9. Les entiers sont pairs ou impairs.  
 $\forall x (Entier(x) \rightarrow (Pair(x) \vee Impair(x)))$
10. Il existe un entier pair

$$\exists x(\text{Entier}(x) \wedge \text{Pair}(x))$$

### Exercice 3.8.3.2\_Solution

$$\forall x(C(x) \rightarrow \text{Courant}(x) \vee \text{Epargne}(x))$$

$$\forall x(\text{Courant}(x) \rightarrow C(x) \wedge \neg \text{Epargne}(x))$$

$$\forall x(\text{Epargne}(x) \rightarrow C(x) \wedge \neg \text{Courant}(x))$$

$$\forall x(C(x) \rightarrow \exists! y(\text{Client}(y) \wedge \text{possede}(y, x)))$$

$$\forall x \forall y(\text{Client}(x), \text{Epargne}(y), \text{possede}(x, y) \\ \rightarrow \exists! y(\text{Epargne}(y) \wedge \text{possede}(x, y)))$$

### Exercice 3.8.3.3\_Solution

Considérons deux personnes appelées Ahmed et Zoubir représentées par des constantes  $a$  et  $z$ , et le film Carnaval Fi Dachra représenté par la constante  $c$ . En utilisant les prédicats

- $F(y)$  :  $y$  est un film

- $H(x)$  :  $x$  est un être humain

- $A(x, y)$  :  $x$  est acteur dans  $y$

- $V(x, y)$  :  $x$  a vu  $y$

et le prédicat  $=$ ,

Formalisation des énoncés :

1. Ahmed a vu Carnaval Fi Dachra

$$V(a, c)$$

2. Quelqu'un a vu Carnaval Fi Dachra

$$\exists x(H(x) \wedge V(x, c))$$

3. Tout le monde a lu Carnaval Fi Dachra

$$\forall x(H(x) \rightarrow V(x, c))$$

4. Tout le monde n'a pas vu Carnaval Fi Dachra

$$\forall x(H(x) \wedge \neg V(x, c))$$

5. Quelqu'un n'a pas vu Carnaval Fi Dachra

$$\exists x(H(x) \wedge \neg V(x, c))$$

6. Ahmed a vu un film

$$\exists y(F(y) \wedge \neg V(a, y))$$

7. Ahmed a vu exactement deux films

$$\exists x \exists y(F(x) \wedge F(y) \wedge V(a, x) \wedge V(a, y) \wedge \neg(x = y) \wedge (\forall z(F(z) \wedge V(a, z)) \\ \rightarrow z = x \vee z = y))$$

8. Zoubir est l'acteur de Carnaval Fi Dachra

$$A(z, c)$$

9. Ahmed a vu tous les films de Zoubir

$$\forall y(F(y) \rightarrow V(a, y))$$

10. Ahmed n'a pas vu tous les films

$$\exists y (F(y) \wedge \neg V(a, y))$$

11. Tout le monde a vu un film de Zoubir

$$\forall x (H(x) \rightarrow (\exists y (F(y) \wedge A(z, y) \wedge V(x, y))))$$

12. Quelqu'un a lu tous les romans de Zoubir

$$\exists x: (H(x) = \forall y (F(y) \wedge A(z, y) \rightarrow V(x; y)))$$

13. Tous ceux qui sont acteurs d'un film ont vu Carnaval Fi Dachra

$$\forall x (H(x) \wedge \exists y (F(y) \wedge A(x, y)) \rightarrow V(x; c))$$

14. Tous les films n'ont pas un seul acteur

$$\neg(\exists x (H(x) \wedge \forall y (F(y) \rightarrow E(x; y))))$$

15. Parmi les films de Zoubir, Ahmed n'a vu que Carnaval Fi Dachra

$$\forall y ((F(y) \wedge A(z; y) \wedge V(a; y)) \rightarrow y = c)$$

### **Exercice 3.8.3.4\_Solution**

Modélisation avec le prédicat:  $A(x, y) : x$  aime  $y$

A.  $\forall x \exists y A(x, y)$

B.  $\exists y \forall x A(x, y)$

C.  $(\forall x \forall y A(x, y) \wedge \neg(\exists x \forall y A(x, y))) \vee (\exists x \forall y A(x, y) \wedge \exists x \forall y \neg A(x, y))$

### **Exercice 3.8.3.5\_Solution**

1.

(a)  $\exists n; \forall m; n \leq m$

(b)  $\neg(\exists n; \forall m; m \leq n) \wedge \forall m; \exists n; (m \leq n \wedge \neg n = m)$

(c)  $\forall n; (Pair(n)) \exists p; \exists q; (n = p + q \wedge Prem(p) \wedge Prem(q))$

(d)  $\forall n; \exists p; (Prem(p) \wedge n \leq p)$

2.

(a) La somme de deux entiers pairs est pair, ce qui est vrai dans le modèle des entiers.

(b) Pour tout entiers  $x$  et  $u$ , il existe  $z$  tel que  $x$  divise  $z$  et  $z$  divise  $y$ . Cette propriété est fausse dans le modèle des entiers, en effet on aurait alors que  $x$  divise  $y$  et il suffit de prendre  $x = 2$

et

$y = 3$  pour que la propriété soit fausse.

3.

(a)  $Pair(x) \equiv \exists y (x = y + y)$

(b)  $Div(y, x) \equiv \exists z (x = y * z)$

(c)  $Prem(x) \equiv (1 + 1 \leq x \wedge \forall y (Div(y, x))(x = y \vee y = 1))$

### **Exercice 3.8.3.6\_Solution**

Modéliser en logique des prédicats l'énoncé :

$$(H_1) \forall x; \text{grippe}(x) \rightarrow \neg \text{travail}(x)$$

$$(H_2) \forall x; \text{fièvre}(x) \wedge \text{tousse}(x) \rightarrow \text{grippe}(x)$$

$$(H_3) \forall x t; \text{temp}(x, t) \wedge \text{sup}(t, 38) \rightarrow \text{fièvre}(x)$$

$$(H_4) \text{tousse}(\text{Pierre}) \wedge \exists t; (\text{temp}(\text{Pierre}, t) \wedge \text{sup}(t, 38))$$

$$(C) \neg \neg(\text{travail}(\text{Pierre}))$$

## 2.4 Solutions des exercices : démonstration en logique des prédicats

### Exercice 3.8.4.1\_Solution

$$a) \forall x P(x) \rightarrow (\exists t Q(t) \vee \exists t C(t)) \equiv \exists x \exists t (\neg P(x) \vee Q(t) \vee \exists t C(t))$$

$$b) \forall x (\forall y P(x, y) \rightarrow \exists z R(x, z)) \equiv \exists x \exists y \exists z (\neg P(x, y) \vee R(x, z))$$

### Exercice 3.8.4.2\_Solution

$$\forall x (P(x) \rightarrow R(x)) \wedge \forall (Q(x) \rightarrow R(x)) \wedge \exists x (P(x)Q(x)) \rightarrow \exists x R(x)$$

On commence par mettre en forme clausale la négation de la formule ci-dessus :

On la met en forme normale :

$$\begin{aligned} & \neg(\forall x(p(x) \rightarrow r(x)) \wedge \forall x(q(x) \rightarrow r(x)) \wedge \exists x(p(x) \vee q(x)) \rightarrow \exists x r(x)) \\ &= \forall x(p(x) \rightarrow r(x)) \wedge \forall x(q(x) \rightarrow r(x)) \wedge \exists x(p(x) \vee q(x)) \wedge \forall x \neg r(x) \\ &= \forall x(\neg p(x) \vee r(x)) \wedge \forall x(\neg q(x) \vee r(x)) \wedge \exists x(p(x) \vee q(x)) \wedge \forall x \neg r(x) \\ & \neg(\forall x(P(x) \rightarrow R(x)) \wedge \forall(Q(x) \rightarrow R(x)) \wedge \exists x(P(x)Q(x)) \rightarrow \exists x R(x)) \\ &\equiv (\forall x(P(x) \rightarrow R(x)) \wedge \forall(Q(x) \rightarrow R(x)) \wedge \exists x(P(x)Q(x)) \rightarrow \exists x R(x)) \end{aligned}$$

On la met sous forme propre :

$$= \forall x(\neg p(x) \vee r(x)) \wedge \forall y(\neg q(y) \vee r(y)) \wedge \exists z(p(z) \vee q(z)) \wedge \forall u \neg r(u)$$

On skolemise et on enlève les quantificateurs universels :

$$(\neg p(x) \vee r(x)) \wedge (\neg q(y) \vee r(y)) \wedge (p(a) \vee q(a)) \wedge \neg r(u)$$

Ce qui donne les quatre clauses :

$$C1 = \neg p(x) \vee r(x); C2 = \neg q(y) \vee r(y); C3 = p(a) \vee q(a); C4 = \neg r(u)$$

De ces quatre clauses, on déduit la clause vide par la preuve :

$$\text{Res}(C1, C3) = \text{Res}(\neg p(x) \vee r(x), p(a) \vee q(a)) = q(a) \vee r(a) = C5; \quad \{x|a\}$$

$$\begin{aligned} \text{Res}(C5, C2) &= \text{Res}(q(a) \vee r(a), \neg q(y) \vee r(y)) = r(a) \vee r(a) = r(a) \\ &= C6; \quad \{y|a\} \end{aligned}$$

$$\text{Res}(C6, C4) = \text{Res}(r(a), \neg r(u)) = r(a) = \emptyset ; \quad \{u|a\}$$

### *Exercice 3.8.4.3\_Solution*

*L'unificateur le plus général*

Soient les deux expressions  $g(h(x), y)$  et  $g(h(f(y)), z)$ .

L'unificateur  $x := f(k(a)), y := k(a), z := k(a)$  est un unificateur mais ce n'est pas l'unificateur le plus général.

En effet l'équation  $g(h(x), y) = g(h(f(y)), z)$  se décompose en :  $h(x) = h(f(y)), y = z$

En décomposant la première équation, on obtient :  $x = f(y), y = z$

En éliminant  $x$  puis  $y$ , on obtient l'unificateur le plus général  $x := f(z), y := z$ .

Si on oriente  $y = z$  en  $z := y$ , on obtient l'autre unificateur plus général  $x := f(y), z := y$ .

*Calcul des unificateurs*

1. L'équation  $f(x, g(h(z), x), y) = f(k(z), y, g(u, z))$  se décompose en :  $x = k(z), g(h(z), x) = y, y = g(u, z)$

Éliminons  $x$ , on obtient :  $x := k(z), g(h(z), k(z)) = y, y = g(u, z)$

Orientons la deuxième équation, puis éliminons  $y$ , on obtient :  $x := k(z), y := g(h(z), k(z)), g(h(z), k(z)) = g(u, z)$

Décomposons la dernière équation, on obtient :

$$x := k(z), y := g(h(z), k(z)), h(z) = u, k(z) = z$$

Il est clair que la dernière équation n'a pas de solution, donc aussi l'équation initiale.

2. L'équation  $f(x, g(h(z), x), y) = f(k(z), y, g(h(j(u)), x))$  se décompose en :

$$x = k(z), g(h(z), x) = y, y = g(h(j(u)), x)$$

Éliminons  $x$ , on obtient :

$$x := k(z), g(h(z), k(z)) = y, y = g(h(j(u)), k(z))$$

Orientons la deuxième équation, puis éliminons  $y$ , on obtient :

$$x = k(z), y := g(h(z), k(z)), g(h(z), k(z)) = g(h(j(u)), k(z))$$

Décomposons la dernière équation, on obtient :

$$x = k(z), y := g(h(z), k(z)), h(z) = h(j(u)), k(z) = k(z)$$

La dernière équation peut être éliminée, et la troisième décomposée, ce qui donne :

$$x = k(z), y := g(h(z), k(z)), z = j(u)$$

En éliminant  $z$ , on obtient :  $x = k(j(u)), y := g(h(j(u)), k(j(u))), z := j(u)$

### *Exercice 3.8.4.4\_Solution*

On met en forme clausale les trois hypothèses, puis la négation de la conclusion :



Mise en forme normale de  $H1$  :

$$\forall x(\neg U(x) \vee V(x) \vee \exists y(S(x, y) \wedge Q(y)))$$

On skolemise et on enlève le quantificateur universel :

$$(\neg U(x) \vee V(x)) \vee (S(x, f(x)) \wedge Q(f(x))).$$

Autrement dit  $H1$  a produit les deux clauses :

$$(C1) \neg U(x) \vee V(x) \vee S(x, f(x))$$

$$(C2) \neg U(x) \vee V(x) \vee Q(f(x))$$

Mise en forme normale de  $H2$  :

$$\exists x (P(x) \wedge U(x) \wedge \forall y(\neg S(x, y) \vee P(y)))$$

On skolemise et on enlève le quantificateur universel :

$$P(a) \wedge U(a) \wedge (\neg S(a, y) \vee P(y))$$

Donc  $H2$  engendre les trois clauses :

$$(C3) P(a)$$

$$(C4) U(a)$$

$$(C5) \neg S(a, y) \vee P(y)$$

Il est évident que  $H3$  engendre la clause :

$$(C6) \neg P(x) \vee \neg V(x)$$

Il est évident que la négation de la conclusion engendre la clause :

$$(C7) \neg P(x) \vee \neg Q(x).$$

Nous obtenons la preuve suivante :

$$Res(C6, C3) = \neg V(a) = C8$$

$$Res(C1, C4) = V(a) \vee S(a, f(a)) = C9$$

$$Res(C8, C9) = S(a, f(a)) = C10$$

$$Res(C4, C2) = V(a) \vee Q(f(a)) = C11$$

$$Res(C6, C3) = \neg V(a) = C12$$

$$Res(C11, C12) = Q(f(a)) = C13$$

$$Res(C7, C5) = \neg Q(x) \vee \neg S(a, x) = C14$$

$$Res(C13, C14) = \neg S(a, f(a)) = C15$$

$$Res(C10, C15) = \perp$$

### *Exercice 3.8.4.5\_Solution*

1.  $W = \{Q(f(a)), g(a), R(y, y)\}$

Ensemble de discordance :  $D0 = \{Q(f(a)), g(a), R(y, y)\}$

Il n'est pas unifiable car pas de substitutions possibles.

2.  $W = \{Q(a), Q(b)\}$

Ensemble de discordance :  $D_1 = \{a, b\}$

Il n'est pas unifiable car on ne peut pas remplacer un terme par un autre

3.  $W = \{Q(a, x), Q(a, a)\}$

Ensemble de discordance :  $D_0 = \{x, a\}$

$k = 0$

$W_0 = W$

$\sigma_0 = \varepsilon$

$\nu_0 = x, t_0 = a$

Cette paire convient car  $\nu_0 = x$  n'apparaît pas dans  $t_0 = a$

$\sigma_1 = \{x|a\}^\circ \varepsilon = \{x|a\}$

$W_1 = \sigma_1^\circ W_0 = \{x|a\}^\circ W_0 = \{Q(a, a)\}$

$W_1$  est un singleton donc est unifiable et donc l'unificateur principale est :

$\sigma_1 = \{x|a\}$

4.  $W = \{Q(f(a), g(a)), Q(y, y)\}$

Ensemble de discordance :  $D_0 = \{f(a), y\}$

Le couple  $(\nu_0, t_0) = (y, f(a))$  convient car  $y$  ne figure pas dans  $f(a)$

$\sigma_1 = \{y|f(a)\}^\circ \varepsilon = \{y|f(a)\}$

$W_1 = \sigma_1^\circ W_0 = \{Q(f(a), g(a)), Q(f(a), f(a))\}$

$W_1$  n'est pas un singleton on continue

$k = 1$

$D_1 = \{g(a), f(a)\}$

*pas de paire qui convient donc  $W$  n'est pas unifiable*

5.  $W = \{Q(a, x, f(x)), Q(a, y, y)\}$

Ensemble de discordance :  $D_0 = \{x, y\}$

Le couple  $(\nu_0, t_0) = (y, x)$

$\sigma_1 = \{y|x\}^\circ \varepsilon = \{y|x\}$

$W_1 = \sigma_1^\circ W_0 = \{Q(a, y, f(y)), Q(a, y, y)\}$

$W_1$  n'est pas un singleton on continue

$k = 1$

$D_1 = \{f(y), y\}$

*pas de paire qui convient donc  $W$  n'est pas unifiable*

6.  $W = \{Q(x, y, z), Q(u, h(v, v), u)\}$

Ensemble de discordance :  $D_0 = \{x, u\}$

Le couple  $(v_0, t_0) = (u, x)$

$$\sigma_1 = \{u|x\}^\circ \varepsilon = \{u|x\}$$

$$W_1 = \sigma_1 \circ W_0 = \{Q(x, y, z), Q(x, h(v, v), x)\}$$

$W_1$  n'est pas un singleton on continue

$$k = 1$$

$$D_1 = \{y, h(v, v)\}$$

Le couple  $(v_1, t_1) = (h(v, v), y)$

$$\sigma_1 = \{y|h(v, v)\}^\circ \varepsilon = \{y|h(v, v)\}$$

$$W_1 = \sigma_1 \circ W_0 = \{Q(x, y, z), Q(x, y, x)\}$$

$$k = 2$$

$$D_2 = \{z, x\}$$

Le couple  $(v_2, t_2) = (x, z)$

$$\sigma_2 = \{z|x\}^\circ \varepsilon = \{z|x\}$$

$$W_2 = \sigma_2 \circ W_2 = \{Q(z, y, z), Q(z, y, z)\}$$

### *Exercice 3.8.4.6\_Solution*

Considérons le langage du premier ordre  $L = \{R, S, =, f, g, a\}$  où

- $R, S$  et  $=$  désignent des symboles prédicats binaires
- $f$  et  $g$  désignent deux symboles de fonctions,  $f$  d'arité 1 et  $g$  d'arité 2
- $a$  désigne un symbole de constante

Soit  $F$  la formule

$$\exists y (\forall z (R(x, w) \rightarrow f(x) = y)) \wedge (z = g(x, y)) \wedge \forall z \exists x (f(z) = g(x, x))$$

Pour chacun des termes suivants  $t$ , écrire  $\sigma[x|t]$  (c'est-à-dire la substitution de  $t$  à la variable  $x$  dans  $\sigma$ ).

- $t = g(a, a)$

$$\exists y (\forall z (R(g(a, a), w) \rightarrow f(g(a, a)) = y)) \wedge (z = g(g(a, a), y)) \wedge \forall z \exists x (f(z) = g(x, x))$$

- $t = g(f(y), w)$

$$\exists t (\forall z (R(g(f(y), w), w) \rightarrow f(g(f(y), w)) = t)) \wedge (z = g(g(f(y), w), y)) \wedge \forall z \exists x (f(z) = g(x, x))$$

On a renommé la variable liée  $y$  en  $t$  dans la formule

- $t = g(x, y)$

$$\exists t (\forall z (R(g(x, y), w) \rightarrow f(g(x, y)) = t)) \wedge (z = g(g(x, y), y)) \wedge \forall z \exists x (f(z) = g(x, x))$$

On a renommé la variable liée  $y$  en  $t$  dans la formule

- $t = g(x, z)$

$$\exists t(\forall z'(R(g(x, z), w) \rightarrow f(g(x, z)) = t)) \wedge (z = g(g(x, z), y)) \wedge \forall z \exists x(f(z) = g(x, x))$$

On a renommé la variable liée  $z$  en  $z'$  dans la formule

### *Exercice 3.8.4.7\_Solution*

Soit l'énoncé suivant :

$H_1$  : Les personnes qui ont la grippe ne doivent pas aller au travail.

$H_2$  : Les personnes qui ont de la fièvre et qui toussent ont la grippe.

$H_3$  : Ceux qui ont une température supérieure à  $38^\circ$  ont de la fièvre.

$H_4$  : Ahmed tousse et a une température supérieure à  $38^\circ$ .

Et  $C$  : Ahmed ne doit pas aller au travail.

On veut montrer en utilisant la méthode de la résolution que  $\{H_1, H_2, H_3, H_4\} \models C$

1. Modéliser en logique des prédicats l'énoncé ci-dessus en utilisant les prédicats suivants :

-*grippe*( $x$ ) :  $x$  a la grippe

-*travail*( $x$ ) :  $x$  doit aller au travail

-*fièvre*( $x$ ) :  $x$  a de la fièvre

-*tousse*( $x$ ) :  $x$  tousse

-*temp*( $x, t$ ) :  $x$  a la température  $t$

-*sup*( $x; y$ ) :  $x$  est supérieur à  $y$

On utilisera également les constantes suivantes :

-38

-Ahmed

Correction :

$$(H_1) \forall x; \text{grippe}(x) \rightarrow \neg \text{travail}(x)$$

$$(H_2) \forall x; \text{fièvre}(x) \wedge \text{tousse}(x) \rightarrow \text{grippe}(x)$$

$$(H_3) \forall x t; \text{temp}(x, t) \wedge \text{sup}(t, 38) \rightarrow \text{fièvre}(x)$$

$$(H_4) \text{tousse}(\text{Pierre}) \wedge \exists t; (\text{temp}(\text{Pierre}, t) \wedge \text{sup}(t, 38))$$

$$(C) \neg \neg(\text{travail}(\text{Pierre}))$$

2. Mettre sous forme préfixe les énoncés ainsi que la négation de la proposition à prouver.

$$(H_1) \forall x; \text{grippe}(x) \rightarrow \neg \text{travail}(x)$$

$$(H_2) \forall x; \text{fièvre}(x) \wedge \text{tousse}(x) \rightarrow \text{grippe}(x)$$

$$(H_3) \forall x \forall t; ((temp(x, t) \wedge sup(t, 38)) \rightarrow fievre(x))$$

$$(H_4) tousse(Pierre) \wedge \exists t; (temp(Pierre, t) \wedge sup(t, 38))$$

$$(C) \neg\neg(travail(Pierre))$$

3. Eliminer les quantificateurs.

Soit  $t_0$  une nouvelle constante :

$$(H_1) grippe(x) \rightarrow \neg travail(x)$$

$$(H_2) (fievre(x) \wedge tousse(x)) \rightarrow grippe(x)$$

$$(H_3) ((temp(x, t) \wedge sup(t; 38)) \rightarrow fievre(x))$$

$$(H_4) tousse(Pierre) \wedge temp(Pierre, t_0) \wedge sup(t_0, 38)$$

$$(C) \neg\neg(travail(Pierre))$$

4. Mettre le problème sous forme de clauses.

$$(C_1) \neg grippe(x) \vee \neg travail(x)$$

$$(C_2) \neg fievre(x) \vee \neg tousse(x) \vee grippe(x)$$

$$(C_3) \neg temp(x, t) \vee \neg sup(t; 38) \vee fievre(x)$$

$$(C_4) tousse(Pierre)$$

$$(C_5) temp(Pierre, t_0)$$

$$(C_6) sup(t_0, 38)$$

$$(C) travail(Pierre)$$

5. Montrer que "Ahmed ne doit pas aller au travail" en utilisant la méthode de résolution avec variables en précisant les différentes étapes.

$$Res(C_1, C) = \neg grippe(x) = C_7$$

$$Res(C_7, C_2) = \neg fievre(x) \vee \neg tousse(x) = C_8$$

$$Res(C_8, C_4) = \neg fievre(x) = C_9$$

$$Res(C_3, C_5) = \neg sup(t; 38) \vee fievre(x) = C_{10}$$

$$Res(C_{10}, C_6) = fievre(Pierre) = C_{10} \text{ (avec } \sigma = \{x/Pierre\})$$

$$Res(C_9, C_{10}) = \emptyset$$

**Exercice 3.8.4.8 \_Solution**

$$\exists x P(x) \wedge \forall x Q(x) \rightarrow \exists x (P(x) \wedge Q(x))$$

On commence par mettre en forme clausale la négation de la formule ci-dessus :

On la met en forme normale :

$$\neg(\exists x P(x) \wedge \forall x Q(x) \rightarrow \exists x (P(x) \wedge Q(x))) \equiv \exists x P(x) \forall x Q(x) \wedge \forall x (\neg P(x) \vee \neg Q(x))$$

On la met sous forme propre :

## Cours-Logique Mathématique

---

$$\equiv \exists x P(x) \forall y Q(y) \wedge \forall z (\neg P(z) \vee \neg Q(z))$$

On skolémise et on enlève les quantificateurs universels :

$$\equiv P(a)Q(y) \wedge (\neg P(z) \vee \neg Q(z))$$

Ce qui donne les trois clauses :

$$C1 = P(a) , C2 = Q(y) , C3 = \neg P(z) \vee \neg Q(z)$$

De ces trois clauses, on déduit la clause vide par la preuve :

$$Res(C1, C3) = \neg Q(a) = C4 \quad (a/z)$$

$$Res(C2, C4) = \emptyset \quad (a/y)$$

Email: ammfellah@yahoo.fr
---------------------------

Merci de me signaler par courrier électronique les (certainement nombreuses) erreurs dans ce document.
--