

FLUORESCIENCES

Informatique

lnf

Joëlle Delacroix
François Barthélémy
Raphaël
Fournier-S'niehotta
Isabelle Gil
Amélie Lambert
Agnès Plateau
Stéphane Rovedakis
Marianne Simonot
Virginie Thion
Emmanuel Waymel

DUNOD

Crédits

Guido van Rossum © Doc Searls

The peasants are revolting © Tower Game Starter Kit 2013 MIT License,
Christer Kaitila (www.myfunkypants.com)

Maurice Wilkes © University of Cambridge, Computer Laboratory Archive
Marc Andreessen © Joi
James Gray © Tony Hey

Droits réservés : malgré nos efforts, il nous a été impossible de joindre certains éditeurs ou ayants-droits pour solliciter leur autorisation de reproduction, mais nous avons naturellement réservé en notre comptabilité des droits usuels.

Création graphique de la couverture : Hokus Pokus Créations
Création graphique de la maquette intérieure : Marse



© Dunod, 2017

11, rue Paul Bert, 92240 Malakoff
www.dunod.com

ISBN 978-2-10-076094-7

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2^e et 3^e al., d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

Table des matières

Se repérer dans le livre	IV
Les selfies des auteurs	VI
Avant-propos	1

PARTIE 1 MATHÉMATIQUES POUR L'INFORMATIQUE 2

1 Logique	4
2 Les ensembles	20
3 Arithmétique	34
4 Calcul matriciel	52
Corrigés	65

PARTIE 2 ALGORITHMIQUE ET PROGRAMMATION 76

5 Programmes et langages	78
6 Les bases de la programmation	94
7 Les fonctions	112
8 Structures de données	136
9 Premiers pas en programmation orientée objet	162
10 Programmation orientée objet avancée	184
11 Algorithmique de graphes	210
Corrigés	241

PARTIE 3 ARCHITECTURE, SYSTÈMES ET RÉSEAUX 254

12 Des premiers ordinateurs à Internet	256
13 Fonctionnement de l'ordinateur	268
14 La programmation assembleur	290
15 Principes des systèmes d'exploitation	310
16 Fonctionnement des réseaux	336
Corrigés	362

PARTIE 4 LES BASES DE DONNÉES 368

17 Découvrir les bases de données	370
18 Le modèle de données relationnel	380
19 L'algèbre relationnelle	404
20 Le langage SQL	428
Corrigés	457
Bibliographie	464
Lexique français-anglais	465
Index	467

Se repérer dans le livre

Ouverture de chapitre

- Prérequis ou questions pour introduire le sujet du chapitre.
 - Le **plan** du cours pour se repérer.
 - Ce que l'on maîtrisera à la fin du chapitre.

Fonctionnement de l'ordinateur

Pour bien démarrer

1. Qu'est-ce qu'un ordinateur ?
2. Dans les 5 Go, il y a environ :
3. Le nombre 12 256 exprimé en base 10 est égal à :
4. L'ensemble des programmes qui sont installés dans une machine, l'utilisateur constitue :

Réponses page XXX

Objectifs de ce chapitre

- Connaitre les principaux composants d'un ordinateur au niveau hardware.
- Comprendre les fonctionnements et le rôle de ces composants.
- Comprendre les principes de la représentation des informations au niveau matériel.

CHAPITRE 13

L'ordinateur est un instrument de la société actuelle, que ce soit dans le cadre d'une activité industrielle, professionnelle ou personnelle. Envoyer un mail, surfer sur le web, compiler un document avec un logiciel de traitement de texte ou exécuter un programme sont des tâches couramment réalisées par les utilisateurs d'ordinateurs. Mais comment ça fonctionne-t-il ? Quels sont les composants qui permettent à l'ordinateur d'effectuer ces tâches ? C'est ce que nous allons voir dans ce chapitre.

Dans ce chapitre nous allons présenter les éléments principaux de la machine matérielle, leur rôle et les principes de leur fonctionnement. Nous introduisons également les concepts liés à la représentation et à la transmission des informations, basée sur le binaire, les systèmes de base et de conversion de représentations.

Le cours

- Le **cours** est illustré par des figures et des exemples et ponctué par des rubriques.
 - Les **Focus** développent les points importants du cours.
 - Les **Méthodes** donnent des conseils méthodologiques.
 - Les **Définitions** à connaître.

1 Rôle et fonctions des systèmes d'exploitation

1.1 Définition

Un système d'exploitation est un ensemble de programmes qui gèrent l'interfaçage entre la matière d'ordinateur et les applications utilisatrices. Il a deux objectifs principaux :

- la construction au dessus du matériau d'une machine virtuelle plus facile d'emploi et plus maniable pour l'utilisateur.
- la prise en charge de la gestion de plus en plus complexe des ressources et de leur partage.

Le diagramme illustre l'architecture d'un système d'exploitation avec les couches suivantes :

- Logiciels utilisateurs** : comprenant le navigateur (IE, Firefox), les programmes bureautiques (Word, OpenOffice), les messagers (Skype, WhatsApp, Outlook, Thunderbird) et les jeux.
- Système d'exploitation** : (Linux, Windows, Unix, MacOs).
- Machine Matérielle** : comprenant le processeur, la mémoire centrale et les périphériques.

Les entrées-sorties sont indiquées à l'entrée et à la sortie du système d'exploitation.

Connue sous le nom de noyau, le système d'exploitation a en charge l'exploitation de la machine pour ou facilier l'accès, la partie et pour l'optimiser.

Les principaux systèmes d'exploitation sont : Windows, Linux, Unix, Mac OS, Android.

1.1.1 Assurer le partage de la machine physique

La machine physique et ses différents composants, tels qu'offrent des mécanismes nécessaires pour faire l'interfaçage entre différents programmes, ne sont pas concus pour appartenir à plusieurs à la fois. C'est pourquoi il faut un système d'exploitation pour créer un environnement intermédiaire qui gère le partage de la machine physique et des ressources matérielles entre les différents programmes. Ce partage des ressources va concerner principalement le processeur, la mémoire physique et les entrées-sorties.

Le rôle de l'administrateur de systèmes d'entrées-sorties. De manière concise, les questions suivantes vont devoir être résolues :

- dans le cadre du partage du processeur : parmi tous les programmes chargés en mémoire allouée, quel doit exécuter ?

• dans le cadre du partage de la mémoire centrale : comment allouer la mémoire centrale aux différents programmes ? Comment donner à deux softwares suffisamment de mémoire pour y placer tous les programmes nécessaires en optimisant le taux d'utilisation du processeur ? Comment assurer la protection* entre ces différents programmes ?

• dans le cadre du partage des périphériques : dans quel ordre traiter les requêtes d'entrées-sorties pour optimiser les transferts ?

1.1.2 Rencontrer la machine physique

Faciliter l'accès à la machine physique constitue le second rôle du système d'exploitation. Il s'agit de faire fonctionner les périphériques matériels primaires attachés à des fonctionnalités qui gèrent elles-mêmes les caractéristiques matérielles sous-jacentes et offrent un service à l'utilisateur : le système d'exploitation gère l'accès à la mémoire, au processeur, aux périphériques, à l'écran et plus encore. Ainsi pour réaliser une opération d'entrées-sorties, l'utilisateur fera appel à une interface programmatique, l'**E/S**, qui sera alors traduite par le système d'exploitation en une commande d'entrées-sorties. Le système d'exploitation est donc la fonction matérielle qui renvoie l'ensemble des entrées-sorties du système d'exploitation à cette primitive qui est née avec les caractéristiques matérielles.

1.2 Fonctions du système d'exploitation

Le système d'exploitation réalise donc une couche logicielle placée entre la machine matérielle et les applications. Le système d'exploitation peut être décrit comme suit :

- La fonctionnalité de gestion du processeur** : le système doit gérer l'allocation de temps aux différents programmes pouvant s'exécuter. Cette allocation de temps est basée sur les besoins et les priorités de chaque programme ou ensemble de programmes. Selon le type de système d'exploitation, l'algorithme d'affectation peut répondre à des objectifs différents (par exemple, offre un temps court d'exécution pour les programmes prioritaires).
- La fonctionnalité de gestion de la mémoire** : le système doit gérer l'allocation de la mémoire physique entre les différents programmes. Il doit également gérer ce que doit être évacué de la mémoire physique et ce qui doit être chargé pour que le chargement puisse se faire dans les meilleures conditions. Les principales méthodes utilisées sont le remplissage et le déplacement.
- La fonctionnalité de gestion des entrées-sorties** : le système doit gérer l'accès aux périphériques, c'est-à-dire faire la liaison entre les appels de haut niveau des programmes utilisateurs (exemple : `getchar()`) et les opérations de hasard de l'unité d'entrée/sortie. La liaison entre périphérique (telle qu'une clé USB) et ce pilote d'entrées-sorties qui assure cette correspondance.

1. Pour proscrire, on entend critiquer ou dénoncer.

2. Par convention, on entend appeler à ce que « un programme donne et accède pas à une page mémoire allouée à un autre programme ».

Figure 11.6 Parcours en largeur et en profondeur de G_1

Figure 11.7 Arborescences liées du parcours de G_2

Remarque

Differentes listes ordonnées peuvent être associées à une même stratégie de parcours depuis longtemps. C'est le cas pour les deux stratégies de parcours en largeur (la liste $P_{\text{L}} = \{x_1, x_2, x_3, x_4, x_5\}$ respectivement $P'_{\text{L}} = \{x_3, x_4, x_5, x_1, x_2\}$) est valide.

Dans l'exemple précédent, nous remarquons que certains arcs ne sont pas utilisés lors d'un parcours en largeur. Ces arcs sont des arcs secondaires. On appelle **arborescence** la partie d'un parcours en largeur qui forme un sous-ensemble d'arcs qui forme un graphe partiel du graphe initial appelé arborescence.

Définition

Dans un graphe orienté, un **chemin** correspond à une suite d'arcs $C = (a_1, a_2, a_3, \dots, a_k)$ pour laquelle pour tout $i = 1, \dots, k - 1$, l'extrémité terminale de l'arc a_i est l'extrémité initiale de l'arc a_{i+1} .

La **longueur** d'un chemin comprend le nombre d'arcs qu'il compose.

Une **arborescence** est un graphe orienté qui comporte un sommet particulier r , appelé **racine**; depuis ce sommet un arc va vers un autre sommet de l'arborescence.

EXEMPLE Arborescences liées aux parcours de G_2

Les arborescences associées aux parcours en profondeur et en largeur de G_2 sont représentées dans la figure 11.7. Elles correspondent aux parcours pour obtenir les listes énumérées de chacun des parcours.

3.2. Stratégie en profondeur d'abord

Le principe de l'algorithme consiste à l'exploration des sommets du graphe en profondeur d'abord pour se réorienter ensuite. Le **sommet courant** (ou en cours de visite) est celui qui a été **précédé le dernier** par tous les sommets pré-visités. À chaque étape, l'algorithme recherche un successeur pour le sommet courant et passe à l'étape suivante du sommet courant. S'il n'en existe pas, alors le sommet courant est **post-visé**. La structure de données support des parcours en profondeur est la **Pile**. La gestion de la Pile nécessite d'appeler à cette fin des opérations telles que **push** dans laquelle on ajoute un élément au sommet et **pop** qui l'utilise et qui enlève l'élément au sommet de la Pile, donc celle empile la dernière.

Méthode Gestion de la structure de données **PILE**

La gestion d'une pile peut être réalisée avec la classe `Stack` du dossier `utilis`, première fois :
Elle est illustrée en figure 11.8. Différentes opérations sont possibles pour manipuler la **Pile** :

- l'opération **depiler** qui consiste à supprimer un élément au sommet de la **Pile** ;
- l'opération **coupler** qui consiste à insérer un élément au sommet de la **Pile** ;

Les selfies des auteurs

Joëlle Delacroix



est maître de conférences au département informatique du CNAM. Ses spécialités concernent le système et l'architecture des ordinateurs. Elle est par ailleurs responsable du DUT informatique du CNAM.

François Barthélémy



est maître de conférences au CNAM et auteur d'un MOOC sur l'initiation à la programmation. Il enseigne la programmation et la théorie des langages. Ses recherches portent sur le traitement automatique des langues (TAL) et sur les techniques de l'information et de la communication dans l'enseignement (TICE).

Raphaël Fournier-S'niehotta



s'est orienté vers la recherche en effectuant un doctorat à l'université Pierre et Marie Curie. Il est depuis 2014 maître de conférences au CNAM où il enseigne les techniques de recherche d'information et les bases de données.

Isabelle Gil



est maître de conférences en mathématiques au CNAM et responsable de la Formation d'ingénieurs en Partenariat d'Exploitation Ferroviaire et de la coordination des mathématiques pour les filières d'ingénieurs de l'antenne alternance du Landy.

Amélie Lambert



est maître de conférences au CNAM. Elle y enseigne la recherche opérationnelle, l'algorithme et la programmation. Elle est membre de l'équipe *Optimisation combinatoire* du laboratoire CEDRIC.

Agnès Plateau



est maître de conférences au CNAM. Elle y enseigne la recherche opérationnelle au travers de l'algorithmique de graphes, la programmation linéaire et l'optimisation combinatoire. Elle est membre de l'équipe *Optimisation combinatoire* du laboratoire CEDRIC.

Stéphane Rovedakis



est maître de conférences en informatique au CNAM Il intervient dans des enseignements de systèmes et réseaux au niveau Licence et Master dans le département informatique du CNAM.

Marianne Simonot



est maître de conférences en informatique à l'IUT de Montrouil- Université de Paris où elle enseigne la programmation et la conception objet. Elle est par ailleurs responsable des études du Département informatique.

Virginie Thion



a d'abord été maître de conférences à l'université Paris Dauphine, puis au CNAM. Depuis 2013, elle est maître de conférences à l'ENSSAT Lannion. Ses enseignements couvrent les systèmes d'exploitation, le développement système Unix, la programmation impérative, fonctionnelle et objet, les bases de données. Ses activités de recherche se concentrent sur la gestion de la qualité des données dans les systèmes d'information.

Emmanuel Waymel



est professeur agrégé de mathématiques à Awty International School, Houston, Texas. Il a enseigné au CNAM, principalement au sein de l'école d'ingénieurs.

Avant propos

L'informatique est présente partout dans notre société. Les parties les plus visibles de cette omniprésence sont le web, les télécommunications, le stockage de données (sur une simple clé ou dans le cloud) ou bien les multiples pilotages automatiques qui sont embarqués dans les objets qui nous entourent.

Toutes ces tâches sont réalisées à l'aide d'applications ou de programmes codés dans un langage de programmation. L'exécution de ces programmes met en œuvre des mécanismes complexes au sein de l'ordinateur.

Le matériel informatique a connu une évolution fulgurante alliant l'augmentation de sa puissance et la miniaturisation de ses composants. Conçu dans les années 50, l'ENIAC (Electronic Numerical Integrator And Computer) est le premier ordinateur entièrement électronique. Il occupait une pièce entière; votre smartphone tient dans votre poche. Il fallait 0,001 seconde à cette grosse machine pour réaliser une multiplication ; votre PC le fait 1000 fois plus vite. L'ENIAC était isolé ; votre tablette est connectée à l'Internet via un réseau sans fil.

En plus de l'exécution d'applications, l'ordinateur et l'informatique permettent une consultation et une conservation d'immenses volumes de données, à l'intérieur des bases de données. Grâce à l'interconnexion des réseaux et à Internet, c'est une quantité d'informations fabuleuse qui est aujourd'hui à notre disposition.

Programmation, fonctionnement de l'ordinateur, réseau-web et bases de données : ce sont les quatre domaines de l'informatique étudiés dans cet ouvrage.

Puisque l'ordinateur compte en base 2 et que la conception de programmes ou de base de données s'appuie sur des outils mathématiques tels que les graphes ou les ensembles, cet ouvrage commence logiquement par ces notions mathématiques.

L'apprentissage du codage avec des programmes écrits en langage Python vient ensuite. Puis nous nous intéressons au fonctionnement de l'ordinateur et à son interaction avec le réseau afin de comprendre comment un programme s'exécute et comment il peut échanger des données via le réseau. Enfin, dans la dernière partie nous présentons les bases de données.

Cet ouvrage va vous permettre d'appréhender ce qu'est l'informatique d'aujourd'hui tout en vous donnant les bases nécessaires pour pouvoir approfondir ce sujet.

Les auteurs remercient Eric Soutil, Geneviève Jomier, Pierre Lointier, Nicolas Jadot et Anna Pappa pour leur relecture attentive de cet ouvrage.



Chirurgiens utilisant l'outil informatique
avant une intervention

Les mathématiques sont partout dans notre entourage dans les domaines scientifiques, dans la vie économique de nos sociétés ou dans de nombreuses applications du quotidien.

Elles sont aussi très présentes dans l'informatique, que ce soit dans la conception et l'analyse d'algorithmes, la sécurité informatique, le fonctionnement des réseaux, l'apprentissage automatique (*machine learning*), la reconnaissance des images....

Les mathématiques et la logique ont également toujours joué un grand rôle dans la conception des ordinateurs afin d'automatiser le maximum de calculs et d'en augmenter la précision et la vitesse. Dans cette partie, vous découvrirez les bases des mathématiques qu'il est important de connaître pour pouvoir comprendre le fonctionnement des ordinateurs.

Mathématiques pour l'informatique

CHAPITRE 1	Logique	4
CHAPITRE 2	Les ensembles	20
CHAPITRE 3	Arithmétique	34
CHAPITRE 4	Calcul matriciel	52

Logique

Pour bien démarrer

1. Quelle est la valeur de vérité de :
“Le 14 juillet est un jour férié” ?
 a. vrai
 b. faux
 C. cela dépend
2. Quelle est la négation de :
“ f est croissante et positive” ?
 a. la fonction est décroissante et négative
 b. la fonction est décroissante ou négative
 c. la fonction n'est pas croissante ou non positive
3. Dans la déduction :
“Si le quadrilatère est un rectangle alors il possède trois angles droits”.
L'affirmation “il possède trois angles droits” est
 a. une condition nécessaire
 b. une condition suffisante
 c. une implication
4. Une écriture avec les quantificateurs de la fonction f est croissante sur \mathbb{R} est :
 a. $\forall x, y \in \mathbb{R}, f(x) < f(y)$
 b. $\forall x \in \mathbb{R}, \exists y \in \mathbb{R}, x < y$ et $f(x) < f(y)$
 c. $\forall x, y \in \mathbb{R}, x < y \Rightarrow f(x) < f(y)$

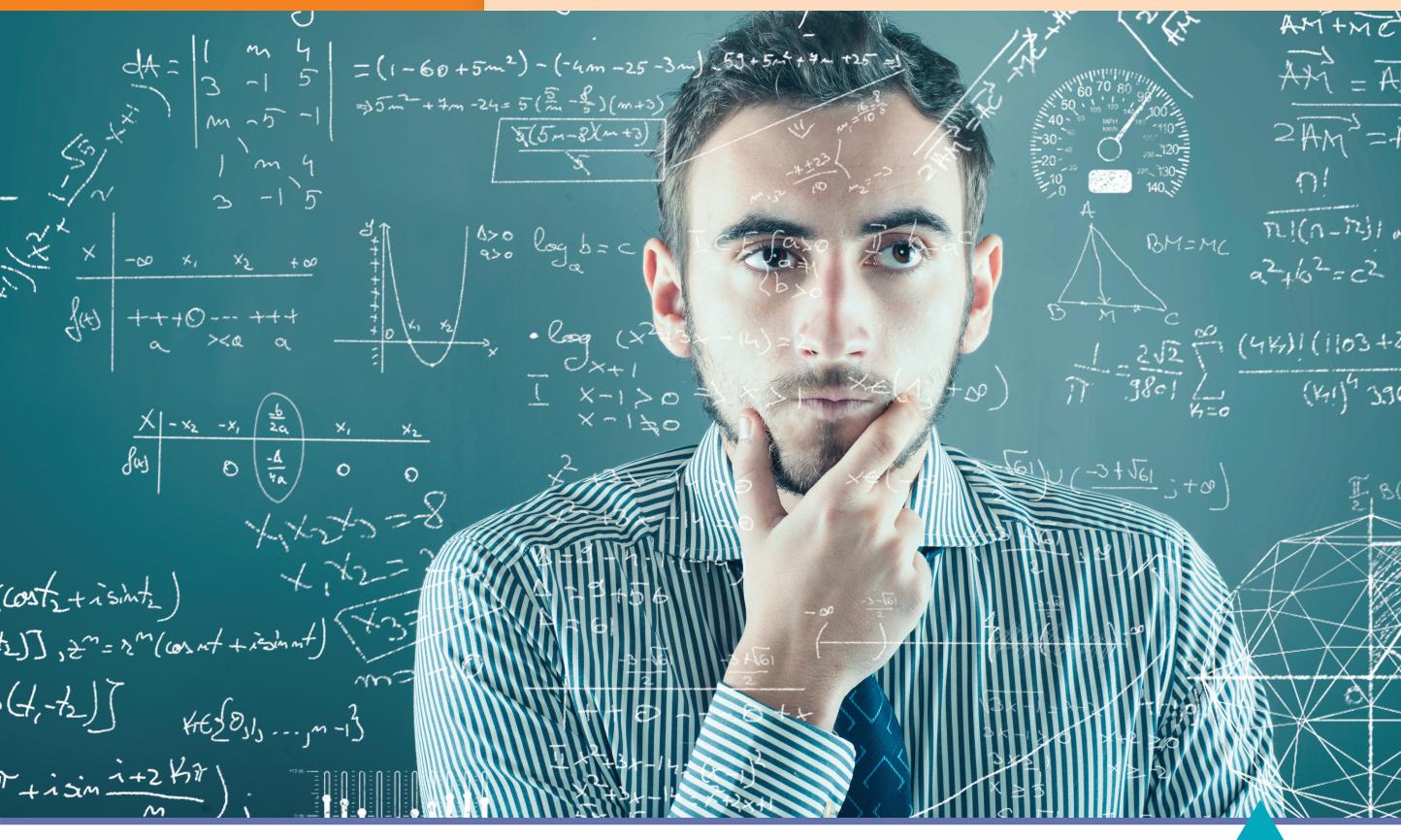
Réponses page 65

Objectifs

- Utiliser correctement les connecteurs logiques “et”, “ou” distinguer leur sens logique et leur sens dans le langage usuel.
- Distinguer condition nécessaire et condition suffisante.
- Connaître et formuler correctement les propositions, leurs réciproques, leurs contraposées, leurs négations.
- Utiliser à bon escient les différents quantificateurs.
- Connaître les principaux types de raisonnements et leurs utilisations.

CHAPITRE

1



De tout temps les penseurs ont cherché à s'assurer que leurs raisonnements, quel qu'en soit le sujet, ne comportaient pas de « faille » qui les fassent aboutir à un résultat inexact. Le besoin s'est rapidement fait sentir d'outils fiables sur lesquels la pensée puisse s'appuyer pour manipuler des concepts, et enchaîner des déductions.

Ce chapitre commence par définir ce qui différencie une proposition « vraie » d'une proposition « fausse ». Il présente ensuite les notions de **prédictats** et de **quantificateurs** avant de passer en revue les différents types de démonstrations.



Pour plus d'infos
sur les syllogismes
et les paradoxes,
consultez les + en
ligne.

Démontrer, c'est partir d'une ou plusieurs hypothèse(s) admise(s) pour aboutir, à l'aide d'un raisonnement, à une "vérité".

On peut par exemple enchaîner des déductions de la forme :

Si Proposition 1 alors Proposition 2, d'où Proposition 3, donc Proposition 4, par conséquent Proposition 5 ...

Mais, certaines propositions sont parfois vraies, toujours vraies ou toujours fausses, et la langue française peut parfois prêter à confusion :

- dans un restaurant on peut avoir au menu "fromage ou dessert", c'est-à-dire l'un ou l'autre mais pas les deux,
- dans un jeu de cartes on avoir "un roi ou un carreau" mais cela n'exclut pas que l'on ait le roi de carreau,
- selon l'heure, la date, le lieu la phrase "il pleut" peut être vraie ou fausse,
- l'affirmation : "si x est un nombre réel, $x^2 + 2x + 1 = (x + 1)^2$ " est toujours vraie,
- l'affirmation : "si x est un nombre réel, $x^2 + 1 = 0$ " est toujours fausse,
- l'affirmation : "si x est un nombre réel, $x - 2x + 1 = 0$ " n'est vraie que si x prend la valeur 1 ...

Dans ce chapitre nous allons dans un premier temps aborder le calcul propositionnel, celui des prédictats et les quantificateurs qui vont nous permettre par la suite d'étudier plusieurs formes classiques de démonstrations et de raisonnements.

FOCUS

La logique du latin "logica" dérivé du grec "logos" (raison), est à l'origine considérée comme l'étude des procédés de raisonnement, la capacité à "raisonner juste". Au coeur des travaux des grands philosophes de l'antiquité essentiellement Aristote dans son *Organon*, cette logique reste longtemps la référence pour les logiciens mettant en avant les bases du syllogisme.

Profitant de la renaissance Bacon met en doute cette logique, pour introduire une logique inductive basée sur l'observation de nombreux cas. Au XVII^e Leibniz cherche à établir une langue bien définie capable de fournir un "langage universel". Il introduit une grande partie de la notation mathématique moderne et généralise l'usage des quantificateurs.

Mais ce n'est qu'au XIX^e siècle que se développe un réel "formalisme logique". Boole introduit le calcul de vérité, les combinaisons logiques (conjonction, disjonction, négation) et les tables de vérité. Frege, Russel... poursuivent ces travaux et développent alors la logique moderne.

Celle-ci devient un vrai sujet de travail pour les mathématiciens qui doivent étudier des problèmes de plus en plus complexes, se trouvent confrontés à la "crise des fondements" (on peut ou non accepter le 5^{ème} postulat d'Euclide et ainsi construire des géométries totalement différentes...) et à l'apparition des paradoxes (paradoxe de Russel ou du barbier...)

1 Calcul propositionnel

Définition

Une **proposition** est une affirmation qui prend toujours la même valeur de vérité (vrai ou faux).

Remarque

C'est ce qu'on appelle parfois le **principe du tiers exclu** : une affirmation est soit vraie soit fausse, mais ne peut pas être en partie vraie et en partie fausse.

EXEMPLES

- l'affirmation “il pleut” n'est pas une proposition,
- l'affirmation “le carré d'un nombre réel est toujours positif ou nul” est une proposition,
- l'affirmation “5 est un multiple entier de 2” est une proposition.

Définition

Un **connecteur** en logique mathématique est un outil qui permet de créer une proposition à partir d'une ou plusieurs propositions initiales.

Définition

La **négation** d'une proposition P , notée $\neg P$ est la proposition obtenue en affirmant son contraire. On dit “non P ”.

EXEMPLES

- La négation de $x \geq 0$ est $x < 0$.
- La négation de “ k est un entier pair” est “ k est un entier impair”.
- La négation de “la fleur est blanche” est “la fleur n'est pas blanche”.

Définition

Soient P et Q deux propositions. La proposition notée $P \wedge Q$ est la proposition qui est vraie uniquement quand les propositions P et Q sont vraies simultanément. On dit que c'est la proposition “ P et Q ”, ou la **conjonction** de P et Q .

EXEMPLES

- “10 est un multiple de 2 et de 5”
- Dire qu'un réel est compris entre -1 et 1 , c'est exactement dire que $x \leq 1$ et $x \geq -1$.

Définition

Soient P et Q deux propositions. La proposition notée $P \vee Q$ est la proposition qui est vraie dès qu'au moins une des deux propositions P ou Q est vraie. On dit que c'est la proposition “ P ou Q ”, ou la **disjonction** de P et Q .

EXEMPLES

- La carte tirée est une figure ou un cœur.
- Un nombre entier est pair ou impair.

Théorème

Les trois premiers connecteurs vérifient les tables de vérité suivantes :

P	Q	$P \wedge Q$
V	V	V
V	F	F
F	V	F
F	F	F

P	Q	$P \vee Q$
V	V	V
V	F	V
F	V	V
F	F	F

P	$\neg P$
V	F
F	V

On en déduit que :

$$\neg(P \wedge Q) = (\neg P) \vee (\neg Q) \quad \text{et} \quad \neg(P \vee Q) = (\neg P) \wedge (\neg Q)$$

Remarque

Les expressions mêlant des propositions et des connecteurs doivent être correctement parenthésées sans quoi le lecteur pourrait interpréter la proposition résultat de plusieurs façons différentes.

Supposons par exemple que :

- P soit la proposition “ x est un nombre divisible par 6”,
- Q soit la proposition “ x est divisible par 2”,
- R soit la proposition “ x est divisible par 3”.

et qu'on écrive $\neg P \wedge \neg Q \vee R$, cela pourrait être :

- $(\neg P) \wedge (\neg(Q \vee R))$ qui peut se dire : x n'est pas divisible par 6 et x n'est ni divisible ni par 2 ni par 3. Dans ce cas $x = 7$ serait une valeur possible.
- $((\neg P) \wedge (\neg Q)) \vee R$ qui peut se dire : x n'est divisible ni par 6 ni par 2 ou est divisible par 3. Dans ce second cas $x = 3$ serait un bon candidat, mais pas $x = 7$.

En règle générale, on admet une seule simplification d'écriture : le connecteur \neg ne porte que sur la première proposition qu'il précède sauf s'il est suivi d'une parenthèse.

Ainsi $\neg P \wedge Q$ signifie $(\neg P) \wedge Q$.

Si on veut écrire la négation de $P \wedge Q$ il faut écrire $\neg(P \wedge Q)$.

Définition

Soient P et Q deux propositions. La proposition qui est vraie si Q est vraie dès que P est vraie est notée $P \rightarrow Q$ c'est l'**implication** de Q par P , on dit que " P implique Q ".

EXEMPLES

- 4 est un nombre pair implique 4 est divisible par 2 est une proposition vraie.
- 6 est multiple de 3 implique 6 est impair est une proposition fausse.
- $1 = 2$ implique Jules César a été général romain est une proposition vraie. En effet, il est vrai que Jules César a été un général romain, et il est évidemment faux que $1 = 2$, en se référant à la dernière ligne du tableau de vérité l'implication est donc vraie.

Théorème

La table de vérité de \rightarrow est :

P	Q	$P \rightarrow Q$
V	V	V
V	F	F
F	V	V
F	F	V

On en déduit que $(P \rightarrow Q)$ a la même table de vérité que $\neg P \vee Q$.

Remarque

Ainsi, dans les raisonnements toutes les propriétés, relations entre propositions peuvent être écrites à partir des trois connecteurs principaux : \wedge, \vee, \neg (voir même uniquement deux d'entre eux : \wedge et \neg ou \vee et \neg). C'est ce qu'a mis en évidence George Boole au *XIX^e siècle* au Royaume Uni.

Définition

Soient P et Q deux propositions.

- Dans l'implication $P \rightarrow Q$ on dit que :
 - Q est une **condition nécessaire** pour que P soit vraie.
 - P est une **condition suffisante** pour que Q soit vraie.
- La **réciproque** de $P \rightarrow Q$ est $Q \rightarrow P$. Si les deux implications sont vraies on dit que P est **équivalente à Q** . On note $P \leftrightarrow Q$.
- La **contraposée** de $P \rightarrow Q$ est $\neg Q \rightarrow \neg P$. Ces deux propositions ont la même table de vérité. On dit qu'elles sont **synonymes** et on note parfois $P \Leftrightarrow Q$.
- La **négation** de $P \rightarrow Q$ est $P \wedge (\neg Q)$.

Définition

Etant donné $P = \{P_1, P_2, \dots, P_n\}$ un ensemble de propositions on appelle **forme propositionnelle** toute proposition construite en combinant une ou plusieurs propositions P_i suivant les principes suivants :

- P_i est une forme propositionnelle,
- $P_i \wedge P_j$ est une forme propositionnelle,
- $P_i \vee P_j$ est une forme propositionnelle,
- $\neg P_i$ est une forme propositionnelle.

Pour toute forme propositionnelle on peut établir une **table de vérité** qui regroupe les différentes valeurs que peut prendre la forme propositionnelle selon la valeur de vérité de chacun des P_i .

Définition

Un choix de valeurs de vérité pour les P_i qui donne la valeur de vérité vraie, pour une forme propositionnelle donnée est un **modèle**.

Deux formes propositionnelles qui ont un modèle commun sont dites **compatibles**, dans le cas contraire elles sont dites **incompatibles** ou **contradictoires**.

Une forme propositionnelle qui prend toujours la valeur vraie est une **tautologie**.

Une forme propositionnelle qui prend toujours la valeur faux est une **antilogie** ou **contradiction**.

Si $P \rightarrow Q$ est une tautologie on dit que Q est une **conséquence logique** de P .

2 Prédicats

Définition

Un **prédicat** noté P associe à chaque élément x d'un ensemble U appelé **univers**, une proposition $P(x)$.

Si U est un ensemble de n -uplets (x_1, x_2, \dots, x_n) le prédicat dépend de n variables x_i on dit qu'il est de **poids n** .

Remarque

Un prédicat de poids $n = 0$, est une proposition.

Les connecteurs logiques s'appliquent aux prédicats tout comme aux propositions.

EXEMPLES

- $P(x)$ défini par “ x est entier naturel pair”, est un prédicat de poids 1 dont l'univers est \mathbb{N} .
- $Q(T)$ défini par “ T est un triangle rectangle” est un prédicat de poids 1 dont l'univers est l'ensemble de tous les triangles du plan.
- $R(m,n)$ défini par “l'entier naturel m est un multiple de l'entier naturel n ” est un prédicat de poids 2, dont l'univers est l'ensemble des couples d'entiers, c'est-à-dire $\mathbb{N} \times \mathbb{N}$.

Remarque

Pour énoncer un prédicat on utilise une ou plusieurs variables : x, y, \dots et le nom donné à ces variables n'a pas de d'importance : on peut parler du triangle isocèle nommé x ou T ou (ABC) sans que cela change le prédicat. On dit que ces variables sont **muettes**.

Cependant si on doit associer des prédicats avec des connecteurs logiques, il faut prendre garde à ce que les variables soient utilisées de façon cohérente dans chaque prédicat.

3 Quantificateurs

Définition

L'affirmation :

“pour tout x appartenant à U la proposition $P(x)$ est vraie”,
décrit une proposition qu'on note : $\forall x \in U, P(x)$ ou $\forall x, P(x)$.

Le symbole \forall est appelé **quantificateur universel**

Définition

L'affirmation :

“il existe (au moins un) x appartenant à U pour lequel la proposition $P(x)$ est vraie”,
décrit une proposition qu'on note : $\exists x \in U, P(x)$ ou $\exists x, P(x)$.

Le symbole \exists est appelé **quantificateur existentiel**.

On peut préciser l'affirmation en disant :

“il existe un unique x appartenant à U pour lequel la proposition $P(x)$ est vraie”,
dans ce cas on note : $\exists! x \in U, P(x)$ ou $\exists! x, P(x)$.

EXEMPLES

- pour tout x réel, le carré de x est un nombre positif ou nul :

$$\forall x \in \mathbb{R}, x^2 \geq 0$$

- il existe un réel tel que $x^2 = 1$ ($x = 1$ ou $x = -1$):

$$\exists x \in \mathbb{R}, x^2 = 1$$

- il existe un unique réel x tel que $x + 1 = 2x - 2$:

$$\exists! x \in \mathbb{R}, x + 1 = 2x - 2$$

Remarque

On peut bien sûr utiliser les quantificateurs pour des prédictats de poids $n > 1$, mais il faut être très vigilant à l'ordre des quantificateurs par rapport à celui des variables.

EXEMPLE

Si on veut écrire que pour tout entier naturel n pair il existe un entier naturel k tel que $n = 2k$, on peut considérer le prédictat : $R(n,k)$ défini par “l’entier naturel n est égal à 2 fois l’entier naturel k ”, on a alors un prédictat de poids 2 dont l’univers est $A \times \mathbb{N}$ où A est l’ensemble des entiers naturels pairs. On écrit alors :

$$\forall x \in A, \exists k \in \mathbb{N} \ R(n,k)$$

Théorème

Soient U un univers et $P(x)$ un prédictat dont les valeurs de vérité dépendent de $x \in U$.

- la négation de $\forall x \in U, P(x)$ est $\exists x \in U, \neg P(x)$
- la négation de $\exists x \in U, P(x)$ est $\forall x \in U, \neg P(x)$

EXEMPLES

- dire que la fonction $f(x)$ n'est pas nulle, c'est-à-dire que l'on n'a pas “ pour tout $x \in \mathbb{R}$, $f(x) = 0$ ”, c'est dire qu'il existe (au moins un) $x \in \mathbb{R}$ tel que $f(x) \neq 0$.

$$\neg(\forall x \in \mathbb{R}, f(x) = 0) \text{ peut s'écrire } \exists x \in \mathbb{R}, f(x) \neq 0$$

- dire qu'il n'existe pas de fraction dont le dénominateur est nul, c'est dire que toute fraction possède un dénominateur non nul.

$$\neg\left(\exists \frac{p}{q} \in \mathbb{Q}, q = 0\right) \text{ peut s'écrire } \forall \frac{p}{q} \in \mathbb{Q}, q \neq 0$$

Théorème

Soient U un univers, $P(x)$ et $Q(x)$ deux prédictats dont les valeurs de vérité dépendent de $x \in U$.

- $(\forall x \in U, P(x) \wedge Q(x)) \Leftrightarrow ((\forall x \in U, P(x)) \wedge (\forall x \in U, Q(x)))$
- $(\exists x \in U, P(x) \vee Q(x)) \Leftrightarrow ((\exists x \in U, P(x)) \vee (\exists x \in U, Q(x)))$

EXEMPLES

- dire que tout entier divisible par 6 est divisible par 2 et par 3 est bien équivalent à dire que tout entier divisible par 6 est divisible par 2 et que tout entier divisible par 6 est divisible par 3.
- dire qu'il existe un entier naturel divisible par 2 ou divisible par 3, est équivalent à dire qu'il existe un entier naturel divisible par 2, et qu'il existe un entier naturel divisible par 3. Il faut remarquer que dans le deuxième prédictat le nombre x divisible par 2 et celui divisible par 3 ne sont pas a priori les mêmes.

Théorème

Soient U un univers, $P(x)$ et $Q(x)$ deux prédictats dont les valeurs de vérité dépendent de $x \in U$.

- $(\exists x \in U, P(x) \wedge Q(x)) \rightarrow ((\exists x \in U, P(x)) \wedge (\exists x \in U, Q(x)))$

La réciproque est fausse.

- $((\forall x \in U, P(x)) \vee (\forall x \in U, Q(x))) \rightarrow (\forall x \in U, P(x) \vee Q(x)).$

La réciproque est fausse.

EXEMPLE

si on compare les deux prédictats : “il existe un réel x tel que $\ln(x) = 0$ ” et “il existe un réel x tel que $\sin(x) = 0$ ”, ils sont bien vrais tous les deux, mais le prédictat “il existe un réel x tel que $\ln(x) = 0$ et $\sin(x) = 0$ ” est faux. Ainsi les deux prédictats ne sont pas équivalents.

4 Les différents types de démonstration

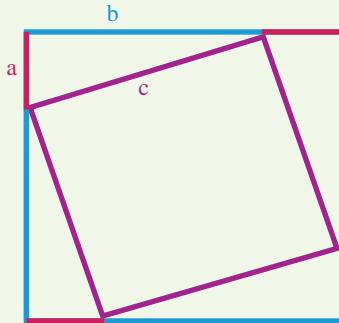
4.1 La démonstration directe

Définition

La **démonstration directe** consiste à démontrer la proposition énoncée en partant directement des hypothèses données, et en arrivant à la conclusion par un enchaînement d'impllications.

EXEMPLES

- Soit n un entier naturel le produit de n par $(n + 1)$ est un entier pair.
En effet, tout entier naturel n est pair ou impair, son successeur est alors impair ou pair.
Ainsi si n est pair, le produit $n(n + 1)$ est alors évidemment pair.
Si n est impair alors $n + 1$ est pair et $n(n + 1)$ est pair.
- Montrer le théorème de Pythagore en utilisant une démonstration géométrique basée sur les surfaces de deux carrés imbriqués.



On considère un triangle rectangle de petit côté a , de grand côté b et d'hypoténuse c .
On construit le carré de côté $a + b$ et le carré imbriqué de côté c comme ci-dessus.

L'aire du grand carré est évidemment $S = (a + b)^2$, mais on peut aussi l'écrire comme la somme de l'aire du petit carré c^2 , et de 4 fois l'aire du triangle rectangle initial $4 \frac{ab}{2}$.
On en déduit :

$$(a + b)^2 = c^2 + 4 \frac{ab}{2} \Leftrightarrow a^2 + 2ab + b^2 = c^2 + 2ab \Leftrightarrow a^2 + b^2 = c^2$$

4.2 La démonstration par contraposée

Définition

La **démonstration par contraposée** consiste à démontrer que : $(\neg Q) \rightarrow (\neg P)$ au lieu de montrer que: $P \rightarrow Q$

Remarque

On peut facilement vérifier que ces deux propositions sont synonymes en écrivant leurs tables de vérité respectives.

EXEMPLES

- Soit x un nombre réel tel que pour tout $\varepsilon > 0$, $x \leq \varepsilon$ alors $x \leq 0$.

Montrons la contraposée de cette implication :

$$(x > 0) \rightarrow (\exists \varepsilon > 0, x > \varepsilon)$$

Cette implication est vraie puisque pour chaque $x > 0$ il suffit de prendre par exemple $\varepsilon = \frac{x}{2}$.

La contraposée est vraie, l'implication initiale est vraie.

- Si n^2 est un entier naturel impair, alors n est un entier naturel impair.

En effet si n est un entier naturel pair, alors il existe k entier naturel tel que $n = 2k$ donc $n^2 = 4k^2 = 2 \cdot 2k^2$ est pair.

4.3 La démonstration par l'absurde

Définition

La **démonstration par l'absurde** consiste à supposer le contraire de la proposition énoncée et à montrer qu'on arrive alors à une contradiction (absurdité, impossibilité).

EXEMPLES

- Montrer que $\sqrt{2}$ est irrationnel.

Supposons par l'absurde que $\sqrt{2}$ est un rationnel, c'est-à-dire qu'il existe une fraction irréductible $\frac{p}{q}$ où p est un entier relatif p et q est un entier naturel non nul tels que $\sqrt{2} = \frac{p}{q}$, alors $2 = \frac{p^2}{q^2} \Leftrightarrow p^2 = 2q^2$.

On en déduit que p^2 est pair donc p lui-même est pair (cf. l'exemple précédent), il existe donc un entier k tel que $p = 2k$, on en déduit

$$p^2 = 2q^2 \Leftrightarrow (2k)^2 = 2q^2 \Leftrightarrow 4k^2 = 2q^2 \Leftrightarrow 2k^2 = q^2$$

ainsi q^2 est pair et donc q est pair.

p et q sont tous les deux pairs, donc la fraction $\frac{p}{q}$ n'est pas irréductible et l'hypothèse de départ est absurde, donc $\sqrt{2}$ est irrationnel.

- Montrer qu'il existe un nombre infini de nombres premiers.

Supposons qu'il existe un nombre fini N de nombres premiers, qu'on peut ranger dans un ordre croissant et noter p_1, p_2, \dots, p_N .

On construit un nouveau nombre, le produit des N nombres premiers auquel on ajoute 1 : $Q = p_1 p_2 \dots p_N + 1$.

N n'est divisible par aucun des p_1, p_2, \dots, p_N (le reste de la division est toujours 1). On a donc deux possibilités :

- soit Q est lui-même premier, on a donc construit un nouveau nombre premier ce qui contredit l'hypothèse de départ,

- soit Q n'est pas premier et il est donc divisible par un autre nombre premier non encore identifié. Ce qui contredit encore l'hypothèse de départ.

Dans les deux cas l'hypothèse de départ est absurde et il existe une infinité de nombres premiers.

4.4 La démonstration par récurrence ou induction

Définition

La **démonstration par récurrence** ne peut se faire que si la propriété $P(n)$ qu'on veut démontrer dépend d'un entier n . On procède en trois étapes :

- on montre que $P(0)$ ou $P(n_0)$ est vraie,
- on suppose que $P(n)$ est vraie et on montre que $P(n + 1)$ est vraie (héritage),
- on en déduit que $P(n)$ est vraie pour tout $n \geq 0$ ou $n \geq n_0$.

EXEMPLES

- Montrer que pour tout réel positif x et tout n entier naturel : $(1 + x)^n \geq 1 + nx$

Montrons que la propriété est vraie pour $n = 0$, en effet on a bien

$$(1 + x)^0 = 1 = 1 + 0x$$

Montrons que si la propriété est vraie au rang n alors elle est vraie au rang $n + 1$, c'est-à-dire que :

$$\text{si } (1 + x)^n \geq 1 + nx \text{ alors } (1 + x)^{n+1} \geq 1 + (n + 1)x$$

On a $(1 + x)^{n+1} = (1 + x)^n(1 + x)$, mais par hypothèse de récurrence au rang n $(1 + x)^n \geq 1 + nx$ et en multipliant les deux membres de l'inégalité par $(1 + x)$ on a :

$$(1 + x)^{n+1} \geq (1 + nx)(1 + x) \Leftrightarrow (1 + x)^{n+1} \geq 1 + (n + 1)x + nx^2$$

On a donc bien $(1 + x)^{n+1} \geq 1 + (n + 1)x$.

Ainsi la propriété est vraie pour $n = 0$, si elle est vraie pour n alors elle est vraie pour $n + 1$, on en déduit que la propriété est vraie pour tous les entiers naturels.

- Montrer que pour n entier naturel : $1 + 2 + \dots + n = \frac{n(n + 1)}{2}$

Montrons que la propriété est vraie pour $n = 1$, en effet on a bien

$$1 = \frac{1(1 + 1)}{2}$$

Montrons que si la propriété est vraie au rang n alors elle est vraie au rang $n + 1$, c'est-à-dire que :

$$\text{si } 1 + 2 + \dots + n = \frac{n(n + 1)}{2} \text{ alors } 1 + 2 + \dots + n + (n + 1) = \frac{(n + 1)(n + 2)}{2}$$

On a $1 + 2 + \dots + n + (n + 1) = (1 + 2 + \dots + n) + (n + 1)$, donc par hypothèse de récurrence au rang n :

$$1 + 2 + \cdots + n + (n+1) = \frac{n(n+1)}{2} + (n+1) = \frac{n(n+1) + 2(n+1)}{2} = \frac{(n+1)(n+2)}{2}$$

On a donc bien $1 + 2 + \cdots + n + (n+1) = \frac{(n+1)(n+2)}{2}$.

Ainsi la propriété est vraie pour $n = 1$, si elle est vraie pour n alors elle est vraie pour $n + 1$, on en déduit que la propriété est vraie pour tous les entiers naturels non nuls.

4.5 La démonstration sans mot

Définition

Une **preuve sans mot, ou démonstration visuelle** est une preuve que l'on fait par un diagramme qui la rend évidente.

Quand le dit diagramme n'illustre qu'un cas particulier, il faut que la généralisation ne demande au lecteur qu'un effort minimal.

EXEMPLES

- Dans le diagramme cartésien d'une relation binaire, si la "diagonale est pleine" alors la relation est reflexive c'est-à-dire que $\forall x, xRx$.

Considérons sur l'ensemble $A = \{1, 2, 3, 4\}$ la relation

$$xRy \Leftrightarrow x + y \text{ est pair}$$

On peut représenter les différents couples en relation en construisant un tableau dit diagramme cartésien où chaque croix indique que x (noté dans la première colonne du diagramme) est en relation avec y (noté dans la première ligne du diagramme)

	1	2	3	4
1	×		×	
2		×		×
3	×		×	
4		×		×

Ce qu'il faut retenir

- Le calcul propositionnel permet de rédiger de façon rigoureuse des expressions dont on connaît la valeur de vérité.
- Si les propositions dépendent d'une ou plusieurs variables on peut appliquer les mêmes principes de calcul avec des prédictats.
- Les quantificateurs ne sont pas de simples abréviations mais déterminent de façon précise les caractéristiques des variables dont dépend le prédicat.
- L'ordre d'apparition des quantificateurs dans l'écriture de la formule est fondamental.
- Plusieurs types de raisonnements ou de démonstrations sont possibles, il faut toujours dans un premier temps isoler les hypothèses qu'on possède, les conclusions auxquelles on souhaite aboutir et se demander alors quelle sera la méthode la plus adaptée et surtout la plus simple à mettre en oeuvre.

- 1. La négation de : “Le menu prévoit fromage et dessert” est :**
 - a. “Le menu ne prévoit ni fromage ni dessert”.
 - b. “Le menu prévoit fromage ou dessert ou les deux”.
 - c. “Le menu ne prévoit que du fromage ou que du dessert”
- 2. Si P est la proposition : “Guillaume vient”, Q est la proposition : “Lucie vient”, R est la proposition : “Astrid vient”, une écriture de la phrase : “Guillaume ne vient pas si Lucie ou Astrid viennent” sous forme de proposition est :**
 - a. $(\neg Q) \vee (\neg R) \vee (\neg P)$
 - b. $\neg((Q \vee R) \wedge P)$
 - c. $\neg(Q \vee (R \wedge P))$
- 3. Ecrire la table de vérité de $\neg P \vee Q$. En déduire qu'une implication et sa contraposée sont synonymes.**
- 4. Ecrire avec les quantificateurs appropriés :**
 - a. la fonction f est nulle pour tout x réel
 - b. il existe un antécédent x pour lequel la fonction f s'annule
 - c. la courbe représentative de f coupe l'axe des abscisses en un seul point d'abscisse x
- 5. Donner la négation, la contraposée, la réciproque de la proposition “Tout entier divisible par 6 est divisible par 2 et 3”.**
- 6. Un problème classique d'après Cervantes Un missionnaire est capturé par des cannibales. Par respect pour sa fonction, les cannibales lui proposent de choisir son sort : il doit faire une courte déclaration, si elle est vraie il sera rôti, si elle est fausse il sera bouilli. Quelle phrase le missionnaire doit-il dire pour sauver sa vie ?**
 - a. je serai rôti
 - b. je serai bouilli
 - c. je demande pitié

Réponses page 65

Exercices

1 Sachant que P , Q et R sont des propositions, écrire les tables de vérité des propositions suivantes, indiquer si ce sont des tautologies ou des antilogies.

- 1) $\neg P \vee (Q \wedge R)$
- 2) $((P \vee R) \wedge Q) \rightarrow (R \vee \neg Q)$
- 3) $(P \rightarrow Q) \wedge (P \wedge \neg Q)$

2 Sachant que P , Q et R sont des propositions, déterminer si les propositions écrites sur une même ligne sont synonymes ou non.

- 1) $P \rightarrow (Q \rightarrow R)$ $(P \rightarrow Q) \rightarrow R$
- 2) $P \rightarrow (Q \wedge R)$ $(P \rightarrow Q) \wedge (P \rightarrow R)$
- 3) $P \rightarrow (Q \vee R)$ $(P \rightarrow Q) \vee (P \rightarrow R)$

3 Pour chacune des 3 assertions suivantes, donner sa valeur de vérité, puis sa négation

- 1) $\exists x \in \mathbb{R}, \forall y \in \mathbb{R}, x - y \leq 0$
- 2) $\forall x \in \mathbb{R}, \exists y \in \mathbb{R}, x - y \leq 0$
- 3) $\exists x \in \mathbb{R}, \exists y \in \mathbb{R}, x - y \leq 0$

4 Donner la réciproque, la contraposée, la négation de chacune des implications suivantes (P , Q , R et S sont des propositions).

- 1) $((\neg Q) \wedge P) \rightarrow (\neg S)$
- 2) $S \rightarrow ((\neg P) \vee Q)$
- 3) $(R \wedge (\neg(S \vee Q))) \rightarrow P$

Réponses page 65