



**École de technologie supérieure**  
Service des enseignements généraux  
Local B-2500 514-396-8938  
Site internet: <http://www.etsmtl.ca/>

# **MAT210**

## **LOGIQUE ET MATHÉMATIQUES DISCRÈTES**

NOTES DE COURS

RÉDIGÉES PAR  
**GENEVIÈVE SAVARD,**  
**ANOUK BERGERON-BRLEK**  
**ET XAVIER PROVENÇAL**

VERSION RÉVISÉE EN DÉCEMBRE 2023

Ce document est mis à disposition selon les termes de la licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 4.0 International.





# Table des matières

<b>1</b>	<b>Logique et ensembles</b>	<b>1</b>
1.1	Logique du premier ordre . . . . .	1
1.1.1	Logique propositionnelle . . . . .	1
1.1.2	Équivalences propositionnelles . . . . .	14
1.1.3	Prédicats et quantificateurs . . . . .	19
1.1.4	Quantificateurs imbriqués . . . . .	28
1.2	Raisonnements . . . . .	34
1.2.1	Règles d'inférence . . . . .	34
1.2.2	Cohérence d'un ensemble de spécifications . . . . .	38
1.2.3	Types de preuve . . . . .	46
1.3	Théorie des ensembles . . . . .	48
1.3.1	Notions de base sur les ensembles . . . . .	48
1.3.2	Ensembles de nombres $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ . . . . .	49
1.3.3	Produit cartésien . . . . .	53
1.3.4	Opérations sur les ensembles $\cap, \cup, \oplus, -$ . . . . .	55
1.3.5	Représentation de sous-ensembles par trains de bits . . . . .	57
1.4	Fonctions . . . . .	62
1.4.1	Fonctions plancher et plafond . . . . .	64
1.4.2	Injection, surjection et bijection . . . . .	65
<b>2</b>	<b>Théorie des nombres</b>	<b>69</b>
2.1	Arithmétique modulaire . . . . .	69
2.1.1	Division entière . . . . .	69
2.1.2	Congruences modulo $m$ . . . . .	73
2.2	Représentation des entiers en base $b$ . . . . .	77
2.2.1	Système de numération décimal . . . . .	77
2.2.2	Système de numération dans une base quelconque . . . . .	77
2.2.3	Changement de base sur Nspire . . . . .	83
2.3	Entiers et algorithmes . . . . .	85
2.3.1	Algorithme d'exponentiation modulaire efficace . . . . .	85
2.3.2	Nombres premiers et PGCD . . . . .	87
2.3.3	Algorithme d'Euclide et théorème de Bézout . . . . .	88
2.3.4	Inverse modulo $m$ . . . . .	91
2.3.5	Résolution de congruence . . . . .	94
2.3.6	Petit théorème de Fermat . . . . .	98
2.4	Cryptographie à clé secrète . . . . .	99
2.4.1	Chiffrement par décalage . . . . .	99
2.4.2	Permutation de l'alphabet . . . . .	100

2.4.3	Masque jetable . . . . .	100
2.4.4	Chiffre affine . . . . .	100
2.5	Cryptographie à clé publique . . . . .	101
2.5.1	Chiffre RSA . . . . .	101
<b>3</b>	<b>Représentation des entiers et manipulations bit à bit</b>	<b>105</b>
3.1	Représentation des entiers à taille fixe . . . . .	105
3.1.1	Entiers non signés à taille fixe . . . . .	105
3.1.2	Entiers signés à taille fixe . . . . .	109
3.2	Opérations bit à bit . . . . .	116
3.3	Masques binaires . . . . .	119
<b>4</b>	<b>Introduction à la complexité des algorithmes</b>	<b>129</b>
4.1	Mesurer un temps de calcul à l'aide d'une fonction . . . . .	130
4.2	Notation grand-O et grand- $\Theta$ . . . . .	134
4.2.1	Grand-O d'une fonction composée . . . . .	142
4.3	Sommations . . . . .	147
4.4	Établir la fonction de complexité d'un algorithme . . . . .	149
4.5	Calculabilité et complexité . . . . .	155
<b>5</b>	<b>Algorithmes récursifs</b>	<b>157</b>
5.1	Définition et exemples d'algorithmes récursifs . . . . .	158
5.2	Fonctions récursives et relations de récurrence . . . . .	161
5.2.1	Résolution de relation de récurrence par la méthode itérative . . . . .	162
5.3	Algorithmes de type diviser pour régner . . . . .	164
5.3.1	Algorithmes et relations de récurrence de type diviser pour régner . . . . .	165
5.3.2	Résolution de relation de type diviser pour régner par la méthode itérative . . . . .	166
<b>6</b>	<b>Preuve par récurrence</b>	<b>171</b>
6.1	Preuve par récurrence simple . . . . .	171
6.2	Preuve par récurrence forte . . . . .	178
6.3	Preuve de validité d'un algorithme récursif . . . . .	182
<b>7</b>	<b>Dénombrement</b>	<b>187</b>
7.1	Notions de base . . . . .	187
7.1.1	Principe du produit . . . . .	187
7.1.2	Principe de la somme . . . . .	188
7.1.3	Principe d'inclusion-exclusion . . . . .	188
7.1.4	Principe des tiroirs . . . . .	189
7.2	Permutations et arrangements . . . . .	189
7.3	Combinaisons . . . . .	191
7.4	Relations de récurrence et dénombrement . . . . .	195
<b>8</b>	<b>Théorie des graphes</b>	<b>199</b>
8.1	Terminologie et types de graphes . . . . .	199
8.2	Représentation des graphes . . . . .	204
8.2.1	Représentation par listes d'adjacence . . . . .	204
8.2.2	Représentation par matrice d'adjacence . . . . .	205
8.3	Chemins dans un graphe . . . . .	207
8.3.1	Chemins, circuits, cycles . . . . .	207

8.3.2	Dénombrement de chemins . . . . .	208
8.3.3	Chemins et circuits eulériens . . . . .	210
8.3.4	Chemins et circuits hamiltoniens . . . . .	217
8.4	Problème du plus court chemin (algorithme de Dijkstra) . . . . .	218
8.4.1	Exemple: plan d'approvisionnement . . . . .	224
<b>Réponses</b>		<b>231</b>
Chapitre 1	. . . . .	231
Chapitre 2	. . . . .	249
Chapitre 3	. . . . .	252
Chapitre 4	. . . . .	256
Chapitre 5	. . . . .	259
Chapitre 6	. . . . .	262
Chapitre 7	. . . . .	273
Chapitre 8	. . . . .	278
<b>Bibliographie</b>		<b>284</b>
<b>Index</b>		<b>285</b>



# Avant-propos

La rédaction de ces notes de cours a débuté tranquillement en 2006. À l'origine, elles n'étaient destinées qu'à résumer certaines notions présentées en classe. Depuis, ces notes ont été peu à peu bonifiées par des exemples variés et des exercices accompagnés de solutions détaillées. Le contenu du cours a été ajusté, notamment pour être mieux arrimé au cours LOG320-*Structures de données et algorithmes*. Cependant, pour un discours complet sur les différents sujets abordés, ainsi que pour avoir accès à davantage d'exercices, nous vous invitons à consulter le livre de référence *Discrete Mathematics and Its Applications* de Rosen, cité par [1] dans ce document.

Le lecteur remarquera que plusieurs exemples ne sont pas solutionnés : on propose aux enseignants de les compléter en classe. Il en va de même des preuves des théorèmes.

Nous remercions notre collègue André Bordeleau pour les graphiques illustrant la croissance des fonctions dans le chapitre sur la notation grand-O.

Nous remercions aussi chaleureusement notre collègue Marie Forest qui a scruté attentivement l'ensemble du texte à l'été 2020, nous signalant plusieurs coquilles et nous donnant des suggestions fort pertinentes.

Finalement, nous vous remercions à l'avance de bien vouloir nous signaler les éventuelles erreurs détectées dans ces notes et de nous donner vos suggestions par courriel, à l'adresse suivante :  
genevieve.savard@etsmtl.ca.

Geneviève Savard,  
Anouk Bergeron-Brlek,  
et Xavier Provençal,  
professeurs enseignants à l'ÉTS

## **Remarques concernant la version d'août 2022**

Cette nouvelle version propose quelques nouveaux exemples et exercices dans les chapitres suivants, ce qui entraîne une modification de la numérotation :

- Chapitre 1, Exercices 1.17, 1.42, 1.49
- Chapitre 2, Exercices 2.17, 2.20
- Chapitre 3, Sections 3.1 et 3.2
- Chapitre 6
- Chapitre 7, Section 7.4

Pour le reste, il s'agit essentiellement de corrections mineures, de reformulations ou de précisions.

## **Remarques concernant la version de décembre 2022**

Cette nouvelle version ne comporte que des modifications mineures visant à corriger les coquilles détectées. La numérotation des exemples et exercices demeure la même.

## **Remarques concernant la version de août 2023**

Cette nouvelle version ne comporte que des modifications mineures visant à corriger les coquilles détectées. La numérotation des exemples et exercices demeure la même.



# Chapitre 1

## Logique et ensembles

### 1.1 Logique du premier ordre

#### 1.1.1 Logique propositionnelle

##### Définition 1.1 : Proposition

Un énoncé qui est soit vrai, soit faux est appelé une **proposition**. La **valeur de vérité** d'une proposition est donc vrai (**vrai**) ou faux (**faux**).

Un énoncé qui n'est pas une proposition (comme un paradoxe, une phrase impérative ou interrogative) sera qualifié d'inacceptable.

##### Exemple 1.1 (à compléter en classe)

Déterminez si la phrase suivante est une proposition ou non. Si oui, précisez s'il s'agit d'une proposition vraie ou d'une proposition fausse.

- (a) Le nombre 7 est pair.
- (b)  $2 + 3 = 6$
- (c) Où est Montréal?
- (d) Dépêchez-vous.
- (e) Il pleut actuellement quelque part sur la ville de Montréal.
- (f)  $x < 5$

- (g) Tous les nombres positifs sont strictement inférieurs à leur carré.
- (h) Cette phrase est fausse.
- (i) L'entier 19 est un nombre premier.
- (j) Il existe un seul nombre premier  $p$  tel que  $p + 1$  est également premier.
- (k) L'entier 427741 est un nombre premier.
- (l) Il existe une infinité de nombres premiers  $p$  tels que  $p + 2$  est également premier.

---

*Il n'est pas nécessaire de connaître la valeur de vérité d'une proposition pour savoir qu'il s'agit bien d'une proposition.* Dans l'exemple précédent, la phrase (k) est une proposition, car soit le nombre 427 741 est premier (et alors la proposition serait vraie), soit il n'est pas premier (et la proposition serait fausse). En fait, ce nombre n'est pas premier, car  $521 \times 821 = 427\,741$ . Il en va de même avec la phrase (l) qui est forcément soit vraie, soit fausse, même si dans l'état actuel de la science, on ne sait pas ce qui en est. Cette question est connue sous le nom de la **conjecture des nombres premiers jumeaux**.

On peut construire de nouvelles propositions à partir de propositions existantes en utilisant des connecteurs logiques. Ces nouvelles propositions forment ce que l'on appelle des **propositions composées**, et les propositions qui ne sont pas formées à partir de connecteur sont appelées **propositions simples**.

**Définition 1.2 : Connecteurs logiques**
 $\neg$     $\wedge$     $\vee$     $\oplus$     $\rightarrow$     $\leftrightarrow$ 

Soient  $p$  et  $q$  des propositions.

$\neg p$  L'énoncé « *il n'est pas vrai que  $p$*  » est une nouvelle proposition. On l'appelle la **négation** de  $p$  et on la note  $\neg p$  (lire « non  $p$  »). La proposition  $\neg p$  est vraie quand  $p$  est fausse et elle est fausse quand  $p$  est vraie.

$p \wedge q$  L'énoncé «  *$p$  et  $q$*  » est une nouvelle proposition. On l'appelle la **conjonction** de  $p$  et de  $q$  et on la note  $p \wedge q$ . La proposition  $p \wedge q$  est vraie uniquement quand  $p$  et  $q$  sont vraies.

$p \vee q$  L'énoncé «  *$p$  ou  $q$*  » est une nouvelle proposition. On l'appelle la **disjonction** de  $p$  et de  $q$  et on la note  $p \vee q$ . La proposition  $p \vee q$  est fausse uniquement quand  $p$  et  $q$  sont fausses. Elle est vraie quand une ou l'autre ou les deux propositions sont vraies.

$p \oplus q$  L'énoncé «  *$p$  ou exclusif  $q$*  » est une nouvelle proposition. On l'appelle la **disjonction exclusive** de  $p$  et de  $q$  et on la note  $p \oplus q$ . La proposition  $p \oplus q$  est vraie uniquement quand une seule des propositions  $p$  et  $q$  est vraie. Ce connecteur peut être défini à partir des précédents par  $(p \wedge \neg q) \vee (\neg p \wedge q)$ .

$p \rightarrow q$  L'énoncé «  *$p$  implique  $q$*  » est une nouvelle proposition. On dit que c'est une **implication** ou **un énoncé conditionnel**, et on le note  $p \rightarrow q$ . La proposition  $p \rightarrow q$  est fausse uniquement quand  $p$  est vraie et  $q$  est fausse. Ce connecteur peut être défini à partir des précédents par  $\neg p \vee q$ .

Voici quelques formulations différentes pour l'implication  $p \rightarrow q$ :

- « si  $p$  alors  $q$  »
- «  $q$  si  $p$  »
- «  $p$  est une condition suffisante pour  $q$  »
- «  $q$  est une condition nécessaire pour  $p$  »
- «  $p$  seulement si  $q$  »

$p \leftrightarrow q$  L'énoncé «  *$p$  si et seulement si  $q$*  » est une nouvelle proposition. On la note  $p \leftrightarrow q$  et on l'appelle **biconditionnelle**. La proposition  $p \leftrightarrow q$  est vraie uniquement quand  $p$  et  $q$  ont les mêmes valeurs de vérité. Ce connecteur peut être défini à partir des précédents par  $(p \rightarrow q) \wedge (q \rightarrow p)$ .

Résumons toute l'information fournie par la définition 1.2 grâce à une **table de vérité**.

$p$	$q$	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
<b>V</b>	<b>V</b>						
<b>V</b>	<b>F</b>						
<b>F</b>	<b>V</b>						
<b>F</b>	<b>F</b>						

**Exemple 1.2** (à compléter en classe)

Voici quatre propositions simples à partir desquelles on peut construire des propositions composées.

$c$ : « Julie étudie en génie de la construction. »

$l$ : « Julie étudie en génie logiciel. »

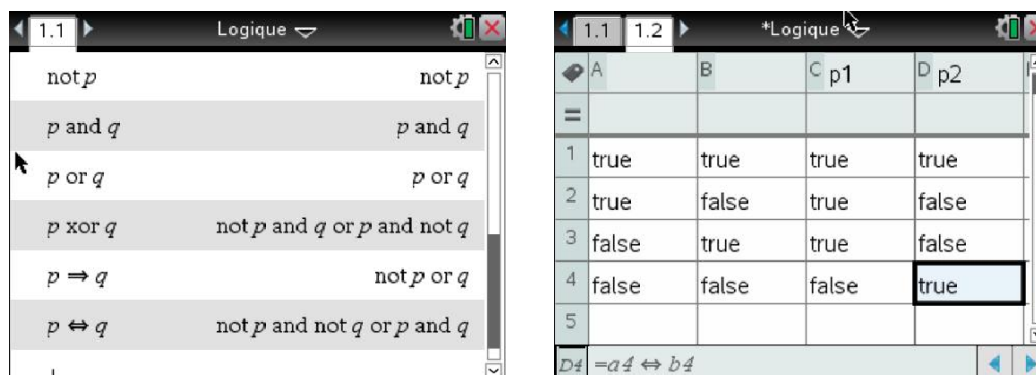
$t$ : « Julie étudie en génie des TI. »

$m$ : « Julie suit le cours MAT210 cet été. »

Traduisez les propositions suivantes en utilisant les propositions simples  $c$ ,  $l$ ,  $t$  et  $m$  ainsi que les connecteurs logiques. *On ne demande pas de dire si les propositions sont vraies ou fausses.*

- (a) Julie n'étudie ni en génie logiciel, ni en génie des TI.
- (b) Julie étudie en génie logiciel ou en génie de la construction, mais elle n'étudie pas dans les deux programmes.
- (c) Si Julie suit le cours de MAT210 cet été, alors elle étudie en génie logiciel ou en génie des TI.
- (d) Si Julie étudie en génie logiciel ou en génie des TI, alors elle suit le cours de MAT210 cet été.
- (e) Si Julie étudie en génie de la construction, alors elle ne suit pas le cours MAT210 cet été.
- (f) Si Julie ne suit pas le cours MAT210 cet été, alors elle étudie en génie de la construction.
- (g) Si Julie suit le cours de MAT210 cet été et n'étudie pas en génie des TI, alors elle étudie en génie logiciel.
- (h) Julie suit le cours de MAT210 cet été seulement si elle étudie en génie des TI ou en génie logiciel.
- (i) Savoir que Julie étudie en génie de la construction est une information *suffisante* pour conclure que Julie ne suit pas le cours de MAT210 cet été.
- (j) Il est nécessaire que Julie n'étudie pas en construction pour qu'elle suive le cours de MAT210 cet été.

Le symbole **V** (vrai) est parfois désigné par **T** (true) ou 1. Le symbole **F** (faux, false) est parfois désigné par 0. Les écrans suivants montrent la syntaxe utilisée par le logiciel et la calculatrice Nspire pour les connecteurs logiques.



### Priorité des connecteurs logiques

La priorité des opérations arithmétiques est une convention qui dicte dans quel ordre effectuer les opérations. Elle permet de réduire le nombre de parenthèses d'une expression sans diminuer sa clarté. Par exemple, on peut omettre les quatre parenthèses dans l'expression  $2 + (3 \times (2^2))$ .

$$2 + 3 \times 2^2 = 2 + (3 \times (2^2)) = 2 + (3 \times 4) = 2 + 12 = 14$$

Il en va de même avec la priorité des connecteurs logiques. Voici les listes de priorité des opérations arithmétiques (un petit rappel!) et des connecteurs logiques.

Priorité des opérations arithmétiques	
1	Parenthèses, de l'intérieur vers l'extérieur.
2	Exposants.
3	$\times$ $\div$
4	$+$ $-$

Priorité des connecteurs logiques	
1	Parenthèses, de l'intérieur vers l'extérieur.
2	$\neg$
3	$\wedge$
4	$\vee$ $\oplus$
5	$\rightarrow$
6	$\leftrightarrow$

Ainsi,  $p \vee q \rightarrow \neg r$  est équivalent à  $(p \vee q) \rightarrow (\neg r)$ . Par ailleurs, bien que certaines parenthèses soient inutiles avec la priorité des connecteurs, il arrive souvent qu'on les ajoute pour un maximum de clarté.

#### Exemple 1.3 (à compléter en classe)

Ajoutez toutes les parenthèses possibles dans les expressions suivantes pour les clarifier sans en changer la valeur de vérité.

- (a)  $\neg p \vee q \wedge r$
- (b)  $p \vee q \wedge r$
- (c)  $\text{not } p \text{ and not } q \text{ or } p \text{ and } q$
- (d)  $\text{not } p \text{ and } q \text{ or } p \text{ and not } q$
- (e)  $\text{not } a \text{ and } b \text{ and not } c \text{ or not } d$

**Exemple 1.4** (à compléter en classe)

Ôtez les parenthèses inutiles dans les expressions suivantes.

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

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

**Exemple 1.5** (à compléter en classe)

Construisez la table de vérité de la proposition composée suivante. De plus, trouvez une expression plus simple ayant la même table de vérité.

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

$p$	$q$	$p \vee q$	$p \wedge q$	$(p \vee q) \oplus (p \wedge q)$
V	V			
V	F			
F	V			
F	F			

(b)  $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow q)$

$p$	$q$	$\neg p$	$p \leftrightarrow q$	$\neg p \leftrightarrow q$	$(p \leftrightarrow q) \oplus (\neg p \leftrightarrow q)$
V	V				
V	F				
F	V				
F	F				

**Distinctions entre le langage courant et le langage mathématique.**

**Dans le langage courant**, le « ou » est souvent considéré exclusif, comme dans la phrase « vous avez droit à une soupe ou à une salade avec votre repas » qui signifie « vous avez droit à une soupe ou à une salade avec votre repas, *mais pas les deux.* »

Cependant, **en mathématique, on considère tous les « ou » comme des disjonctions inclusives** : «  $a$  ou  $b$  » sera vraie si  $a$  ou  $b$  ou ( $a$  et  $b$ ) sont vraies.

De même, le « si » du langage courant est souvent utilisé au lieu du « si et seulement si », comme dans l'exemple « Tu auras un cadeau si tu es sage » qui signifie en fait « tu auras un cadeau si *et seulement* si tu es sage ». Ou encore « je vous embaucherai si vous faites votre stage ici », qui peut sous-entendre « je vous embaucherai si et seulement si vous faites votre stage ici ». Dans ce recueil de notes, le « si » correspondra toujours à une implication et non à une biconditionnelle.

**Définition 1.3 : Réciproque, contraposée et inverse**

La **réciproque** de la proposition  $p \rightarrow q$  est la proposition  $q \rightarrow p$ .

La **contraposée** de la proposition  $p \rightarrow q$  est la proposition  $\neg q \rightarrow \neg p$ .

L'**inverse** de la proposition  $p \rightarrow q$  est la proposition  $\neg p \rightarrow \neg q$ .

**Exemple 1.6 (à compléter en classe)**

Soit les propositions

$v$ : « Je viens à l'ÉTS à vélo. »

$b$ : « Il fait beau. »

- (a) Écrivez les propositions suivantes en utilisant les lettres  $v$ ,  $b$  et les connecteurs logiques.

P1: « Je viens à l'ÉTS à vélo s'il fait beau. »

P2: « Si je ne viens pas à l'ÉTS à vélo, alors il ne fait pas beau. »

P3: « Je viens à l'ÉTS à vélo seulement s'il fait beau. »

P4: « S'il ne fait pas beau, je ne viens pas à l'ÉTS à vélo. »

P5: « Je viens à l'ÉTS à vélo si et seulement s'il fait beau. »

- (b) Indiquez laquelle des propositions est la réciproque de la proposition P1.

- (c) Indiquez laquelle des propositions est la contraposée de P1.

- (d) Indiquez laquelle des propositions est l'inverse de P1.

**Exemple 1.7** (à compléter en classe)

Considérons le détecteur d'erreurs très rudimentaire suivant: lors de la transmission d'un message, disons par paquets de 7 bits, on ajoute un bit de parité au train de bits: 0 si la somme des bits est paire et 1 si elle est impaire. Par exemple,

111 0001 devient 0111 0001 et 101 0001 devient 1101 0001

Lors de la réception, on vérifie si le bit de parité correspond ou non à la parité du message reçu. Si le bit de parité n'est pas bon, alors on sait qu'il y a eu au moins une erreur au cours de la transmission. Soit les propositions

$b$ : « Le bit de parité est bon ».

$e$ : « Il y a eu au moins une erreur au cours de la transmission. »

$i$ : « Il y a eu un nombre impair d'erreurs au cours de la transmission. »

- (a) Écrivez les propositions composées suivantes en utilisant les propositions  $b$ ,  $e$  et  $i$  et les connecteurs logiques.

P1: « Si le bit de parité n'est pas bon, alors il y a eu au moins une erreur au cours de la transmission. »

P2: « Si le bit de parité est bon, alors il n'y a pas eu d'erreur au cours de la transmission. »

P3: « S'il n'y a pas eu d'erreur au cours de la transmission, alors le bit de parité est bon. »

P4: « S'il y a eu au moins une erreur au cours de la transmission, alors le bit de parité n'est pas bon. »

P5: « S'il y a eu un nombre impair d'erreurs au cours de la transmission, alors le bit de parité n'est pas bon. »

- (b) Déterminez les propositions qui sont vraies parmi P1 à P5.  
 (c) Indiquez laquelle des propositions P2 à P5 est la réciproque de P1.  
 (d) Indiquez laquelle des propositions P2 à P5 est la contraposée de P1.



**Exemple 1.8** (à compléter en classe)

L'extrait de code suivant fait intervenir les variables booléennes  $p$ ,  $q$  et  $r$ . Chacune de ces variables peut prendre les valeurs **vrai** ou **faux**. Pour chaque bloc indiqué, donnez toutes les valeurs possibles pour  $p$ ,  $q$  et  $r$  au moment où le bloc est atteint.

Notation (utilisée notamment en C/C++, C# et Java) :

- l'opérateur de conjonction  $\wedge$  est noté `&&`,
- l'opérateur de disjonction  $\vee$  est noté `||`.

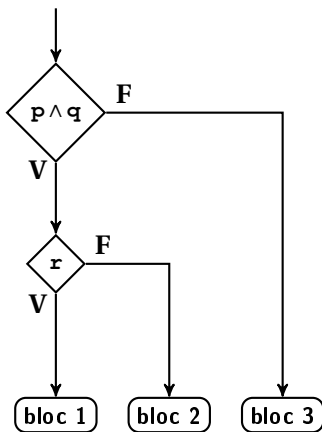
```

if ( p && q ) {
  if( r ) {
    // bloc (1)
  } else {
    // bloc (2)
  }
} else {
  // bloc (3)
}

```

**Solution :**

Il est utile, mais pas nécessaire, de représenter le code du programme par un organigramme de programmation puis de compléter une table de vérité correspondant aux conditions menant à chacun des blocs.



p	q	r				
V	V	V				
V	V	F				
V	F	V				
V	F	F				
F	V	V				
F	V	F				
F	F	V				
F	F	F				

## Exercices

**1.1** Les extraits de code suivants font intervenir les variables booléennes  $p$ ,  $q$  et  $r$ . Chacune de ces variables peut prendre les valeurs **vrai** ou **faux**. Pour chaque bloc indiqué, donnez toutes les valeurs possibles pour  $p$ ,  $q$  et  $r$  au moment où le bloc est atteint.

Notation:

- l'opérateur de conjonction  $\wedge$  est noté `&&`,
- l'opérateur de disjonction  $\vee$  est noté `||`.

(a)

```
if ( p && q || r ) {  
    // bloc (1)  
} else {  
    // bloc (2)  
}
```

(b)

```
if ( p ) {  
    if ( q ) {  
        if ( r ) {  
            // bloc (1)  
        }  
    } else {  
        // bloc (2)  
    }  
}
```

(c)

```
if ( p && q ) {  
    // bloc (1)  
} else {  
    while ( r ) {  
        // bloc (2)  
    }  
    // bloc (3)  
}
```

**1.2** Soit  $p$  et  $q$  les propositions

$p$ : « La température est au-dessous de  $0^\circ$  Celsius. »

$q$ : « L'eau gèle. »

Écrivez les propositions suivantes à l'aide des propositions  $p$  et  $q$  et des connecteurs logiques.

- La température est au-dessous de  $0^\circ$  Celsius et l'eau gèle.
- Si l'eau gèle, alors la température est au-dessous de  $0^\circ$  Celsius.
- La température est au-dessous de  $0^\circ$  Celsius, mais l'eau ne gèle pas.
- La température est au-dessous de  $0^\circ$  Celsius ou l'eau gèle.
- Si la température est au-dessous de  $0^\circ$  Celsius, alors l'eau gèle.
- La température est de  $0^\circ$  Celsius ou plus, mais l'eau ne gèle pas.
- Il est nécessaire et suffisant que la température soit au-dessous de  $0^\circ$  Celsius pour que l'eau gèle.
- Si la température est de  $0^\circ$  Celsius ou plus, alors l'eau ne gèle pas.

**1.3** Parmi les phrases de l'exercice 1.2, y en a-t-il une qui soit la réciproque de l'énoncé (e)? Sa contraposée? Son inverse? Si oui, identifiez ces phrases.

**1.4** Les phrases suivantes sont-elles vraies si on les complète avec « *nécessaire et suffisante* »? Sinon, lequel des termes parmi « *nécessaire* » ou « *suffisante* » faut-il choisir pour que la phrase soit vraie?

- La condition  $x > 5$  est \_\_\_\_\_ pour que le nombre  $x$  soit positif<sup>1</sup>.
- La condition  $x > -5$  est \_\_\_\_\_ pour que  $x$  soit positif.
- La condition  $x > -1$  est \_\_\_\_\_ pour que l'entier  $x$  soit positif.
- Pour que le nombre  $x$  soit premier, il est \_\_\_\_\_ qu'il soit supérieur ou égal à 2.
- Pour qu'un triangle soit rectangle, il est \_\_\_\_\_ que ses côtés mesurent respectivement 3, 4 et 5 centimètres.

**1.5** Réécrivez chacune des phrases de l'exercice précédent sous la forme « il est nécessaire que... pour que... », « il suffit que... pour que... » ou encore « il faut et il suffit que... pour que... ».

**1.6** Vrai ou faux?

- Il est *nécessaire* que les quatre angles d'un quadrilatère  $Q$  mesurent  $90^\circ$  pour que  $Q$  soit un carré.
- Il *suffit* que le quadrilatère  $Q$  ait deux paires de côtés parallèles pour qu'il soit un rectangle.

---

1. On considère que 0 est un nombre positif et négatif.

**1.7** Soit  $p$  et  $q$  les propositions

$p$ : « Vous avez fait tous les exercices du cours. »

$q$ : « Vous avez obtenu la note A+. »

Écrivez les propositions suivantes à l'aide de  $p$ , de  $q$  et des connecteurs logiques.

- (a) Il est nécessaire que vous ayez fait tous les exercices du cours pour avoir obtenu la note A+.
- (b) Vous avez fait tous les exercices du cours, mais vous n'avez pas obtenu la note A+.
- (c) Si vous avez fait tous les exercices du cours, alors vous avez obtenu la note A+.
- (d) Si vous n'avez pas fait tous les exercices du cours, alors vous n'avez pas obtenu la note A+.
- (e) Vous avez obtenu la note A+, mais vous n'avez pas fait tous les exercices du cours.
- (f) Si vous avez obtenu la note A+, alors vous avez fait tous les exercices du cours.
- (g) Si vous n'avez pas obtenu la note A+, on peut en déduire que vous n'avez pas fait tous les exercices du cours.
- (h) Vous n'avez pas fait tous les exercices du cours et vous n'avez pas obtenu la note A+.
- (i) Vous avez fait tous les exercices du cours ou vous n'avez pas obtenu la note A+.
- (j) Vous avez obtenu la note A+ si et seulement si vous avez fait tous les exercices du cours.

**1.8** Parmi les phrases de l'exercice 1.7, identifiez la réciproque, la contraposée et l'inverse de l'énoncé (c): « Si vous avez fait tous les exercices du cours, alors vous avez obtenu la note A+. »

**1.9** Déterminez si  $p$  est une condition *nécessaire et suffisante* pour  $q$ . Dans le cas contraire, déterminez si  $p$  est *suffisante* pour  $q$  ou si elle est *nécessaire* pour  $q$ . Les domaines des variables sont sous-entendus.

- (a)  $p$ : «  $x$  est un nombre supérieur ou égal à 0 »,  $q$ : «  $x$  est le carré d'un nombre réel. »
- (b)  $p$ : «  $t$  est un triangle rectangle »,  
 $q$ : «  $t$  est un triangle dont un angle mesure  $60^\circ$  et un autre mesure  $30^\circ$ . »
- (c)  $p$ : «  $t$  est un triangle rectangle »,  $q$ : «  $t$  est un triangle dont un des angles mesure  $90^\circ$ . »
- (d)  $p$ : «  $t$  est une partie de tic-tac-toe terminée »,  $q$ : «  $t$  comporte au moins 5 symboles X ou O. »
- (e)  $p$ : «  $t$  est une partie de tic-tac-toe terminée »,  $q$ : «  $t$  comporte au moins 9 symboles X ou O. »
- (f)  $p$ : «  $x$  suit le cours de Mathématiques discrètes de l'ÉTS et fait le présent exercice »,  
 $q$ : «  $x$  étudie à l'ÉTS ou aide gentiment un étudiant de l'ÉTS. »
- (g)  $p$ : «  $x$  est un carré »,  $q$ : «  $x$  est un rectangle. »

**1.10** Construisez une table de vérité pour les propositions suivantes.

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

$p$	$q$	$p \rightarrow q$	$\neg q$	$(p \rightarrow q) \wedge \neg q$
V	V			
V	F			
F	V			
F	F			

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

$p$	$q$	$p \rightarrow q$	$\neg p \vee q$	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$
V	V			
V	F			
F	V			
F	F			

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

$p$	$q$	$r$	$p \rightarrow q$	$(p \rightarrow q) \rightarrow r$
V	V	V		
V	V	F		
V	F	V		
V	F	F		
F	V	V		
F	V	F		
F	F	V		
F	F	F		

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

$p$	$q$	$r$		
V	V	V		
V	V	F		
V	F	V		
V	F	F		
F	V	V		
F	V	F		
F	F	V		
F	F	F		

### 1.1.2 Équivalences propositionnelles

#### Définition 1.4 : Tautologie, contradiction, contingence

Une proposition composée qui est toujours vraie, quelle que soit la valeur de vérité des propositions qui la composent, est appelée une **tautologie**.

Une proposition composée qui est toujours fausse est appelée une **contradiction**.

Une proposition qui n'est ni une tautologie ni une contradiction est appelée une **contingence**.

#### Exemple 1.9 (à compléter en classe)

Déterminez si chacune des propositions suivantes est une tautologie, une contradiction ou une contingence.

(a)  $p \vee \neg p$

(b)  $p \rightarrow q$

(c)  $p \wedge \neg p$

#### Définition 1.5 : Équivalence de propositions

Les propositions  $p$  et  $q$  sont dites **logiquement équivalentes** si la proposition  $p \leftrightarrow q$  est une tautologie. Ainsi, deux propositions sont logiquement équivalentes si elles ont la même table de vérité, c'est-à-dire la même valeur de vérité dans tous les cas possibles.

Les notations  $p \equiv q$  et  $p \Leftrightarrow q$  signifient que  $p$  et  $q$  sont logiquement équivalentes.

#### Exemple 1.10 (à compléter en classe)

Vérifiez l'équivalence suivante à l'aide d'une table de vérité.

$$p \rightarrow q \equiv \neg p \vee q$$

#### Solution :

Afin de vérifier l'équivalence, il faut s'assurer que la proposition  $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$  est toujours vraie. Construisons sa table de vérité.

$p$	$q$	$p \rightarrow q$	$\neg p \vee q$	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$
V	V			
V	F			
F	V			
F	F			

**Exemple 1.11** (à compléter en classe)

Vérifiez l'équivalence suivante.

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

**Solution :**

Construisons la table de vérité de chacune des deux propositions afin de montrer qu'elles ont la même valeur de vérité dans tous les cas possibles.

$p$	$q$	$p \vee q$	$\neg(p \vee q)$
V	V		
V	F		
F	V		
F	F		

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

Pour gagner du temps, on note les équivalences fréquemment utilisées dans une table et on leur donne un nom ou un numéro afin d'y faire référence. Par exemple, l'équivalence de l'exemple 1.11 est appelée *loi de De Morgan*, en l'honneur du mathématicien britannique Auguste De Morgan qui vécut au 19<sup>e</sup> siècle. On la retrouve à la 8<sup>e</sup> ligne de la table 1. Quant à l'équivalence de l'exemple 1.10, elle se retrouve à la première ligne de la table 2. Pour gagner du temps, on note les équivalences fréquemment utilisées dans une table et on leur donne un nom ou un numéro afin d'y faire référence. Par exemple, l'équivalence de l'exemple 1.11 est appelée *loi de De Morgan*, en l'honneur du mathématicien britannique Auguste De Morgan qui vécut au 19<sup>e</sup> siècle. On la retrouve à la 8<sup>e</sup> ligne de la table 1. Quant à l'équivalence de l'exemple 1.10, elle se retrouve à la première ligne de la table 2.

**TABLE 1** Équivalences logiques

1	$p \wedge \mathbf{V} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identité
2	$p \vee \mathbf{V} \equiv \mathbf{V}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination
3	$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotence
4	$\neg(\neg p) \equiv p$	Double négation
5	$p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$	Commutativité
6	$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associativité
7	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributivité
8	$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	Lois de De Morgan
9	$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption
10	$p \vee \neg p \equiv \mathbf{V}$ $p \wedge \neg p \equiv \mathbf{F}$	Négation

**TABLE 2** Équivalences logiques (implications)

1	$p \rightarrow q \equiv \neg p \vee q$
2	$p \rightarrow q \equiv \neg q \rightarrow \neg p$
3	$p \vee q \equiv \neg p \rightarrow q$
4	$p \wedge q \equiv \neg(p \rightarrow \neg q)$
5	$\neg(p \rightarrow q) \equiv p \wedge \neg q$
6	$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
7	$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
8	$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
9	$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

**TABLE 3** Équivalences logiques (biconditionnelles)

1	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
2	$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
3	$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
4	$p \leftrightarrow q \equiv \neg(p \wedge \neg q) \wedge \neg(\neg p \wedge q)$
5	$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

**Exemple 1.12**

Vérifiez que la proposition

$$\neg(p \rightarrow q) \rightarrow \neg q$$

est une tautologie

- à l'aide d'une table de vérité;
- sans l'aide d'une table de vérité, en utilisant les tables d'équivalences;

**Solution :**

- À l'aide d'une table de vérité.

$p$	$q$	$p \rightarrow q$	$\neg(p \rightarrow q)$	$\neg q$	$\neg(p \rightarrow q) \rightarrow \neg q$
<b>V</b>	<b>V</b>	<b>V</b>	<b>F</b>	<b>F</b>	<b>V</b>
<b>V</b>	<b>F</b>	<b>F</b>	<b>V</b>	<b>V</b>	<b>V</b>
<b>F</b>	<b>V</b>	<b>V</b>	<b>F</b>	<b>F</b>	<b>V</b>
<b>F</b>	<b>F</b>	<b>V</b>	<b>F</b>	<b>V</b>	<b>V</b>



(b) En utilisant les tables d'équivalences

$$\begin{aligned}
 \neg(p \rightarrow q) \rightarrow \neg q &\equiv \neg(\neg(p \rightarrow q)) \vee \neg q && \text{Table 2.1} \\
 &\equiv (p \rightarrow q) \vee \neg q && \text{Double négation} \\
 &\equiv (\neg p \vee q) \vee \neg q && \text{Table 2.1} \\
 &\equiv \neg p \vee (q \vee \neg q) && \text{Associativité} \\
 &\equiv \neg p \vee \mathbf{V} && \text{Négation} \\
 &\equiv \mathbf{V} && \text{Domination}
 \end{aligned}$$

## Exercices

**1.11** À l'aide d'une table de vérité, déterminez si la proposition suivante est une tautologie, une contingence ou une contradiction.

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

**1.12** Vérifiez les équivalences suivantes à l'aide d'une table de vérité.

- (a)  $p \rightarrow q \equiv \neg q \rightarrow \neg p$
- (b)  $\neg(p \vee q) \equiv \neg p \wedge \neg q$

**1.13** Utilisez les lois de De Morgan pour formuler la négation des énoncés suivants.

- (a) Ce programme est rapide et efficace.
- (b) Le programme doit être en C+ ou en Java.

**1.14** Montrez que, pour les langages Java ou C, les expressions suivantes sont équivalentes.

$$!(x < 10 \ || \ x > 20) \quad \text{et} \quad x \geq 10 \ \&\& \ x \leq 20$$

Rappel:

symbole en Java ou C	symbole logique
!	$\neg$
$\geq$	$\geq$
$\leq$	$\leq$
	$\vee$
&&	$\wedge$

**1.15** Prouvez que

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

- (a) à l'aide d'une table de vérité;
- (b) en utilisant les tables d'équivalences;
- (c) en entrant l'expression sur Nspire.

#### Propositions équivalentes ou non?

Pour démontrer que les propositions **ne sont pas** équivalentes, il suffit de fournir des valeurs de  $p$ ,  $q$  et  $r$  pour lesquelles elles diffèrent.

Pour démontrer que les propositions **sont** équivalentes, on peut procéder de l'une des trois façons suivantes.

1. Fournir leur table de vérité.
2. Utiliser les tables d'équivalence logique de la page 14.
3. Formuler une explication en mots qui montre que les deux propositions sont vraies, ou encore que les deux sont fausses, exactement pour les mêmes combinaisons de valeur de vérité des variables propositionnelles.

**1.16** Déterminez si les propositions suivantes sont logiquement équivalentes (consultez l'encadré ci-dessus).

- (a)  $(p \rightarrow q) \rightarrow r$  et  $p \rightarrow (q \rightarrow r)$
- (b)  $(p \rightarrow r) \vee (q \rightarrow r)$  et  $(p \wedge q) \rightarrow r$  (sans utiliser la table 2, ligne 9).

**1.17** Vérifiez les équivalences suivantes à l'aide des tables d'équivalences.

- (a)  $(\neg q \rightarrow p) \rightarrow (r \vee p) \equiv (\neg p \wedge q) \rightarrow r$
- (b)  $((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r) \equiv \mathbf{vrai}$  (tautologie)

**1.18** Une entreprise doit recruter une petite équipe pour réaliser un projet à l'extérieur du pays. Suite aux entrevues de 8 postulants, à l'analyse des compétences et des besoins, ainsi qu'aux questions de bonne entente, les contraintes suivantes ont été identifiées.

1. Au moins un des postulants numéro 2, 4 ou 5 doit être embauché, car eux seuls ont des connaissances en électricité.
2. Si le postulant 2 est embauché ou le postulant 4 est embauché, alors le postulant 5 ne doit pas l'être.
3. Si les postulants 2 et 4 sont embauchés tous les deux, alors le postulant 5 ne doit pas l'être.
4. Les postulants 3 et 5 forment un couple: ils accepteront l'affectation seulement si les deux sont embauchés.
5. Il est impossible d'embaucher à la fois les postulants 2 et 5, et il est impossible d'embaucher à la fois les postulants 4 et 5.
6. Il est impossible d'embaucher le trio de postulants 2, 4 et 5: on peut en choisir un, deux ou aucun, mais pas les trois.

7. Il faut absolument embaucher au moins 2 personnes parmi les postulants 1 à 4.
8. Il faut embaucher les postulants 1 et 3, ou embaucher le postulant 2 et au moins un des postulants parmi 1, 3 et 4, ou encore embaucher le postulant 4 et au moins un des postulants parmi 1, 2 et 3.

(a) En utilisant les propositions  $p_1$  à  $p_8$ , où

$p_i$  : « le postulant  $i$  est embauché »,

traduisez chacune des contraintes à l'aide de connecteurs logiques.

(b) Déterminez quelles contraintes sont équivalentes et démontrez-le à l'aide des règles des tables 1 à 3 de la page 16

### 1.1.3 Prédicats et quantificateurs

Un énoncé contenant une ou plusieurs variables tel que

$$x < 10 \quad \text{ou} \quad x + 2 = 7 - y$$

n'est pas une proposition puisque, tant que la valeur de  $x$  ou  $y$  n'est pas connue, on ne peut dire s'il est vrai ou faux.

#### Terminologie

Dans l'énoncé «  $x < 10$  »,  $x$  est le **sujet**, et « est inférieur à 10 » est le **prédicat**. Notons  $P(x)$  l'énoncé  $x < 10$ . On dit que  $P$  est une **fonction propositionnelle**.

Une fonction propositionnelle  $P(x)$  prend la valeur vrai ou faux quand  $x$  est précisé. Par exemple :

- $P(8)$  est une proposition vraie. On écrira parfois  $P(8)$  est vrai (au masculin, en sous-entendant l'énoncé est vrai », ou même  $P(8) \equiv \mathbf{V}$ ).
- $P(13)$  est une proposition fausse.
- $P(\text{Julie})$  n'est pas une proposition, car Julie n'est pas une valeur possible pour la variable  $x$ .

L'ensemble des valeurs possibles pour la variable  $x$  est appelé **univers du discours**, ou **domaine** de la fonction  $P$ .

**Définition 1.6 : Quantificateurs**

$\forall$  : quantificateur universel     $\exists$  : quantificateur existentiel

La proposition  $\forall x P(x)$  signifie « Pour toutes les valeurs de  $x$  dans l'univers du discours,  $P(x)$  ». Ou encore « Quel que soit  $x$  (dans l'univers du discours),  $P(x)$ . »

La proposition  $\exists x P(x)$  signifie « Il existe un élément  $x$  de l'univers du discours tel que  $P(x)$  ». Ou encore « Il y a au moins un  $x$  (dans l'univers du discours) tel que  $P(x)$ . » Ou encore « Pour un certain  $x$  (dans l'univers du discours),  $P(x)$ . »

**Notation.** Certains auteurs mettent une virgule avant la fonction propositionnelle, surtout quand celle-ci est composée. Par exemple:  $\forall x, (P_1(x) \rightarrow P_2(x) \vee P_3(x))$ . Par ailleurs, si l'ensemble  $U$  n'a pas déjà été identifié, on peut préciser que la variable  $x$  prendra des valeurs dans l'ensemble  $U$  ainsi:  $\exists x \in U, P(x)$ .

Lorsque l'univers du discours est un ensemble fini  $\{x_1, x_2, \dots, x_n\}$ , on a les équivalences logiques suivantes:

$$\forall x P(x) \equiv P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n),$$

$$\exists x P(x) \equiv P(x_1) \vee P(x_2) \vee \dots \vee P(x_n).$$

La quantification universelle  $\forall x P(x)$  est vraie quand  $P(x)$  est vraie pour toutes les valeurs de  $x$  dans l'univers du discours  $U$ . Elle est donc fausse s'il existe un  $x$  de  $U$  pour lequel  $P(x)$  est fausse. Un tel élément est appelé un **contre-exemple** de  $\forall x P(x)$ .

La quantification existentielle  $\exists x P(x)$  est vraie s'il existe au moins une valeur  $x$  dans l'univers du discours telle que  $P(x)$  est vraie. Elle est fausse si  $P(x)$  est fausse pour toutes les valeurs possibles de  $x$ .

Ainsi, pour prouver qu'un énoncé de la forme  $\forall x P(x)$  est vrai, fournir un exemple de  $x$  tel que  $P(x)$  est vrai ne suffit pas. Il faut montrer que la proposition  $P(x)$  est vraie pour toutes les valeurs de  $x$ , ce qui peut s'avérer particulièrement **difficile lorsque  $U$  est un ensemble infini**. Il en va de même lorsqu'on veut prouver qu'un énoncé de la forme  $\exists x P(x)$  est faux. Le tableau 1.1 résume les différents cas possibles.

Pour prouver que	est <b>vrai</b>	est <b>faux</b>
$\exists x P(x)$	il suffit de fournir <b>un exemple</b> : un $x$ de $U$ tel que $P(x)$ est vrai.	il faut fournir un argument général pour montrer que $P(x)$ est faux quelque soit $x$ de $U$ .
$\forall x P(x)$	il faut fournir un argument général pour montrer que $P(x)$ est vrai quelque soit $x$ de $U$ .	il suffit de fournir <b>un contre-exemple</b> : un $x$ de $U$ tel que $P(x)$ est faux.

Tableau 1.1 Comment prouver qu'un énoncé quantifié est vrai ou faux quand l'univers du discours  $U$  est infini.

**Exemple 1.13** (à compléter en classe)

Si l'univers du discours est l'ensemble des nombres réels et

$P(x)$  désigne «  $x \geq 0$  »

$Q(x)$  désigne «  $x$  est un nombre premier »

$R(x)$  désigne «  $3^x + 4^x = 5^x$  »

$S(x)$  désigne «  $x \geq 100$  »,

dites si chacun des énoncés suivants est une proposition vraie, une proposition fausse ou n'est pas une proposition. Donnez un exemple ou un contre-exemple le cas échéant. Dans le cas contraire, indiquez qu'un argument général est requis.

(a)  $\forall x P(x)$

(b)  $\forall x \neg P(x)$

(c)  $\forall x P(x^2)$

(d)  $\exists x P(x)$

(e)  $\exists x \neg P(x)$

(f)  $\exists x Q(x)$

(g)  $\exists x Q(x^2)$

(h)  $\forall x R(x)$

(i)  $P(x)$

(j)  $\forall x (S(x) \rightarrow P(x))$

(k)  $(\forall x P(x)) \rightarrow (\forall x S(x))$

(l)  $\forall x S(x + 100)$

(m)  $\forall x S(x^2 + 100)$

**Théorème 1.1 : Lois de De Morgan pour les quantificateurs**

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

**Exemple 1.14** (à compléter en classe)

Si l'univers du discours est l'ensemble des employés de l'ÉTS et  $M(x)$  désigne l'énoncé « L'employé  $x$  peut modifier les fichiers du répertoire  $U$  », traduisez clairement les propositions suivantes à l'aide des quantificateurs.

- (a) Tous les employés de l'ÉTS peuvent modifier les fichiers du répertoire  $U$ .
- (b) Il est faux que tous les employés de l'ÉTS peuvent modifier les fichiers du répertoire  $U$ .
- (c) Au moins un employé de l'ÉTS peut modifier les fichiers du répertoire  $U$ .
- (d) Il est faux qu'au moins un employé de l'ÉTS peut modifier les fichiers du répertoire  $U$ .
- (e) Aucun employé de l'ÉTS ne peut modifier les fichiers du répertoire  $U$ .
- (f) Au moins un employé de l'ÉTS ne peut pas modifier les fichiers du répertoire  $U$ .

De plus, déterminez les propositions ci-dessus qui sont équivalentes.

**Exemple 1.15** (à compléter en classe)

Si l'univers du discours est l'ensemble des billes contenues dans un bol, et si

$G(x)$  désigne « la bille  $x$  est grosse »     $R(x)$  désigne « la bille  $x$  est rouge »  
 $J(x)$  désigne « la bille  $x$  est jaune »     $B(x)$  désigne « la bille  $x$  est bleue »

traduisez clairement les propositions suivantes en prenant soin de bien formuler les phrases.

- (a)  $\forall x (R(x) \vee J(x))$
- (b)  $(\forall x R(x)) \vee (\forall x J(x))$
- (c) Les propositions (a) et (b) sont-elles équivalentes?
- (d)  $\exists x B(x)$

(e)  $\neg (\exists x B(x))$

(f) Utilisez le quantificateur universel  $\forall$  pour écrire une proposition équivalente à la précédente.

(g)  $\neg (\forall x R(x))$

(h) Utilisez le quantificateur existentiel  $\exists$  pour écrire une proposition équivalente à la précédente.

(i)  $\forall x (G(x) \rightarrow B(x))$

(j)  $\exists x (G(x) \wedge B(x))$

(k)  $(\exists x G(x)) \wedge (\exists x B(x))$

(l) Les deux propositions précédentes sont-elles équivalentes?

(m) Les deux propositions suivantes sont-elles équivalentes?

$$(\exists x R(x)) \vee (\exists x J(x)) \quad \text{et} \quad \exists x (R(x) \vee J(x))$$

(n) Les deux propositions suivantes sont-elles équivalentes?

$$(\forall x R(x)) \wedge (\forall x G(x)) \quad \text{et} \quad \forall x (R(x) \wedge G(x))$$

---

Les équivalences des deux dernières sous-questions se retrouvent dans la table 4 : Équivalences logiques (énoncés quantifiés).

1	$\forall x (P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$
2	$\exists x (P(x) \vee Q(x)) \equiv \exists x P(x) \vee \exists x Q(x)$
3	$\neg \exists x P(x) \equiv \forall x \neg P(x)$
4	$\neg \forall x P(x) \equiv \exists x \neg P(x)$

### Exercices

**1.19** Soit  $P(x)$  l'énoncé «  $x = x^2$  ». Si l'univers du discours est l'ensemble des nombres réels, quelles sont les valeurs de vérité des propositions suivantes ?

**De plus**, pour chacun des énoncés, dites si un exemple ou un contre-exemple suffit à justifier votre réponse ou si un argument général serait requis.

(a)  $\exists x P(x)$

(b)  $\forall x P(x)$

(c)  $\exists x \neg P(x)$

(d)  $\forall x \neg P(x)$

**1.20** Supposons que l'univers du discours d'une fonction propositionnelle  $P(x)$  est l'ensemble des entiers  $\{0, 1, 2\}$ . Écrivez chacune des propositions suivantes sans avoir recours à un quantificateur.

(a)  $\exists x P(x)$

(b)  $\forall x P(x)$

(c)  $\neg \exists x P(x)$

(d)  $\neg \forall x P(x)$

(e)  $\exists x \neg P(x)$

(f)  $\forall x \neg P(x)$



**1.21** Associez chacune des phrases à sa traduction ou à une formulation équivalente utilisant les quantificateurs et les prédicats suivants, avec l'ensemble des humains comme univers du discours,  $I(x)$  : «  $x$  est un ingénieur »,  $P(x)$  : «  $x$  sait programmer ». *Le même choix de réponse peut être utilisé plus d'une fois.*

- (a) Il existe au moins un ingénieur qui sait programmer.
- (b) Tous les ingénieurs savent programmer.
- (c) Il existe un ingénieur qui ne sait pas programmer.
- (d) Aucun humain ne sait programmer.
- (e) Aucun ingénieur ne sait programmer.
- (f) Il n'existe aucun ingénieur qui ne sait pas programmer.
- (g) Il existe quelqu'un qui sait programmer, mais n'est pas ingénieur.
- (h) Les ingénieurs ne savent pas programmer.
- (i) Les programmeurs ne sont pas tous ingénieurs.

- |  |  |   |
|--|--|---|
| 1. $\exists x (I(x) \wedge P(x))$      | 4. $\exists x (I(x) \wedge \neg P(x))$ | 7. $\forall x \neg P(x)$                  |
| 2. $\exists x (I(x) \vee P(x))$        | 5. $\exists x I(x)$                    | 8. $\forall x (I(x) \rightarrow P(x))$    |
| 3. $\exists x (P(x) \wedge \neg I(x))$ | 6. $\exists x \neg I(x)$               | 9. $\forall x (\neg I(x) \vee \neg P(x))$ |

**1.22** Reprenons le contexte de l'exercice précédent, où l'univers du discours est cette fois l'ensemble des six employés d'une petite entreprise. Pour chacune des propositions, dites si elle est **vraie ou fausse** en vous fiant au tableau des caractéristiques des employés: le 1 dans une case désigne que l'employé possède la caractéristique, tandis qu'une case vide signifie le contraire.

Employés	Alain	Baba	Claudia	Denis	Émile	France
caractéristiques						
ingénieur.e		1	1		1	
sait programmer	1	1	1		1	

- (a)  $\exists x (I(x) \wedge P(x))$
- (b)  $\forall x (I(x) \vee P(x))$
- (c)  $\forall x \neg (I(x) \vee P(x))$
- (d)  $\exists x \neg (I(x) \vee P(x))$
- (e)  $\exists x (P(x) \wedge \neg I(x))$
- (f)  $\forall x \neg P(x)$
- (g)  $\forall x (I(x) \rightarrow P(x))$
- (h)  $\forall x (P(x) \rightarrow I(x))$
- (i)  $\forall x (I(x) \leftrightarrow P(x))$

**1.23** Soit  $M(x)$  l'énoncé «  $x$  a étudié au cégep Maisonneuve » et soit  $J(x)$  l'énoncé «  $x$  connaît le langage Java », où l'univers du discours est l'ensemble des étudiants de l'ÉTS. Exprimez chacune des phrases suivantes en fonctions de  $M(x)$ , de  $J(x)$ , de quantificateurs et de connecteurs logiques.

- (a) Il y a un étudiant de l'ÉTS qui a étudié au cégep Maisonneuve et qui connaît le langage Java.
- (b) Il y a un étudiant de l'ÉTS qui a étudié au cégep Maisonneuve, mais ne connaît pas le langage Java.
- (c) Chaque étudiant de l'ÉTS connaît Java.
- (d) Aucun étudiant de l'ÉTS a étudié au cégep Maisonneuve ou connaît le langage Java.
- (e) Tous les étudiants de l'ÉTS ayant étudié au cégep Maisonneuve connaissent le langage Java.

**1.24** Déterminez la valeur de vérité de chacun des énoncés suivants si l'univers du discours est l'ensemble des nombres entiers. **De plus**, pour chacun des énoncés, dites si un exemple ou un contre-exemple suffit à justifier votre réponse ou si un argument général serait requis.

- (a)  $\exists n (n = -2n)$
- (b)  $\forall n (2n \leq 3n)$
- (c)  $\exists n (n^2 < n)$
- (d)  $\forall n (n^2 > 0)$
- (e)  $\forall n (n < n + 2)$

**1.25** Traduisez chacun des énoncés suivants en expressions logiques à l'aide de quantificateurs, de connecteurs logiques et des fonctions propositionnelles suivantes.  $A(x)$  désigne «  $x$  est votre ami »;  $V(x)$  désigne «  $x$  a une voiture électrique »; l'univers du discours est l'ensemble des humains.

- (a) Au moins un de vos amis n'a pas de voiture électrique.
- (b) Tous ceux qui ont une voiture électrique sont vos amis.
- (c) Personne n'a de voiture électrique.
- (d) Ce n'est pas tout le monde qui a une voiture électrique.
- (e) Tous vos amis ont une voiture électrique.
- (f) Au moins un de vos amis a une voiture électrique.
- (g) Tout le monde est votre ami et a une voiture électrique.
- (h) Ce n'est pas tout le monde qui est votre ami ou il y a quelqu'un qui n'a pas de voiture électrique.

**1.26** Considérez les prédicats suivants, où l'univers du discours est l'ensemble des dix chiffres:  
 $C = \{0, 1, 2, 3, \dots, 9\}$ .

$$P(x) : \langle (x^5 \bmod 10) = x \rangle$$

$$T(x) : \langle (x^3 \bmod 10) = x \rangle.$$

Pour le lecteur qui ne connaît pas encore le modulo, disons simplement que le résultat modulo 10 d'un nombre naturel est le dernier chiffre du nombre (le chiffre des unités) :  $173 \bmod 10 = 3$ .

Quelles sont les valeurs de vérité des propositions suivantes? Justifiez.

- (a)  $P(2) \wedge T(4)$
- (b)  $\forall x T(x)$
- (c)  $\exists x T(x)$
- (d)  $\exists x \neg T(x)$
- (e)  $\forall x P(x)$
- (f)  $\exists x \neg P(x)$

**1.27** Considérez la proposition suivante: « Quel que soit le nombre naturel  $x$ ,  $(x^5 \bmod 10) = (x \bmod 10)$ . »

Complétez la phrase suivante avec un ou plusieurs des choix de réponses. « Pour déterminer les valeurs de vérité de la proposition  $\forall x P(x)$ , ... »

1. il suffit de fournir un exemple pour montrer qu'elle est vraie;
2. il suffit de fournir un contre-exemple pour montrer qu'elle est fautive;
3. il faut présenter une preuve générale pour montrer qu'elle est vraie, car l'univers du discours est infini;
4. il faut présenter une preuve générale pour montrer qu'elle est fautive, car l'univers du discours est infini.

### 1.1.4 Quantificateurs imbriqués

Pour les quantifications à deux variables, on a les équivalences logiques suivantes :

$$\forall x \forall y P(x, y) \equiv \forall y \forall x P(x, y) \quad \text{et} \quad \exists x \exists y P(x, y) \equiv \exists y \exists x P(x, y).$$

Par contre, il faut faire attention à l'ordre lorsque les quantificateurs universels et existentiels sont imbriqués. En général,

$$\forall x \exists y P(x, y) \not\equiv \exists y \forall x P(x, y).$$

#### Exemple 1.16

Soit  $E$  un ensemble<sup>2</sup> d'employés et  $R$  l'ensemble des répertoires de leur système informatique.

Désignons par  $M(x, B)$  l'énoncé « l'employé  $x$  peut **modifier** le répertoire  $B$  » et par  $L(x, B)$  l'énoncé « l'employé  $x$  peut **lire** le répertoire  $B$  ».

Exprimez la proposition suivante à l'aide de quantificateurs et des fonctions propositionnelles  $M$  et  $L$ .

« Il existe un répertoire pouvant être lu par tous les employés »

#### Solution :

$$\exists B \in R, \forall x \in E, L(x, B) \quad \text{ou} \quad \exists B \forall x L(x, B)$$

Dans la deuxième expression, on peut déduire quels sont les ensembles de références en observant la position de la variable dans le prédicat. Première variable : employé. Deuxième variable : répertoire.

Plusieurs étudiants éprouvent des difficultés avec les quantificateurs universel et existentiel, particulièrement lorsqu'ils sont imbriqués. Voici deux conseils, suivis de quelques exemples utilisant les mêmes fonctions propositionnelles  $M$  et  $L$ . Mais attention de ne pas toujours chercher le truc, il faut vraiment réfléchir au sens de la phrase.

1. **Réécrire la phrase sous forme passive avant de traduire du français à l'expression logique.**
2. **Placer les sujets du prédicat au début de la phrase.**

Dans les exemples suivants, on demande encore d'exprimer la proposition à l'aide de quantificateurs et des fonctions propositionnelles  $M$  et  $L$ . De plus, **le symbole de négation ne doit apparaître que devant les propositions simples** (pas devant un quantificateur ni devant une proposition contenant un ou des connecteurs).

2. La notion d'ensemble ainsi que le symbole d'appartenance  $\in$  sont définis à la page 48.

**Exemple 1.17**

Chaque employé peut modifier au moins un répertoire.

**Solution :**

Récrivons d'abord cette phrase sous forme passive avec les sujets du prédicat au début. Ajoutons aussi les variables  $x$  et  $B$ .

Pour chaque employé  $x$ , il existe au moins un répertoire  $B$  (qui dépend de l'employé  $x$ ) qui peut être modifié par l'employé  $x$ . Passons ensuite à l'expression logique.

$$\forall x \in E, \exists B \in R, M(x, B)$$

**Attention!** Puisque le répertoire  $B$  dépend de l'employé  $x$ ,  $B$  apparaît après  $x$ . Chaque employé peut modifier un répertoire, mais il ne s'agit peut-être pas du même répertoire.

**Erreur type:** inversion des quantificateurs.  $\exists B \in R, \forall x \in E, M(x, B)$  Ceci signifie: *il existe au moins un répertoire  $B$  tel que, pour chaque employé  $x$ ,  $x$  peut modifier  $B$ .* Ou encore: *il existe au moins un répertoire pouvant être modifié par tous les employés.*

**Exemple 1.18**

Il est faux de dire que chaque employé peut modifier au moins un répertoire.

**Solution :**

Comme nous l'avons vu avec les lois de De Morgan (1.1), on construit la négation d'une proposition contenant un quantificateur en changeant  $\forall$  pour  $\exists$  ou réciproquement puis en remplaçant la proposition quantifiée par sa négation.

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

Ainsi, en appliquant deux fois cette règle, on obtient

$$\begin{aligned} & \neg(\forall x \in E, \exists B \in R, M(x, B)) \\ & \equiv \exists x \in E \neg(\exists B \in R, M(x, B)) \\ & \equiv \exists x \in E \forall B \in R, \neg M(x, B) \end{aligned}$$

**Exemple 1.19**

Guy peut modifier tous les répertoires que Manon peut lire.

**Solution :**

Forme passive: *Chaque répertoire qui peut être lu par Manon peut être modifié par Guy.*

Puisqu'il s'agit d'une implication, introduisons les mots *si... alors...*

*Pour chaque répertoire  $B$ , si ce répertoire  $B$  peut être lu par Manon, alors ce répertoire  $B$  peut être modifié par Guy.*

$$\forall B \in R, L(\text{Manon}, B) \rightarrow M(\text{Guy}, B)$$

**Exemple 1.20**

*Il est faux de dire que Guy peut modifier tous les répertoires que Manon peut lire.*

**Solution :**

Appliquons une des lois de De Morgan puis utilisons ensuite les tables des équivalences logiques pour trouver une proposition équivalente où le symbole de négation n'apparaît que devant les propositions simples:  $\neg(p \rightarrow q) \equiv p \wedge \neg q$ .

$$\begin{aligned} & \neg(\forall B \in R, L(\text{Manon}, B) \rightarrow M(\text{Guy}, B)) \\ & \equiv \exists B \in R, \neg(L(\text{Manon}, B) \rightarrow M(\text{Guy}, B)) \\ & \equiv \exists B \in R, (L(\text{Manon}, B) \wedge \neg M(\text{Guy}, B)) \end{aligned}$$

**Exemple 1.21**

*Les répertoires qui ne peuvent être modifiés par Guy ne peuvent l'être par Manon.*

**Solution :**

Remarquons que l'expression « Les répertoires » a ici le sens de « tous les répertoires » ou « chaque répertoire ».

Passons à la forme passive et introduisons clairement les mots clés *si.. alors...*

*Pour chaque répertoire B, si ce répertoire B ne peut pas être modifié par Guy alors ce répertoire B ne peut pas être modifié par Manon.*

$$\forall B \in R, \neg M(\text{Guy}, B) \rightarrow \neg M(\text{Manon}, B)$$

**Exemple 1.22**

*Il existe un employé qui peut modifier tous les répertoires.*

**Solution :**

Forme passive où les sujets apparaissent au début de la phrase :

*Il existe un employé x tel que chaque répertoire B peut être modifié par cet employé x.*

$$\exists x \in E, \forall B \in R, M(x, B)$$

**Voici deux erreurs types.** La première consiste à copier la structure de la phrase initiale. *Il existe un employé qui peut modifier tous les répertoires.*

$$\exists x \in E, M(x, \forall B)$$

Mais un quantificateur ne peut pas être sujet d'un prédicat: pas de  $\forall$  ou  $\exists$  à l'intérieur du  $M$ .

La deuxième erreur consiste à inverser l'ordre des quantificateurs, ce qui modifie complètement le sens de la phrase.

$$\forall B \in R, \exists x \in E, M(x, B)$$

Cette proposition signifie que pour chaque répertoire  $B$ , il existe au moins un employé  $x$  (qui dépend du répertoire  $B$ ) tel que  $x$  peut modifier  $B$ . Chaque répertoire peut donc être modifié par au moins un employé, mais ce n'est pas nécessairement le même employé qui peut modifier tous les répertoires.

## Exercices

**1.28** Soit  $X$  l'ensemble des 6 employés d'une petite équipe et  $R$  l'ensemble des 5 répertoires de leur système informatique. Désignons par  $M(x, B)$  l'énoncé « l'employé  $x$  peut **modifier** le répertoire  $B$  » et par  $L(x, B)$  l'énoncé « l'employé  $x$  peut **lire** le répertoire  $B$  ».

Écrivez les propositions suivantes **en mots**. De plus, pour chacune des propositions, dites si elle est **vraie ou fausse** en vous fiant au tableau de caractéristiques des employés: un L dans une case désigne que l'employé peut lire le répertoire de cette ligne, un M signifie qu'il peut le modifier, tandis que l'absence d'une lettre L ou M signifie que l'employé ne peut lire ou modifier le répertoire.

(a)  $\forall B \in R, \exists x \in X, M(x, B)$

(i)  $\exists B \in R, \exists! x \in X, M(x, B)$

(b)  $\exists x \in X, \forall B \in R, M(x, B)$

(j)  $\exists! B \in R, \exists x \in X, M(x, B)$

(c)  $\exists B \in R, \forall x \in X, M(x, B)$

(k)  $\exists! B \in R, \exists! x \in X, L(x, B)$

(d)  $\exists B \in R, \forall x \in X, L(x, B)$

(e)  $\exists B \in R, \forall x \in X, \neg L(x, B)$

(f)  $\forall x \in X, \exists B \in R, \neg L(x, B)$

(g)  $\forall B \in R, (L(\text{Manon}, B) \rightarrow M(\text{Guy}, B))$

(h)  $\forall B \in R, \forall x \in X, (M(x, B) \rightarrow L(x, B))$

$R \backslash X$	Alice	Bartez	Guy	Julien	Manon	Ugo
$K$		$L$	$L, M$			
$J$	$L, M$					
$P$	$L$	$L$	$L, M$	$L, M$	$L, M$	$L, M$
$S$			$L, M$		$L, M$	
$Z$	$L$	$L$	$L, M$	$L$	$L$	$L$

**1.29** Soit  $C(x, y)$  l'énoncé «  $x$  connaît  $y$  », où l'univers du discours de  $x$  et de  $y$  est l'ensemble de tous les habitants de la Terre. Utilisez des quantificateurs pour exprimer chacun des énoncés suivants :

- (a) Ève connaît tout le monde.
- (b) Il y a quelqu'un que Julie ne connaît pas.
- (c) Il y a quelqu'un que personne ne connaît.
- (d) Chaque personne se connaît elle-même.
- (e) Il y a quelqu'un qui ne connaît personne d'autre que lui-même.
- (f) Tout le monde connaît Karim.
- (g) Tout le monde connaît quelqu'un.
- (h) Il y a quelqu'un que tout le monde connaît.
- (i) Personne ne connaît tout le monde.

**1.30** Soit  $F(u, p)$  l'énoncé « l'usine  $u$  fabrique le produit  $p$  », où l'univers du discours de  $u$  est l'ensemble  $U$  des 5 usines d'une compagnie et l'univers du discours de  $p$  est l'ensemble  $P$  des 7 produits fabriqués par cette même compagnie. De façon pratique, les 7 produits sont identifiés par les lettres  $a, b, c, d, e, f, g$  et une usine est identifiée par la ville où elle se trouve. Écrivez les propositions suivantes **en mots**. De plus, pour chacune des propositions, dites si elle est **vraie ou fausse** en vous fiant au tableau de production de la compagnie : le 1 dans une case désigne que l'usine fabrique le produit, tandis qu'une case vide signifie le contraire.

- (a)  $\exists u \in U, F(u, c)$
- (b)  $\forall p \in P, (\neg F(\text{Montréal}, p) \vee \neg F(\text{Chicoutimi}, p))$
- (c)  $\forall p \in P, (F(\text{Montréal}, p) \rightarrow \neg F(\text{Gatineau}, p))$
- (d)  $\exists p \in P, (\neg F(\text{Montréal}, p) \wedge \neg F(\text{Québec}, p))$
- (e)  $\exists p \in P, \forall u \in U, F(u, p)$
- (f)  $\forall u \in U, \exists p \in P, F(u, p)$
- (g)  $\forall u \in U, \forall p \in P, F(u, p)$
- (h)  $\forall u \in U, (F(u, b) \rightarrow \neg F(u, d) \wedge \neg F(u, e))$
- (i)  $\forall u \in U, (F(u, a) \rightarrow F(u, f) \vee F(u, g))$
- (j)  $\forall p \in P, \exists u \in U, F(u, p)$
- (k)  $\forall u \in U, (\neg F(u, c) \vee \neg F(u, d))$
- (l)  $\forall u \in U, (F(u, a) \leftrightarrow F(u, b))$
- (m)  $\exists u_1 \in U, \exists u_2 \in U, F(u_1, a) \wedge F(u_2, a) \wedge u_1 \neq u_2$
- (n)  $\exists p \in P, \exists! u \in U, F(u, p)$

$P \backslash U$	Amos	Chicoutimi	Gatineau	Montréal	Québec
$a$		1		1	1
$b$		1		1	
$c$	1	1			
$d$	1		1		
$e$	1				1
$f$	1	1		1	
$g$			1		



**1.31** ★ Considérons le jeu de Sudoku de 9 lignes et 9 colonnes (voir figure 1.1). Désignons  $C(i, j, k)$  l'énoncé «la case située dans la ligne  $i$  et la colonne  $j$  contient le nombre  $k$ . L'univers du discours pour les variables  $i, j$  et  $k$  est l'ensemble des nombres entiers compris entre 1 et 9 inclusivement.

	2		5		1			9
8			2		3			6
	3			6				7
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

Figure 1.1 Exemple de grille du jeu de Sudoku.

Traduisez clairement les règles suivantes à l'aide des quantificateurs.

- Il y a au moins un chiffre par case.
- Il y a au plus un chiffre par case.
- Les chiffres 1 à 9 apparaissent dans chaque colonne.
- Les chiffres 1 à 9 apparaissent dans chaque ligne.
- Chaque chiffre apparaît une seule fois sur une ligne.
- Chaque chiffre apparaît une seule fois sur une colonne.
- ★ ★ Les chiffres 1 à 9 apparaissent dans chaque bloc. (La grille est constituée de 9 blocs 3 par 3, voir figure 1.1.)
- ★ ★ Chaque chiffre apparaît une seule fois dans chaque bloc.

**1.32** Soit  $P(x, y)$  le prédicat désignant « $x < y$ ». L'univers du discours pour les variables  $x$  et  $y$  est l'ensemble des nombres naturels. Pour chacune des propositions suivantes, déterminez sa valeur de vérité (**vrai** ou **faux**). Justifiez en donnant une démonstration, un exemple ou un contre-exemple selon le cas.

- $\exists x P(x, 4)$
- $\exists x P(x, 0)$
- $\forall x \exists y P(x, y)$
- $\neg \forall x \forall y P(x, y)$
- $\neg \exists x \forall y P(x, y)$
- $\exists x \forall y ((x \neq y) \rightarrow P(x, y))$
- $\forall x \forall y (P(x, y) \vee P(y, x))$

**1.33** Trouvez un contre-exemple, si possible, pour chacune des quantifications suivantes, où l'univers du discours des variables est l'ensemble des nombres entiers.

- $\forall x \forall y (x^2 = y^2 \rightarrow x = y)$
- $\forall x \forall y (xy \geq x)$
- $\forall x \exists y (xy = 1)$

## 1.2 Raisonnements

Un **raisonnement** est un énoncé de la forme  $(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \longrightarrow q$ , où  $p_1, p_2, \dots, p_n$  sont les prémisses (ou hypothèses) et  $q$  est la conclusion. On utilise la notation

$$\begin{array}{c} p_1 \\ p_2 \\ \vdots \\ p_n \\ \hline q \end{array}$$

### 1.2.1 Règles d'inférence

La table 5 de la page 35 résume les principales **règles d'inférence**. Ces formes de raisonnement sont valides : elles garantissent la véracité de la conclusion quand toutes les prémisses (ou hypothèses) sont vraies. Plus formellement, un raisonnement est **valide** si  $(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \longrightarrow q$  est une tautologie.

Les hypothèses  $p_i$  ne sont pas toujours des propositions simples. Par exemple, voici la règle d'inférence appelée *sylogisme hypothétique*.

$$\begin{array}{l} p_1 : p \rightarrow s \\ p_2 : s \rightarrow r \\ \hline q : p \rightarrow r \end{array}$$

Les tables suivantes présentent des raisonnements valides auxquels nous ferons référence par la suite.

$\frac{p}{p \rightarrow q}$ $q$	Modus ponens
$\frac{\neg q}{p \rightarrow q}$ $\neg p$	Modus tollens
$\frac{p \rightarrow q}{q \rightarrow r}$ $p \rightarrow r$	Syllogisme hypothétique
$\frac{p \vee q}{\neg p}$ $q$	Syllogisme disjonctif
$\frac{p}{p \vee q}$	Addition
$\frac{p \wedge q}{p}$	Simplification
$\frac{p}{p \wedge q}$ $q$	Conjonction
$\frac{p \vee q}{\neg p \vee r}$ $q \vee r$	Résolution

$\frac{\forall x P(x)}{P(c)}$	Instanciation universelle
$\frac{P(c) \text{ pour } c \text{ quelconque}}{\forall x P(x)}$	Généralisation universelle
$\frac{\exists x P(x)}{P(c) \text{ pour un certain } c}$	Instanciation existentielle
$\frac{P(c) \text{ pour un certain } c}{\exists x P(x)}$	Généralisation existentielle

**Exemple 1.23** (à compléter en classe)

Analysez chacun des raisonnements suivants pour voir s'il est valide ou non. Indiquez sa structure à l'aide de connecteurs logiques. Si le raisonnement est valide, tentez de trouver sa structure dans les tables 5 et 6.

- (a) *S'il pleut, alors les trottoirs sont mouillés.  
Les trottoirs ne sont pas mouillés.  
Donc, il ne pleut pas.*

**Solution :**

Le raisonnement est de la forme suivante, appelée *modus tollens*.

$$\frac{p \rightarrow m}{\neg m}$$

$$\hline \neg p$$

Valide

Le *modus tollens* est valide. En effet, la table de vérité suivante montre que la proposition  $((p \rightarrow m) \wedge \neg m) \rightarrow \neg p$  est une tautologie.

$p$	$m$	$\neg m$	$p \rightarrow m$	$(p \rightarrow m) \wedge \neg m$	$\neg p$	$((p \rightarrow m) \wedge \neg m) \rightarrow \neg p$
V	V	F	V	F	F	V
V	F	V	F	F	F	V
F	V	F	V	F	V	V
F	F	V	V	V	V	V

- (b) *S'il pleut, alors les trottoirs sont mouillés.  
Les trottoirs sont mouillés.  
Donc, il pleut.*

**Solution :**

Ce raisonnement (invalide) est de la forme

$$\frac{p \rightarrow m}{m} \quad \frac{m}{p}$$

Invalide

où  $p$  désigne « il pleut »  
et  $m$  désigne « les trottoirs sont mouillés ».

Ce raisonnement est invalide. En effet, les prémisses  $(p \rightarrow m)$  et  $(m)$  peuvent être toutes les deux vraies sans que la conclusion  $p$  ne soit vraie, comme le montre la table de vérité suivante.

$p$	$m$	$p \rightarrow m$	$(p \rightarrow m) \wedge m$	$((p \rightarrow m) \wedge m) \rightarrow p$
F	V	V	V	F

- (c) *Sarah étudie en LOG ou en TI.  
Sarah n'étudie pas en LOG.  
Donc Sarah étudie en TI.*
- (d) *Si je suis à Montréal, alors je suis au Québec.  
Je ne suis pas à Montréal.  
Donc je ne suis pas au Québec.*
- (e) *Si je suis à Montréal, alors je suis au Québec.  
Je ne suis pas au Québec.  
Donc je ne suis pas à Montréal.*

- (f) *Si je suis à Montréal, alors je suis au Québec.  
Je suis à Montréal.  
Donc je suis au Québec.*
- (g) *Les grandes villes ont des gratte-ciel.  
Montréal est une grande ville.  
Donc Montréal a des gratte-ciel.*
- (h) *Tous les chats sont mortels.  
Socrate est mortel.  
Donc Socrate est un chat.*
- (i) *Les martiens sont verts.  
Xip est un martien.  
Donc Xip est vert.*
- (j) *Alex fume.  
Certains fumeurs développeront un cancer.  
Donc Alex développera un cancer.*
-

### 1.2.2 Cohérence d'un ensemble de spécifications

Dans la conception d'un logiciel, l'ingénieur.e logiciel doit décrire un ensemble de spécifications (tâches qui devront être exécutées par l'ordinateur). Traduire ces spécifications (phrases du langage courant) en expressions logiques permet d'éviter toute ambiguïté dans le développement du logiciel.

Cet ensemble de spécifications doit être **cohérent**, c'est-à-dire qu'il ne doit mener à aucune contradiction. Pour vérifier qu'un ensemble de spécifications est cohérent, on traduit premièrement les spécifications en expressions logiques composées de propositions simples (aussi appelées variables) et on s'assure ensuite qu'elles puissent être toutes vraies simultanément, pour certaines valeurs de vérité des variables. On dit alors qu'il existe une **affectation** des variables qui satisfait l'ensemble de spécifications.

Plusieurs approches permettent de résoudre ce type de problèmes, par exemple :

1. Vérifier si la table de vérité des spécifications contient une unique ligne où les propositions sont vraies en même temps.
2. Utiliser les règles d'inférences ainsi que les équivalences logiques pour déduire les valeurs des variables puis vérifier si elles satisfont bien chacune des spécifications.

Dans cette deuxième approche, le niveau de détails de la justification fournie peut être plus ou moins grand. Soit on explique son raisonnement en quelques mots, soit on le formalise en citant chacune des règles logiques utilisées.

La vérification de la cohérence d'un ensemble de spécifications devient rapidement très longue quand le nombre de variables augmente: c'est un problème qui suscite encore beaucoup de recherche. Dans le cours, nous travaillerons avec de petits ensembles de spécifications composées d'un petit nombre de variables. Plusieurs énigmes et jeux mathématiques peuvent aussi être résolus grâce à une analyse de satisfaisabilité.

**Exemple 1.24**

Un étudiant entre au Pub *Le 100 génies* et fait cinq déclarations.

1. Les jours où je ne fais pas de maths et je dors, je suis content.
2. Les jours où je fais des maths, je suis content et je dors.
3. Les jours où je ne bois pas de café, ou bien je suis content, ou bien je dors, ou les deux.
4. Les jours où je bois du café, je fais des maths ou je ne suis pas content, ou les deux.
5. Je ne suis pas content aujourd'hui.

En supposant que les cinq déclarations sont vraies, avez-vous suffisamment d'informations pour déterminer si cet étudiant a fait des maths aujourd'hui? S'il a bu du café? S'il a dormi? Si oui, répondez aux questions en expliquant votre raisonnement. Sinon, dites pourquoi.

**Solution :**

Identifions les propositions élémentaires suivantes (aussi appelées variables).

$m$ : « l'étudiant a fait des maths »;

$b$ : « l'étudiant a bu du café »;

$c$ : « l'étudiant est content »;

$d$ : « l'étudiant a dormi ».

Nous savons que les cinq propositions suivantes sont vraies.

$$p1: (\neg m \wedge d) \rightarrow c$$

$$p2: m \rightarrow (c \wedge d)$$

$$p3: \neg b \rightarrow (c \vee d)$$

$$p4: b \rightarrow (m \vee \neg c)$$

$$p5: \neg c$$

Voyons si nous pouvons trouver une unique affectation des propositions  $m$ ,  $b$ ,  $d$  et  $c$  qui satisfont les propositions 1 à 5, c'est-à-dire qui font en sorte que  $p1$  à  $p5$  soient vraies.

D'abord, pour que  $p5$  soit vraie, il est nécessaire que la variable  $c$  soit fausse, ce qui entraîne alors que  $(c \wedge d)$  est fausse. Ainsi, l'affirmation  $p2$  nous permet de conclure que  $m$  doit être fausse: l'étudiant n'a pas fait de maths.

Ensuite, pour que  $p1$  soit vraie tout en ayant  $c$  fausse, il faut que  $(\neg m \wedge d)$  soit fausse, et donc que  $d$  soit fausse (car  $\neg m$  est vraie): l'étudiant n'a pas dormi.

Puis, pour que  $p3$  soit vraie sachant que  $(c \vee d)$  est fausse, il est nécessaire que  $\neg b$  soit fausse: l'étudiant a bu du café.

Finalement, nous vérifions que l'affectation trouvée, c'est-à-dire  $b = \mathbf{V}$ ,  $c = \mathbf{F}$ ,  $d = \mathbf{F}$  et  $m = \mathbf{F}$ , satisfait aussi  $p4$ : comme  $b$  est vraie, il faut que  $(\neg c \vee m)$  soit vraie, ce qui est le cas, car  $\neg c$  est vraie.

**Réponse.** Oui, nous pouvons déduire que l'étudiant a bu du café, n'a pas dormi et n'a pas fait de maths.

Voici une solution plus détaillée citant chacune des règles d'inférences et des formules d'équivalence logique utilisée, ainsi que chacune des propositions satisfaites (énoncés de l'étudiant).

proposition	justification	affectation	proposition satisfaite
1. $(\neg m \wedge d) \rightarrow c$			
2. $m \rightarrow (c \wedge d)$			
3. $\neg b \rightarrow (c \vee d)$			
4. $b \rightarrow (\neg c \vee m)$			
5. $\neg c$		$c = \mathbf{F}$	$p5$
6. $\neg c \vee \neg d$	par <i>addition</i> à la proposition de la ligne 5.		
7. $\neg(c \wedge d)$	par <i>De Morgan</i> de 6.		
8. $\neg m$	par <i>modus tollens</i> de 2 et 7.	$m = \mathbf{F}$	$p2$
9. $\neg(\neg m \wedge d)$	par <i>modus tollens</i> de 1 et 5.		
10. $\neg\neg m \vee \neg d$	par <i>De Morgan</i> de 9.		
11. $m \vee \neg d$	par <i>double négation</i> de 10.		
12. $\neg d$	par <i>syllogisme disjonctif</i> de 8 et 11.	$d = \mathbf{F}$	$p1$
13. $\neg c \wedge \neg d$	par <i>conjonction</i> de 5 et 12.		
14. $\neg(c \vee d)$	par <i>De Morgan</i> de 13.		
15. $\neg\neg b$	par <i>modus tollens</i> de 14 et 3.		
16. $b$	par <i>double négation</i> de 15.	$b = \mathbf{V}$	$p3$
17. $b \wedge \neg c \wedge \neg d \wedge \neg m$	par <i>conjonction</i> de 5, 8, 12 et 16.		

Finalement, il faut vérifier que l'affectation trouvée, c'est-à-dire  $b = \mathbf{V}$ ,  $c = \mathbf{F}$ ,  $d = \mathbf{F}$  et  $m = \mathbf{F}$ , satisfait le 4<sup>e</sup> énoncé de l'étudiant (laissé au lecteur).

---



**Exemple 1.25**

Si l'étudiant de l'exemple précédent entre au pub et refait les quatre premières déclarations, mais affirme cette fois « Je n'ai pas bu de café aujourd'hui, mais je suis content », avez-vous suffisamment d'informations pour déterminer si cet étudiant a fait des maths aujourd'hui et s'il a dormi? Si oui, répondez aux questions en expliquant votre raisonnement. Sinon, dites pourquoi.

**Solution :**

Gardons les mêmes propositions simples  $b, c, d$  et  $m$  (aussi appelées variables), ainsi que les quatre premières propositions (énoncées par l'étudiant) et posons :

$$p5: \neg b \wedge c$$

Voyons si nous pouvons trouver une unique affectation des variables qui satisfont les propositions 1 à 5.

	proposition	justification	affectation	proposition satisfaite
1.	$(\neg m \wedge d) \rightarrow c$			
2.	$m \rightarrow (c \wedge d)$			
3.	$\neg b \rightarrow (c \vee d)$			
4.	$b \rightarrow (\neg c \vee m)$			
5.	$\neg b \wedge c$			
6.	$c$	par <i>simplification</i> de 5.	$c = \mathbf{V}$	$p1$
7.	$\neg b$	par <i>simplification</i> de 5.	$b = \mathbf{F}$	$p4, p5$
8.	$c \vee d$	par <i>addition</i> à 6.		$p3$
9.	$\neg m \vee (c \wedge d)$	équivalent à 2 par table 2.1		
10.	$\neg m \vee d$	$c \wedge d = d$ par <i>identité</i> , car $c = \mathbf{V}$		

Il est impossible d'aller plus loin et de déterminer si l'étudiant a fait des maths aujourd'hui et s'il a dormi, car il y a plusieurs possibilités pour les valeurs de vérité de  $m$  et  $d$  qui satisfont les cinq énoncés :

$$m = \mathbf{F} \text{ et } d = \mathbf{V} \quad \text{ou} \quad m = \mathbf{F} \text{ et } d = \mathbf{F} \quad \text{ou} \quad m = \mathbf{V} \text{ et } d = \mathbf{V}.$$

Cependant, nous pouvons conclure que si l'étudiant a fait des maths, alors il a dormi! En effet :

$$\neg m \vee d \equiv m \rightarrow d.$$

**Exercices**

**1.34** Analysez chacun des raisonnements suivants pour voir s'il est valide ou non. Indiquez sa structure à l'aide de connecteurs logiques. Si le raisonnement est valide, tentez de trouver sa structure dans la table de la page 35.

- (a) S'il pleut, alors les trottoirs sont mouillés. Il ne pleut pas. Donc les trottoirs ne sont pas mouillés.
- (b) S'il pleut, alors les trottoirs sont mouillés. Il pleut. Donc les trottoirs sont mouillés.
- (c) Si tu fais tous les exercices du livre de référence, tu réussiras le cours. Tu as réussi le cours. Donc, tu as fait tous les exercices du livre.
- (d) Ou bien la science peut expliquer ce phénomène, ou bien c'est un miracle. La science ne peut expliquer ce phénomène. C'est donc un miracle.
- (e) Ce sandwich ne contient pas de viande ou contient des champignons. Ce sandwich contient de la viande ou de la mayonnaise. Donc ce sandwich contient des champignons ou de la mayonnaise.

**1.35** En utilisant les quantificateurs  $\exists$  et  $\forall$ , traduisez les énoncés suivants.

- (a) Il existe un et un seul élément de l'univers du discours pour lequel le prédicat  $P$  est vrai.  
*Remarque: on utilise parfois la forme  $\exists!x P(x)$  pour désigner cet énoncé, mais on demande ici de l'exprimer sans le symbole!*
- (b) Il existe au moins deux éléments de l'univers du discours pour lesquels  $P$  est vrai.

**1.36** Annie se rend à l'ÉTS et elle se rend compte qu'elle a oublié son téléphone chez elle. Elle se fait les réflexions suivantes:

1. Si mon téléphone est sur ma table de chevet, alors je l'ai vu en me levant
2. Si j'ai été réveillée par la sonnerie de mon téléphone, alors il se trouve sur ma table de chevet.
3. Si j'ai parlé au téléphone dans la cuisine, alors mon téléphone est sur le comptoir de la cuisine.
4. J'ai été réveillée par la sonnerie de mon téléphone ou j'ai parlé au téléphone dans la cuisine.
5. Je n'ai pas vu mon téléphone en me levant.

Où est son téléphone?

**1.37** Montrer que les quatre hypothèses ci-dessous mènent à la conclusion:  $\neg a \wedge b$ .

1.  $a \vee \neg b \rightarrow \neg c \wedge d$
2.  $e \rightarrow c \vee \neg d$
3.  $\neg f$
4.  $e \vee f$

**1.38** Traduisez les informations suivantes concernant un vol. Utilisez  $P(x)$  pour «  $x$  était présent » et  $C(x)$  pour «  $x$  est coupable ».

1. Il y a un et un seul coupable.
2. Le ou les coupables étaient forcément présents.
3. Au moins une des personnes qui étaient présentes n'est pas coupable.
4. Si Alain et Diane étaient présents, alors Claude est coupable.
5. Si Alain ou Claude étaient présents, Benoît est coupable.
6. Diane n'était pas présente.
7. Si Diane n'est pas coupable, alors Alain non plus.
8. Alain était présent si et seulement si Claude ne l'était pas.

**1.39** À l'aide des informations de l'exercice précédent et sachant que seules quatre personnes pouvaient être présentes (Alain, Benoît, Claude et Diane), est-il possible de déterminer qui est le coupable? Si oui, déterminez qui est coupable et justifiez votre raisonnement.

**1.40** Montrez que les hypothèses :

1. Un étudiant inscrit au cours de Mathématiques discrètes n'a pas fait les exercices suggérés.
2. Tous les étudiants inscrits au cours de Mathématiques discrètes ont réussi l'intra.

mènent à la conclusion: « Il y a un étudiant qui a réussi l'intra, mais n'a pas fait les exercices suggérés. »

**1.41** Analysez chacun des raisonnements suivants pour voir s'il est valide ou non.

- |   |   |
|---|---|
| <p>(a) Tous les hommes sont mortels.<br/>Socrate est un homme.<br/>Donc Socrate est mortel.</p> | <p>(c) Quelques hommes sont bleus.<br/>Socrate est un homme.<br/>Donc Socrate est bleu.</p>     |
| <p>(b) Tous les hommes sont bleus.<br/>Socrate est un homme.<br/>Donc Socrate est bleu.</p>     | <p>(d) Certains hommes sont mortels.<br/>Socrate est un homme.<br/>Donc Socrate est mortel.</p> |

**1.42** Parmi les trois raisonnements suivants, deux sont valides et un est invalide.

A	B	C
1. $a \rightarrow (b \wedge c)$ 2. $\neg c$ <hr style="width: 50%; margin-left: 0;"/> $\neg a$	1. $a \rightarrow b$ 2. $\neg a$ <hr style="width: 50%; margin-left: 0;"/> $\neg b$	1. $(a \vee b) \wedge c$ 2. $d \rightarrow \neg c$ 3. $d \vee \neg a$ <hr style="width: 50%; margin-left: 0;"/> $b$

- (a) Déterminez quel raisonnement est invalide et expliquez pourquoi.
- (b) Démontrez que les deux autres raisonnements sont valides en utilisant les règles d'inférence et les tables d'équivalences.

**1.43** L'ensemble de spécifications suivant est-il cohérent?

1. La communication est fluide si le logiciel Communik est installé.
2. Pour que le logiciel Communik soit installé, il est nécessaire que la version 10 ou plus du système d'exploitation soit installée.
3. C'est la version 9 du système d'exploitation qui est installée.
4. La communication est fluide .

**1.44** L'ensemble de spécifications de l'exercice précédent est-il cohérent si on modifie la première spécification ainsi?

1. La communication est fluide seulement si le logiciel Communik est installé.

**1.45** L'ensemble de spécifications suivant est-il cohérent?

1. La communication est fluide seulement si le logiciel Communik est installé.
2. Pour que le logiciel Communik soit installé, il est nécessaire que la version 10 ou plus du système d'exploitation soit installée.
3. la version 10 ou plus du système d'exploitation assure une communication fluide.
4. C'est la version 9 du système d'exploitation qui est installée.

**1.46** Si  $a$ ,  $b$ ,  $c$  et  $d$  désignent des propositions, l'ensemble de spécifications suivant est-il cohérent?

1.  $a \rightarrow (b \wedge \neg c)$
2.  $b \rightarrow \neg d$
3.  $c \rightarrow b$
4.  $d \leftrightarrow \neg a$

**1.47** Trouvez toutes les assignations des variables  $a$ ,  $b$ ,  $c$  et  $d$  qui satisfont l'ensemble de spécifications de l'exercice précédent.

**1.48** Si  $a$ ,  $b$ ,  $c$  et  $d$  désignent des propositions, l'ensemble de spécifications suivant est-il cohérent?

1.  $b \rightarrow c$
2.  $\neg a \rightarrow d$
3.  $\neg c \vee d$
4.  $a \leftrightarrow b$
5.  $\neg d$

**1.49** Déterminer si le système de spécifications ci-dessous est cohérent. Si oui, donner toutes les valeurs de vérité des propositions  $p$ ,  $q$ ,  $r$ ,  $s$  et  $t$  qui satisfont chacune des spécifications.

1.  $r \wedge (\neg p \rightarrow q)$
2.  $\neg q \vee s$
3.  $t \rightarrow q \vee \neg s$
4.  $\neg(\neg q \wedge s) \rightarrow t$
5.  $s \rightarrow \neg r$

**1.50** Quatre amis sont identifiés comme les suspects d'un crime. Voici leurs déclarations:

1. Anouk dit: «Xavier est coupable.»
2. Geneviève dit: «Je ne suis pas coupable.»
3. Xavier dit: «Stéphane est coupable.»
4. Stéphane dit: «Xavier a menti quand il a dit que j'étais coupable.»

Sachant qu'il y a exactement un suspect qui dit la vérité et qu'il n'y a qu'un seul coupable, déterminez qui est coupable.

**1.51** Trois amis ont été identifiés comme les suspects d'un crime et sont interrogés. Le coupable ment et les autres disent la vérité. Voici leurs déclarations:

1. Anouk a dit: «Xavier est coupable.»
2. Geneviève a dit: «Xavier ment.»
3. Xavier a dit: «Geneviève est coupable.»

Déterminez qui est coupable. *Conseil: complétez un tableau des états possibles (sincère ou menteur) et analysez la cohérence des énoncés en fonction des états.*

**1.52** Anouk ment systématiquement du dimanche au mardi et dit la vérité les autres jours de la semaine, tandis que son amie Geneviève ment toujours du mercredi au vendredi et est sincère du samedi au mardi. Xavier rencontre ses amies. Anouk lui dit «hier, j'ai menti» et Geneviève ajoute «moi aussi». Quel jour de la semaine a lieu cette discussion? *Conseil: complétez un tableau des états possibles selon les journées (sincère ou menteur) et analysez la cohérence des énoncés en fonction des états.*

**1.53** Sur une île vivent deux catégories d'habitants: les sincères qui disent toujours la vérité, et les menteurs qui mentent toujours. Vous rencontrez deux habitants,  $A$  et  $B$ . Déterminez, si possible, de quelles catégories d'habitants sont  $A$  et  $B$  si ces deux personnes s'adressent à vous de la manière suivante.

- (a)  $A$  dit: «Au moins un de nous deux est un menteur.» et  $B$  ne dit rien.
- (b)  $A$  dit: «Nous sommes les deux sincères.» et  $B$  dit: « $A$  est un menteur.»
- (c)  $A$  dit: «Je suis un menteur ou  $B$  est sincère.» et  $B$  ne dit rien.
- (d)  $A$  et  $B$  disent chacun: «Je suis sincère.»
- (e)  $A$  dit: «Nous sommes les deux des menteurs.» et  $B$  ne dit rien.

**1.54** ★ Deux gardiens, l'un toujours sincère et l'autre toujours menteur, sont devant deux portes. L'une mène au cours de Mathématiques discrètes, l'autre en enfer. Vous avez le droit de poser une seule question à un seul gardien, sans savoir lequel est sincère. Formulez une question qui vous permettra de déterminer quelle porte mène au cours de Mathématiques discrètes.

### 1.2.3 Types de preuve

#### Définition 1.7 : Théorème, démonstration, axiome

Un **théorème** est un énoncé que l'on peut démontrer.

Une **démonstration** (on dit aussi une **preuve**) consiste à déduire que l'énoncé est vrai en utilisant un raisonnement logique (règles d'inférence) reposant sur des **axiomes** ou **postulats** (énoncés considérés vrais sans démonstration) ou sur des théorèmes déjà démontrés.

#### Définition 1.8 : Conjecture

Une **conjecture** est une proposition que l'intuition ou l'observation nous porte à croire vraie, mais qui n'a pas encore été formellement démontrée (ni réfutée). Si elle est démontrée, la conjecture devient un théorème.

Le travail des mathématiciens consiste à formuler des conjectures puis à démontrer qu'elles sont vraies (ou fausses). L'équivalent en informatique consiste à inventer un algorithme, ou un programme, puis à démontrer qu'il produit toujours le résultat attendu.

Au cours de la session, vous serez amenés à lire différentes preuves et à en produire vous-mêmes. Démontrer un résultat n'est pas facile; il n'y a pas de recette magique. Par contre, connaître les principaux types de preuves est un atout précieux.

### Types de preuves

- **Preuve directe** de  $p \rightarrow q$

On suppose que  $p$  est vrai et l'on démontre (en se basant sur des définitions, des axiomes, des théorèmes déjà démontrés et des règles d'inférence), qu'il s'en suit forcément que  $q$  est vrai aussi.

- **Preuve directe** de  $\forall x \in U P(x) \rightarrow Q(x)$

On suppose que  $x$  est un élément arbitraire de  $U$  tel que  $P(x)$  est vrai. On démontre qu'il s'en suit forcément que  $Q(x)$  est vrai aussi.

- **Preuve indirecte** de  $p \rightarrow q$ , aussi appelée **preuve par contraposée**

On suppose que  $q$  est faux et l'on démontre qu'il s'en suit forcément que  $p$  est faux.

$$\frac{\neg q \rightarrow \neg p}{p \rightarrow q}$$

- **Preuve par contradiction** de  $p$ , aussi appelée **preuve par l'absurde**

On suppose que  $p$  est faux et l'on montre que cela entraîne une contradiction. Ainsi,  $p$  doit être vrai.

$$\frac{\neg p \rightarrow F}{p}$$

- **Preuve d'équivalence** de  $p \leftrightarrow q$

On montre que  $p \rightarrow q$  et  $q \rightarrow p$  sont vrais. Ainsi,  $p \leftrightarrow q$  est vrai.

- **Preuve par cas** de  $p \rightarrow q$

On démontre que  $p$  entraîne un nombre fini de possibilités. On étudie chacune séparément pour vérifier qu'elle entraîne  $q$ . On conclut donc que  $p$  entraîne  $q$ .

$$\begin{array}{l} p \rightarrow (p_1 \vee p_2 \vee \dots \vee p_n) \\ p_1 \rightarrow q \\ p_2 \rightarrow q \\ \vdots \\ p_n \rightarrow q \\ \hline p \rightarrow q \end{array}$$

- **Preuve triviale** de  $p \rightarrow q$

On montre que  $q$  est vrai. Ainsi, l'implication  $p \rightarrow q$  est vraie (peu importe la valeur de  $p$ ).

$$\frac{q}{p \rightarrow q}$$

- **Preuve exhaustive** de  $\forall x \in U P(x)$

Si l'ensemble  $U$  possède un nombre fini d'éléments, disons  $U = \{x_1, x_2, \dots, x_n\}$ , et que l'on démontre que pour chacun d'eux la propriété  $P$  est vraie, alors on peut conclure que la propriété  $P$  est vraie pour tous les éléments de  $U$ .

$$\frac{\begin{array}{l} U = \{x_1, x_2, \dots, x_n\} \\ P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n) \end{array}}{\forall x \in U P(x)}$$

- **Preuve constructive** de  $\exists x \in U P(x)$

Il suffit de trouver un élément  $a$  de l'ensemble  $U$  tel que  $P(a)$  est vrai.

$$\frac{\begin{array}{l} a \in U \\ P(a) \end{array}}{\exists x \in U P(x)}$$

- **Preuve par récurrence** de  $\forall n \in \mathbb{N} P(n)$

Pour démontrer que la propriété  $P$  est vraie pour tous les nombres naturels, on démontre qu'elle est vraie pour le nombre 0 (cas de base), et que si elle est vraie pour un nombre  $n$ , alors elle est aussi vraie pour le nombre suivant (étape de récurrence).

$$\frac{\begin{array}{l} P(0) \\ \forall n \in \mathbb{N} (P(n) \rightarrow P(n+1)) \end{array}}{\forall n \in \mathbb{N} P(n)}$$

De nombreux exemples de preuves de chaque type seront donnés en classe, en commençant par des preuves assez courtes et « simples » semblables à celle de l'exercice de la page 58. Les preuves par récurrence seront abordées dans la deuxième moitié de la session.

## 1.3 Théorie des ensembles

### 1.3.1 Notions de base sur les ensembles

#### Définition 1.9 : Ensemble, élément

Un **ensemble** est une collection non ordonnée d'objets. Les objets sont appelés **éléments** de l'ensemble et on dit qu'ils appartiennent à l'ensemble.

Notation:  $x \in F$  signifie que  $x$  est un élément de l'ensemble  $F$ . On dit aussi que  $x$  appartient à l'ensemble  $F$ .

#### Définition 1.10 : Ensemble fini ou infini, cardinalité

Soit  $A$  un ensemble composé de  $n$  éléments distincts. On dit que  $A$  est un **ensemble fini** de **cardinalité**  $n$  et on note  $|A| = n$ . Un ensemble est dit **infini** s'il n'est pas fini.

#### Exemple 1.26

Soit

$$F = \{2, \pi, 7\}.$$

Utilisez les symboles introduits pour traduire les énoncés suivants: l'ensemble  $F$  contient 3 éléments,  $\pi$  appartient à  $F$ , 5 n'appartient pas à  $F$ .

**Solution :**

$$|F| = 3, \quad \pi \in F, \quad 5 \notin F.$$

On peut décrire un ensemble **en extension** (on énumère ses éléments que l'on place entre accolades)

$$A = \{5, 7, 9, 11\} \quad B = \{1, 8, 27, 64\}$$

ou **en compréhension**, comme ceci:

$$A = \{x \in \mathbb{N} \mid (x \text{ est impair}) \wedge (5 \leq x \leq 11)\} \quad B = \{x \in \mathbb{N} \mid (x \leq 64) \wedge (\exists y \in \mathbb{N}, y^3 = x)\}$$



1.3.2 Ensembles de nombres  $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ 

$\emptyset = \{\}$	Ensemble vide (ne contient aucun élément $ \emptyset  = 0$ )
$\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$	Ensemble des nombres entiers
$\mathbb{Z}^* = \{\dots, -2, -1, 1, 2, \dots\}$	Ensemble des entiers non nuls
$\mathbb{N} = \{0, 1, 2, 3, \dots\}$	Ensemble des nombres naturels
$\mathbb{N}^* = \{1, 2, 3, \dots\}$	Ensemble des nombres naturels strictement positifs
$\mathbb{Q} = \left\{ \frac{p}{q} \mid p \in \mathbb{Z}, q \in \mathbb{Z} \text{ et } q \neq 0 \right\}$	Ensemble des nombres rationnels
$\mathbb{R}$	Ensemble des nombres réels
$\mathbb{R}^+ = \{x \in \mathbb{R} \mid x \geq 0\}$	Ensemble des nombres réels positifs
$\mathbb{C} = \{a + bi \mid a \in \mathbb{R} \text{ et } b \in \mathbb{R}\}$ avec $i = \sqrt{-1}$ , donc $i^2 = -1$	Ensemble des nombres complexes

**TABLE 6** Propriétés des ensembles de nombres  $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$ 

Si $A$ est un ensemble de nombre parmi $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ et $a, b, c \in A$ , alors	
$a + b \in A$	Clos pour l'addition
$a \cdot b \in A$	Clos pour la multiplication
$a + b = b + a$	Commutativité de l'addition
$a \cdot b = b \cdot a$	Commutativité de la multiplication
$a(b + c) = ab + ac$	Distributivité de la multiplication sur l'addition
$a \cdot b = 0 \leftrightarrow (a = 0 \vee b = 0)$	Équation produit-nul
Si $a, b \in \mathbb{N}$ , alors	
$a + b = 0 \rightarrow (a = 0 \wedge b = 0)$	Absence d'un inverse additif
Si $a, b \in \mathbb{Z}$ , alors	
$a \cdot b = 1 \rightarrow ((a = 1 \wedge b = 1) \vee (a = -1 \wedge b = -1))$	Absence d'un inverse multiplicatif
Si $A$ est un ensemble de nombre parmi $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ et $a \in A$ , alors	
$a - b = a + (-1) \cdot b$	Notation
$a - a = 0$	Inverse additif
Si $a, b, c, d \in \mathbb{Z}$ , $b \neq 0$ et $d \neq 0$ , alors $\frac{a}{b}, \frac{c}{d} \in \mathbb{Q}$ et	
$\frac{a}{b} = \frac{c}{d} \leftrightarrow ad = bc$	Égalité de fractions
$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$	Addition de fractions
$\frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}$	Multiplication de fractions

**Définition 1.11 : Nombres pairs et impairs**

- Un nombre  $x$  est **pair** s'il existe  $n \in \mathbb{Z}$  tel que  $x = 2n$ .
- Un nombre  $x$  est **impair** s'il existe  $n \in \mathbb{Z}$  tel que  $x = 2n + 1$ .

**Exemple 1.27**

Montrez que le carré d'un nombre impair est impair.

**Solution :**

On veut montrer: «  $\forall x \in \mathbb{Z}, (x \text{ est impair}) \rightarrow (x^2 \text{ est impair})$  »,

1.  $x$  est impair
2.  $x = 2n + 1, n \in \mathbb{Z}$  (définition d'*impair*)
3.  $x^2 = (2n + 1)^2$  (par 2.)
4.  $x^2 = 4n^2 + 4n + 1$  (distributivité)
5.  $x^2 = 2(2n^2 + 2n) + 1$  (distributivité)
6.  $x^2 = 2m + 1$ , où  $m = 2n^2 + 2n \in \mathbb{Z}$  ( $2, n \in \mathbb{Z}$  et  $\mathbb{Z}$  est clos pour la multiplication)
7.  $x^2$  est impair (définition d'*impair*)

**Théorème 1.2 : Représentation réduite des fractions**

Pour tout élément  $x \in \mathbb{Q}$ , il existe une unique paire  $p \in \mathbb{Z}, q \in \mathbb{N}^*$ , telle que  $x = \frac{p}{q}$  et l'unique diviseur commun à  $p$  et  $q$  est 1.

**Exemple 1.28**

Montrez que le nombre  $\sqrt{2}$  n'est pas rationnel.

(Aide,  $\sqrt{2}$  est défini comme étant un nombre qui lorsque multiplié par lui-même donne 2.)

**Solution :**

Démontrer que le nombre  $\sqrt{2}$  n'est pas rationnel n'est pas si simple que cela! Nous allons procéder par contradiction, c'est-à-dire que nous allons démontrer l'implication suivante :

$$(\sqrt{2} \in \mathbb{Q}) \rightarrow \mathbf{F}.$$

Ainsi, nous pourrions conclure que

$$(\sqrt{2} \in \mathbb{Q}) \equiv \mathbf{F},$$

ou encore que

$$\sqrt{2} \notin \mathbb{Q}.$$

Par contradiction, on suppose que  $\sqrt{2} \in \mathbb{Q}$ . Par le théorème 1.2, on a qu'il existe une unique fraction  $\frac{p}{q} = \sqrt{2}$  telle que  $p \in \mathbb{Z}$ ,  $q \in \mathbb{N}^*$  et l'unique diviseur commun à  $p$  et  $q$  est 1. De manière plus formelle, cette dernière propriété se formule de la façon suivante:  $\forall a \in \mathbb{N}, \forall p' \in \mathbb{Z}, \forall q' \in \mathbb{Z}, (p = ap' \wedge q = aq') \rightarrow a = 1$ .

1.  $p \in \mathbb{Z}$
2.  $q \in \mathbb{N}^*$
3.  $\left(\frac{p}{q}\right)^2 = 2$
4.  $\forall a \in \mathbb{N}, \forall p' \in \mathbb{Z}, \forall q' \in \mathbb{Z}, (p = ap' \wedge q = aq') \rightarrow a = 1$ , c'est-à-dire que  $\text{pgcd}(p, q) = 1$
5.  $\frac{p^2}{q^2} = 2$  (par 3. et multiplication de fractions)
6.  $p^2 = 2q^2 = 2k$ , où  $k = q^2 \in \mathbb{Z}$  (par 5. et  $\mathbb{N}^* \subset \mathbb{Z}$  est clos pour la multiplication.)
7.  $p^2$  est pair (par 6. et définition de *pair*)
8.  $p^2$  pair  $\rightarrow p$  pair (contraposée de l'exemple 1.27)
9.  $p$  est pair (modus ponens 7. et 8.)
10.  $p = 2l$ , où  $l \in \mathbb{Z}$  (9. et définition de *pair*)
11.  $p^2 = (2l)^2 = 2 \cdot 2l^2$  (par 10.)
12.  $2q^2 = p^2 = 2 \cdot 2l^2$  (par 6. et 11.)
13.  $q^2 = 2l^2 = 2m$ , où  $m \in \mathbb{Z}$  (par 12. et  $\mathbb{Z}$  clos par multiplication)
14.  $q^2$  est pair (par 13. et définition de *pair*)
15.  $q^2$  pair  $\rightarrow q$  pair (contraposée de l'exercice 1.27)
16.  $q$  est pair (modus ponens 14. et 15.)
17.  $q = 2n$ ,  $n \in \mathbb{Z}$  (par 16. et définition de *pair*)
18.  $p = 2l \wedge q = 2n$  (conjonction de 10. et 17.)
19.  $(p = 2l \wedge q = 2n) \rightarrow 2 = 1$  (instanciation exist. de 4. avec  $a = 2$ ,  $p' = l$ ,  $q' = n$ )
20.  $2 = 1$  (modus ponens 18. et 19.)
21. **Faux** (logiquement équivalent à 20.)

### Définition 1.12 : Égalité d'ensembles

Deux ensembles sont dits égaux si et seulement s'ils contiennent exactement les mêmes éléments.

$$A = B \longleftrightarrow \forall x(x \in A \leftrightarrow x \in B)$$

**Exemple 1.29**

$$\{1, 3, 5\} = \{3, 5, 1\}$$

$\{1, 3, 5\} \neq \{\{1\}, \{3\}, \{5\}\}$ , puisque l'élément 1 n'est pas égal à l'élément  $\{1\}$ .

**Définition 1.13 : Sous-ensemble**

L'ensemble  $A$  est **sous-ensemble** de l'ensemble  $B$  si et seulement si tous les éléments de  $A$  sont aussi des éléments de  $B$ :

$$A \subseteq B \longleftrightarrow \forall x(x \in A \rightarrow x \in B).$$

L'ensemble  $A$  est **sous-ensemble strict (ou propre)** de l'ensemble  $B$  si et seulement si tous les éléments de  $A$  sont aussi des éléments de  $B$  et  $A$  n'est pas égal à  $B$ :

$$A \subset B \longleftrightarrow A \subseteq B \wedge A \neq B.$$

**Exemple 1.30**

$$\{1, 2\} \subseteq \{1, 2, 3, 4, 5\}$$

$$\{1, 2\} \subset \{1, 2, 3, 4, 5\}$$

$$\{2k \mid k \in \mathbb{N}\} = \{0, 2, 4, 6, \dots\} \subset \mathbb{N}$$

**Théorème 1.3**

Pour tout ensemble  $A$ , on a

1.  $\emptyset \subseteq A$
2.  $A \subseteq A$

**Théorème 1.4**

$A = B$  si et seulement si  $A \subseteq B$  et  $B \subseteq A$ .

Rappelons qu'un théorème est un énoncé que l'on a *démontré*. Or les démonstrations des théorèmes n'apparaissent toujours pas dans ce texte. Certaines sont présentées en classe, d'autres sont laissées en exercice au lecteur.

### 1.3.3 Produit cartésien

#### Définition 1.14 : Produit cartésien

Le **produit cartésien** des ensembles  $A$  et  $B$ , noté  $A \times B$ , est l'ensemble de tous les couples (paires ordonnées) dont le premier élément appartient à  $A$  et le second, à  $B$ :

$$A \times B = \{(a, b) \mid a \in A \text{ et } b \in B\}.$$

On généralise cette définition au produit cartésien de  $n$  ensembles:

$$A_1 \times A_2 \times \dots \times A_n = \{(a_1, a_2, \dots, a_n) \mid a_1 \in A_1, \dots, a_n \in A_n\}$$

#### Exemple 1.31

Décrivez en extension les produits cartésiens  $A \times B$  et  $B \times A$ , où  $A = \{0, 1, 2\}$  et  $B = \{a, c\}$ .

#### Solution :

$$A \times B = \{(0, a), (0, c), (1, a), (1, c), (2, a), (2, c)\}$$

$$B \times A = \{(a, 0), (c, 0), (a, 1), (c, 1), (a, 2), (c, 2)\}$$

#### Définition 1.15 : Relation

Une **relation** entre les ensembles  $A$  et  $B$  est un sous-ensemble du produit cartésien  $A \times B$ .

#### Exemple 1.32

Soit  $A = \{0, 1, 2\}$  et  $B = \{a, c\}$ . L'ensemble

$$R = \{(0, a), (1, c), (2, a)\} \subseteq A \times B$$

est une relation de  $A$  dans  $B$ .

#### Définition 1.16 : Ensemble des parties

L'**ensemble des parties de  $A$** , noté  $\wp(A)$ , est l'ensemble de tous les sous-ensembles de  $A$ .

$$B \in \wp(A) \iff B \subseteq A$$

**Exemple 1.33**

Décrivez  $\wp(A)$ , l'ensemble des parties de  $A$ , où  $A = \{0, 1, 2\}$ .

**Solution :**

$k$	Sous-ensembles de $A$ ayant $k$ éléments	nombre de sous-ensembles
0	$\emptyset$	1
1	$\{0\}, \{1\}, \{2\}$	3
2	$\{0, 1\}, \{0, 2\}, \{1, 2\}$	3
3	$\{0, 1, 2\}$	1

Ainsi,  $\wp(A)$  contient 8 éléments, soit  $2^{|A|}$ .

$$\wp(A) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}$$

**Exemple 1.34**

Décrivez l'ensemble des parties de  $A$ , où  $A = \{0, 1, 2, 3\}$ .

**Solution :**

$k$	Sous-ensembles de $A$ ayant $k$ éléments	nombre de sous-ensembles
0	$\emptyset$	1
1	$\{0\}, \{1\}, \{2\}, \{3\}$	4
2	$\{0, 1\}, \{0, 2\}, \{0, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}$	6
3	$\{0, 1, 2\}, \{0, 1, 3\}, \{0, 2, 3\}, \{1, 2, 3\}$	4
4	$\{0, 1, 2, 3\}$	1

Ainsi,  $\wp(A)$  contient 16 éléments, soit  $2^{|A|}$ .

$$\wp(A) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{3\}, \{0, 1\}, \{0, 2\}, \{0, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{0, 1, 2\}, \{0, 1, 3\}, \{0, 2, 3\}, \{1, 2, 3\}, \{0, 1, 2, 3\}\}$$

**Exemple 1.35**

Décrivez l'ensemble des parties de  $A$ , où  $A = \{\} = \emptyset$ .

**Solution :**

$k$	Sous-ensembles de $A$ ayant $k$ éléments	nombre de sous-ensembles
0	$\emptyset$	1

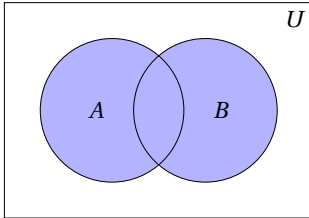
Ainsi, bien que  $A$  ne possède aucun élément,  $\wp(A)$  en contient un:  $\wp(A) = \{\emptyset\}$

### 1.3.4 Opérations sur les ensembles $\cap, \cup, \oplus, -$

Soit  $U$  l'ensemble universel et  $A$  et  $B$  des sous-ensembles de  $U$ . Les opérations suivantes génèrent des sous-ensembles de  $U$ .

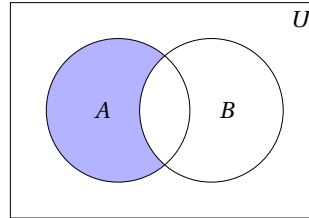
**Union:**

$$A \cup B = \{x \in U \mid x \in A \vee x \in B\}$$



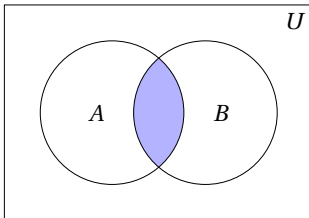
**Différence:**

$$A - B = \{x \in U \mid x \in A \wedge x \notin B\} = A \setminus B$$



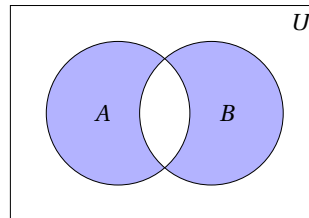
**Intersection:**

$$A \cap B = \{x \in U \mid x \in A \wedge x \in B\}$$



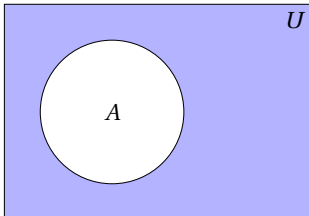
**Différence symétrique:**

$$A \oplus B = \{x \in U \mid x \in A \oplus x \in B\}$$



**Complément:**

$$\bar{A} = \{x \in U \mid x \notin A\} = U - A$$



La table suivante présente les propriétés des opérations sur les ensembles.

TABLE 8 Propriétés des ensembles	
$A \cap U = A$ $A \cup \emptyset = A$	Identité
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination
$A \cup A = A$ $A \cap A = A$	Idempotence
$\overline{\overline{A}} = A$	Double négation
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutativité
$A \cup (B \cap C) = (A \cup B) \cap C$ $A \cap (B \cup C) = (A \cap B) \cup C$	Associativité
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributivité
$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	Lois de De Morgan
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	Négation

### Exemple 1.36

Utilisez la table des propriétés des ensembles pour obtenir une expression simplifiée de l'ensemble

$$\overline{\overline{A \cap B \cap \overline{C} \cup \overline{A} \cap B}}$$

### Solution :

$$\begin{aligned} \overline{\overline{A \cap B \cap \overline{C} \cup \overline{A} \cap B}} &= \overline{\overline{A \cap B \cap \overline{C}} \cap \overline{\overline{A} \cap B}} \\ &= (A \cap B \cap \overline{C}) \cap (\overline{\overline{A} \cap B}) \\ &= (A \cap \overline{A}) \cap B \cap \overline{C} \cap B \\ &= \emptyset \cap B \cap \overline{C} \cap B \\ &= \emptyset \end{aligned}$$

De Morgan

Complément du complément:  $\overline{\overline{A}} = A$

Commutativité et associativité de l'intersection

Complément:  $A \cap \overline{A} = \emptyset$

Domination: intersection avec  $\emptyset$  donne  $\emptyset$ .



### 1.3.5 Représentation de sous-ensembles par trains de bits

On peut représenter un sous-ensemble d'un ensemble à l'aide d'un train de bits. Pour ce faire, il faut fixer un ordre pour les éléments de  $A$ , puis construire le train de bits en posant 1 à la position  $j$  si le  $j$ -ième élément appartient au sous-ensemble et 0 sinon.

Par exemple, si  $A = \{0, 1, 2, 4\}$ , en choisissant l'ordre croissant, on obtient les représentations suivantes.

train de bits	sous-ensemble de $A$
0000	$\emptyset$
1010	$\{0, 2\} = B$
0011	$\{2, 4\} = C$
0010	$\{2\} = D$
1011	$\{0, 2, 4\} = E$
1100	$\{0, 1\} = F$

Remarquons que les opérations d'union, d'intersection et de complément sur les sous-ensembles peuvent alors être effectuées directement sur les trains de bits correspondants.

$$\begin{array}{rclcl}
 1010 & \wedge & 0011 & = & 0010 \\
 B & \cap & C & = & D
 \end{array}$$

$$\begin{array}{rclcl}
 1010 & \vee & 0011 & = & 1011 \\
 B & \cup & C & = & E
 \end{array}$$

$$\begin{array}{rcl}
 \sim 0011 & = & 1100 \\
 \overline{C} & = & F
 \end{array}$$

**Exercices**

L'exercice suivant n'est pas facile. Il vous permettra de vous exercer à traduire des énoncés à l'aide des quantificateurs existentiels et universels (voir page 20) et à **rédigier différents types de preuves** (voir page 46), le tout avec différents ensembles de nombres.

**1.55** Réécrivez chacun des énoncés à l'aide de quantificateurs, de connecteurs et des prédicats suivants:  $Q(x)$ :  $x$  est rationnel,  $P(x)$ :  $x$  est pair. Sauf pour le (a), l'univers du discours est l'ensemble  $\mathbb{R}$  des nombres réels.

De plus, pour chaque énoncé, dites s'il est **vrai ou faux**. Justifiez en donnant une preuve, un exemple ou un contre-exemple selon le cas.

Rappelons que  $\sqrt{2}$  est irrationnel, tel que prouvé à la page 50.

- (a) Pour tout nombre entier  $n$ , si  $n^2$  est pair, alors  $n$  est pair.
- (b) La somme d'un nombre rationnel et d'un nombre irrationnel est irrationnelle.
- (c) Quand on multiplie un nombre rationnel non nul avec un nombre irrationnel, on obtient un nombre irrationnel.
- (d) Le nombre  $\frac{\sqrt{2}}{3}$  est irrationnel.
- (e) Le nombre  $\frac{1}{\sqrt{2}}$  est irrationnel.
- (f) Si un nombre est irrationnel, alors son inverse l'est aussi. Rappel: l'inverse de  $x$  est  $1/x$ .
- (g) La somme de deux nombres irrationnels est irrationnelle.
- (h) Le produit de deux nombres irrationnels est irrationnel.
- (i) L'inverse d'un nombre rationnel non nul est rationnel.
- (j) L'inverse d'un nombre irrationnel est irrationnel.

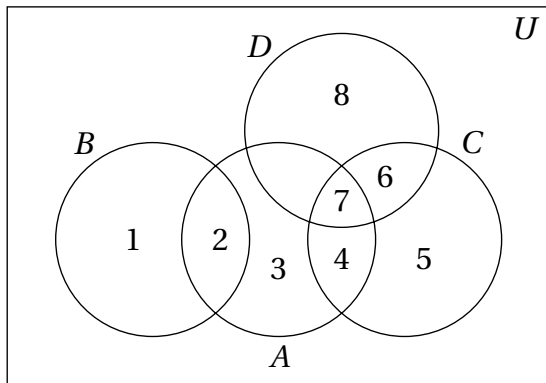
**1.56** Vrai ou faux.

- |                                       |                                       |   |  |
|---------------------------------------|---------------------------------------|---|--|
| (a) $\mathbb{N} \in \mathbb{Z}$       | (d) $\emptyset \in \mathbb{N}$        | (g) $\emptyset \subset \mathbb{N}$        | (j) $\{\{1\}\} \subseteq \wp(\{1, 2, 3\})$ |
| (b) $\mathbb{N} \subseteq \mathbb{Z}$ | (e) $\emptyset \subseteq \mathbb{N}$  | (h) $\{1, 2\} \subseteq \wp(\{1, 2, 3\})$ |  |
| (c) $\mathbb{N} \subset \mathbb{Z}$   | (f) $\mathbb{N} \subseteq \mathbb{N}$ | (i) $\{1, 2\} \in \wp(\{1, 2, 3\})$       |  |

**1.57** Vrai ou faux.

- |                         |                           |                                 |                           |
|-------------------------|---------------------------|---------------------------------|---------------------------|
| (a) $x \in \{x\}$       | (c) $\{x\} \subset \{x\}$ | (e) $\{x\} \subseteq \{\{x\}\}$ | (g) $\emptyset \in \{x\}$ |
| (b) $x \subseteq \{x\}$ | (d) $\{x\} \in \{x\}$     | (f) $\emptyset \subseteq \{x\}$ |                           |

**1.58** Le diagramme de Venn suivant représente fidèlement les ensembles  $A, B, C, D$  et  $U$ . Décrivez en extension les ensembles suivants.



- (a)  $A \cup B$
- (b)  $A \cap B$
- (c)  $A \cap \bar{C}$
- (d)  $A \cap C \cap D$
- (e)  $\varnothing(B)$
- (f)  $\bar{U}$
- (g)  $\overline{A \cup C}$
- (h)  $B \cap D$
- (i)  $C - A$
- (j)  $D - B$
- (k)  $A \oplus C$
- (l)  $D \times B$

**1.59** Soit  $X$  l'ensemble des 6 employés d'une petite équipe et  $R$  l'ensemble des 5 répertoires de leur système informatique. Désignons par  $M(x, B)$  l'énoncé « l'employé  $x$  peut **modifier** le répertoire  $B$  » et par  $L(x, B)$  l'énoncé « l'employé  $x$  peut **lire** le répertoire  $B$  ». Le tableau suivant indique les caractéristiques des employés : un L dans une case désigne que l'employé peut lire le répertoire de cette ligne, un M signifie qu'il peut le modifier, tandis que l'absence d'une lettre L ou M signifie que l'employé ne peut lire ou modifier le répertoire.

Décrivez chacun des ensembles suivants en extension. De plus, pour chacun des ensembles, donnez sa cardinalité.

- (a)  $A = \{r \in R \mid M(\text{Alice}, r)\}$
- (b)  $B = \{x \in X \mid M(x, P)\}$
- (c)  $C = \{x \in X \mid L(x, K)\}$
- (d)  $D = \{r \in R \mid \forall x \in X, L(x, r)\}$
- (e)  $E = \{r \in R \mid \exists x \in X, \neg L(x, r)\}$
- (f)  $D \cup E$
- (g)  $D \cap E$
- (h)  $B \cap C$
- (i)  $X - B$
- (j)  $\bar{E}$

	$X$	Alice	Bartez	Guy	Julien	Manon	Ugo
$R$	$K$		L	L, M			
$J$	L, M						
$P$	L	L	L, M	L, M	L, M	L, M	L, M
$S$			L, M		L, M		
$Z$	L	L	L, M	L	L	L	L

**1.60** Soit  $M = \{m_1, m_2, m_3, m_4, m_5\}$  l'ensemble des magasins d'une bannière et  $P = \{p_1, p_2, \dots, p_7\}$  l'ensemble des produits vendus. Dans tableau suivant, le symbole 1 dans une case désigne que le produit est vendu par le magasin, tandis qu'une case vide signifie le contraire.

Notons  $L_i \subseteq P$  l'ensemble des produits vendus par le magasin  $i$ . Par exemple, on a  $L_1 = \{p_3, p_4, p_5, p_6\}$ .

Décrivez chacun des ensembles suivants en extension. De plus, pour chacun des ensembles, donnez sa cardinalité.

- (a)  $L_1 \cup L_2$
- (b)  $L_2 \cup L_4$
- (c)  $(L_1 \cup L_2) \cap L_4$
- (d)  $L_2 - L_4$
- (e)  $\overline{L_1 \cup L_2}$
- (f)  $\bigcup_{i=2}^4 L_i$
- (g)  $\bigcap_{i=3}^5 L_i$

$P \backslash M$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
$p_1$		1		1	1
$p_2$		1		1	
$p_3$	1	1			
$p_4$	1		1		
$p_5$	1				1
$p_6$	1	1		1	
$p_7$			1		

**1.61** Dans le contexte de l'exercice 1.60, déterminez si chacun des énoncés suivants est vrai ou faux.

- (a)  $L_4 \subset L_2$
- (b)  $L_4 \subseteq L_2$
- (c)  $L_1 = L_2$
- (d)  $p_4 \subset L_3$
- (e)  $p_4 \in L_3$
- (f)  $m_4 \in M$
- (g)  $L_2 \in P$
- (h)  $L_2 \subset P$
- (i)  $\bigcup_{i=2}^5 L_i = P$
- (j)  $\{p_2, p_4\} \subseteq P$

$P \backslash M$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
$p_1$		1		1	1
$p_2$		1		1	
$p_3$	1	1			
$p_4$	1		1		
$p_5$	1				1
$p_6$	1	1		1	
$p_7$			1		

**1.62** Dans le contexte de l'exercice 1.60, on définit

$$x_{ij} = \begin{cases} 1 & \text{si le produit } p_i \text{ est vendu par le magasin } m_j \\ 0 & \text{sinon} \end{cases}.$$

Associez chacun des énoncés suivants à une des 14 phrases de la liste. De plus, déterminez si chacun des énoncés est vrai ou faux.

(a)  $\sum_{i=1}^7 x_{i3} \geq 3$

(d)  $\forall i \in \{1, 2, 3, 4, 5, 6, 7\}, \sum_{j=1}^5 x_{ij} \geq 3$

(b)  $\sum_{j=1}^5 x_{3j} \geq 3$

(e)  $\exists j \in \{1, 2, 3, 4, 5\}, x_{3j} = 0$

(f)  $\sum_{j=1}^5 x_{3j} \leq 3$

(c)  $\forall j \in \{1, 2, 3, 4, 5\}, \sum_{i=1}^7 x_{ij} \geq 3$

(g)  $\exists j \in \{1, 2, 3, 4, 5\}, \sum_{i=1}^7 x_{ij} > 3$

- |   |   |
|---|---|
| 1. Le magasin 3 vend au moins 3 produits.           | 9. Le produit 3 est vendu dans au plus 3 magasins.              |
| 2. Le magasin 3 vend moins de 3 produits.           | 10. Au moins un des magasins ne vend pas le produit 3.          |
| 3. Le magasin 3 vend au plus 3 produits.            | 11. Au moins un des produits n'est pas vendu dans le magasin 3. |
| 4. Le magasin 3 vend plus de 3 produits.            | 12. Au moins un des magasins vend plus de 3 produits.           |
| 5. Le magasin 3 vend au moins 3 produits.           | 13. Chaque produit est disponible dans au moins 3 magasins.     |
| 6. Le magasin 3 vend moins de 3 produits.           | 14. Chaque magasin vend au moins 3 produits.                    |
| 7. Le produit 3 est vendu dans moins de 3 magasins. |   |
| 8. Le produit 3 est vendu dans au moins 3 magasins. |   |

**1.63** Chacune des expressions suivantes désigne un ensemble qui peut être décrit par une expression plus simple. Utilisez la table des propriétés des ensembles pour obtenir une expression simplifiée. Indiquez la propriété utilisée à chaque étape.

(a)  $\overline{A \cap B \cap C} \cup C$

(b)  $A \cap (H \cup D \cup A)$

(c)  $\overline{(A \cap B) \cup \overline{B}}$

(d)  $\overline{(A \cup B) \cup \overline{B}}$

(e)  $\overline{(A \cup \overline{B}) \cup (C \cup \overline{A})}$

**1.64** Écrivez une expression équivalente à celle présentée en utilisant uniquement les connecteurs logiques  $\wedge$  ou  $\vee$ , les symboles d'inégalités  $\leq$  ou  $\geq$  et le symbole d'égalité  $=$ .

(a)  $x \in [2, 4]$

(b)  $x \in [2, \infty[$

(c)  $x \in [2, 4] \cup [7, 9]$

(d)  $x \in ]-\infty, 0] \cup [10, 15]$

(e)  $x \in [2, 4[ \cup ]4, 8]$

## 1.4 Fonctions

### Définition 1.17 : Fonction

Une **fonction**  $f$  d'un ensemble  $A$  vers un ensemble  $B$  est une règle qui, à chaque élément  $a$  de l'ensemble  $A$ , associe un et un seul élément  $b$  de l'ensemble  $B$ . Cet élément  $b$  est noté  $f(a)$ . On écrit alors parfois  $(a, b) \in f$ .

La notation usuelle pour désigner une fonction  $f$  d'un ensemble  $A$  vers un ensemble  $B$  est

$$f : A \longrightarrow B$$

L'ensemble  $A$  est appelé le **domaine** de la fonction  $f$ , noté  $\text{Dom}(f)$ , et le sous-ensemble de  $B$  formé des éléments atteints par  $f$  est appelé l'**image** de  $f$ , noté  $\text{Im}(f)$ .

$$\text{Im}(f) = \{b \in B \mid \exists a \in A, f(a) = b\} \subseteq B$$

Par ailleurs, on peut aussi voir une fonction  $f$  de  $A$  vers  $B$  comme un sous-ensemble du produit cartésien  $A \times B$  ayant la propriété suivante :

$$\forall a \in A, \exists! b \in B, (a, b) \in f$$

où le symbole  $\exists!$  désigne « **il existe un et un seul** ».

### Exemple 1.37

Considérons  $T_8$ , l'ensemble des trains de bits de longueur 8 et la fonction  $f : T_8 \longrightarrow \mathbb{N}$  définie par

$$f(t) = \text{nombre de 0 dans le train de bits } t.$$

Par exemple,  $f(11001011) = 3$ . Donnez le domaine et l'image de la fonction  $f$ .

### Solution :

On a

$$\text{Dom}(f) = T_8 \quad \text{et} \quad \text{Im}(f) = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}.$$

**Exemple 1.38**

Soit  $M = \{m_1, m_2, m_3, m_4, m_5\}$  l'ensemble des magasins d'une bannière et  $P = \{p_1, p_2, \dots, p_7\}$  l'ensemble des produits vendus. Étant donné un magasin  $m \in M$  et un produit  $p \in P$ , on définit

$$f(m, p) = \begin{cases} 1 & \text{si le magasin } m \text{ vend le produit } p \\ 0 & \text{sinon} \end{cases}.$$

$P \backslash M$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
$p_1$	0	1	0	1	1
$p_2$	0	1	0	1	0
$p_3$	1	1	0	0	0
$p_4$	1	0	1	0	0
$p_5$	1	0	0	0	1
$p_6$	1	1	0	1	0
$p_7$	0	0	1	0	0

Donnez le domaine et l'image de la fonction  $f$ .

De plus, donnez  $f(m_3, p_4)$  si les images de la fonction  $f$  sont données par le tableau ci-dessus.

**Solution :**

On a

$$\text{Dom}(f) = M \times P \quad \text{et} \quad \text{Im}(f) = \{0, 1\}.$$

De plus,  $f(m_3, p_4) = 1$ .

**Exemple 1.39**

Dans le contexte de l'exemple précédent, étant donné un magasin  $m \in M$ , on définit

$$v(m) = \text{ensemble des produits vendus dans le magasin } m.$$

Donnez  $v(m_2)$ .

De plus, déterminez si l'image de la fonction  $v$  est un sous-ensemble de  $P$  ou de l'ensemble des parties de  $P$ .

**Solution :**

On a

$$v(m_2) = \{p_1, p_2, p_3, p_6\} \in \wp(P).$$

Ainsi, l'image de la fonction  $v$  est un sous-ensemble de l'ensemble des parties de  $P$ :

$$\text{Im}(v) \subseteq \wp(P).$$

### 1.4.1 Fonctions plancher et plafond

#### Définition 1.18 : Fonctions plancher et plafond

La fonction **plancher** associe à tout nombre réel  $x$ , le plus grand entier  $n$  tel que  $n \leq x$ . On note  $\lfloor x \rfloor = n$ . La fonction **plafond** associe à tout nombre réel  $x$ , le plus petit entier  $n$  tel que  $n \geq x$ . On note  $\lceil x \rceil = n$ .

#### Exemple 1.40

$$\left\lfloor \frac{1}{3} \right\rfloor = 0, \quad \left\lceil \frac{1}{3} \right\rceil = 1, \quad \lfloor -9.2 \rfloor = -10 \quad \text{et} \quad \lceil -9.2 \rceil = -9.$$

#### Théorème 1.5 : Propriétés des fonctions plancher et plafond

1.  $\lfloor x \rfloor = n \leftrightarrow n \leq x < n + 1$
2.  $\lceil x \rceil = n \leftrightarrow n - 1 < x \leq n$
3.  $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$

#### Exemple 1.41

Un magasin vend de la farine en sacs de 2 kg uniquement, au coût de 5\$ le sac.

Donnez une formule décrivant chacune des fonctions suivantes :

- la fonction  $m$  qui retourne le nombre minimal de sacs que l'on doit acheter si l'on a besoin de  $x$  kg de farine;
- la fonction  $s$  qui retourne le nombre maximal de sacs que l'on peut acheter avec  $d$  dollars;
- la fonction  $c$  qui retourne la quantité de farine à commander si l'on a besoin de  $x$  kg de farine.

**Solution :**

$$m(x) = \left\lceil \frac{x}{2} \right\rceil \quad s(d) = \left\lfloor \frac{d}{5} \right\rfloor \quad c(x) = 2 \left\lceil \frac{x}{2} \right\rceil.$$



## 1.4.2 Injection, surjection et bijection

**Définition 1.19 : Fonction injective, surjective, bijective**

Soit  $f : A \rightarrow B$  une fonction. On dit que

**$f$  est injective** si elle n'associe jamais la même image à deux éléments distincts :

$$\forall a_1 \in A, \forall a_2 \in A, (a_1 \neq a_2) \rightarrow (f(a_1) \neq f(a_2))$$

**$f$  est surjective** si son image est l'ensemble  $B$  au complet, c'est-à-dire si tous les éléments de  $B$  sont atteints :

$$\forall b \in B, \exists a \in A, f(a) = b$$

**$f$  est bijective** si elle est injective et surjective :

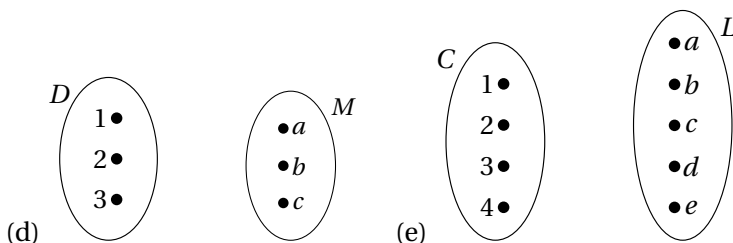
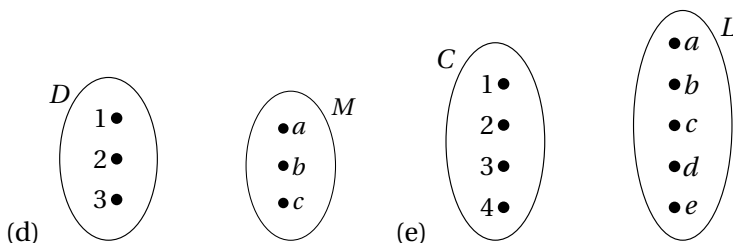
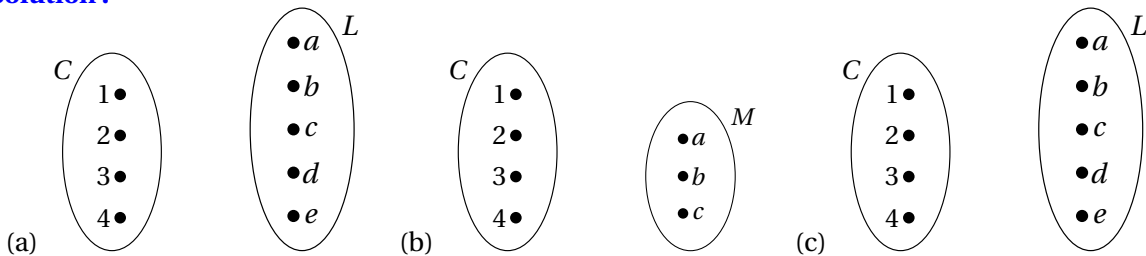
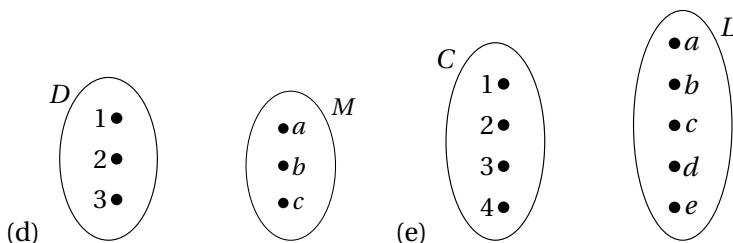
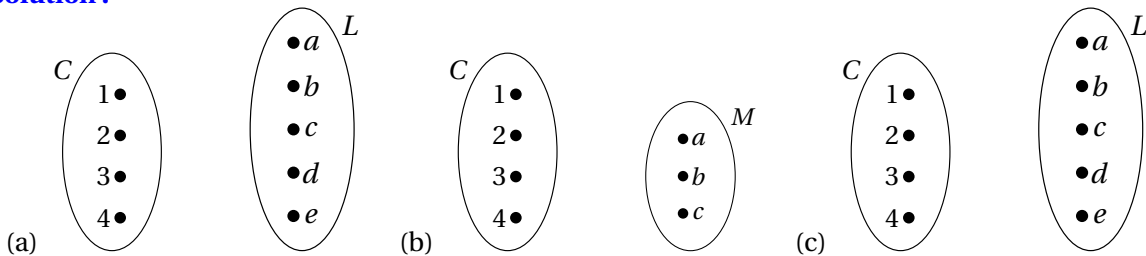
$$\forall b \in B, \exists! a \in A, f(a) = b$$

**Exemple 1.42 (à compléter en classe)**

On considère un sous-ensemble  $f$  du produit cartésien de deux ensemble. Dans chaque cas, tracez son graphe sagittal puis déterminez s'il s'agit d'une fonction ou non. De plus, si  $f$  est une fonction, déterminez si elle est injective, surjective ou bijective.

Ici,  $L = \{a, b, c, d, e\}$ ,  $M = \{a, b, c\}$ ,  $C = \{1, 2, 3, 4\}$  et  $D = \{1, 2, 3\}$ .

- (a)  $f = \{(1, a), (2, d), (3, c), (4, e)\} \subseteq C \times L$       (d)  $f = \{(1, c), (2, a), (3, b)\} \subseteq D \times M$   
 (b)  $f = \{(1, a), (2, a), (3, c), (4, b)\} \subseteq C \times M$       (e)  $f = \{(1, a), (2, a), (3, a), (4, a)\} \subseteq C \times L$   
 (c)  $f = \{(1, a), (2, d), (3, c), (4, e), (1, b)\} \subseteq C \times L$

**Solution :**

**Exemple 1.43** (à compléter en classe)

La fonction  $f : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$  définie par  $f(x_1, x_2) = x_1 + x_2$  est-elle oui ou non injective? Est-elle oui ou non surjective? Est-elle oui ou non bijective?

**Exercices**

**1.65** Déterminez si la fonction  $f$  est injective, surjective ou bijective.

$T_8$  désigne ici l'ensemble des trains de bits de longueur 8.

- (a)  $f : T_8 \rightarrow \mathbb{N}$ ,  $f(t) =$  nombre de 0 dans le train de bits  $t$ .
- (b)  $f : T_8 \rightarrow T_8$ ,  $f(t) = \sim t$  le train formé en inversant la valeur de chacun des bits de  $t$ .
- (c)  $f : T_8 \rightarrow T_8$ ,  $f(t) = (t \vee 11110000)$ , où  $\vee$  est appliqué bit à bit.
- (d)  $f : T_8 \rightarrow \{0, 1, 2, 3, \dots, 8\}$ ,  $f(t) =$  nombre de 1 dans le train de bits  $t$ .

**1.66** Déterminez si la fonction  $f$  est injective, surjective ou bijective.

- (a)  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,  $f(x) = 2x + 1$ .
- (b)  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,  $f(x) = x^2 + 1$ .
- (c)  $f : \mathbb{Z} \rightarrow \mathbb{Q}$ ,  $f(x) = \frac{x}{1}$ .
- (d)  $f : \mathbb{Q} \rightarrow \mathbb{Z}$ ,  $f(x) = p + q$  où  $\frac{p}{q}$  est l'unique représentation réduite du rationnel  $x$ .
- (e)  $f : \mathbb{R} \rightarrow \mathbb{Z}$ ,  $f(x) = \lceil x \rceil$ .

**1.67** Considérez les ensembles suivants.

$$\begin{aligned} A &= \mathbb{Z} \\ B &= \{x \in \mathbb{Q} \mid 0 < x \leq 1\} \end{aligned}$$

Soit  $f : A \rightarrow B$  la fonction définie par

$$f(x) = \frac{1}{(x-1)^2 + 1}$$

- (a) La fonction  $f$  est-elle surjective? Justifiez.
- (b) La fonction  $f$  est-elle injective? Justifiez.

**1.68** Considérez les ensembles suivants.

$$A = \mathbb{N}$$

$$B = \{x \in \mathbb{Q} \mid 0 < x \leq 1\}$$

Soit  $f : A \rightarrow B$  la fonction définie par

$$f(x) = \frac{1}{(x-1)^2 + 1}$$

- (a) La fonction  $f$  est-elle surjective? Justifiez.
- (b) La fonction  $f$  est-elle injective? Justifiez.

**1.69** Considérez les ensembles suivants.

$$A = \mathbb{R}$$

$$B = \{x \in \mathbb{R} \mid 0 < x \leq 1\}$$

Soit  $f : A \rightarrow B$  la fonction définie par

$$f(x) = \frac{1}{(x-1)^2 + 1}$$

- (a) La fonction  $f$  est-elle surjective? Justifiez.
- (b) La fonction  $f$  est-elle injective? Justifiez.



# Chapitre 2

## Théorie des nombres

### 2.1 Arithmétique modulaire

#### 2.1.1 Division entière

##### Définition 2.1 : Divisibilité

Si  $a \in \mathbb{Z}$ ,  $b \in \mathbb{Z}$  et  $a \neq 0$ , on dit que  $a$  **divise**  $b$  s'il existe un entier  $c$  tel que  $b = ac$ . L'entier  $a$  est alors appelé un **facteur** de  $b$ .

Notation :

$$\begin{aligned} a \mid b &\longleftrightarrow \exists c \in \mathbb{Z}, ac = b \longleftrightarrow \frac{b}{a} \in \mathbb{Z} \\ a \nmid b &\longleftrightarrow \neg(a \mid b) \end{aligned}$$

Attention, la barre verticale utilisée pour signifier que  $a$  divise  $b$ ,  $a \mid b$ , ne doit pas être confondue avec le symbole de substitution de Nspire, le « tel que » :  $(x^2 \mid x = 2) = 4$ . Notons aussi que  $-2 \mid 6$  n'est *pas* une opération arithmétique dont le résultat est  $-3$  : c'est une proposition dont la valeur de vérité est **V**.

##### Théorème 2.1 : Divisibilité

Soient  $a, b$  et  $c$  des nombres entiers quelconques, avec  $a \neq 0$ .

1. Si  $a \mid b$  et  $a \mid c$  alors  $a \mid (b + c)$  et  $a \mid (b - c)$ .
2. Si  $a \mid b$  alors  $a \mid (bc)$ .
3. Si  $a \mid b$  et  $b \mid c$  alors  $a \mid c$ .

**Exemple 2.1** (à compléter en classe)

Vrai ou faux? Justifiez en invoquant une définition, un théorème, en donnant une preuve ou un contre-exemple.

(a)  $7 \mid 10$

(b)  $-5 \mid 10$

(c)  $100 \mid 10$

(d)  $5 \mid -10$

(e)  $\forall x \in \mathbb{N}, x \mid 17 \rightarrow x \mid 85$

(f)  $\forall x \in \mathbb{N}, x \mid bc \rightarrow (x \mid b \vee x \mid c)$

**Théorème 2.2**

Soient  $a$  et  $d$  des entiers, avec  $d > 0$ . Il existe une seule paire d'entiers  $q$  et  $r$  satisfaisant

$$0 \leq r < d \quad \text{et} \quad a = dq + r.$$

**Définition 2.2 : Diviseur, dividende, quotient, reste, modulo**

Considérons  $a$  et  $d$  des entiers, avec  $d > 0$ . Le théorème précédent stipule qu'il existe une seule paire d'entiers  $q$  et  $r$  satisfaisant

$$a = dq + r \quad \text{et} \quad 0 \leq r < d.$$

- L'entier  $d$  est appelé le **diviseur**.
- L'entier  $a$  est appelé le **dividende**.
- L'entier  $q$  est appelé le **quotient** (notation :  $q = a \text{ div } d$ ).
- L'entier  $r$  est appelé le **reste** (notation :  $r = a \text{ mod } d$ ). On dit que  $r$  est égal à  $a$  **modulo**  $m$ .

**Exemple 2.2**

Par exemple si on divise  $a = 17$  par  $d = 3$ , on obtient

$$17 = 3 \cdot 5 + 2 \quad \text{et} \quad 0 \leq 2 < 3.$$

- L'entier  $d = 3$  est appelé le **diviseur**.
  - L'entier  $a = 17$  est appelé le **dividende**.
  - L'entier  $q = 5$  est appelé le **quotient**.
  - L'entier  $r = 2$  est appelé le **reste**.
- On dit aussi que 2 est égal à 17 **modulo** 3 (notation :  $2 = 17 \text{ mod } 3$ ).

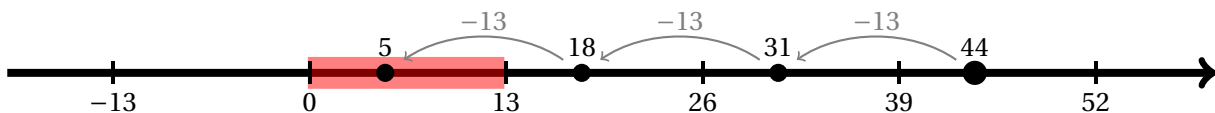
Deux images sont très utiles pour comprendre ce qu'est le « reste modulo  $m$  ».

- Celle d'une droite réelle sur laquelle on fait des bonds de longueur  $m$  pour atteindre une valeur sur l'intervalle allant de 0 inclusivement à  $m$  exclusivement :  $[0, m[$ .
- Celle d'une horloge circulaire de  $m$  heures identifiées de 0 à  $m - 1$ .

Les deux exemples suivants illustrent ces analogies.

**Exemple 2.3**

Pour déterminer le reste de la division entière de 44 par  $m = 13$ , on peut considérer la droite réelle sur laquelle on identifie le nombre 44 et l'intervalle  $[0, 13[$ . À partir de 44, on se déplace en effectuant des bonds de taille 13 jusqu'à atterrir dans l'intervalle  $[0, 13[$ .



En procédant de la sorte, on arrive finalement sur le nombre 5, ce qui correspond bien à  $44 \text{ mod } 13$  :

$$44 \text{ mod } 13 = 5 \quad \text{car} \quad 44 = 13 \cdot 3 + 5 \quad \text{et} \quad 0 \leq 5 < 13.$$

**Exemple 2.4**

Calculez les valeurs de  $-8 \bmod 5$  et  $9 \bmod 5$  et illustrez les résultats en dessinant une aiguille sur un cadran de 5 heures.

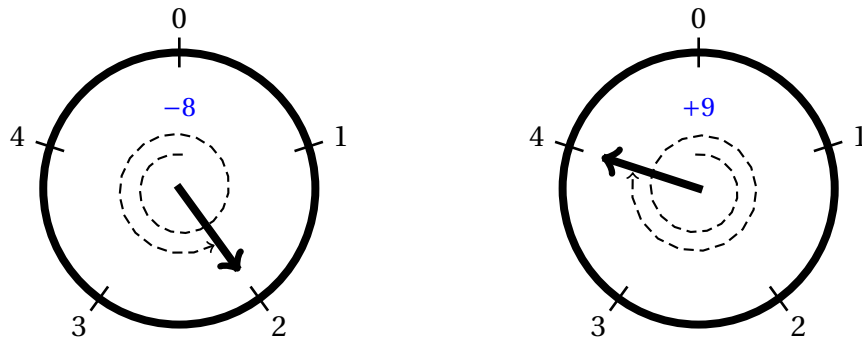
**Solution :**

On a

$$-8 \bmod 5 = 2 \quad \text{car} \quad -8 = -2 \cdot 5 + 2 \text{ et } 0 \leq 2 < 5.$$

$$9 \bmod 5 = 4 \quad \text{car} \quad 9 = 1 \cdot 5 + 4 \text{ et } 0 \leq 4 < 5.$$

Sur une horloge de 5 heures, si l'aiguille est en position 0 et recule de 8 h, elle arrive à la position 2 h. Si l'aiguille est en position 0 et avance de 9 h, elle arrive à la position 4 h.



De manière générale, pour calculer le quotient  $q$  et le reste  $r$  de la division de  $a$  par l'entier positif  $d$ , il faut compter le nombre de fois ( $q$ ) que l'on peut soustraire  $d$  de  $|a|$  tout en conservant un résultat positif ( $r$ ). **Si  $a$  est positif**, le diviseur est  $q$  et le reste est  $r$ . **Si  $a$  est négatif**, on procède à un petit ajustement ( $r := d - r$  et  $q := -(q + 1)$ ).

**Exemple 2.5 (à compléter en classe)**

Calculez les valeurs demandées.

(a)  $5 \bmod 2$

(b)  $2 \bmod 5$

(c)  $44 \bmod 10$

(d)  $-44 \bmod 10$



(e)  $-44 \pmod{3}$ 

**Remarque:** En (d), nous pourrions être tentés de dire que  $r = -44 \pmod{10} = -4$  puisque  $-44 = 10(-4) + (-4)$ , mais comme dans la définition nous voulons  $0 \leq r < 10$ , alors le réajustement suivant doit être fait :

$$r := d - r = 10 - 4 = 6 \quad \text{et} \quad q := -(q+1) := -(4+1) = -5.$$

Nous allons voir à la section suivante qu'en fait, les entiers  $-4$  et  $6$  sont congrus modulo  $10$ .

### Exercices

**2.1** Si nous sommes mardi, quel jour de la semaine serons-nous dans 30 jours ?

**2.2** Il est 15 heures. Quelle heure sera-t-il dans 60 heures ?

### 2.1.2 Congruences modulo $m$

#### Définition 2.3 : Congruence modulo $m$

Soient  $a$  et  $b$  des entiers et  $m \in \mathbb{N}^*$ . On dit que  $a$  est congru à  $b$  modulo  $m$  si et seulement si  $m$  divise  $a - b$ . Notation

$$a \equiv b \pmod{m} \iff m \mid (a - b)$$

**Attention :** le symbole  $\equiv$  est aussi utilisé en logique, mais il a ici un sens différent.

**Remarque :** si  $a$  est congru à  $b$  modulo  $m$ , c'est que  $m$  divise  $(a - b)$ , mais alors  $m$  divise aussi  $(b - a)$ , et donc, par définition,  $b$  est congru à  $a$  modulo  $m$ . On peut donc dire que  **$a$  et  $b$  sont congrus modulo  $m$** .

#### Exemple 2.6

Les nombres  $73$  et  $23$  sont congrus modulo  $10$  car  $10$  divise leur différence.

$$\begin{aligned} 73 \equiv 23 \pmod{10} &\iff 10 \mid (73 - 23) \\ &\iff 10 \mid 50 \\ &\iff \frac{50}{10} \in \mathbb{Z} \\ &\iff 5 \in \mathbb{Z} \end{aligned}$$

Les nombres  $-8$  et  $7$  sont congrus modulo  $5$  car  $5$  divise leur différence.

$$\begin{aligned} -8 \equiv 7 \pmod{5} &\iff 5 \mid (-8 - 7) \\ &\iff 5 \mid (-15) \\ &\iff \frac{-15}{5} \in \mathbb{Z} \\ &\iff -3 \in \mathbb{Z} \end{aligned}$$

**Théorème 2.3 : congruences**

Soient  $a$  et  $b$  des entiers et  $m \in \mathbb{N}^*$ . Les énoncés suivants sont équivalents.

1.  $a \equiv b \pmod{m}$
2.  $m \mid (a - b)$
3.  $(a \bmod m) = (b \bmod m)$
4.  $\exists k \in \mathbb{Z}, a = b + km$

**Exemple 2.7**

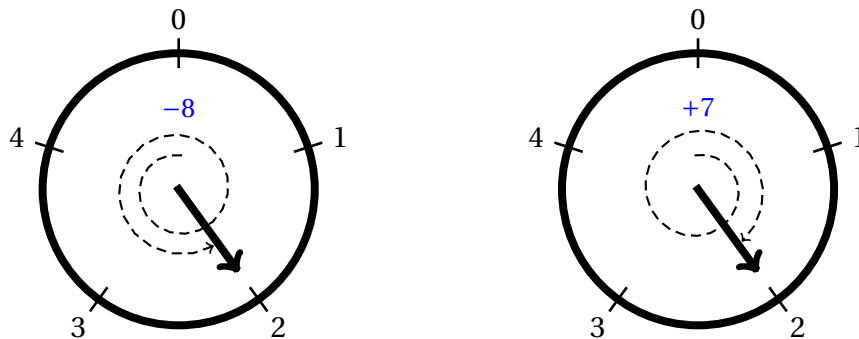
Illustrons le théorème 2.3 avec les nombres  $a = -8$ ,  $b = 7$  et  $m = 5$ . Le théorème stipule que les quatre énoncés sont équivalents: ils sont donc tous vrais ou tous faux. Pour les valeurs de  $a$  et  $b$  choisies, ils sont tous vrais.

1.  $-8 \equiv 7 \pmod{5}$
2.  $5 \mid (-8 - 7)$
3.  $(-8 \bmod 5) = (7 \bmod 5)$

En effet, on a

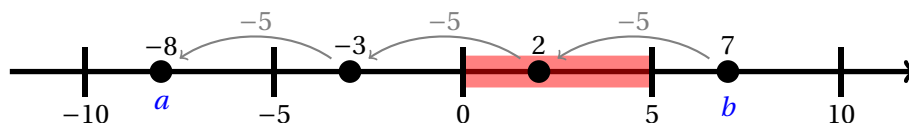
$$(-8 \bmod 5) = 2 \quad \text{et} \quad (7 \bmod 5) = 2.$$

Sur une horloge de 5 heures, si l'aiguille est en position 0 et recule de 8 h, elle arrive à la position 2 h. Si l'aiguille est en position 0 et avance de 7 h, elle arrive aussi à la position 2 h.



4. En prenant  $k = -3$ , on obtient  $-8 = 7 + (-3) \cdot 5$ .

Visuellement, cela signifie que l'on peut passer de  $b = 7$  à  $a = -8$  en se déplaçant par bonds de longueur  $m = 5$  sur la droite réelle.



Le théorème suivant est extrêmement important. Il permet de simplifier et d'accélérer de nombreux calculs comme le montrent les exemples qui suivent.

#### Théorème 2.4

Soient  $m \in \mathbb{N}^*$  et  $x \in \mathbb{Z}$ . Si  $a_1 \equiv a_2 \pmod{m}$  alors

$$x + a_1 \equiv x + a_2 \pmod{m} \quad \text{et} \quad x \cdot a_1 \equiv x \cdot a_2 \pmod{m}$$

Ainsi, dans toute expression arithmétique utilisant seulement les opérations  $+$  et  $\cdot$ , on peut remplacer un nombre  $a_1$  par n'importe quel nombre  $a_2$  qui lui est congru modulo  $m$  et le résultat de l'expression originale sera inchangé. En arithmétique modulaire, il est donc préférable de calculer d'abord les restes modulo  $m$ , puis d'effectuer les multiplications et les additions de manière à toujours manipuler des nombres inférieurs à  $m$ .

#### Exemple 2.8

Utilisez le théorème 2.4 pour calculer

$$(118 + 84) \pmod{10} \quad \text{et} \quad (118 \cdot 84) \pmod{10}.$$

#### Solution :

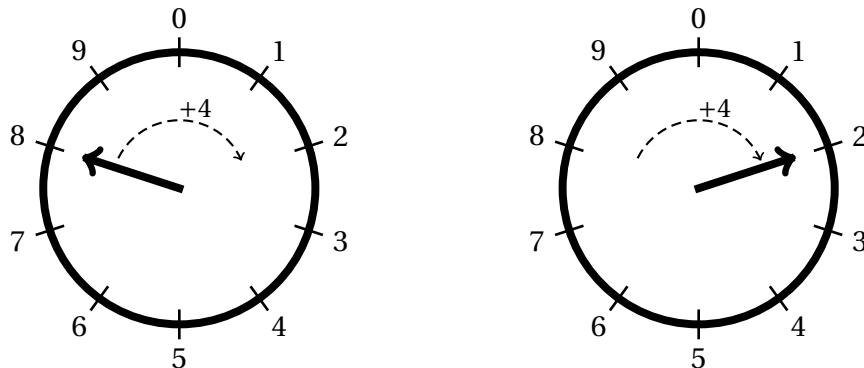
$$\begin{aligned} 118 + 84 &\equiv 8 + 4 \pmod{10} \\ &\equiv 12 \pmod{10} \\ &\equiv 2 \pmod{10} \end{aligned}$$

$$\begin{aligned} 118 \cdot 84 &\equiv 8 \cdot 4 \pmod{10} \\ &\equiv 32 \pmod{10} \\ &\equiv 2 \pmod{10} \end{aligned}$$

Donc

$$(118 + 84) \pmod{10} = 2 \quad \text{et} \quad (118 \cdot 84) \pmod{10} = 2.$$

Illustrons le fait que  $8 + 4 \equiv 2 \pmod{10}$ . On considère un cadran circulaire doté de  $m = 10$  positions. Sur ce cadran, si l'aiguille est en position 8 et qu'on la déplace de 4 unités vers la droite, alors l'aiguille se retrouve en position 2. Ceci illustre le fait que le reste de la division entière de  $8 + 4$  par 10 est 2.



**Exemple 2.9** (à compléter en classe)

Calculez **sans** utiliser la calculatrice. Soyez stratégiques : utilisez le théorème précédent!

- (a)  $(26 \cdot 19) \bmod 12$
- (b)  $(26 + 19) \bmod 12$
- (c)  $(136 \cdot 882 \cdot 14 + 12) \bmod 5$
- (d)  $((-912)^3 + 999 \cdot (82)) \bmod 10$

**Exercices**

**2.3** Vrai ou faux?

- (a)  $7 \equiv 13 \pmod{6}$
- (b)  $-2 \equiv 10 \pmod{6}$
- (c)  $0 \equiv 14 \pmod{6}$
- (d)  $30 \equiv 600 \pmod{6}$
- (e)  $5^3 \equiv 5 \pmod{6}$

**2.4** Calculez **sans** calculatrice. Vérifiez avec la commande  $\text{mod}(a, m)$  de la TI qui calcule  $a \bmod m$ .

- (a)  $-2 \bmod 11$
- (b)  $(-2)^3 \bmod 11$
- (c)  $(55 \cdot 13 + 8 \cdot 47) \bmod 11$
- (d)  $9^2 \bmod 11$
- (e)  $9^4 \bmod 11$  N. B. Vous n'avez *pas* à évaluer  $9^4$  puis à effectuer la division par 11. Utilisez le résultat précédent et le théorème 2.4.
- (f)  $9^8 \bmod 11$
- (g)  $9^{16} \bmod 11$

## 2.2 Représentation des entiers en base $b$

Un *système de numération* est un ensemble de règles qui permettent de représenter des nombres. Le plus ancien est probablement le système *unaire* où le symbole | représente l'entier un, || représente l'entier deux, ||| pour trois, |||| pour quatre et ainsi de suite. Ce système atteint vite ses limites, mais il permet de mettre en évidence le fait qu'il existe plusieurs façons de représenter les entiers.

Nom français	Système unaire	Système décimal	Chiffres romains
Zéro		0	
Un		1	I
Deux		2	II
Trois		3	III
⋮	⋮	⋮	⋮
Douze		12	XII
⋮	⋮	⋮	⋮

Dans la table ci-dessus, on remarque que sur une ligne donnée, on retrouve quatre manières différentes de représenter le même entier. Pour le reste de cette section, il sera important de dissocier la **représentation** d'un nombre et sa **valeur**.

### 2.2.1 Système de numération décimal

Il s'agit du système de numération le plus utilisé dans notre société. On peut le résumer avec les trois règles suivantes.

- On se dote de 10 symboles ordonnés qu'on nomme des *chiffres*: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Chaque chiffre représente une unité de plus que le précédent et le premier, *zéro*, désigne l'absence de quantité.
- Lorsqu'on écrit un nombre avec plusieurs chiffres, la *valeur* d'un chiffre est 10 fois plus importante que celle du chiffre à sa droite et le dernier représente les unités.

Ainsi, l'écriture « 197 281 » signifie :

$$197\,281 = 1 \cdot 10^5 + 9 \cdot 10^4 + 7 \cdot 10^3 + 2 \cdot 10^2 + 8 \cdot 10^1 + 1 \cdot 10^0.$$

### 2.2.2 Système de numération dans une base quelconque

Les trois règles du système décimal peuvent facilement être généralisées à une base  $b$  quelconque,  $b \in \mathbb{Z}$ ,  $b \geq 2$ . Les trois règles se traduisent comme suit :

- On se dote de  $b$  symboles ordonnés qu'on nomme des *chiffres*: 0, 1, 2, ...,  $(b - 1)$ .
- Chaque chiffre représente une unité de plus que le précédent et le premier, *zéro*, désigne l'absence de quantité.
- Lorsqu'on écrit un nombre avec plusieurs chiffres, la *valeur* d'un chiffre est  $b$  fois plus importante que celle du chiffre suivant et le dernier chiffre représente les unités.

Ainsi, en base  $b$ , l'écriture «  $a_n a_{n-1} \cdots a_2 a_1 a_0$  » où chaque  $a_i \in \{0, 1, \dots, b - 1\}$  est un chiffre, représente l'entier :

$$a_n b^n + a_{n-1} b^{n-1} + \cdots + a_2 b^2 + a_1 b + a_0.$$

**Notation**: lorsque plusieurs bases interviennent dans un même contexte, on écrit  $(a_n \cdots a_1 a_0)_b$  pour indiquer que le nombre est représenté en base  $b$ .

Le tableau ci-dessous liste les bases les plus fréquemment utilisées en informatique.

Nom	Base	Chiffres
Binaire	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Décimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadécimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

On remarque qu'en base 16, les dix chiffres de 0 à 9 ne suffisent pas. Il faut donc se doter de 6 symboles additionnels. On utilise les lettres de A à F avec la signification suivante :

$$(A)_{16} = (10)_{10}, \quad (B)_{16} = (11)_{10}, \quad (C)_{16} = (12)_{10}, \quad (D)_{16} = (13)_{10}, \quad (E)_{16} = (14)_{10}, \quad (F)_{16} = (15)_{10}.$$

### Exemple 2.10

Vérifiez que les représentations  $(1100\ 0100)_2$ ,  $(304)_8$ ,  $(196)_{10}$  et  $(C4)_{16}$  désignent toutes le même nombre.

#### Solution :

- Calcul de la valeur représentée par  $(1100\ 0100)_2$  :

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2 + 0 = 196.$$

- Calcul de la valeur représentée par  $(304)_8$  :

$$3 \cdot 8^2 + 0 \cdot 8 + 4 = 196.$$

- La représentation  $(196)_{10}$  est déjà en base 10, il n'y a rien à faire.
- Calcul de la valeur représentée par  $(C4)_{16}$  :  $12 \cdot 16 + 4 = 196$ .

Le théorème suivant stipule que, peu importe la base  $b \geq 2$ , chaque entier admet une représentation unique.

#### Théorème 2.5 : Représentation des entiers (développement de $x$ en base $b$ )

Soient  $b \in \mathbb{Z}$ ,  $b \geq 2$  et  $x \in \mathbb{N}^*$ . Il existe une unique suite  $a_n \cdots a_2 a_1 a_0$  avec chaque  $a_i \in \{0, 1, \dots, b-1\}$  et  $a_n \neq 0$  telle que

$$x = a_n b^n + a_{n-1} b^{n-1} + \cdots + a_2 b^2 + a_1 b + a_0.$$

▷ **Démonstration** On commence par montrer que tout entier  $x \in \mathbb{N}^*$  possède une représentation en base  $b$  et on montre ensuite que cette représentation est unique.

##### 1. Existence

Pour montrer l'existence d'une représentation en base  $b$ , on donne un algorithme qui construit une telle représentation et on prouve sa validité. Tout d'abord, on définit la suite infinie  $\{a_i\}_{i \geq 0}$  avec  $a_0 = x$  et  $\forall i \geq 1$ ,  $a_i = 0$ , puis on considère l'algorithme suivant :

**tant que**  $\exists i$  tel que  $a_i \geq b$  **faire**  
  Soustraire  $b$  à  $a_i$   
  Ajouter 1 à  $a_{i+1}$   
**fin tant que**

On appelle un **invariant de boucle** une propriété qui est vraie avant et après chacune des itérations d'une boucle. Cet algorithme présente deux invariants :

Invariant 1.  $\forall i, a_i \geq 0$ .

Invariant 2.  $\sum_{i \geq 0} a_i b^i = x$ .

Le premier invariant tient du fait qu'on soustrait  $b$  à  $a_i$  seulement si celui-ci est supérieur ou égal à  $b$ . Le deuxième invariant est justifié par la suite d'égalités suivantes :

$$a_i b^i + a_{i+1} b^{i+1} = (a_i \underbrace{-b+b}_{=0}) b^i + a_{i+1} b^{i+1} = (a_i - b) b^i + b b^i + a_{i+1} b^{i+1} = (a_i - b) b^i + (a_{i+1} + 1) b^{i+1}.$$

On montre maintenant que cet algorithme termine après un nombre fini d'itérations. Pour cela, on considère la somme  $\sum_{i \geq 0} a_i$ . Il s'agit d'un entier qui décroît à chaque itération de la boucle, or le premier invariant garantit que cette somme ne peut pas être négative. La terminaison de l'algorithme est donc assurée.

Comme l'algorithme termine seulement lorsque la condition du **tant que** est fautive, par De Morgan cela signifie que lorsque l'algorithme termine,  $\forall i, a_i < b$  et donc chaque  $a_i$  appartient à l'ensemble  $\{0, 1, \dots, b-1\}$ . Soit  $n$  le plus grand indice tel que  $a_n \neq 0$ , le deuxième invariant implique

$$x = \sum_{i=0}^n a_i b^i$$

Ceci conclut la preuve de l'existence d'une représentation de l'entier  $x$  en base  $b$ .

## 2. Unicité

Pour montrer l'unicité de la représentation, on procède par contradiction. On suppose qu'il existe deux suites distinctes  $x_n \cdots x_2 x_1 x_0$  et  $y_n \cdots y_2 y_1 y_0$  telles que tous les  $x_i, y_i \in \{0, 1, \dots, b-1\}$  et telles que

$$x_n b^n + x_{n-1} b^{n-1} + \cdots + x_2 b^2 + x_1 b + x_0 = y_n b^n + y_{n-1} b^{n-1} + \cdots + y_2 b^2 + y_1 b + y_0.$$

Pour tout  $i$ , on pose  $z_i = x_i - y_i$ . Par l'égalité précédente, on a

$$z_n b^n + z_{n-1} b^{n-1} + \cdots + z_2 b^2 + z_1 b + z_0 = 0.$$

Soit  $k$  le plus grand indice tel que  $z_k \neq 0$ . Sans perte de généralité, on suppose que  $z_k \geq 1$  (si  $z_k$  est négatif, il suffit de multiplier les deux côtés de l'égalité par  $-1$ ). Dans l'égalité précédente, on isole le terme  $z_k b^k$  afin d'obtenir :

$$z_k b^k = (-z_{k-1}) b^{k-1} + \cdots + (-z_2) b^2 + (-z_1) b + (-z_0) \quad (2.1)$$

Il ne reste plus qu'à montrer que cette égalité est impossible. Tout d'abord, comme  $z_k \geq 1$ , le membre de gauche satisfait l'inégalité :

$$z_k b^k \geq b^k.$$

D'un autre côté, pour tout  $i \in \{0, 1, \dots, k-1\}$ , la manière dont  $z_i$  est défini implique que  $|z_i| \leq b-1$ . Ainsi, le membre de droite de l'égalité 2.1 satisfait l'inégalité :

$$(-z_{k-1}) b^{k-1} + \cdots + (-z_2) b^2 + (-z_1) b + (-z_0) \leq (b-1) b^{k-1} + \cdots + (b-1) b^2 + (b-1) b + (b-1)$$

Dans le membre de droite, on effectue une mise en évidence du terme  $(b-1)$  afin d'obtenir une progression géométrique qui peut être résolue à l'aide du théorème 4.7(d) présenté au chapitre 4.

$$\begin{aligned}(b-1)b^{k-1} + \dots + (b-1)b^2 + (b-1)b + (b-1) &= (b-1) \left( b^{k-1} + \dots + b^2 + b + 1 \right) \\ &= (b-1) \left( \frac{b^k - 1}{b-1} \right) \\ &= b^k - 1.\end{aligned}$$

Ceci montre que l'égalité 2.1 est forcément fautive, car :

$$z_k b^k \geq b^k > b^k - 1 \geq (-z_{k-1})b^{k-1} + \dots + (-z_2)b^2 + (-z_1)b + (-z_0)$$

L'hypothèse voulant qu'il existe deux représentations distinctes pour le même entier mène donc à une contradiction. Il faut rejeter cette hypothèse, ce qui prouve l'unicité.

◁

### Algorithme 2.1 : Développement en base $b$

Pour calculer la représentation d'un entier  $x > 0$  dans une base  $b$  quelconque, on utilise l'algorithme suivant afin de calculer les chiffres  $a_i$  tels que  $(a_n a_{n-1} \dots a_1 a_0)_b = x$ ,

```
1:  $i := 0$ 
2: tant que  $x > 0$  faire
3:    $a_i := x \bmod b$ 
4:    $x := \lfloor x/b \rfloor$ 
5:    $i := i + 1$ 
6: fin tant que
```

### Exemple 2.11

Développez  $(222)_{10}$  en base 2.

#### Solution :

Effectuons une trace de l'algorithme avec  $x = 222$  et  $b = 2$ . Les lignes du tableau ci-dessous contiennent l'état des variables à la fin de chacune des itérations de la boucle *tant que*.

	$a_i$	$x$	$i$
Initialisation		222	0
Itération 1	$a_0 = 0$	111	1
Itération 2	$a_1 = 1$	55	2
⋮	$a_2 = 1$	27	3
	$a_3 = 1$	13	4
	$a_4 = 1$	6	5
	$a_5 = 0$	3	6
	$a_6 = 1$	1	7
	$a_7 = 1$	0	8



Ainsi  $(222)_{10} = (11011110)_2$ . Vérifions ce dernier résultat :

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2 + 0 = 222.$$

### Exemple 2.12

Développez  $(3658)_{10}$  en base 16.

#### Solution :

Effectuons une trace de l'algorithme avec  $x = 3658$  et  $b = 16$ .

	$a_i$	$x$	$i$
Initialisation		3658	0
Itération 1	$a_0 = 10$	228	1
Itération 2	$a_1 = 4$	14	2
Itération 3	$a_2 = 14$	0	3

Ainsi  $(1032)_{10} = (E4A)_{16}$ . Vérifions ce dernier résultat :

$$14 \cdot 16^2 + 4 \cdot 16 + 10 = 3658.$$

L'algorithme 2.1 permet de calculer la représentation d'un entier dans n'importe quelle base  $b \geq 2$ . Cela dit, dans certains cas il existe une manière plus simple. Par exemple, en informatique, il est fréquent d'utiliser le système hexadécimal, car celui-ci permet une conversion facile vers le binaire et vice-versa. Cette facilité est due au fait que 16 est une puissance de 2.

#### Passage d'une base $b$ à la base $b^m$

Étant donné  $x = (a_n \cdots a_1 a_0)_b$ , l'écriture de  $x$  en base  $b^m$  peut être obtenue en traitant son écriture en base  $b$  bloc par bloc, en considérant des blocs de taille  $m$ .

**Attention :**  $n + 1$ , le nombre de chiffres, doit être un multiple de  $m$  de manière à ce que tous les blocs aient exactement la taille  $m$ . Si ce n'est pas le cas, on ajoute des zéros à gauche pour corriger.

### Exemple 2.13

#### Binaire vers hexadécimal

Donnez le développement en base 16 du nombre  $x = (1001\ 0011\ 1010\ 0101\ 0001\ 1110\ 0101\ 1100)_2$ .

#### Solution :

Étant donné que  $16 = 2^4$ , il suffit de considérer l'écriture binaire de  $x$  par blocs de taille 4. Chaque bloc étant calculé de manière indépendante.

$$\begin{array}{cccccccc} \underbrace{1001} & \underbrace{0011} & \underbrace{1010} & \underbrace{0101} & \underbrace{0001} & \underbrace{1110} & \underbrace{0101} & \underbrace{1100} \\ (9)_{10}=(9)_{16} & (3)_{10}=(3)_{16} & (10)_{10}=(A)_{16} & (5)_{10}=(5)_{16} & (1)_{10}=(1)_{16} & (14)_{10}=(E)_{16} & (5)_{10}=(5)_{16} & (12)_{10}=(C)_{16} \end{array}$$

L'écriture hexadécimale est donc  $x = (93A51E5C)_{16}$ .

**Exemple 2.14****Hexadécimal vers binaire**

Donnez le développement en base 2 du nombre  $x = (3AF8)_{16}$ .

**Solution :**

Chaque chiffre du développement hexadécimal de  $x$  est converti en un bloc de 4 bits.

$$\begin{array}{cccc} \underbrace{3} & \underbrace{A} & \underbrace{F} & \underbrace{8} \\ (3)_{10}=(0011)_2 & (10)_{10}=(1010)_2 & (15)_{10}=(1111)_2 & (8)_{10}=(1000)_2 \end{array}$$

L'écriture binaire est  $x = (0011\ 1010\ 1111\ 1000)_2$ .

**Justification**

Le calcul suivant montre pourquoi, lorsqu'on effectue un changement de base de  $b$  vers  $b^m$ , il est correct de considérer chaque bloc de taille  $m$  indépendamment des autres. On considère l'entier  $x$  dont l'écriture dans la base  $b$  est  $(a_n \cdots a_1 a_0)_b$  :

$$x = a_0 + a_1 b + a_2 b^2 + \cdots + a_{n-1} b^{n-1} + a_n b^n.$$

L'entier  $x$  est décrit comme étant la somme de  $n+1$  termes, chacun étant de la forme  $a_i b^i$ . On considère  $S_k$  la somme des  $m$  termes consécutifs à partir de  $a_{km} b^{km}$  :

$$S_k = a_{km} b^{km} + a_{km+1} b^{km+1} + \cdots + a_{km+(m-2)} b^{km+(m-2)} + a_{km+(m-1)} b^{km+(m-1)}.$$

On met en évidence le facteur  $b^{km}$  et on définit  $a'_k$  de la manière suivante :

$$S_k = \underbrace{(a_{km} + a_{km+1} b + \cdots + a_{km+(m-2)} b^{(m-2)} + a_{km+(m-1)} b^{(m-1)})}_{a'_k} b^{km}.$$

Le terme  $S_k$  s'écrit alors comme étant simplement :

$$S_k = a'_k (b^m)^k$$

Pour finir, il reste à montrer que  $a'_k$  est bien un *chiffre* pour le système de numération en base  $b^m$ , c'est-à-dire que  $a'_k \in \{0, 1, \dots, b^m - 1\}$ . On procède en deux étapes, on montre d'abord que  $a'_k \geq 0$  puis que  $a'_k < b^m$ .

- $a'_k \geq 0$  car  $a'_k$  est la somme de produits d'entiers positifs ou nuls.
- $a'_k < b^m$  car chaque  $a_i$  est inférieur ou égal à  $b-1$ .

$$\begin{aligned} a'_k &= a_{km} + a_{km+1} b + \cdots + a_{km+(m-2)} b^{(m-2)} + a_{km+(m-1)} b^{(m-1)} \\ &\leq (b-1) + (b-1)b + \cdots + (b-1)b^{(m-2)} + (b-1)b^{(m-1)} \\ &= (b-1)(1 + b + \cdots + b^{(m-2)} + b^{(m-1)}) \\ &= (b-1) \frac{b^m - 1}{b-1} \\ &= b^m - 1 \end{aligned}$$

Le passage de la troisième à la quatrième ligne utilise le théorème 4.7(d) présenté au chapitre 4.

### 2.2.3 Changement de base sur Nspire

La calculatrice Nspire supporte les bases 2, 10 et 16, tel qu'illustré à la Figure 2.1. La figure 2.2 montre comment accéder aux fonctions permettant d'afficher la représentation d'un entier en base 2 et en base 16.

©Conversion base 10 vers bases 2 et 16	
196►Base2	0b11000100
196►Base16	0hC4
©Saisie d'un nombre en base 2	
0b11000100	196
©Saisie d'un nombre en base 16	
0hC4	196

Figure 2.1 Exemples de changements de base sur Nspire. Le préfixe **0b** indique la saisie d'un entier écrit en base 2, alors que le préfixe **0h** indique la saisie d'un entier écrit en base 16.

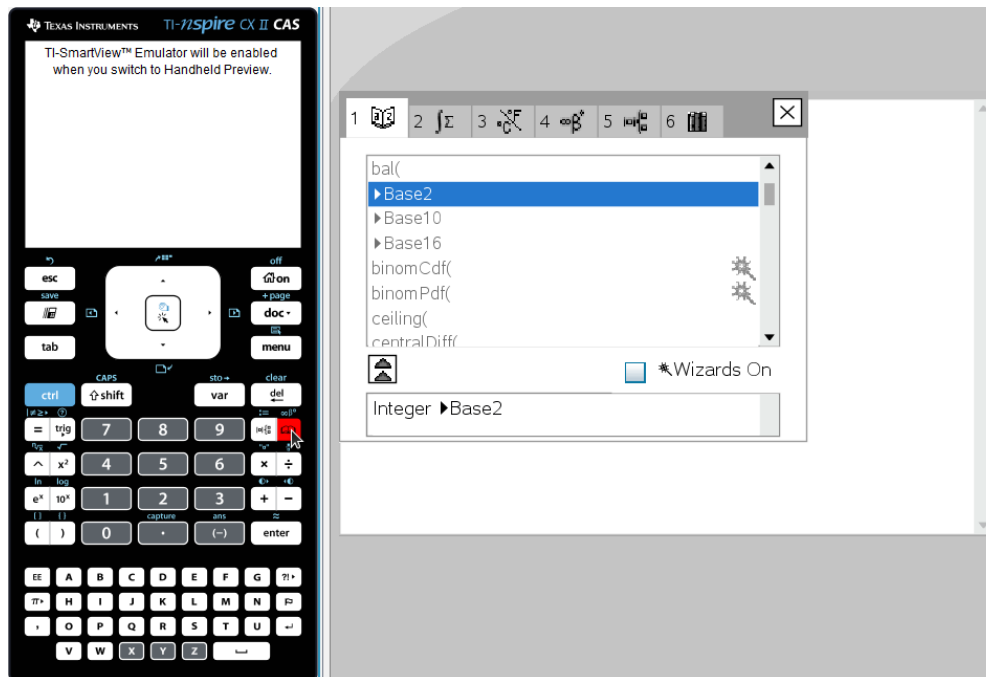


Figure 2.2 Sur Nspire, les fonctions de changement de base sont listées dans le catalogue.

**Exercices**

**2.5** Donnez la valeur des nombres suivants en base 10.

- (a)  $(110100)_2$  (c)  $(333)_7$   
(b)  $(110100)_3$  (d)  $(253)_9$

**2.6** Montrez que pour toute base  $b \geq 3$ , l'égalité suivante est valide :

$$(121)_b = ((11)_b)^2.$$

**2.7** Écrivez les nombres suivants dans la base demandée.

- (a)  $(511)_{10}$  en base 8. (d)  $(100110011)_2$  en base 3.  
(b)  $(1330)_{10}$  en base 11. (e)  $(100110011)_3$  en base 2.  
(c)  $(2748)_{10}$  en base 16.

**2.8** Calculez les changements de base demandés.

- (a) Développez  $(1111\ 0000\ 1001\ 1101)_2$  en base 8.  
(b) Développez  $(1231231)_4$  en base 16.  
(c) Développez  $(776105)_8$  en base 16.  
(d) Développez  $(AABB)_{16}$  en base 8.

## 2.3 Entiers et algorithmes

La prochaine section présente un algorithme permettant de calculer  $b^n \bmod m$  beaucoup plus efficacement que par la méthode naïve.

### 2.3.1 Algorithme d'exponentiation modulaire efficace

#### Algorithme 2.2 : Exponentiation modulaire $b^n \bmod m$

1. Calculer le développement binaire de l'exposant:  $n = (a_k \dots a_1 a_0)_2$ .
2. Calculer successivement les valeurs modulo  $m$  de  $b, b^2, b^4, b^8, \dots, b^{2^k}$ .
3. Multiplier les termes  $b^{2^i}$  pour lesquels  $a_i = 1$  afin d'obtenir  $b^n$  modulo  $m$ , en accord avec la loi des exposants

$$b^n = b^{(a_k 2^k + \dots + a_2 2^2 + a_1 2^1 + a_0)} = b^{a_k 2^k} \cdot \dots \cdot b^{a_3 2^3} \cdot b^{a_2 2^2} \cdot b^{a_1 2^1} \cdot b^{a_0}.$$

#### Exemple 2.15 (à compléter en classe)

Calculez  $5^{21} \bmod 14$  **à la main**, en utilisant l'algorithme d'exponentiation modulaire.

**Solution :**

**Exemple 2.16**

Calculez  $7^{222} \pmod{11}$  à la main.

**Solution :**

Comme nous l'avons vu à l'exemple 2.11,

$$\begin{aligned} 222 &= 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 \\ &= 2^7 + 2^6 + 2^4 + 2^3 + 2^2 + 2^1. \end{aligned}$$

Ainsi, nous pouvons écrire

$$\begin{aligned} 7^{222} &= 7^{(2^7+2^6+2^4+2^3+2^2+2^1)} \\ &= 7^{2^7} \cdot 7^{2^6} \cdot 7^{2^4} \cdot 7^{2^3} \cdot 7^{2^2} \cdot 7^{2^1} \\ &= 7^{128} \cdot 7^{64} \cdot 7^{16} \cdot 7^8 \cdot 7^4 \cdot 7^2 \end{aligned}$$

et calculer les puissances de  $7^{2^k} \pmod{11}$  successivement

$$\begin{aligned} 7^2 &\equiv 7 \cdot 7 \equiv 49 \equiv 4 \cdot 11 + 5 \equiv 0 + 5 \equiv 5 \pmod{11} \\ 7^4 &\equiv 7^2 \cdot 7^2 \equiv 5 \cdot 5 \equiv 25 \equiv 2 \cdot 11 + 3 \equiv 0 + 3 \equiv 3 \pmod{11} \\ 7^8 &\equiv 7^4 \cdot 7^4 \equiv 3 \cdot 3 \equiv 9 \pmod{11} \\ 7^{16} &\equiv 7^8 \cdot 7^8 \equiv 9 \cdot 9 \equiv 81 \equiv 7 \cdot 11 + 4 \equiv 0 + 4 \equiv 4 \pmod{11} \\ 7^{32} &\equiv 7^{16} \cdot 7^{16} \equiv 4 \cdot 4 \equiv 16 \equiv 11 + 5 \equiv 0 + 5 \equiv 5 \pmod{11} \\ 7^{64} &\equiv 7^{32} \cdot 7^{32} \equiv 5 \cdot 5 \equiv 25 \equiv 3 \pmod{11} \\ 7^{128} &\equiv 7^{64} \cdot 7^{64} \equiv 3 \cdot 3 \equiv 9 \pmod{11} \end{aligned}$$

pour obtenir

$$\begin{aligned} 7^{222} &= 7^{128} \cdot 7^{64} \cdot 7^{16} \cdot 7^8 \cdot 7^4 \cdot 7^2 \\ &\equiv (((9 \cdot 3) \cdot 4) \cdot 9) \cdot 3 \cdot 5 && \text{toutes les congruences sont modulo 11} \\ &\equiv (((5 \cdot 4) \cdot 9) \cdot 3) \cdot 5 && \text{car } 9 \cdot 3 \equiv 27 \equiv 5 \pmod{11} \\ &\equiv ((9 \cdot 9) \cdot 3) \cdot 5 && \text{car } 5 \cdot 4 \equiv 20 \equiv 9 \pmod{11} \\ &\equiv (4 \cdot 3) \cdot 5 && \text{car } 9 \cdot 9 \equiv 81 \equiv 4 \pmod{11} \\ &\equiv 1 \cdot 5 && \text{car } 4 \cdot 3 \equiv 12 \equiv 1 \pmod{11} \\ &\equiv 5. \end{aligned}$$

Ainsi, le résultat de  $7^{222} \pmod{11}$  est 5.

**Exercices**

**2.9** À l'aide de l'algorithme d'exponentiation modulaire, calculez **à la main** :

(a)  $13^{24} \bmod 9$

(c)  $3^7 \bmod 10$

(b)  $17^6 \bmod 20$

(d)  $3^{33} \bmod 10$

**2.10** Utilisez l'algorithme d'exponentiation modulaire pour calculer les valeurs suivantes (indiquez toutes les étapes de l'algorithme).

(a)  $7^{1323} \bmod 11$

(b)  $1819^{323} \bmod 245363$

**2.3.2 Nombres premiers et PGCD****Définition 2.4 : Nombre premier**

Un entier  $p > 1$  est dit **premier** si ses seuls facteurs positifs sont 1 et  $p$ .

**Exemple 2.17**

Les entiers

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29$$

sont des nombres premiers puisqu'ils ne sont divisibles que par 1 et eux-mêmes.

**Théorème 2.6 : Théorème fondamental de l'arithmétique**

Tout entier positif supérieur à 1 peut être écrit de façon unique soit comme un nombre premier, soit comme un produit de nombres premiers où les facteurs sont écrits en ordre croissant.

Nous allons prouver ce théorème par récurrence en deuxième moitié de trimestre. Nous n'avons pas encore étudié ce type de preuve!

**Exemple 2.18**

Les entiers 84 et 150 se décomposent en facteurs premiers comme ceci :

$$\begin{aligned} 84 &= 2^2 \cdot 3 \cdot 7 \\ 150 &= 2 \cdot 3 \cdot 5^2. \end{aligned}$$

**Théorème 2.7**

Il y a une infinité de nombres premiers.

**Définition 2.5 : PGCD et PPCM**

Soient  $a$  et  $b$  des entiers. Le plus grand entier  $d$  qui divise  $a$  et qui divise  $b$  est appelé le **plus grand commun diviseur** de  $a$  et  $b$ . On le note  $\text{PGCD}(a, b)$  en français et  $\text{GCD}(a, b)$  en anglais.

Le plus petit entier  $m$  qui est à la fois un multiple de  $a$  et un multiple de  $b$  est appelé le **plus petit commun multiple** de  $a$  et  $b$ . On le note  $\text{PPCM}(a, b)$  en français et  $\text{LCM}(a, b)$  en anglais.

N.B.  $\text{PGCD}(0, 0)$  n'est pas défini.

**Comment calculer  $\text{PGCD}(a, b)$  ?**

Rappelons que si les décompositions en facteurs premiers des entiers  $a$  et  $b$  sont connues, on peut obtenir rapidement le  $\text{PGCD}(a, b)$  en prenant les minimums de chacun des exposants, et le  $\text{PPCM}(a, b)$  en prenant les maximums de chacun des exposants.

**Exemple 2.19**

Calculez  $\text{PGCD}(96, 28)$  et  $\text{PPCM}(96, 28)$ .

**Solution :**

Comme les décompositions en facteurs premiers des entiers 96 et 28 sont

$$96 = 2^5 \cdot 3$$

$$28 = 2^2 \cdot 7$$

on trouve que  $\text{PGCD}(96, 28) = 2^2 \cdot 3^0 \cdot 7^0 = 4$  et  $\text{PPCM}(96, 28) = 2^5 \cdot 3^1 \cdot 7^1 = 672$ .

**2.3.3 Algorithme d'Euclide et théorème de Bézout**

Quand les décompositions en facteurs premiers ne sont pas déjà connues, il est plus rapide d'utiliser l'algorithme d'Euclide pour calculer le plus grand commun diviseur.

**Algorithme 2.3 : Algorithme d'Euclide (calcul de  $\text{PGCD}(a, b)$ )**

L'algorithme d'Euclide calcule  $\text{PGCD}(a, b)$  où  $a \geq b$

$$a = bq_1 + r_2$$

$$b = r_2q_2 + r_3$$

$$r_2 = r_3q_3 + r_4$$

$$\vdots$$

$$r_{n-2} = r_{n-1}q_{n-1} + r_n \quad \text{le dernier reste non nul, } r_n, \text{ est le pgcd cherché}$$

$$r_{n-1} = r_nq_n + 0 \quad \text{Le reste est 0: fin.}$$

L'algorithme précédent repose sur le fait que

$$\text{PGCD}(a, b) = \text{PGCD}(b, r_2) = \dots = \text{PGCD}(r_{n-1}, r_n) = \text{PGCD}(r_n, 0) = r_n.$$

On peut démontrer ces égalités à l'aide du théorème de la page 69.



**Exemple 2.20**

Calculez le plus grand commun diviseur de 96 et 28 en utilisant l'algorithme d'Euclide.

**Solution :**

Nous devons diviser premièrement 96 par 28, et ensuite successivement les diviseurs par les restes, jusqu'à l'obtention d'un reste nul :

$$\begin{aligned} 96 &= 28 \cdot 3 + 12 \\ 28 &= 12 \cdot 2 + 4 \quad \text{le dernier reste non nul est 4} \\ 12 &= 4 \cdot 3 + 0 \quad \text{Le reste est 0 : fin.} \end{aligned}$$

Le dernier reste non nul est 4, donc  $\text{PGCD}(96, 28) = 4$ .

**Théorème 2.8 : Théorème de Bézout**

Si  $a, b \in \mathbb{N}^*$  alors il existe des entiers  $s$  et  $t$  tel que

$$\text{PGCD}(a, b) = sa + tb.$$

**Exemple 2.21**

Exprimez le plus grand commun diviseur de 96 et 28 comme combinaison linéaire de 96 et 28, c'est-à-dire trouvez des entiers  $s$  et  $t$  tels que

$$\text{PGCD}(96, 28) = s \cdot 96 + t \cdot 28$$

**Remarque :** le Théorème de Bézout stipule qu'il est toujours possible de trouver une telle paire d'entiers  $s$  et  $t$  peu importe les entiers choisis.

**Solution :**

Reprenons les étapes de l'algorithme d'Euclide de l'exemple 2.20 et isolons les restes dans chacune des équations :

$$\begin{aligned} 96 &= 28 \cdot 3 + 12 & \longleftrightarrow & 12 = 96 - 3 \cdot 28 & (1) \\ 28 &= 12 \cdot 2 + 4 & \longleftrightarrow & 4 = 28 - 2 \cdot 12 & (2) \\ 12 &= 4 \cdot 3 + 0 \end{aligned}$$

Ainsi,

$$\begin{aligned} \text{PGCD}(96, 28) &= 4 = 28 - 2 \cdot 12 && \text{par (2)} \\ &= 28 - 2 \cdot (96 - 3 \cdot 28) && \text{par (1)} \\ &= 28 - 2 \cdot 96 + 6 \cdot 28 && \text{en distribuant } -2 \text{ sur la parenthèse} \\ &= \underbrace{7}_{t} \cdot 28 + \underbrace{(-2)}_s \cdot 96 && \text{en additionnant } 28 \text{ à } 6 \cdot 28 \end{aligned}$$

```

numtheory\bezout(96,28)
-----
au+bv=d
u=-2, v=7, d=4
-----
{-2,7,4}

```

Figure 2.3 Obtention des coefficients  $s$  et  $t$  du théorème de Bézout avec Nspire.

La figure 2.3 montre comment utiliser la librairie (bibliothèque) `numtheory` de Nspire pour obtenir les coefficients  $s$  et  $t$  du théorème de Bézout, notés ici  $u$  et  $v$ , ainsi que le PGCD, noté  $d$ .

### Exercices

**2.11** Vrai ou faux? Justifiez.

- (a) 205 est un nombre premier.
- (b) 19 est un nombre premier.
- (c) -19 est un nombre premier.
- (d) Tout nombre pair entre 4 et 20 est la somme de deux nombres premiers.

**2.12** Calculez **à la main** le plus grand commun diviseur de  $a$  et  $b$ .

Vérifiez ensuite vos résultats avec la commande `gcd(a, b)` de la TI.

- (a)  $a = 3^5 5^4 7^3$  et  $b = 3^8 5^2 11^2$ .
- (b)  $a = 504$  et  $b = 480$ , grâce à l'algorithme d'Euclide.
- (c)  $a = 80$  et  $b = 185$ , grâce à l'algorithme d'Euclide.

**2.13** Calculez **à la main** le plus grand commun diviseur de  $a$  et  $b$  en utilisant l'algorithme d'Euclide. Exprimez ensuite le PGCD comme combinaison linéaire de  $a$  et  $b$ :

$$\text{PGCD}(a, b) = sa + tb.$$

Vérifiez vos résultats avec la commande `bezout(a, b)` du package `numtheory` de la TI.

- (a)  $a = 9$  et  $b = 4$
- (b)  $a = 121$  et  $b = 33$
- (c)  $a = 32$  et  $b = 14$
- (d)  $a = 35$  et  $b = 15$
- (e)  $a = 55$  et  $b = 81$

### 2.3.4 Inverse modulo $m$

#### Définition 2.6 : Inverse modulo $m$

Soient  $a$  et  $b$  des entiers et  $m \in \mathbb{N}^*$ . On dit que  $b$  est **l'inverse de  $a$  modulo  $m$**  si et seulement si

$$ab \equiv 1 \pmod{m}$$

#### Théorème 2.9

Si  $a$  et  $m$  sont relativement premiers, i.e.  $\text{pgcd}(a, m) = 1$ , alors l'inverse de  $a$  modulo  $m$  existe et est unique modulo  $m$ . On le note  $a^{-1}$ . Il y a donc un seul nombre entier  $a^{-1}$  entre 1 et  $m$  tel que  $aa^{-1} \equiv 1 \pmod{m}$ .

Si  $a$  et  $m$  ne sont pas relativement premiers, i.e.  $\text{pgcd}(a, m) \neq 1$ , alors  $a$  n'est *pas* inversible modulo  $m$ .

#### Exemple 2.22

Le nombre 4 est-il inversible modulo 5? Si oui, quel est son inverse?

Le nombre 3 est-il inversible modulo 9? Si oui, quel est son inverse?

Nous verrons bientôt un algorithme pour obtenir l'inverse. Pour l'instant, concentrons-nous sur la définition et cherchons l'inverse par essais et erreurs.

#### Solution :

Le nombre 4 est-il inversible modulo 9? Oui car  $\text{PGCD}(4, 9) = 1$ . Son inverse est 7.

En effet,

$$4 \cdot 7 \equiv 1 \pmod{9}.$$

Le nombre 3 est-il inversible modulo 9? Non, car  $\text{PGCD}(3, 9) = 3 \neq 1$ .

Le nombre 4 est-il inversible modulo 5? Oui, car  $\text{PGCD}(4, 5) = 1$ . Quel est son inverse? Lui-même!

$$4 \cdot 4 \equiv 1 \pmod{5}.$$

**Comment calculer l'inverse de  $a$  modulo  $m$  ?**

Si  $\text{PGCD}(a, m) = 1$ , alors  $a$  est inversible modulo  $m$ . Pour déterminer son inverse, on pourrait tester toutes les possibilités pour  $b$  allant de 1 à  $m$  jusqu'à ce que la congruence suivante soit vérifiée :

$$ab \equiv 1 \pmod{m}$$

Pour de grand  $m$ , cette façon de faire serait peu efficace. Voici une façon de procéder qui est plus efficace.

1. Exprimer 1 comme combinaison linéaire de  $a$  et  $m$  en utilisant les décompositions de l'algorithme d'Euclide (Théorème de Bézout) :

$$1 = sa + tm$$

2. Passer modulo  $m$  :

$$1 \equiv sa \pmod{m}$$

Ainsi, en arithmétique modulo  $m$ ,  $s$  est l'inverse de  $a$ .

**Exemple 2.23**

Trouvez l'inverse de 4 modulo 9.

**Solution :**

1. Exprimons  $\text{PGCD}(4, 9) = 1$  comme  $s \cdot 4 + t \cdot 9$  :

$$\begin{aligned} 9 &= 4 \cdot 2 + 1 & \longleftrightarrow & \quad 1 = 9 - 2 \cdot 4 = \overbrace{(1)}^t \cdot 9 + \overbrace{(-2)}^s \cdot 4 \\ 4 &= 1 \cdot 4 + 0 \end{aligned}$$

2. Passons modulo 9 :

$$\begin{aligned} 1 &\equiv 1 \cdot 9 + (-2) \cdot 4 \pmod{9} \\ &\equiv 0 + (-2) \cdot 4 \pmod{9} \\ &\equiv (-2) \cdot 4 \pmod{9} \\ &\equiv 7 \cdot 4 \pmod{9} \end{aligned}$$

L'unique inverse de 4 modulo 9 est donc l'entier 7.

**Exemple 2.24** (à compléter en classe)

Trouvez l'inverse de 55 modulo 81.

**Solution :**

**Exercices**

**2.14** À partir de la définition d'inverse modulo  $m$ , par essais et erreurs pour b) à e).

- (a) Vrai ou faux? L'inverse de 28 modulo 81 est 55.
- (b) Trouvez, s'il existe, l'inverse de 2 modulo 3.
- (c) Trouvez, s'il existe, l'inverse de 2 modulo 4.
- (d) Trouvez, s'ils existent, l'inverse des nombres 1 à 4 modulo 5.
- (e) Vrai ou faux? Les congruences sont toutes modulo 4.

$$ab \equiv 0 \longrightarrow (a \equiv 0 \vee b \equiv 0)$$

- (f) Résolvez la congruence  $3x \equiv 4 \pmod{5}$ .

**2.15** Calculez, s'ils existent, les inverses suivants **à la main**, en utilisant les théorèmes d'Euclide et de Bézout.

- (a) L'inverse de 15 modulo 55.                                      (d) L'inverse de 21 modulo 30.  
 (b) L'inverse de 3 modulo 13.                                      (e) L'inverse de 23 modulo 30.  
 (c) L'inverse de 11 modulo 12.

### 2.3.5 Résolution de congruence

#### Théorème 2.10 : Existence de solution d'une congruence

La congruence

$$ax \equiv b \pmod{m}$$

- possède une solution unique modulo  $m$  si  $a$  est inversible modulo  $m$ , c'est-à-dire si

$$\text{PGCD}(a, m) = 1$$

- possède plusieurs solutions modulo  $m$  si

$$\text{PGCD}(a, m) = u \neq 1 \quad \text{et} \quad u \mid b$$

L'écart entre chacune des solutions sera alors de  $\frac{m}{u}$ .

- ne possède aucune solution si

$$\text{PGCD}(a, m) = u \neq 1 \quad \text{et} \quad u \nmid b$$

#### Exemple 2.25

Pour quelles valeurs entières de  $x$  la congruence suivante est-elle satisfaite?

$$55x \equiv 2 \pmod{81}$$

#### Solution :

Pour résoudre l'équation

$$55x = 2$$

avec  $x \in \mathbb{R}$ , il suffirait de diviser chaque côté de l'égalité par 55, ou de multiplier par  $55^{-1}$ . On aurait  $x = \frac{2}{55}$ .

Pour résoudre les congruences, on ne peut pas diviser, mais on peut multiplier par l'inverse d'un nombre s'il existe. Ici, rappelons que 55 est inversible modulo 81 et que son inverse est 28, car  $55 \cdot 28 = 1540 \equiv 1 \pmod{81}$ . Ainsi,

$$\begin{aligned} 55x &\equiv 2 \pmod{81} \\ 55^{-1} \cdot (55x) &\equiv 55^{-1} \cdot 2 \pmod{81} \\ (55^{-1} \cdot 55)x &\equiv 28 \cdot 2 \pmod{81} \\ 1x &\equiv 28 \cdot 2 \pmod{81} \\ x &\equiv 56 \pmod{81} \end{aligned}$$

Les solutions sont donc tous les nombres congrus à 56 modulo 81 : ...,  $x = -106$ ,  $x = -25$ ,  $x = 56$ ,  $x = 137$ ,  $x = 218$ , ... On dit que la congruence possède une *solution unique* modulo 81.

**Exemple 2.26**

Pour quelles valeurs entières de  $x$  la congruence suivante est-elle satisfaite?

$$2x \equiv 5 \pmod{6}$$

**Solution :**

J'aimerais isoler  $x$  en divisant par 2, c'est-à-dire en multipliant par l'inverse de 2. Mais  $\text{PGCD}(2,6) = 2 \neq 1$  donc 2 n'est pas inversible modulo 6. Pour résoudre la congruence, nous pouvons toujours tester toutes les valeurs possibles pour  $x$ :

$$\begin{array}{lll} 2 \cdot 0 \equiv 0 & 2 \cdot 1 \equiv 2 & 2 \cdot 2 \equiv 4 \\ 2 \cdot 3 \equiv 0 & 2 \cdot 4 \equiv 2 & 2 \cdot 5 \equiv 4 \end{array}$$

La congruence  $2x \equiv 5 \pmod{6}$  ne possède donc *aucune solution*.

Remarque: le théorème 2.10 permet de déduire que la congruence ne possède aucune solution sans avoir à tester les possibilités. En effet, puisque

$$\text{PGCD}(2,6) = 2 \neq 1 \quad \text{et} \quad 2 \nmid 5$$

la congruence ne possède aucune solution.

**Exemple 2.27**

Pour quelles valeurs entières de  $x$  la congruence suivante est-elle satisfaite?

$$2x \equiv 4 \pmod{6}$$

**Solution :**

Ici encore, il est impossible d'isoler  $x$  en multipliant par l'inverse de 2. En testant toutes les possibilités, on constate que la congruence possède *plusieurs solutions*:

$$\begin{array}{lll} 2 \cdot 0 \equiv 0 & 2 \cdot 1 \equiv 2 & 2 \cdot 2 \equiv 4 \\ 2 \cdot 3 \equiv 0 & 2 \cdot 4 \equiv 2 & 2 \cdot 5 \equiv 4 \end{array}$$

Les solutions sont

$$x \equiv 2 \pmod{6} \quad \text{ou} \quad x \equiv 5 \pmod{6}.$$

Remarquons que l'écart entre les solutions est de 3, soit le résultat de la division de 6 par le  $\text{PGCD}(6,2)$ . Ainsi, tous les entiers congrus à 2 ou 5 modulo 6 sont solutions de la congruence  $2x \equiv 4 \pmod{6}$ :

$$x = 2, x = 5, x = 8, x = 11, x = 14, x = 17, \dots$$

$$x = -1, x = -4, x = -7, x = -10, \dots$$

Remarque: le théorème 2.10 permet de déduire que la congruence possède plusieurs solutions sans avoir à tester les possibilités. En effet, puisque

$$\text{PGCD}(2,6) = 2 \neq 1 \quad \text{et} \quad 2 \mid 4$$

la congruence possède plusieurs solutions.

### Étapes de résolution d'une congruence $ax \equiv b \pmod{m}$

1. Calculer  $u = \text{PGCD}(a, m)$ .
2. (a) **Si  $u = 1$** , alors il y a une unique solution modulo  $m$ :

$$x \equiv a^{-1}b \pmod{m}.$$

- (b) **Sinon, si  $u \neq 1$  et  $u \mid b$** , alors il y a  $u$  solutions modulo  $m$ . Pour obtenir ces solutions, on peut procéder en suivant les étapes ci-dessous ou en utilisant la technique présentée dans l'exemple 2.28.

- Résoudre la congruence

$$\left(\frac{a}{u}\right)x \equiv \left(\frac{b}{u}\right) \pmod{\left(\frac{m}{u}\right)}$$

où  $\text{PGCD}\left(\frac{a}{u}, \frac{m}{u}\right) = 1$ . On obtient une solution unique

$$x_0 \equiv \left(\frac{a}{u}\right)^{-1} \left(\frac{b}{u}\right) \pmod{\left(\frac{m}{u}\right)}.$$

(voir 2. (a))

- Regrouper en classes modulo  $m$ :

$$x_k \equiv x_0 + k \cdot \frac{m}{u} \pmod{m}, \quad \text{pour } k \in \{0, 1, \dots, u-1\}$$

- (c) **Sinon, si  $u \neq 1$  et  $u \nmid b$** , alors il n'y a aucune solution.

### Exemple 2.28

Résolvez la congruence suivante (autrement dit, trouvez toutes les solutions).

$$42x + 3 \equiv 15 \pmod{90}$$

#### Solution :

Les solutions de la congruence que l'on veut résoudre sont les mêmes que celles de la congruence suivante:

$$42x \equiv 12 \pmod{90}$$

qui est de la forme

$$ax \equiv b \pmod{m}.$$

1. Calculer  $u = \text{PGCD}(a, m) = \text{PGCD}(42, 90)$  avec l'algorithme d'Euclide.

$$90 = 42 \cdot 2 + 6 \quad \text{le dernier reste non nul est 6, donc } \text{PGCD}(90, 42) = 6$$

$$42 = 6 \cdot 7 + 0$$

2. Ainsi,  $u = 6$  et  $b = 12$ , donc  $u \mid b$ . D'après le théorème 2.10, la congruence possède donc 6 solutions distinctes comprises entre 0 et  $(m - 1)$ , donc entre 0 et 89, espacées par une distance de 15:

$$\text{écart entre les solutions} = \frac{m}{u} = \frac{90}{6} = 15.$$



Pour obtenir ces solutions, on peut procéder en suivant les étapes de l'encadré précédent, ou encore utiliser le théorème de Bézout et quelques simplifications comme voici.

Le théorème de Bézout permet d'exprimer  $\text{pgcd}(a, m)$ , noté  $u$ , comme combinaison linéaire de  $a$  et  $m$ :

$$\begin{aligned}u &= sa + tm \\6 &= (-2) \cdot 42 + 1 \cdot 90.\end{aligned}$$

Ainsi, quand on passe en **congruence modulo**  $m$ , on voit que  $u$  est un multiple de  $a$ :

$$\begin{aligned}u &\equiv sa \pmod{m} \quad (\text{car } m \bmod m = 0) \\6 &\equiv (-2) \cdot 42 \pmod{90}.\end{aligned}$$

Or, puisque  $u|b$ , on peut écrire  $b$  comme un multiple de  $u$ : disons  $b = cu$ .

De retour à la congruence, ceci donne:

$$\begin{aligned}ax &\equiv b && \text{toutes les congruences sont modulo } m \\ \rightarrow ax &\equiv cu && \text{car } b = cu \\ \rightarrow ax &\equiv c(sa) && \text{car } u \equiv sa \\ \rightarrow ax &\equiv (cs)a && \text{pas associativité de la multiplication.}\end{aligned}$$

Donc  $x_0 \equiv cs$  sera l'une des solutions de la congruence. Dans notre exemple, ceci donne donc:

$$\begin{aligned}42x &\equiv 12 && \text{toutes les congruences sont modulo } 90 \\ \rightarrow 42x &\equiv 2 \cdot 6 \\ \rightarrow 42x &\equiv 2(-2) \cdot 42 && \text{car } 6 = (-2) \cdot 42 \\ \rightarrow x_0 &\equiv -4 && \text{est une solution de la congruence.} \\ \rightarrow x_0 &= 86 && \text{est une solution de la congruence.}\end{aligned}$$

Pour trouver les autres solutions, on ajoute ou retranche des bonds de longueur 15 à  $x_0$ .

$$\begin{aligned}x_0 &= 86 \\ x_1 &= (x_0 + 15) \bmod 90 = (101) \bmod 90 = 11 \\ x_2 &= (x_1 + 15) \bmod 90 = 26 \\ x_3 &= (x_2 + 15) \bmod 90 = 41 \\ x_4 &= (x_3 + 15) \bmod 90 = 56 \\ x_5 &= (x_4 + 15) \bmod 90 = 71 \\ x_6 &= (x_5 + 15) \bmod 90 = 86 = x_0\end{aligned}$$


---

**Remarque.** Dans l'exemple précédent, puisque nous avons travaillé en parallèle avec les paramètres  $a$ ,  $b$ ,  $m$  et  $u$  et des valeurs numériques particulières, nous avons en fait démontré une partie du théorème 2.10 (2<sup>e</sup> cas) :

lorsque  $u|b$ , la congruence  $ax \equiv b$  possède au moins une solution :  $x_0 = cs$ .

Pour compléter la preuve du théorème 2.10 (2<sup>e</sup> cas), il reste à démontrer que lorsqu'on ajoute un multiple de  $\frac{m}{u}$  à la solution  $x_0$ , la nouvelle valeur satisfait encore la congruence. Toutes les congruences suivantes sont modulo  $m$  :

$$\begin{aligned} ax_0 \equiv b &\rightarrow a\left(x_0 + k \cdot \frac{m}{u}\right) \equiv ax_0 + \frac{akm}{u} \quad \text{où } k \in \mathbb{Z} \\ &\rightarrow a\left(x_0 + k \cdot \frac{m}{u}\right) \equiv b + \frac{akm}{u} \quad \text{car } ax_0 \equiv b \\ &\rightarrow a\left(x_0 + k \cdot \frac{m}{u}\right) \equiv b + \frac{a}{u} \cdot km \\ &\rightarrow a\left(x_0 + k \cdot \frac{m}{u}\right) \equiv b + d \cdot km \quad \text{car } a \text{ est un multiple de } u, \text{ disons } a = du \\ &\rightarrow a\left(x_0 + k \cdot \frac{m}{u}\right) \equiv b + d \cdot k \cdot 0 \quad \text{car } m \equiv 0 \pmod{m} \\ &\rightarrow a\left(x_0 + k \cdot \frac{m}{u}\right) \equiv b \end{aligned}$$

Ainsi, lorsqu'on ajoute un multiple de  $\frac{m}{u}$  à la solution  $x_0$ , la nouvelle valeur satisfait encore la congruence. Ceci complète la preuve du théorème 2.10 (2<sup>e</sup> cas). Les autres cas seront traités en classe.

### Exercices

**2.16** Résolvez les congruences suivantes (trouvez toutes les solutions).

- |                                 |                                  |
|---------------------------------|----------------------------------|
| (a) $15x \equiv 6 \pmod{28}$    | (e) $4x - 1 \equiv 5 \pmod{10}$  |
| (b) $13x \equiv 3 \pmod{20}$    | (f) $15x + 2 \equiv 8 \pmod{25}$ |
| (c) $9x \equiv 7 \pmod{12}$     | (g) $15x \equiv 31 \pmod{24}$    |
| (d) $9x + 2 \equiv 8 \pmod{12}$ | (h) $42x \equiv 18 \pmod{90}$    |

### 2.3.6 Petit théorème de Fermat

Le théorème suivant sera utilisé lorsque nous verrons comment déchiffrer un message qui a été chiffré par le système cryptographique RSA.

#### Théorème 2.11 : Petit théorème de Fermat

Soit  $p$  un nombre premier. Si  $a$  est un entier qui n'est pas divisible par  $p$ , alors

$$a^{p-1} \equiv 1 \pmod{p}.$$

De plus, quel que soit l'entier  $a$ ,

$$a^p \equiv a \pmod{p}.$$

## 2.4 Cryptographie à clé secrète

La cryptographie est l'étude des techniques d'envoi de messages chiffrés. Cette section présente trois systèmes cryptographiques qui reposent sur les notions vues dans les sections précédentes.

Pour cette section, on considère des systèmes cryptographiques pour lesquels Alice et Bob partagent un *secret*.

### 2.4.1 Chiffrement par décalage

Ce système fonctionne par un décalage des lettres de l'alphabet. Il peut être défini en utilisant des opérations sur les entiers de la manière suivante. On assigne premièrement un entier à chaque lettre de l'alphabet:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Chaque lettre  $m$  du message clair (message original) est ensuite chiffrée en utilisant la fonction  $f$ . Pour déchiffrer le message, on utilise la réciproque  $f^{-1}$  sur chacune des lettres chiffrées  $c$ .

- **Clé** (secrète):  $k \in \{1, 2, \dots, 25\}$
- **Chiffrement** (d'une lettre  $m$ )

$$f: \{0, 1, 2, \dots, 25\} \rightarrow \{0, 1, 2, \dots, 25\}$$

$$f(m) = (m + k) \bmod 26$$

- **Déchiffrement** (d'une lettre  $c$ )

$$f^{-1}: \{0, 1, 2, \dots, 25\} \rightarrow \{0, 1, 2, \dots, 25\}$$

$$f^{-1}(c) = (c - k) \bmod 26$$

Lorsque l'on connaît la clé de chiffrement, il est rapide de décoder le message. Le chiffrement par décalage est un système que l'on peut casser facilement. Il suffit de tester les 26 décalages possibles ou encore d'analyser la fréquence des lettres, certaines lettres apparaissant plus souvent dans la langue française (par exemple la lettre E apparaît plus souvent que la lettre X).

#### Exemple 2.29 (à compléter en classe)

Jules César utilisait dans les correspondances avec ses généraux un chiffrement par décalage de 3 vers la droite ( $k = 3$ ). Déchiffrez le message suivant envoyé à César:

DVWHULA HW REHOLA.

### 2.4.2 Permutation de l'alphabet

Ce système fonctionne de la même manière que le précédent à la différence qu'au lieu d'appliquer un simple *décalage*, on applique une des  $26!$  permutations aux lettres de l'alphabet. Le nombre de permutations étant très élevé ( $26! \approx 4 \cdot 10^{26}$ ), l'attaque *brute force* consistant à énumérer toutes les clés est impossible. Cela dit, une simple analyse statistique de la fréquence des lettres permet d'identifier la permutation.

### 2.4.3 Masque jetable

- **Clé:**  $k$ , une suite de lettres aléatoires.
- **Chiffrement:** chaque lettre  $M[i]$  du message est remplacée par

$$C[i] = (M[i] + k[i]) \bmod 26.$$

- **Déchiffrement:** chaque lettre  $C[i]$  du message chiffré est remplacée par

$$M[i] = (C[i] - k[i]) \bmod 26.$$

**Avantage:** on peut garantir que les messages ainsi chiffrés sont indéchiffrables pour quelqu'un qui ne connaît pas la clé.

**Inconvénient:** la clé doit être au moins aussi longue que le message. De plus, la clé ne doit être utilisée qu'une seule fois, car sinon on s'expose à des attaques statistiques.

### 2.4.4 Chiffre affine

Ce système est une généralisation du chiffrement par décalage. Il utilise les notions d'inverse modulo 26 et de résolution de congruence. Soit  $m$  une lettre du message clair et  $c$  une lettre du message chiffré.

- **Clé (secrète):**  $(a, b)$ , où  $a, b \in \mathbb{Z}$  et  $\text{pgcd}(a, 26) = 1$  (dans ce cas,  $a$  est inversible modulo 26).
- **Chiffrement** (d'une lettre  $m$ )

$$f : \{0, 1, 2, \dots, 25\} \rightarrow \{0, 1, 2, \dots, 25\}$$

$$f(m) = am + b \bmod 26$$

- **Déchiffrement** (d'une lettre  $c$ )

$$f^{-1} : \{0, 1, 2, \dots, 25\} \rightarrow \{0, 1, 2, \dots, 25\}$$

$$f^{-1}(c) = a^{-1}(c - b) \bmod 26$$

Rappel:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

**Exemple 2.30** (à compléter en classe)

Déchiffrez le message TCR si le chiffrement est affine, avec la clé (3, 4).

**Exercices**

**2.17** Dites si chacune des fonctions suivantes est une fonction de chiffrement affine. Si c'est le cas, trouvez la fonction de déchiffrement.

- (a)  $f(m) = 21m + 10 \pmod{26}$
- (b)  $f(m) = 9m + 21 \pmod{26}$
- (c)  $f(m) = 2m + 3 \pmod{26}$
- (d)  $f(m) = 11m + 12 \pmod{26}$

## 2.5 Cryptographie à clé publique

On considère des systèmes de cryptographie qui ne supposent pas qu'Alice et Bob partagent de secret. Ainsi, toutes les opérations que Bob fera pour chiffrer son message sont connues de tous.

### 2.5.1 Chiffre RSA

Le système RSA (de Ronald Rivest, Adi Shamir, Leonard Adleman) est basé sur l'exponentiation modulaire modulo le produit de deux très grands nombres premiers. Tout ce système repose sur le fait qu'il est très difficile d'obtenir la décomposition en nombres premiers de grands nombres (utilisée pour décrypter).

Le fait de savoir comment chiffrer un message n'implique pas qu'on puisse le déchiffrer. La clé de déchiffrement est maintenue secrète et seul le destinataire du message peut le déchiffrer. Sans clé, le déchiffrement exige une énorme quantité de travail.

Ce système utilise les notions d'exponentiation modulaire, de décomposition en nombres premiers, du calcul d'inverse modulo  $m$  et la preuve de son fonctionnement repose sur le Petit théorème de Fermat.

**Chiffrement:** Alice veut envoyer un message à Bob. Pour ce faire, elle suit les étapes suivantes.

1. Alice regarde le registre et trouve la **clé publique** de Bob composée de deux entiers:  $n$  et  $e$ .

N.B. Bob n'a pas choisi sa clé au hasard! Le nombre  $n$  est le produit de 2 grands nombres premiers  $p$  et  $q$  que Bob ne divulgue pas,  $n = pq$ , et le nombre  $e$  est choisi de sorte qu'il soit **inversible** modulo  $s$ , où  $s$  est le produit des nombres  $(p - 1)$  et  $(q - 1)$ . Ainsi, par le théorème 2.9, il faut que:

$$\text{pgcd}(e, s) = 1 \quad \text{où} \quad s = (p - 1)(q - 1).$$

2. Alice traduit chaque lettre de son message en entiers en respectant une convention préétablie. Elle obtient ainsi une chaîne de chiffres, notée  $M$ .

N.B. Dans le cadre de notre cours, il n'y a que 26 symboles; pas d'espace ni autres caractères: (A:00, B:01, C:02, ..., Z:25).

3. Alice découpe sa chaîne  $M$  en blocs, en ajoutant des lettres au besoin pour compléter le dernier bloc:

$$M = M_1 M_2 M_3 \dots$$

N.B. Dans le cadre de notre cours, la longueur des blocs sera indiquée. En général, la longueur des blocs est fixée ainsi:  $2N$  chiffres, où  $2N$  est le plus grand entier pair tel que  $\underbrace{2525\dots25}_{2N \text{ chiffres}} < n$ .

4. Pour chacun des blocs  $M_i$ , Alice calcule  $C_i$  (en programmant l'algorithme d'exponentiation modulaire qu'elle a appris dans son cours de Mathématiques discrètes):

$$C_i = (M_i^e \bmod n).$$

5. Alice utilise un canal public pour envoyer le message  $C$  à Bob. Ce message  $C$  est la concaténation des blocs  $C_i$ :

$$C = C_1 C_2 C_3 \dots$$

Seul Bob (ou un malin mathématicien fort en factorisation) pourra déchiffrer  $C$  en utilisant sa clé de déchiffrement.

**Déchiffrement:** Bob reçoit le message  $C$  de Alice. Pour le décoder, il utilise sa clé de déchiffrement  $d$  en suivant les étapes suivantes.

N.B. Bob a obtenu sa clé de déchiffrement  $d$  (**clé privée**) en calculant l'inverse modulo  $s$  de sa clé de chiffrement  $e$  (en utilisant l'algorithme d'Euclide et le théorème de Bézout):

$$d = (e^{-1} \bmod s), \quad \text{où} \quad s = (p - 1)(q - 1).$$

1. Bob divise le message  $C$  en blocs:

$$C = C_1 C_2 C_3 \dots$$

N.B. Dans le cadre de notre cours, la longueur des blocs sera indiquée.

2. Pour chacun des blocs  $C_i$ , Bob calcule  $M_i$  ainsi:

$$M_i = (C_i^d \bmod n).$$

3. Bob traduit la suite de chiffres en lettres en suivant la convention (dictionnaire).

Le système RSA repose sur l'égalité suivante, que nous pouvons démontrer avec le petit théorème de Fermat et quelques astuces.

$$\begin{aligned} C_i^d \bmod n &= (M_i^e \bmod n)^d \bmod n && \text{par construction des blocs} \\ &= M_i && \text{égalité à démontrer.} \end{aligned}$$

Il sera plus simple de travailler avec les congruences modulo  $n$  :

$$\begin{aligned} C_i^d &\equiv (M_i^e)^d && \text{par construction des blocs} \\ &\equiv (M_i^{ed}) && \text{par la loi des exposants} \\ &\equiv M_i && \text{congruence à démontrer.} \end{aligned}$$

### Exercices

**2.18** Quand le nombre  $n$  n'est pas très grand et qu'on arrive à le factoriser, il est alors possible pour un espion de retrouver la clé de déchiffrement  $d$  à partir de la clé publique  $(n, e)$ . L'espion peut ensuite déchiffrer le message  $C$ . C'est ce que vous devez faire ici.

- Déchiffrez le message 0309 5434, sachant qu'il a été chiffré de la façon suivante: tableau de lettres de la page 99, découpage en blocs de 4 chiffres, RSA avec la clé publique suivante:  $n = 8051$  et  $e = 3125$ .
- Déchiffrez le message 6953 7519, sachant qu'il a été chiffré de la façon suivante: tableau de lettres de la page 99, découpage en blocs de 4 chiffres, RSA avec la clé publique suivante:  $n = 8633$  et  $e = 3125$ .

### 2.19

- Chiffrez le message STOP avec la clé (2537, 13).
- Chiffrez le message STOP avec la clé (245363, 323).
- Déchiffrez le message 2081 2182 sachant qu'il a été chiffré avec la clé (2537, 13).
- Déchiffrez le message 239815 087533 sachant qu'il a été chiffré avec la clé (245363, 323).

**2.20** Alice veut envoyer un message à Bob en utilisant le système RSA. Elle regarde dans le bottin des clés publiques et trouve celle de Bob :

$$(n, e) = (73813, 13).$$

Elle chiffre son message en utilisant cette clé et envoie ainsi 61880 22967 à Bob.

- Quelle est la clé  $(n, d)$  que doit utiliser Bob pour déchiffrer le message?
- Quel est le message d'origine envoyé par Alice?





## Chapitre 3

# Représentation des entiers et manipulations bit à bit

### 3.1 Représentation des entiers à taille fixe

Dans un ordinateur, les entiers sont représentés à l'aide de trains de bits. La manière la plus courante consiste à fixer une taille  $n$  et ensuite, chaque entier est représenté par un train de bits de taille  $n$ . Pour des raisons pratiques, il est courant d'utiliser  $n = 16, 32$ , ou  $64$ , mais d'autres valeurs sont aussi possibles.

Lorsque des entiers sont représentés par des trains de bits à taille fixe, il est primordial de distinguer deux cas :

- **les entiers non signés** : les nombres représentés sont positifs ou nuls,
- **les entiers signés** : les nombres représentés sont positifs, négatifs ou nuls.

La table 3.1 regroupe les différents symboles qui seront définis tout au long de ce chapitre.

#### 3.1.1 Entiers non signés à taille fixe

Un entier  $x$  est représenté par un train de bits de longueur  $n$  où chaque bit représente un chiffre de l'écriture binaire de  $x$ .

##### Définition 3.1

La **représentation binaire non signée** sur  $n$  bits d'un entier  $x \in \{0, 1, \dots, 2^n - 1\}$ , est le train de bits  $t \in T_n$  correspondant à l'écriture de  $x$  en base 2.

On écrit alors :  $\text{Rep}_n(x) = t$  et  $(t)_2 = x$ .

Symbole	Signification	Définition	Exemple
$T_n$	Ensemble des trains de bits de longueur $n$ .	$T_n = \{000 \dots 000, \dots, 111 \dots 111\}$	$0110, 1010 \in T_4$
$T_*$	Ensemble des trains de bits de toutes les longueurs.	$T_* = \bigcup_{i \geq 0} T_i$	$0, 10, 1110 \in T_*$
$(\cdot)_2$	Entier non signé représenté en base 2.	$(\cdot)_2 : T_* \rightarrow \mathbb{N}$	$(1101)_2 = (13)_{10}$
$(\cdot)_{\pm 2}$	Entier signé représenté par la méthode du complément à deux.	$(\cdot)_{\pm 2} : T_* \rightarrow \mathbb{Z}$	$(1101)_{\pm 2} = (-3)_{10}$
$\text{Rep}_n$	Représentation binaire non signée de taille $n$ .	$\text{Rep}_n : \{0, 1, \dots, 2^n - 1\} \rightarrow T_n$	$\text{Rep}_4(13) = 1101$
$\text{Rep}_{\pm n}$	Représentation binaire signée de taille $n$ .	$\text{Rep}_{\pm n} : \{-2^{n-1}, \dots, 2^{n-1} - 1\} \rightarrow T_n$	$\text{Rep}_{\pm 4}(-3) = 1101$
$\wedge$	Opérateur « <b>et</b> » appliqué bit à bit.	$\wedge : T_n \times T_n \rightarrow T_n$	$0011 \wedge 0110 = 0010$
$\vee$	Opérateur « <b>ou</b> » appliqué bit à bit.	$\vee : T_n \times T_n \rightarrow T_n$	$0011 \vee 0110 = 0111$
$\oplus$	Opérateur « <b>ou exclusif</b> » appliqué bit à bit.	$\oplus : T_n \times T_n \rightarrow T_n$	$0011 \oplus 0110 = 0101$
$\sim$	Opérateur de <b>négation</b> appliqué bit à bit.	$\sim : T_n \rightarrow T_n$	$\sim 0011 = 1100$
$\ll$	Opérateur de <b>décalage à gauche</b> .	$\ll : T_n \rightarrow T_n$	$0011 \ll = 0110$
$\gg$	Opérateur de <b>décalage à droite</b> .	$\gg : T_n \rightarrow T_n$	$0011 \gg = 0001$

Tableau 3.1 Récapitulatif des symboles utilisés. Ceux-ci sont introduits tout au long du chapitre.



**Exemple 3.2**

Si les entiers  $x = 9$  et  $y = 13$  sont représentés par des trains de bits à taille fixe sur 4 bits, que vaut  $x + y$ ?

**Solution :**

Les représentations sur 4 bits de  $x$  et  $y$  sont  $\text{Rep}_4(x) = 1001$  et  $\text{Rep}_4(y) = 1101$ .

$$\begin{array}{r}
 1001 \\
 + 1101 \\
 \hline
 1\ 0110 \\
 \uparrow \\
 \boxed{\begin{array}{c} \text{Valeur} \\ \text{perdue} \end{array}}
 \end{array}$$

Comme  $(0110)_2 = 6$ , on conclut que  $x + y = 6$ , ce qui ne correspond pas à l'addition dans  $\mathbb{N}$  :  $9 + 13 = 22$ . Cette différence est due au fait qu'il y a eu un dépassement d'entier. On aurait pu obtenir le même résultat plus rapidement en utilisant le modulo :

$$(9 + 13) \bmod 2^4 = 22 \bmod 16 = 6.$$

**Exemple 3.3**

Déterminez ce qui est affiché à l'exécution de ce programme écrit dans le langage C. Le type `uint8_t` désigne des entiers non signés représentés sur 8 bits.

```

#include <stdio.h>
#include <stdint.h>
int main() {
    uint8_t x, y, z;
    x = 100;
    y = 200;
    z = x+y;
    printf("%u + %u = %u\n", x, y, z);
}

```

**Solution :**

L'arithmétique avec des entiers non signés représentés sur 8 bits est équivalente à l'arithmétique modulo  $2^8 = 256$ . Ainsi, la valeur de  $z$  est

$$z = (100 + 200) \bmod 256 = 300 \bmod 256 = 44.$$

Le programme affiche donc :

```
100 + 200 = 44
```

Ce résultat est confirmé lorsqu'on effectue l'addition bit à bit sur 8 bits. On a :

- $\text{Rep}_8(100) = 0110\ 0100$ ,
- $\text{Rep}_8(200) = 1100\ 1000$ .

L'addition est donc :

$$\begin{array}{r}
 0110\ 0100 \\
 + \quad 1100\ 1000 \\
 \hline
 \mathbf{1}\ 0010\ 1100 \\
 \uparrow \\
 \boxed{\begin{array}{l} \text{Valeur} \\ \text{perdue} \end{array}}
 \end{array}$$

Il y a un dépassement d'entier et la valeur de  $z$  est  $(0010\ 1100)_2 = 44$ .

### 3.1.2 Entiers signés à taille fixe

La base 2 permet de représenter les nombres entiers positifs d'une manière naturelle à l'aide de trains de bits. Par contre, *comment représenter les nombres entiers négatifs?* Bien qu'il y ait plusieurs façons de faire, pratiquement tous les systèmes informatiques représentent les nombres entiers négatifs par la méthode du **complément à deux**. Cette méthode exploite le principe du dépassement d'entier de manière à représenter les entiers positifs et négatifs sur un nombre fixe de bits en respectant les prémisses suivantes :

**Prémisse 1.** Le bit le plus à gauche indique le signe : 0 si positif ou nul, 1 si négatif.

**Prémisse 2.** La représentation des nombres positifs ou nuls est la même que dans le cas non signé.

**Prémisse 3.** La représentation des nombres négatifs est telle que si  $x$  est positif et que  $x + y = 0$  alors  $y = -x$ .

La troisième prémisse est particulièrement importante, car elle impose la manière dont les nombres négatifs sont représentés.

#### Exemple 3.4

Utilisez la troisième prémisse afin de déterminer la représentation sur 8 bits de  $-1$ .

#### Solution :

On cherche un train de bits  $t$  tel que

$$\begin{array}{r}
 \phantom{+} \phantom{0000} \phantom{0001} \phantom{0000} \\
 + \quad \phantom{0000} \phantom{0001} \phantom{0000} \phantom{0000} \\
 \hline
 \phantom{0000} \phantom{0001} \phantom{0000} \phantom{0000}
 \end{array}$$

Il est évident que le dernier bit de  $t$  doit être 1 afin d'obtenir un 0 à la dernière position de la somme. L'addition des deux derniers bits sera  $1 + 1 = 2$  ou, en binaire,  $1 + 1 = 10$ . Il y a aura donc une retenue au niveau de l'avant-dernier bit. Conséquemment, l'avant-dernier bit de  $t$  doit être également 1. En répétant ce raisonnement sur tous les bits de  $t$ , on conclut que  $t = 1111\ 1111$ . L'addition est donc :

$$\begin{array}{r}
 1111\ 1111 \\
 + \quad 0000\ 0001 \\
 \hline
 \mathbf{1}\ 0000\ 0000 \\
 \uparrow \\
 \boxed{\begin{array}{l} \text{Valeur} \\ \text{perdue} \end{array}}
 \end{array}$$

Ainsi, le train de bits  $1111\ 1111$  représente l'entier  $-1$ .

La définition qui suit présente la méthode du complément à deux de manière formelle. Par abus de notation, étant donné un train de bits  $t \in T_n$ , on se permet d'écrire  $t + 1$  pour représenter l'addition :

$$t + \underbrace{000 \cdots 01}_{n \text{ bits}}$$

De plus, on note  $\sim t$  le train de bits obtenu en inversant la valeur tous les bits de  $t$ . Par exemple,  $\sim 0110 = 1001$ .

### Définition 3.2 : Complément à deux sur $n$ bits

Soit  $x \in \{-2^{n-1}, \dots, -1, 0, 1, \dots, 2^{n-1} - 1\}$ , et  $t = \text{Rep}_n(|x|)$ . La **représentation binaire signée** de  $x$  sur  $n$  bits par la méthode du **complément à deux** est :

$$\text{Rep}_{\pm n}(x) = \begin{cases} t & \text{si } x \geq 0, \\ \sim t + 1 & \text{si } x < 0. \end{cases}$$

Inversement, la valeur du train de bits  $u = \text{Rep}_{\pm n}(x)$  est notée  $(u)_{\pm 2} = x$ .

La manière dont les entiers négatifs sont représentés est une conséquence de la troisième prémisse de la méthode du complément à deux. En effet, soit  $t \in T_n$ , si on additionne  $t$  et  $\sim t$ , on obtient un train de bits dont tous les bits valent 1 :

$$t + (\sim t) = 111 \cdots 1.$$

Ceci est dû au fait que lorsqu'on additionne les bits de  $t$  et de  $\sim t$ , on additionne toujours un bit à 1 avec un bit à 0 ou un bit à 0 avec un bit à 1. Dans les deux cas, le résultat est un bit à 1. Il est évident que lorsqu'on additionne 1 à un tel train de bits, il y a automatiquement un dépassement d'entier et le résultat est 0. La troisième prémisse impose donc que  $t + \sim t = -1$ . Ainsi,

$$\begin{aligned} -1 &= t + (\sim t), \\ -t - 1 &= \sim t, \\ -t &= \sim t + 1. \end{aligned}$$

Au-delà de la définition de la méthode du complément à deux, cette observation fournit une méthode systématique pour inverser le signe d'un entier.

**Inverser le signe d'un entier représenté par le complément à deux**

Soit  $t$  la représentation sur  $n$  bits de l'entier  $x$  par la méthode du complément à deux. La représentation de l'entier  $-x$  est:

$$\text{Rep}_{\pm n}(-x) = \sim t + 1.$$

On revient sur l'Exemple 3.4. On a vu que sur 8 bits, 1 est représenté par 0000 0001 et  $-1$  par 1111 1111.

Ceci est cohérent avec la définition, car:

$$\text{Rep}_{\pm 8}(-1) = \sim \text{Rep}_{\pm 8}(1) + 1 = \sim 0000\ 0001 + 1 = 1111\ 1110 + 0000\ 0001 = 1111\ 1111.$$

De plus, si on inverse le signe de la représentation de  $-1$ , on obtient la représentation de 1,

$$\text{Rep}_{\pm 8}(1) = \sim \text{Rep}_{\pm 8}(-1) + 1 = \sim 1111\ 1111 + 1 = 0000\ 0000 + 0000\ 0001 = 0000\ 0001.$$

**Exemple 3.5**

Déterminez la représentation binaire signée sur 8 bits de 108 et  $-108$ . De plus, vérifiez que l'addition des deux trains de bits obtenus donne bien zéro.

**Solution :**

1. Comme 108 est positif, sa représentation binaire signée est la même que dans le cas non signé.

$$\text{Rep}_{\pm 8}(108) = \text{Rep}_8(108) = 0110\ 1100.$$

2. La représentation de  $-108$  est obtenue en inversant le signe de celle de 108.

$$\begin{aligned} \text{Rep}_{\pm 8}(-108) &= \sim \text{Rep}_{\pm 8}(108) + 1 \\ &= \sim 0110\ 1100 + 0000\ 0001 \\ &= 1001\ 0011 + 0000\ 0001 \\ &= 1001\ 0100. \end{aligned}$$

Finalement, on vérifie que l'addition des deux trains de bits donne bien 0.

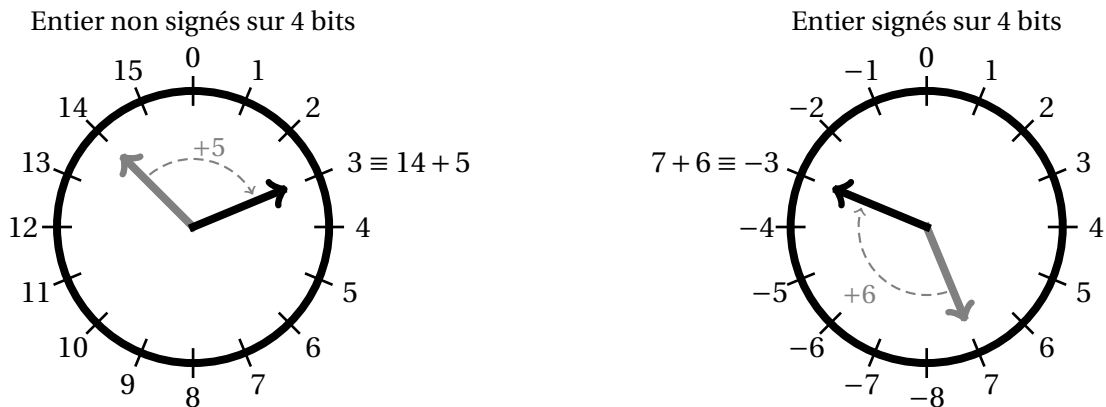
$$\begin{array}{r} 0110\ 1100 \\ + 1001\ 0100 \\ \hline \mathbf{1}\ 0000\ 0000 \\ \uparrow \\ \text{Valeur} \\ \text{perdue} \end{array}$$

**Exemple 3.6**

La table qui suit donne la valeur de l'entier signé et non signé représenté par chacun des 16 trains de bits de longueur 4.

Train de bits	Valeur non signée	Valeur signée	Train de bits	Valeur non signée	Valeur signée
0000	0	0	1000	8	-8
0001	1	1	1001	9	-7
0010	2	2	1010	10	-6
0011	3	3	1011	11	-5
0100	4	4	1100	12	-4
0101	5	5	1101	13	-3
0110	6	6	1110	14	-2
0111	7	7	1111	15	-1

On utilise l'analogie voulant que faire des calculs modulo 16 soit équivalent à considérer les nombres sur un cadran circulaire gradué de 0 à 15. La représentation des nombres négatifs par le complément à deux consiste à faire exactement la même chose, mais on *décide* que la moitié des entiers sont considérés comme étant des nombres négatifs.



La figure ci-dessus illustre des dépassements d'entier lors de l'addition d'entiers non signés (à gauche) et signés (à droite) tous représentés sur 4 bits.

- À gauche, on additionne les entiers non signés 14 et 5. Il y a un dépassement d'entier et le résultat est  $19 \bmod 2^4 = 3$ .
- À droite, on additionne les entiers signés 7 et 6. Il y a également un dépassement d'entier et le résultat de l'addition se retrouve du côté négatif. On remarque que la valeur obtenue,  $-3$ , est cohérente avec la congruence modulo  $2^4$ :

$$7 + 6 = 13 \equiv -3 \bmod 2^4.$$

**Exemple 3.7**

Déterminez la valeur représentée par chacun des trains de bits suivants avec la méthode du complément à deux.

- 0101
- 1101 0100
- 1001 1100 0011 0110



**Solution :**

(a) 0101

Le premier bit étant 0, il s'agit d'un nombre positif. Le train de bit correspond donc à l'écriture en base 2.

$$(0101)_{\pm 2} = 2^2 + 2^0 = 5.$$

(b) 1101 0100

Le premier bit étant 1, il s'agit d'un nombre négatif. Il y a deux manières de procéder.

**Calculer la valeur absolue.** On sait que si on inverse le signe d'un nombre négatif, on obtient sa valeur absolue. Ainsi, on utilise le fait que  $-t = \sim t + 1$  pour calculer la valeur absolue du nombre négatif représenté par 1101 0100.

$$\sim(1101\ 0100) + 1 = 0010\ 1011 + 0000\ 0001 = 0010\ 1100$$

On calcule maintenant le nombre positif représenté par 0010 1100.

$$(0010\ 1100)_{\pm 2} = 2^5 + 2^3 + 2^2 = 44.$$

Comme  $(0010\ 1100)_{\pm 2} = 44$ , on conclut que  $(1101\ 0100)_{\pm 2} = -44$ .

**Utiliser la congruence modulo  $2^n$ .** On considère l'entier non signé représenté par le train de bits 1101 0100.

$$(1101\ 0100)_2 = 2^7 + 2^6 + 2^4 + 2^2 = 212.$$

On sait que l'entier signé représenté par 1101 0100 est négatif et congru à 212 modulo  $2^8 = 256$ . Pour être plus précis, il s'agit du plus petit entier, en valeur absolue, qui est négatif et congru à 212 modulo 256. Pour obtenir ce nombre, il suffit de soustraire 256 à 212.

$$\begin{aligned} (1101\ 0100)_{\pm 2} &= (1101\ 0100)_2 - 256 \\ &= 212 - 256 \\ &= -44. \end{aligned}$$

(c) 1001 1100 0011 0110

Il s'agit encore une fois d'un nombre négatif, car le premier bit est 1. On présente encore une fois les deux manières de calculer la valeur représentée.

**Calculer la valeur absolue.**

1. Inverser le signe.

$$\begin{aligned} \sim 1001\ 1100\ 0011\ 0110 + 1 &= 0110\ 0011\ 1100\ 1001 + 1 \\ &= 0110\ 0011\ 1100\ 1010 \end{aligned}$$

2. Évaluer l'entier positif.

$$(0110\ 0011\ 1100\ 1010)_{\pm 2} = 2^{14} + 2^{13} + 2^9 + 2^8 + 2^7 + 2^6 + 2^3 + 2^1 = 25546.$$

3. L'entier représenté est donc

$$(1001\ 1100\ 0011\ 0101)_{\pm 2} = -25546.$$





### 3.2 Opérations bit à bit

Certaines opérations binaires (sur 0 et 1) effectuées par les ordinateurs correspondent exactement à celles des connecteurs logiques  $\wedge$ ,  $\vee$  et  $\oplus$ , comme l'indique le tableau suivant :

$x$	$y$	$x \wedge y$	$x \vee y$	$x \oplus y$
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

Afin d'accélérer le traitement des données, les ordinateurs modernes ne manipulent jamais un seul bit à la fois. La majorité des ordinateurs supportent uniquement la manipulation de trains de bits de taille 8, 16, 32 ou 64. Les opérateurs binaires  $\wedge$ ,  $\vee$  et  $\oplus$  sont alors effectués simultanément sur chacun des bits, le premier avec le premier, le deuxième avec le deuxième et ainsi de suite, d'où le nom *bit à bit*.

#### Exemple 3.8

Illustrons les opérateurs logiques  $\wedge$ ,  $\vee$  et  $\oplus$  employés bit à bit.

$$\begin{array}{r}
 1000\ 0101 \\
 \wedge\ 1110\ 1100 \\
 \hline
 1000\ 0100
 \end{array}
 \qquad
 \begin{array}{r}
 1000\ 0101 \\
 \vee\ 1110\ 1100 \\
 \hline
 1110\ 1101
 \end{array}
 \qquad
 \begin{array}{r}
 1000\ 0101 \\
 \oplus\ 1110\ 1100 \\
 \hline
 0110\ 1001
 \end{array}$$

D'autres opérateurs bit à bit s'appliquent à un seul train de bits. On les appelle les *opérateurs unaires*. En voici trois :

- **Négation**, notée  $\sim$  en préfixe, inverse la valeur de chacun des bits.
- **Décalage à gauche**, noté  $\ll$  en suffixe, supprime le bit le plus à gauche et ajoute un zéro à droite.
- **Décalage à droite**, noté  $\gg$  en suffixe, supprime le bit le plus à droite et ajoute un bit à gauche. Pour  $\gg$ , la valeur du bit ajouté dépend du train de bit :
  - s'il s'agit d'un entier non signé, alors un 0 est ajouté à gauche.
  - s'il s'agit d'un entier signé positif ou nul, alors un 0 est ajouté à gauche.
  - s'il s'agit d'un entier signé négatif, alors un 1 est ajouté à gauche.

Il y a une bonne raison pour que l'opérateur  $\gg$  agisse différemment sur les nombres négatifs. En effet, en procédant de la sorte, les opérateurs de décalage à gauche et à droite ont l'interprétation arithmétique suivante :

- L'opérateur de **décalage à gauche** ( $\ll$ ) **double la valeur** de l'entier, à condition qu'il n'y ait pas de dépassement d'entier.
- L'opérateur de **décalage à droite** ( $\gg$ ) **divise par deux** la valeur de l'entier, arrondie à l'entier inférieur.

Par abus de notation, lorsqu'il n'y a pas d'ambiguïté sur le type de représentation, on se permet d'appliquer les opérateurs bit à bit directement sur les entiers.

**Exemple 3.9**

Illustration des opérateurs bit à bit négation ( $\sim$ ), décalage à gauche ( $\ll$ ) et décalage à droite ( $\gg$ ) avec des entiers signés représentés sur 8 bits.

Écriture en base 10	Représentation binaire
$\sim 58 = -59$	$\sim(0011\ 1010) = 1100\ 0101$
$58 \ll = 116$	$(0011\ 1010) \ll = 0111\ 0100$
$58 \gg = 29$	$(0011\ 1010) \gg = 0001\ 1101$
$\sim(-58) = 57$	$\sim(1100\ 0110) = 0011\ 1001$
$(-58) \ll = -116$	$(1100\ 0110) \ll = 1000\ 1100$
$(-58) \gg = -29$	$(1100\ 0110) \gg = 1110\ 0011$

La capture d'écran suivante montre les mêmes calculs effectués sur Nspire.

not 58	-59
shift(58,1)	116
shift(58,-1)	29
not -58	57
shift(-58,1)	-116
shift(-58,-1)	-29

Il est fréquent d'appliquer les opérateurs de décalage plusieurs fois de suite. Afin d'alléger les notations, l'écriture  $t \ll k$  signifie que l'opérateur de décalage  $\ll$  est appliqué  $k$  fois sur le train de bits  $t$ . Il en va de même pour l'opérateur  $\gg$ .

$$t \ll k \text{ signifie } t \underbrace{\ll \ll \dots \ll}_{k \text{ fois}}.$$

**Exemple 3.10**

Évaluez les expressions suivantes :

- (a)  $133 \ll 4$ , entier non signé représenté sur 8 bits.

**Solution :**

$\text{Rep}_8(133) = 1000\ 0101$  et le décalage vers la gauche ajoute des zéros à droite :

$$1000\ 0101 \ll 4 = 0101\ 0000.$$

Ainsi,  $133 \ll 4 = (0101\ 0000)_2 = 80$ , ce qui correspond bien au calcul  $(133 \cdot 2^4) \bmod 2^8 = 80$ .

- (b)  $133 \gg 2$ , entier non signé représenté sur 8 bits.

**Solution :**

La représentation binaire est la même qu'en (a). Le décalage vers la droite ajoute des 0 à gauche, car l'entier est non signé :

$$1000\ 0101 \gg 2 = 0010\ 0001.$$

Ainsi,  $133 \gg 2 = (0010\ 0001)_2 = 33$ , ce qui correspond bien au calcul  $\lfloor 133/2^2 \rfloor = 33$ .

- (c)  $-123 \gg 2$ , entier signé représenté sur 8 bits.

**Solution :**

$\text{Rep}_{\pm 8}(-123) = 1000\ 0101$ , on remarque que ce train de bits est le même qu'en (b) mais cette fois-ci il est considéré comme étant un nombre négatif. Le décalage à droite ajoute donc des 1 à gauche.

$$1000\ 0101 \gg 2 = 1110\ 0001.$$

On sait que l'entier représenté par  $1110\ 0001$  est négatif. On calcule sa valeur en utilisant la congruence modulo  $2^8 = 256$ .

$$(1110\ 0001)_{\pm 2} = (1110\ 0001)_2 - 2^8 = 225 - 256 = -31.$$

Le résultat obtenu correspond bien au calcul :

$$\lfloor -123/2^2 \rfloor = -31.$$

**Exercices**

**3.1** Effectuez les opérations bit à bit suivantes **à la main**, considérant qu'il s'agit d'entiers signés représentés sur 8 bits.

- (a)  $\sim(0011\ 1100 \oplus 1010\ 0011)$   
 (b)  $1010\ 1010 \vee (0000\ 0001 \ll 5)$   
 (c)  $\sim(1101\ 1011 \wedge (\sim 0001\ 0111))$   
 (d)  $(1100\ 1011 \gg 3) \oplus (0101\ 0100 \ll 2)$



**Exemple 3.11**

La couleur violette peut être obtenue en fixant le rouge à 93%, le vert à 51% et le bleu à 93%. On calcule sa représentation en un train de 24 bits.

$$\begin{aligned} \text{rouge} & : \lfloor 93/100 \cdot 255 \rfloor = 237 = (1110\ 1101)_2, \\ \text{vert} & : \lfloor 51/100 \cdot 255 \rfloor = 130 = (1000\ 0010)_2, \\ \text{bleu} & : \lfloor 93/100 \cdot 255 \rfloor = 237 = (1110\ 1101)_2. \end{aligned}$$

La couleur violette est représentée par le train de bits:  $\underbrace{1110\ 1101}_{\text{rouge}} \underbrace{1000\ 0010}_{\text{vert}} \underbrace{1110\ 1101}_{\text{bleu}}$ .

Étant donné un train de 24 bits  $t$  représentant une couleur au format RGB, comment peut-on en extraire la valeur de l'intensité de chacune des trois couleurs primaires: rouge, vert et bleu?

Une solution courante consiste à construire un **masque binaire**. Il s'agit d'un train de bit  $m$  ayant la même taille que  $t$  qui possède des 1 aux positions des bits à extraire et des 0 partout ailleurs. Pour l'extraction, on applique le masque  $m$  sur le train de bits  $t$  en effectuant un  $\wedge$  bit à bit. Les trois masques suivants permettent d'extraire chacune des trois couleurs:

$$\begin{aligned} \text{Masque rouge} & : m_r = 1111\ 1111\ 0000\ 0000\ 0000\ 0000. \\ \text{Masque vert} & : m_v = 0000\ 0000\ 1111\ 1111\ 0000\ 0000. \\ \text{Masque bleu} & : m_b = 0000\ 0000\ 0000\ 0000\ 1111\ 1111. \end{aligned}$$

Après avoir appliqué le masque, il ne reste plus qu'à décaler les bits vers la droite de manière à obtenir un nombre de 0 à 255 décrivant l'intensité de la couleur primaire.

**Exemple 3.12**

Le train de bit  $t = 1110\ 1111\ 0011\ 0111\ 0011\ 0000$  décrit une couleur au format RGB. Détaillez les opérations à effectuer pour extraire la valeur de chacune des trois couleurs.

**Solution :**

- Extraction de l'intensité du rouge:

1. Appliquer le masque  $m_r$ .

On calcule  $t \wedge m_r$ :

$$\begin{array}{r} 1110\ 1111\ 0011\ 0111\ 0011\ 0000 \\ \wedge\ 1111\ 1111\ 0000\ 0000\ 0000\ 0000 \\ \hline 1110\ 1111\ 0000\ 0000\ 0000\ 0000 \end{array}$$

2. Décaler de 16 bits vers la droite.

On calcule  $t \gg 16$ :

$$1110\ 1111\ 0000\ 0000\ 0000\ 0000 \gg 16 = 0000\ 0000\ 0000\ 0000\ 1110\ 1111$$

La valeur du rouge est donc:  $(1110\ 1111)_2 = 239$ .

- Extraction de l'intensité du vert:

1. Appliquer le masque  $m_v$ .

On calcule  $t \wedge m_v$ :



$$\begin{array}{r}
 1110\ 1111\ 0011\ 0111\ 0011\ 0000 \\
 \wedge\ 0000\ 0000\ 1111\ 1111\ 0000\ 0000 \\
 \hline
 0000\ 0000\ 0011\ 0111\ 0000\ 0000
 \end{array}$$

2. Décaler de 8 bits vers la droite.

On calcule  $t \gg 8$ :

$$0000\ 0000\ 0011\ 0111\ 0000\ 0000 \gg 8 = 0000\ 0000\ 0000\ 0000\ 0011\ 0111$$

La valeur du vert est donc:  $(0011\ 0111)_2 = 55$ .

• Extraction de l'intensité du bleu:

1. Appliquer le masque  $m_b$ .

On calcule  $t \wedge m_r$ :

$$\begin{array}{r}
 1110\ 1111\ 0011\ 0111\ 0011\ 0000 \\
 \wedge\ 0000\ 0000\ 0000\ 0000\ 1111\ 1111 \\
 \hline
 0000\ 0000\ 0000\ 0000\ 0011\ 0000
 \end{array}$$

La valeur du bleu est donc:  $(11\ 0000)_2 = 48$ .

La couleur RGB est (239,55,48), une manière de visualiser cette couleur est d'utiliser l'outil *Colour picker* de Google, tel qu'illustré à la figure 3.1. La figure 3.2 montre comment effectuer ce calcul avec Nspire.

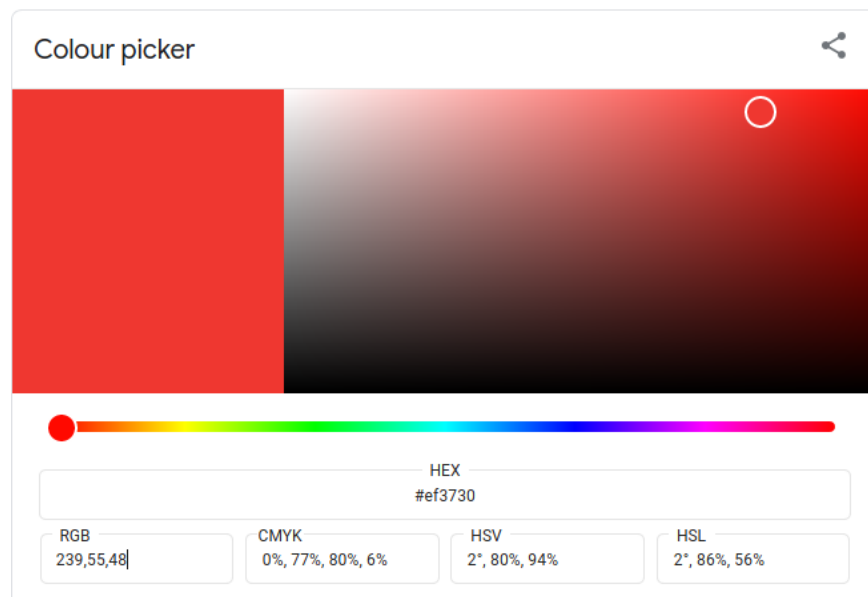


Figure 3.1 Illustration de la couleur RGB (239,55,48) à l'aide de l'outil *Colour picker* de Google. La description en hexadécimal de cette couleur #ef3730 correspond bien au train de bits initial  $t = 1110\ 1111\ 0011\ 0111\ 0011\ 0000$  car  $(E)_{16} = (1110)_2$ ,  $(F)_{16} = (1111)_2$ ,  $(3)_{16} = (0011)_2$ ,  $(7)_{16} = (0111)_2$ ,  $(3)_{16} = (0011)_2$ ,  $(0)_{16} = (0000)_2$ .

$t := 0b111011110011011100110000$	15677232
$m_r := 0b111111110000000000000000$	16711680
$rouge := \text{shift}(t \text{ and } m_r, -16)$	239
$m_v := 0b1111111100000000$	65280
$vert := \text{shift}(t \text{ and } m_v, -8)$	55
$m_b := 0b11111111$	255
$bleu := t \text{ and } m_b$	48
$\begin{bmatrix} rouge \\ vert \\ bleu \end{bmatrix}$	$\begin{bmatrix} 239 \\ 55 \\ 48 \end{bmatrix}$

Figure 3.2 Extraction des couleurs rouge, bleu et vert à partir d'un train de bit  $t$  représentant une couleur RGB. Les valeurs calculées sont présentées sous la forme d'un vecteur de dimension 3.

**3.4** Sur Npsire, écrivez une fonction  $f(r, g, b)$  qui prend en entrée trois nombres  $r, g, b$  de 0 à 255 et retourne le nombre correspondant au train de bit de longueur 24 décrivant cette couleur.

### Construire un masque binaire

Il existe de nombreuses manières pour décrire un masque binaire. Selon le système utilisé, on peut :

- donner son écriture binaire (cette méthode est illustrée sur Npsire à la figure 3.2),
- donner son écriture hexadécimale,
- donner son écriture décimale.

Dans certains cas, on obtient une meilleure lisibilité utilisant des opérateurs bit à bit. Par exemple, à la Figure 3.2 le masque  $m_r$  utilisé pour extraire la valeur du rouge dans un triplet RGB est problématique. Lorsqu'on écrit un tel masque à la main, il y a de bonnes chances qu'une faute de frappe fasse en sorte que le nombre de 0 ne soit pas exactement 16. Par contre, en utilisant le fait que l'écriture binaire de 255 est 1111 1111, les masques  $m_r, m_v$  et  $m_b$  s'écrivent alors de manière beaucoup plus lisible comme étant :

$$m_r = 255 \ll 16$$

$$m_v = 255 \ll 8$$

$$m_b = 255$$

La figure 3.3 illustre cette façon de faire sur Npsire. De plus, on vérifie que les masques  $m_r, m_v$  et  $m_b$  sont rigoureusement les mêmes que ceux construits à l'exemple 3.12.

$m_r := \text{shift}(255, 16)$	16711680
$m_r = 0b111111110000000000000000$	true
$m_v := \text{shift}(255, 8)$	65280
$m_v = 0b1111111100000000$	true
$m_b := 255$	255
$m_b = 0b11111111$	true

Figure 3.3 Les masques binaires utilisés à la figure 3.2 peuvent être définis de manière beaucoup plus lisible à l'aide de nombres écrits en base 10 et l'opérateur de décalage.

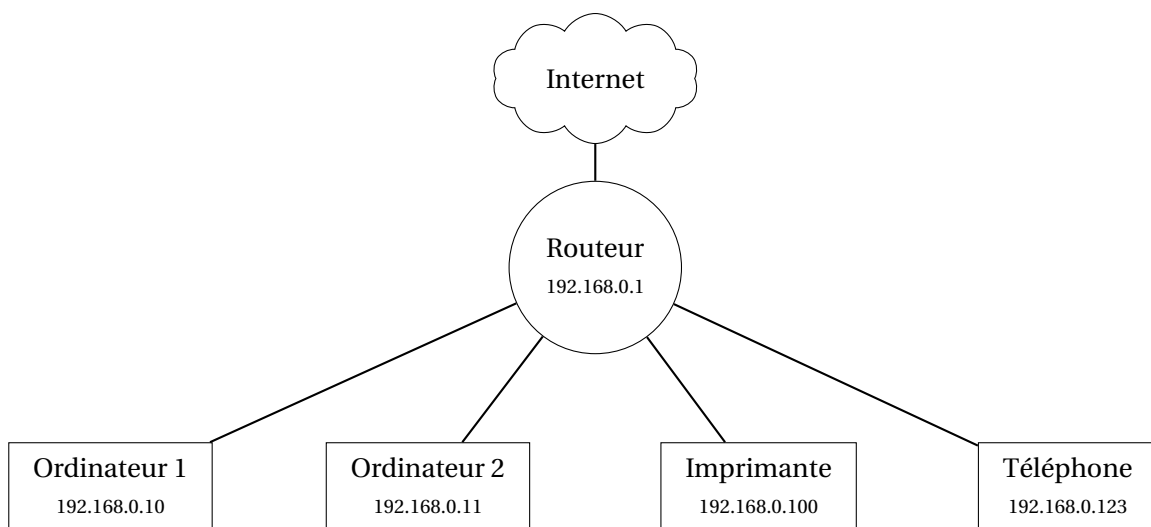
### Exemple 3.13

#### Les adresses IPv4 et les masques de sous-réseau

Avec le protocole de communication IPv4, chaque machine est identifiée par une adresse codée sur 32 bits. Ces adresses sont habituellement représentées avec la notation **décimale pointée** qui consiste à écrire l'adresse comme une suite de quatre nombres, chacun de ces nombres étant un entier non signé codé sur 8 bits. La plus grande valeur représentable sur 8 bits étant  $2^8 - 1 = 255$ , chacun de ces nombres est dans l'intervalle 0 à 255. Par exemple, l'adresse associée au nom de domaine `etsmt1.ca` est :

$$\underbrace{142}_{1000\ 1110} \cdot \underbrace{137}_{1000\ 1001} \cdot \underbrace{248}_{1111\ 1000} \cdot \underbrace{90}_{0101\ 1010}$$

Le protocole IPv4 prévoit la présence de plusieurs sous-réseaux interconnectés. Les 32 bits d'une adresse sont divisés en deux parties: les premiers bits de l'adresse servent à identifier le sous-réseau alors que les autres identifient une machine dans ce sous-réseau. Sans entrer dans les détails de ce protocole, considérons l'exemple d'un sous-réseau résidentiel :



Dans cet exemple, et c'est souvent le cas avec les réseaux résidentiels, le sous-réseau est identifié par les 24 premiers bits d'une adresse. Par exemple, pour l'imprimante :

Adresse	192	168	0	100
Binaire	1100 0000	1010 1000	0000 0000	0110 0100
	Identifiant du sous-réseau			Identifiant de la machine

### Problème 1. Est-ce que la destination appartient au sous-réseau ?

Lorsque l'*Ordinateur 1* souhaite communiquer avec une autre machine, un *paquet* dont l'entête contient l'adresse de la destination est envoyé au routeur. Celui-ci doit déterminer si ce paquet est à destination d'une des machines du sous-réseau ou s'il doit être transmis à un autre sous-réseau. Sur le schéma, les autres sous-réseaux sont représentés par le nuage *Internet*.

Pour résoudre ce problème, on peut utiliser différentes approches. Le routeur pourrait, par exemple, tester si l'adresse de la destination est l'une des adresses parmi 192.168.0.10, 192.168.0.11, 192.168.0.100 et 192.168.0.123. Cette méthode présente plusieurs inconvénients. Entre autres, le nombre de tests à effectuer est proportionnel au nombre de machines connectées au sous-réseau. Comme ce test doit être effectué pour chaque paquet émis par chacune des machines du sous-réseau, le routeur doit l'effectuer très fréquemment. Il est donc important de simplifier ce test autant que possible afin de ne pas surcharger le routeur<sup>1</sup>.

Une méthode plus efficace consiste simplement à tester si les 24 premiers bits de l'adresse correspondent à 192.168.0, l'identifiant du sous-réseau, ou, exprimé en binaire, 1100 0000.1010 1000.0000 0000. Comment effectuer ce test? À l'aide d'un masque binaire dont tous les 24 premiers bits valent 1 et les huit derniers valent 0. Ce masque est donc :

$$m = 11111111 11111111 11111111 00000000.$$

En utilisant la notation décimale pointée, ce masque s'écrit: 255.255.255.0.

Pour tester si une adresse  $A$  appartient au sous-réseau, il suffit de vérifier si  $A \wedge m = 192.168.0.0$ . En effet, le masque a été construit de sorte que, d'un côté, les bits à 1 sont alignés avec l'identifiant du sous-réseau et, de l'autre, les 0 sont alignés avec l'identifiant de la machine destinataire.

Supposons que l'*Ordinateur 1* envoie un paquet à l'*Imprimante*. L'adresse de destination du paquet est  $A = 192.168.0.100$  et le routeur effectue le calcul  $A \wedge m$ .

Décimale-pointée	Binaire
192 . 168 . 0 . 100	1100 0000 . 1010 1000 . 0000 0000 . 0110 0100
$\wedge$ 255 . 255 . 255 . 0	$\wedge$ 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
= 192 . 168 . 0 . 0	= 1100 0000 . 1010 1000 . 0000 0000 . 0000 0000

Comme le résultat étant bien 192.168.0.0, le routeur sait maintenant que la destination fait partie de son sous-réseau.

### Problème 2. Identifier le destinataire.

Dans l'exemple précédent, le routeur a déterminé que la machine destinataire fait partie de son sous-réseau en effectuant un **et** bit à bit entre l'adresse du destinataire et le masque  $m$ . Le routeur doit maintenant déterminer à quelle machine de son sous-réseau ce paquet est destiné. Dans l'adresse 192.168.0.100 il faut donc extraire l'identifiant de la machine. Pour cela, on utilise un second masque binaire qui est le complément bit à bit du masque  $m$ .

1. En général, les sous-réseaux peuvent être beaucoup plus complexes et la méthode consistant à tester l'égalité avec toutes les adresses du sous-réseau est encore plus problématique.

$$\sim m = 00000000\ 00000000\ 00000000\ 11111111.$$

La machine destinataire est donc identifiée par le calcul:

Notation décimale	Notation binaire
192 . 168 . 0 . 100	1100 0000 . 1010 1000 . 0000 0000 . 0110 0100
$\wedge$ 0 . 0 . 0 . 255	$\wedge$ 0000 0000 . 0000 0000 . 0000 0000 . 1111 1111
= 0 . 0 . 0 . 100	= 0000 0000 . 0000 0000 . 0000 0000 . 0110 0100

L'identifiant de l'imprimante sur ce sous-réseau est donc: 0.0.0.100.

### Exercices

**3.5** Pour chacune des adresses suivantes, donner l'identifiant du sous-réseau ainsi que l'identifiant de la machine en notation décimale pointée, sachant que le sous-réseau est identifié par les 20 premiers bits.

- (a) 192.168.17.42
- (b) 192.168.153.205
- (c) 142.137.248.90

**Exemple 3.14****Lecture d'un capteur**

Lorsqu'un capteur communique des données avec un ordinateur, il est fréquent que celui-ci transmette des trains de bits de taille fixe. Plus précisément, à chaque fois que le capteur effectue une lecture, il transmet un train de bit à l'ordinateur. Il faut ensuite extraire les informations pertinentes de ce train de bits.

Pour cet exemple, considérons un capteur qui mesure la température ainsi que le niveau d'humidité relative dans l'air, tel qu'illustré à la figure 3.4. L'intervalle des températures mesurables est de  $-40^{\circ}\text{C}$  à  $85^{\circ}\text{C}$  alors que pour l'humidité relative les valeurs possibles vont de 0% à 100%.

Lorsque ce capteur effectue une lecture, il communique un 8 octets (64 bits) à l'ordinateur. Parmi ces octets, seulement quatre sont utilisés: les octets 2 et 3 pour l'humidité et les octets 4 et 5 pour la température, tel qu'illustré à la Figure 3.5. Dans les deux cas, le capteur communique un entier non signé écrit sur 16 bits, soit de 0 à 65535. La valeur de cet entier est proportionnelle à la valeur mesurée dans leurs intervalles respectifs.

Température		Humidité	
0	$-40^{\circ}\text{C}$	0	0%
⋮	⋮	⋮	⋮
32768	$22.5^{\circ}\text{C}$	32768	50%
⋮	⋮	⋮	⋮
65535	$85^{\circ}\text{C}$	65535	100%

De plus, il faut savoir que l'ordre des octets utilisés pour représenter les entiers n'est pas le même sur le capteur que sur l'ordinateur<sup>2</sup>. En conclusion, pour chacune des deux mesures (température et humidité), l'ordinateur doit effectuer quatre opérations:

1. extraire l'octet faible et la placer en position 0,
2. extraire l'octet fort et la placer en position 1,
3. combiner les deux octets de manière à former un entier sur 16 bits,
4. appliquer une transformation affine pour ramener l'entier formé à l'étape 3 dans l'intervalle correspondant ( $-40^{\circ}\text{C}$  à  $85^{\circ}\text{C}$  pour la température et 0% à 100% pour l'humidité).



Figure 3.4 Un capteur capable de mesurer la température et le niveau d'humidité relative dans l'air. Les données sont communiquées à un ordinateur via un port USB.

- (a) Décrivez chacune des opérations requises pour obtenir la température.
- (b) Décrivez chacune des opérations requises pour obtenir l'humidité.

2. Ce problème, appelé le **boutisme** (*endianness* en anglais), est omniprésent lorsque des machines ayant des architectures différentes communiquent entre elles.

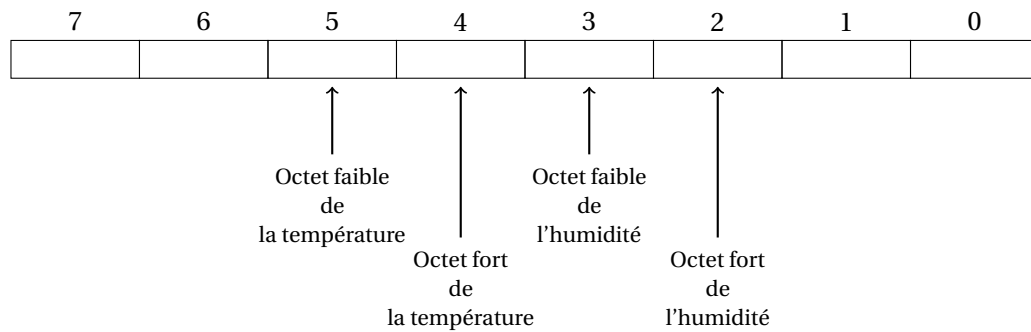


Figure 3.5 Disposition des octets de données dans le bloc transmis par le capteur. Pour un nombre écrit sur deux octets, l'**octet faible** contient les huit bits les moins significatifs alors que l'**octet fort** contient les huit bits les plus significatifs. Par exemple, si l'octet faible est 0011 1100 et l'octet fort est 0000 1111 alors le nombre représenté est  $(0000\ 1111\ 0011\ 1100)_2 = (3900)_{10}$ .

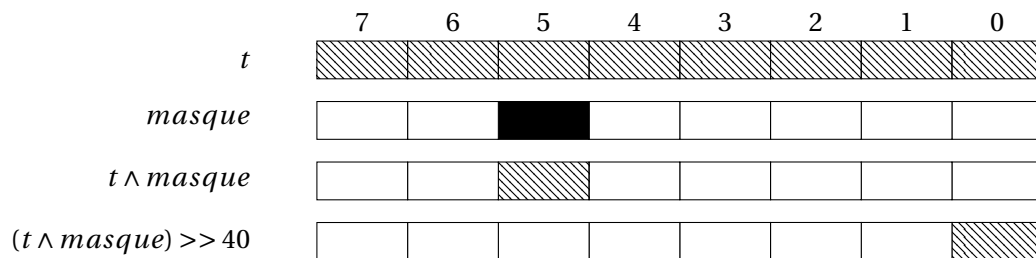
### Solution :

Soit  $t$  le train de 64 bits transmis par le capteur.

(a) Calcul de la température.

#### 1. Extraire l'octet faible de la température.

On utilise un masque binaire construit à partir du nombre 255. Sachant que 255 extrait les huit bits de l'octet 0, pour extraire les bits de l'octet 5, il faut le décaler vers la gauche de  $5 \cdot 8 = 40$  bits. Ensuite, il faut décaler les bits extraits vers la droite de sorte qu'ils occupent l'octet 0.

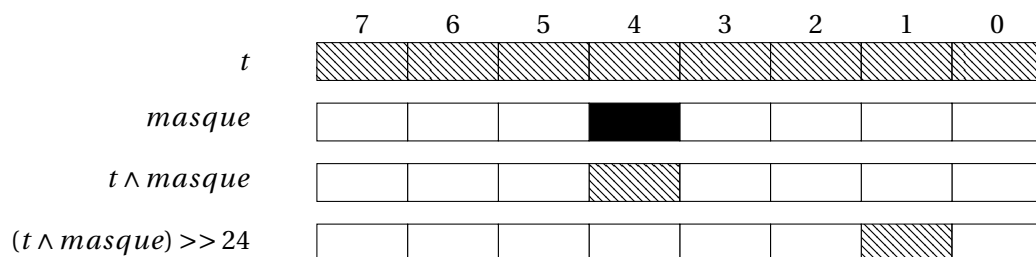


Le calcul de l'octet faible de la température est donc :

$$\begin{aligned} \text{masque} &:= 255 \ll 40 \\ T_{\text{faible}} &:= (t \wedge \text{masque}) \gg 40. \end{aligned}$$

#### 2. Extraire l'octet fort de la température.

Le calcul semblable à celui de l'octet faible, mais cette fois-ci, lorsqu'on décale l'octet vers la droite, fait en sorte qu'il occupe la position de l'octet 1.

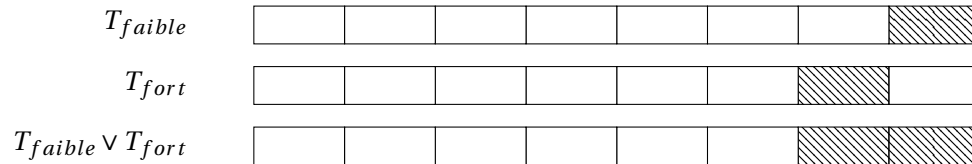


Le calcul de l'octet faible de la température est donc :

$$\begin{aligned} \text{masque} &:= 255 \ll 32 \\ T_{fort} &:= (t \wedge \text{masque}) \gg 24. \end{aligned}$$

### 3. Combiner les deux octets.

On utilise l'opérateur  $\vee$  pour former un entier sur 16 bits.



L'entier 16 bits de la température est :

$$T = T_{faible} \vee T_{fort}.$$

### 4. Appliquer la transformation affine.

On cherche la valeur dans l'intervalle  $[-40, 85]$  qui correspond à position de  $T$  dans l'intervalle  $[0, 65535]$ .

$$\text{température} := \frac{(85 - (-40))}{65535} T - 40.$$

(b) Calcul de l'humidité.

#### 1. Extraire l'octet faible de l'humidité.

$$H_{faible} := (t \wedge (255 \ll 24)) \gg 24.$$

#### 2. Extraire l'octet fort de l'humidité.

$$H_{fort} := (t \wedge (255 \ll 16)) \gg 8.$$

#### 3. Combiner les deux octets.

$$H = H_{faible} \vee H_{fort}.$$

#### 4. Appliquer la transformation affine.

Il suffit d'exprimer la fraction  $H/65535$  sous la forme d'un pourcentage.

$$\text{humidité} := \frac{100}{65535} H.$$


---



## Chapitre 4

# Introduction à la complexité des algorithmes

En informatique, on considère habituellement qu'un algorithme est une suite d'opérations permettant de résoudre un problème. Cette suite d'opérations peut être *implémentée* de sorte à obtenir un programme qui résout ce problème de manière effective.

En général, il existe plusieurs algorithmes pour résoudre un problème donné. Par exemple, étant donné deux entiers positifs  $x$  et  $y$ , pour calculer  $x$  modulo  $y$ , on peut :

- **Algorithme A.** Soustraire  $y$  à  $x$  jusqu'à obtenir une valeur dans l'intervalle  $[0, y[$ . Cette valeur est la solution.
- **Algorithme B.** Effectuer la division de  $x$  par  $y$  avec la méthode du *crochet*<sup>1</sup>. Le reste de la division est la solution.

### Exemple 4.1 (à compléter en classe)

Calculez 63 modulo 5 en utilisant d'abord l'algorithme A (soustractions successives), puis l'algorithme B (division par crochet).

### Solution :

---

Même si dans certains cas l'algorithme A est plus rapide que le B, dans la grande majorité des cas c'est l'algorithme B qui est le plus rapide. En effet, il est évident que pour calculer  $100\,000 \bmod 7$ , l'algorithme A est beaucoup plus lent.

---

1. Il s'agit de la méthode habituellement enseignée au primaire.

Dans le cas de cet exemple, il est facile de déterminer quel algorithme est le plus efficace, mais de manière générale *comment comparer l'efficacité de deux algorithmes?*

Cette simple question a donné naissance à un domaine d'études appelé la **théorie de la complexité**, dont voici quelques principes clés:

- le **temps** est mesuré en **nombre d'opérations**,
- on s'intéresse à la **croissance** du nombre d'opérations en fonction de la taille de l'entrée,
- pour une taille de données fixée, on s'intéresse toujours au **pire cas** possible.

Ce chapitre d'introduction à la théorie de la complexité se divise en deux parties. Nous verrons d'abord comment représenter la complexité d'un algorithme à l'aide d'une fonction  $f(n)$ , où  $n$  correspond à la taille des données à traiter et  $f(n)$  correspond aux nombres d'opérations qui seront effectuées par l'algorithme. Comparer la rapidité de deux algorithmes reviendra donc à comparer deux fonctions.

À la deuxième section, nous introduirons les notations grand-O et grand- $\Theta$ . Celles-ci sont en quelque sorte analogues aux symboles  $\leq$  et  $=$  car elles permettent de comparer « l'ordre de grandeur » des fonctions. Cela nous permettra de comparer les algorithmes afin d'identifier lesquels sont plus efficaces pour traiter des données de grande taille.

## 4.1 Mesurer un temps de calcul à l'aide d'une fonction

Vous souvenez-vous comment additionner et multiplier des nombres à la main? L'algorithme enseigné au primaire pour additionner deux nombres va comme suit (expliqué ici à un humain et non à une machine):

1. superposer les 2 nombres de façon à ce que les chiffres des unités soient alignés
2. tracer une ligne horizontale sous le nombre du bas
3. débiter par la colonne des unités (colonne de droite):
  - (a) additionner les 2 chiffres de la colonne
  - (b) inscrire le chiffre des unités du résultat en dessous en notant la retenue au-dessus de la colonne située juste à gauche s'il y a lieu
4. passer à la colonne suivante
  - (a) additionner les chiffres de la colonne (possibilité de 3 chiffres, car retenue possible)
  - (b) inscrire le chiffre des unités du résultat en dessous en notant la retenue au-dessus de la colonne située juste à gauche s'il y a lieu
5. répéter le processus jusqu'à la dernière colonne (colonne la plus à gauche)
6. abaisser la dernière retenue potentielle, c'est-à-dire réécrire ce chiffre sous la ligne horizontale
7. lire le résultat sous la ligne horizontale

**Exemple 4.2** (à compléter en classe)

Calculez  $3186 + 5916$  en utilisant l'algorithme décrit ci-dessus. De plus, déterminez le nombre d'additions simples de 2 chiffres que l'on doit-on effectuer pour obtenir le résultat.

**Solution :****Exemple 4.3** (à compléter en classe)

Si les 2 nombres à additionner ont chacun 10 chiffres, combien d'additions simples de 2 chiffres doit-on effectuer pour obtenir le résultat?

Et si les nombres ont 100 chiffres chacun?

De façon plus générale, si les 2 nombres à additionner ont chacun  $n$  chiffres, combien d'additions simples de 2 chiffres doit-on effectuer pour obtenir le résultat?

**Solution :****Exemple 4.4** (à compléter en classe)

Considérons maintenant l'algorithme de multiplication de 2 nombres de  $n$  chiffres enseigné au primaire, similaire à celui exposé ci-dessus pour l'addition. Combien cet algorithme nécessite-t-il de multiplications simples de 2 chiffres?

**Solution :**

Le **temps d'exécution** d'un algorithme donné dépend principalement de

- la machine utilisée : langage, système d'exploitation, compilateur, vitesse de calcul de la machine, etc.
- les données auxquelles l'algorithme est appliqué.

Nous allons utiliser une mesure plus abstraite, qui ne dépend pas de la machine ni des données elles-mêmes, mais plutôt de la *taille* des données. Par exemple, le temps d'exécution d'un algorithme de tri dépend de la longueur de la liste à trier. Le temps d'exécution d'une multiplication de deux nombres dépend du nombre de chiffres de ces nombres. Nous utiliserons donc une fonction pour décrire la complexité d'un algorithme (voir figure 4.1).

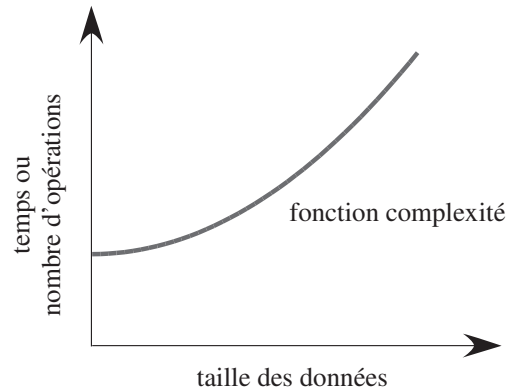


Figure 4.1 On utilise une fonction pour décrire la complexité d'un algorithme.

**Exemple 4.5** (à compléter en classe)

Considérez l'algorithme de fouille séquentielle dans une liste quelconque (fonction **recherchelin** à la figure 4.2) et l'algorithme de fouille dichotomique **dans une liste ordonnée en ordre croissant** (fonction **recherchebin** à la figure 4.2). Nous choisissons ici d'étudier la complexité dans *le pire des cas*.

- Pour chacun des algorithmes, trouvez le nombre de comparaisons requises pour traiter une liste de taille  $n$ . Considérez autant les comparaisons de nombres entiers que les comparaisons d'éléments de la liste avec l'élément recherché. Lequel des deux algorithmes requiert au plus  $f(n) = 2n + 2$  comparaisons pour traiter une liste de taille  $n$ ? Et lequel en nécessite au plus  $g(n) = 2\log_2(n) + 2$ ?
- Lequel des deux algorithmes sera plus efficace pour traiter de grandes listes de données préalablement ordonnées en ordre croissant?

---

```

Define recherchein ( $t,x$ )=
Func
©  $t$  est un tableau, i.e. une liste, indexé par 1, 2, ...,  $n$ .
©  $x$  est l'élément recherché dans le tableau  $t$ .
© Retourne l'indice, i.e. la position, de l'élément  $x$  s'il est dans le tableau  $t$ 
© et retourne 0 si  $x$  n'est pas dans le tableau.
Local  $n,i,position$ 
 $n:=dim(t)$ 
 $i:=1$ 
While  $i\leq n$ 
  If  $x=t[i]$ :Exit
   $i:=i+1$ 
EndWhile
If  $i\leq n$  Then
   $position:=i$ 
Else
   $position:=0$ 
EndIf
Return  $position$ 
EndFunc

```

---

```

Define recherchebin ( $t,x$ )=
Func
©  $t$  est un tableau, i.e. une liste, ORDONNÉ indexé par 1, 2, ...,  $n$ .
©  $x$  est l'élément recherché dans le tableau  $t$ .
© Retourne l'indice, i.e. la position, de l'élément  $x$  s'il est dans le tableau  $t$ 
© et retourne 0 si  $x$  n'est pas dans le tableau.
Local  $n,a,b,position,m$ 
 $n:=dim(t)$ 
 $a:=1$ 
 $b:=n+1$ 
While  $b-a>1$ 
   $m:=\text{floor}\left(\frac{a+b}{2}\right)$ 
  If  $t[m]\leq x$  Then
     $a:=m$ 
  Else
     $b:=m$ 
  EndIf
EndWhile
If  $x=t[a]$  Then
   $position:=a$ 
Else
   $position:=0$ 
EndIf
Return  $position$ 
EndFunc

```

---

Figure 4.2 Fouille séquentielle et fouille dichotomique en TI Nspire.

**Exemple 4.6** (à compléter en classe)

Considérez l'algorithme  $A$  qui nécessite  $f(n)$  opérations d'un certain type pour résoudre un problème de taille  $n$  et l'algorithme  $B$  qui en nécessite  $g(n)$ . Utilisez vos connaissances mathématiques préalables afin de déterminer lequel, selon vous, sera plus efficace pour résoudre les gros problèmes si

(a)  $f(n) = n^3 + 3$  et  $g(n) = 25 + n^2$ ;

(b)  $f(n) = 2^n + 4n^3$  et  $g(n) = 10n^4$ .

## 4.2 Notation grand-O et grand- $\Theta$

Il y a plusieurs facteurs qui influencent la vitesse à laquelle un programme exécute une tâche précise :

- l'ordinateur sur lequel il est lancé;
- le système d'exploitation;
- la taille des données à traiter pour cette tâche;
- l'utilisation de la mémoire;
- etc.

Nous nous concentrerons ici uniquement sur la fonction de complexité d'un algorithme, c'est-à-dire le **nombre d'opérations** requises par l'algorithme pour traiter un problème. Si la taille des données à traiter double, par exemple pour trier une liste de 1000 ou de 2000 entrées, le nombre d'opérations requises demeurera-t-il constant? Doublera-t-il? Sera-t-il mis au carré? Au cube?

Concrètement, il peut s'avérer fastidieux de compter précisément toutes les opérations effectuées lors de l'exécution d'un algorithme. Afin de nous simplifier la tâche dans le cadre de ce chapitre, **l'opération la plus significative** sera sélectionnée et nous compterons uniquement le nombre de fois que cette opération est effectuée.

Le rappel de la méthode d'additions présentée à la section 4.1 nous a conduits au fait que le nombre d'additions de chiffres requises pour additionner deux nombres de  $n$  chiffres est au plus  $f(n) = 2n - 1$ . L'addition de deux nombres de 100 chiffres requiert donc au plus 199 additions, tandis que l'addition de deux nombres de 200 chiffres requiert au plus 399 additions. Quand la taille des données augmente, le nombre d'opérations requises pour les traiter (les additionner) ne demeure pas constant. Il augmente environ du même facteur.

Pour des algorithmes plus compliqués, il est souvent difficile et même impossible de déterminer précisément la fonction de complexité. En fait, pour comparer deux algorithmes, il n'est pas toujours nécessaire de connaître exactement le nombre d'opérations effectuées par chacun d'eux: bien souvent, il suffit de connaître leur ordre de grandeur. Il nous faut donc un outil qui permet de déterminer et de comparer ces ordres de grandeur sur les fonctions de complexités. Cet outil existe; il s'agit de la notation grand-O. La notation grand-O regroupe en sous-ensemble les fonctions dont la croissance est comparable. On peut ensuite les trier, de la croissance la plus lente à la croissance la plus rapide. Chacun des sous-ensembles est identifié par son représentant le plus simple possible.

- Les fonctions constantes, ou qui sont bornées par une fonction constante, sont regroupées en un sous-ensemble qui est identifié par  $O(1)$ , car la fonction constante la plus « simple » possible est  $f(n) = 1$ . Les fonctions de complexité de l'ensemble  $O(1)$  correspondent à des algorithmes qui ne prennent pas plus de temps pour traiter un problème dont la taille est grande ou petite. La figure 4.3 illustre 3 fonctions de l'ensemble  $O(1)$ .
- Les fonctions qui sont bornées par un polynôme de degré 1 forment le sous-ensemble  $O(n)$ . La figure 4.4 illustre 3 fonctions de l'ensemble  $O(n)$ .
- Les fonctions qui sont bornées par un polynôme de degré 2 forment le sous-ensemble  $O(n^2)$ .

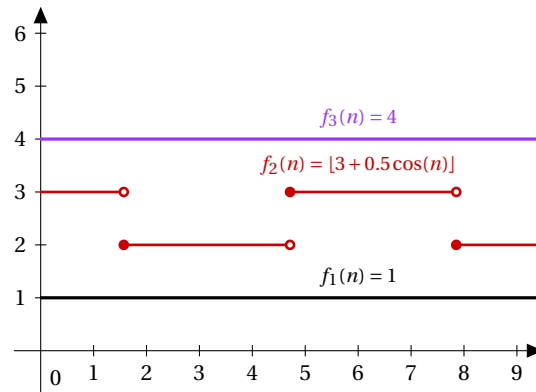


Figure 4.3 Les fonctions qui sont bornées par une fonction constante forment le sous-ensemble qui est identifié par  $O(1)$ . Par exemple, on a  $f_1 \in O(1)$ ,  $f_2 \in O(1)$ ,  $f_3 \in O(1)$ .

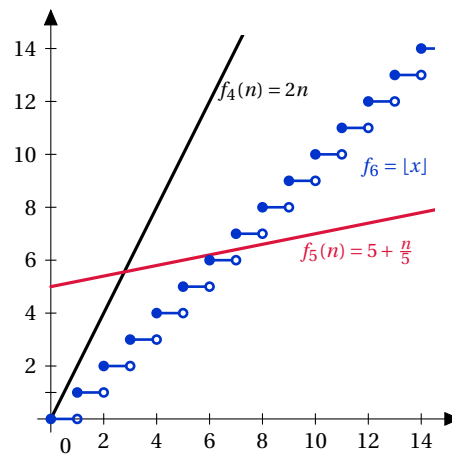


Figure 4.4 Les fonctions qui sont bornées par un polynôme de degré 1 forment le sous-ensemble qui est identifié par  $O(n)$ . Par exemple, on a  $f_4 \in O(n)$ ,  $f_5 \in O(n)$ ,  $f_6 \in O(n)$ .

**Définition 4.1 : Grand-O, facteur, témoin**

Soit  $f$  et  $g$  deux fonctions de  $\mathbb{R}$  vers  $\mathbb{R}$  ou de  $\mathbb{N}$  vers  $\mathbb{R}$ . On dit que  $f(x)$  est **grand-O** de  $g(x)$ , que l'on note  $f(x) \in O(g(x))$ , si, en valeur absolue,  $f$  est éventuellement dépassée par un multiple de  $g$ . Ainsi,

$$f(x) \in O(g(x)) \iff \exists C \in \mathbb{R}, \exists k \in \mathbb{R}, \forall x \in \text{dom}(f), x > k \rightarrow |f(x)| \leq C|g(x)|$$

Cela signifie donc que la fonction  $|f|$  est bornée par un multiple de la fonction  $|g|$  à partir d'une certaine valeur  $k$  du domaine (voir figure 4.5).

La borne  $k$ , appelée **seuil**, permet d'ignorer le comportement des fonctions pour les données de petite taille (dans ces cas, la complexité de l'algorithme est souvent dominée par des opérations d'initialisation qui deviennent négligeables pour des données plus grandes).

La constante  $C$ , appelée **facteur**, permet de faire abstraction de la vitesse de la machine utilisée.

Les nombres  $k$  et  $C$  sont appelés **témoins** de la relation  $f(x) \in O(g(x))$ .

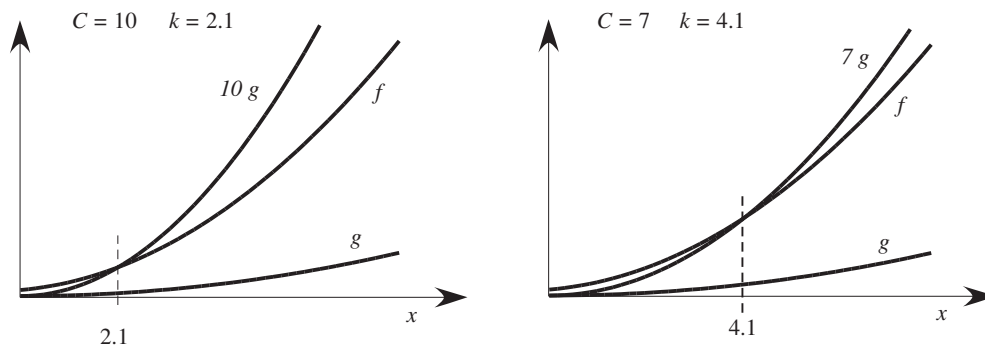


Figure 4.5 Pour montrer que  $f(x) \in O(g(x))$ , il suffit de trouver une paire de témoins  $C$  et  $k$ , mais il en existe une infinité. Nous en présentons deux paires ici. Le but est de montrer qu'il existe un multiple de  $g$  qui dépasse  $f$  pour toutes les valeurs de  $x$  à partir d'un certain seuil.

Notons que la définition repose sur une inégalité. Ainsi, les sous-ensembles de fonctions définis par cette notation sont imbriqués. Par exemple, on a

$$O(1) \subseteq O(n) \subseteq O(n^2).$$

**Définition 4.2 : Grand-Θ**

Soit  $f$  et  $g$  deux fonctions de  $\mathbb{R}$  vers  $\mathbb{R}$  ou de  $\mathbb{N}$  vers  $\mathbb{R}$ . La notation « **grand thêta** » est utilisée pour désigner que chacune des 2 fonctions est grand-O de l'autre. On dit alors que les fonctions sont **du même ordre**.

$$f(x) \in O(g(x)) \text{ et } g(x) \in O(f(x)) \iff f(x) \in \Theta(g(x))$$



**Exemple 4.7**

Considérez les deux fonctions de  $\mathbb{R}$  vers  $\mathbb{R}$  suivantes :

$$f(x) = 5x^2 + 6x + 9 \quad \text{et} \quad g(x) = x^2.$$

Montrez que  $f(x) \in \Theta(g(x))$  en fournissant les témoins obtenus algébriquement et non par l'observation d'un graphique.

**Solution :**

Nous devons donc démontrer que  $f(x) \in O(g(x))$  et que  $g(x) \in O(f(x))$ . Notons d'abord que ces deux fonctions prennent des valeurs positives lorsque  $x \geq 0$ . Nous pouvons donc laisser tomber les valeurs absolues dans les inégalités à vérifier.

Pour montrer que  $f(x) \in O(g(x))$ , nous devons montrer que  $f(x)$  est inférieure ou égale à un multiple de  $g(x)$  à partir d'une certaine valeur du domaine. On sait que

$$\begin{aligned} x \geq 3 &\rightarrow 9 \leq x^2; \\ x \geq 6 &\rightarrow 6x \leq x^2. \end{aligned}$$

En additionnant ces inégalités et en en additionnant  $5x^2$  de chaque côté, on obtient

$$x \geq 6 \rightarrow 5x^2 + 6x + 9 \leq 5x^2 + x^2 + x^2 = 7x^2.$$

Nous avons donc trouvé une paire de témoins de la relation  $f(x) \in O(g(x))$  :  $C = 7$  et  $k = 6$ . En effet,

$$\forall x \in \mathbb{R}, x \geq 6 \rightarrow f(x) \leq 7g(x).$$

Le graphique 4.6 illustre la situation.

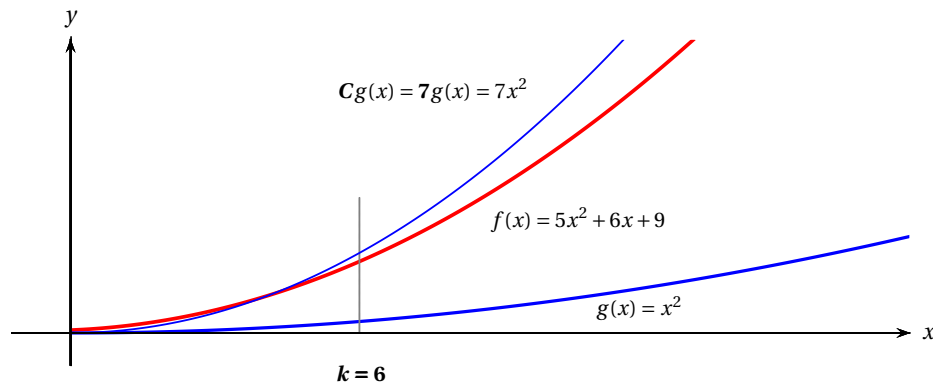


Figure 4.6 Illustration de la relation  $f(x) \in O(g(x))$ , avec les témoins  $k = 6$  et  $C = 7$ .

Pour montrer que  $g(x) \in O(f(x))$ , nous devons montrer que  $g(x)$  est inférieure ou égale à un multiple de  $f(x)$  à partir d'une certaine valeur du domaine. Or, nous n'avons même pas à multiplier  $f$  par une constante pour dépasser  $g$ . On sait que

$$x \geq 0 \rightarrow x^2 \leq 5x^2 + 6x + 9.$$

Nous avons donc trouvé une paire de témoins de la relation  $g(x) \in O(f(x))$  :  $C = 1$  et  $k = 0$ . En effet,

$$\forall x \in \mathbb{R}, x \geq 0 \rightarrow g(x) \leq 1f(x).$$

En général, il est laborieux de prouver la validité des témoins  $k$  et  $C$  d'une relation grand-O et, dans la pratique, les valeurs de ces témoins n'ont pas d'importance: l'important est de savoir que  $f(x)$  est  $O(g(x))$ . Les théorèmes suivants permettent d'établir une relation grand-O tout en évitant la recherche de témoins.

### Théorème 4.1

Soit  $f$  et  $g$  deux fonctions de  $\mathbb{R}$  vers  $\mathbb{R}$  ou de  $\mathbb{N}$  vers  $\mathbb{R}$ .

- Si

$$\lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| = b \text{ avec } 0 < b < \infty$$

alors les fonctions  $f$  et  $g$  sont du **même ordre de grandeur**:

$$f(x) \in O(g(x)) \text{ et } g(x) \in O(f(x)).$$

- Si

$$\lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| = 0$$

alors les fonctions  $f$  et  $g$  **ne sont pas du même ordre de grandeur**:

$f(x) \in O(g(x))$  mais  $g(x) \notin O(f(x))$ . La fonction  $f$  est donc négligeable face à la fonction  $g$  quand  $x$  tend vers l'infini.

- Si

$$\lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| = \infty$$

alors les fonctions  $f$  et  $g$  **ne sont pas du même ordre de grandeur**:

$g(x) \in O(f(x))$  mais  $f(x) \notin O(g(x))$ . La fonction  $g$  est donc négligeable face à la fonction  $f$  quand  $x$  tend vers l'infini.

### Exemple 4.8

Soit  $f(x) = 7x^2$ ,  $g(x) = x^3$  et  $h(x) = x^2$ . Est-ce que  $f(x) \in O(g(x))$ ? Est-ce que  $g(x) \in O(f(x))$ ?

Est-ce que  $f(x) \in O(h(x))$ ? Est-ce que  $h(x) \in O(f(x))$ ?

#### Solution :

Calculons la limite

$$\begin{aligned} \lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| &= \lim_{x \rightarrow \infty} \frac{7x^2}{x^3} \\ &= \lim_{x \rightarrow \infty} \frac{7}{x} \\ &= 0. \end{aligned}$$

Par le théorème 4.1, les fonctions  $f$  et  $g$  ne sont donc pas du même ordre :

$$7x^2 \in O(x^3) \text{ mais } x^3 \notin O(7x^2).$$

Calculons la limite

$$\begin{aligned} \lim_{x \rightarrow \infty} \left| \frac{f(x)}{h(x)} \right| &= \lim_{x \rightarrow \infty} \frac{7x^2}{x^2} \\ &= \lim_{x \rightarrow \infty} 7 \\ &= 7 \end{aligned}$$

Par le théorème 4.1, les fonctions  $f$  et  $h$  sont du même ordre :

$$7x^2 \in O(x^2) \text{ et } x^2 \in O(7x^2).$$

Notez que l'on souhaite habituellement trouver **le plus petit sous-ensemble** de fonctions auxquelles appartient la fonction de complexité que l'on étudie, disons  $f(x) = 7x^2$ . Pour cela, il faut trouver la fonction aussi simple que possible et ayant la complexité la moins grande qui est du même ordre de grandeur que la fonction  $f$ . Ainsi, même s'il est vrai que

$$7x^2 \in O(x^3),$$

cette proposition nous renseigne moins que celle-ci :

$$7x^2 \in O(x^2).$$

### Exemple 4.9

Soit  $f(x) = 5x^2 + 2x + 7$  et  $g(x) = x^2$ . Est-ce que  $f(x) \in O(g(x))$ ? Est-ce que  $g(x) \in O(f(x))$ ?

**Solution :**

$$\begin{aligned} \lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| &= \lim_{x \rightarrow \infty} \frac{5x^2 + 2x + 7}{x^2} \\ &= \lim_{x \rightarrow \infty} \left( 5 + \frac{2}{x} + \frac{7}{x^2} \right) \\ &= 5 \end{aligned}$$

Ainsi, par le théorème 4.1, les fonctions  $f$  et  $g$  sont du même ordre de grandeur :

$$5x^2 + 2x + 7 \in O(x^2) \quad \text{et} \quad x^2 \in O(5x^2 + 2x + 7).$$

### Exemple 4.10

Soit  $f(n) = \log_2(n)$  et  $g(n) = n$ . Est-ce que  $f(n) \in O(g(n))$ ? Est-ce que  $g(n) \in O(f(n))$ ?

**Solution :**

$$\begin{aligned} \lim_{n \rightarrow \infty} \left| \frac{f(n)}{g(n)} \right| &= \lim_{n \rightarrow \infty} \frac{\log_2(n)}{n} \quad \text{qui est de la forme } \frac{\infty}{\infty} \\ &= \lim_{n \rightarrow \infty} \frac{1}{n \ln(2)} \quad \text{par la règle de l'Hopital} \\ &= 0 \end{aligned}$$

donc, par le théorème 4.1, les fonctions  $f$  et  $g$  ne sont pas du même ordre de grandeur :

$$\log_2(n) \in O(n) \quad \text{mais} \quad n \notin O(\log_2(n)).$$

La même démarche fonctionne pour un logarithme de n'importe quelle base.

**Exemple 4.11**

Soit  $f(n) = \log_2(n)$  et  $g(n) = \log(n)$ . Est-ce que  $f(n) \in O(g(n))$ ? Est-ce que  $g(n) \in O(f(n))$ ?

**Solution :**

Grâce à la loi des logarithmes (changement de base), on a

$$\log_2(n) = \frac{\log_{10}(n)}{\log_{10}(2)}.$$

Donc

$$\lim_{x \rightarrow \infty} \left| \frac{f(n)}{g(n)} \right| = \frac{1}{\log_{10}(2)}.$$

Ainsi, par le théorème 4.1, les fonctions  $f$  et  $g$  sont du même ordre de grandeur :

$$\log_2(n) \in O(\log_{10}(n)) \quad \text{et} \quad \log_{10}(n) \in O(\log_2(n)).$$

**Théorème 4.2**

Dans la liste de fonctions ci-dessous, chaque fonction est grand-O des fonctions qui sont situées plus à droite, mais n'est pas grand-O des fonctions situées plus à gauche.

$$1, \log(n), \sqrt{n}, n, n \log(n), n^2, n^3, n^4, \dots, 2^n, 3^n, 4^n, \dots, n!, n^n$$

Autrement dit :

$$\begin{aligned} O(1) &\subset O(\log(n)) \subset O(\sqrt{n}) \subset O(n) \subset O(n \log(n)) \subset O(n^2) \subset O(n^3) \subset O(n^4) \\ &\subset \dots \subset O(2^n) \subset O(3^n) \subset O(4^n) \subset \dots \subset O(n!) \subset O(n^n). \end{aligned}$$

La figure 4.7 illustre l'imbrication de quelques-uns des sous-ensembles du théorème 4.2. La figure 4.8 illustre que les sous-ensembles  $\Theta(1), \Theta(n^2), \Theta(2^n), \dots, \Theta(n^n)$  sont disjoints, contrairement aux sous-ensembles  $O(1), O(n^2), O(2^n), \dots, O(n^n)$  qui sont emboîtés.

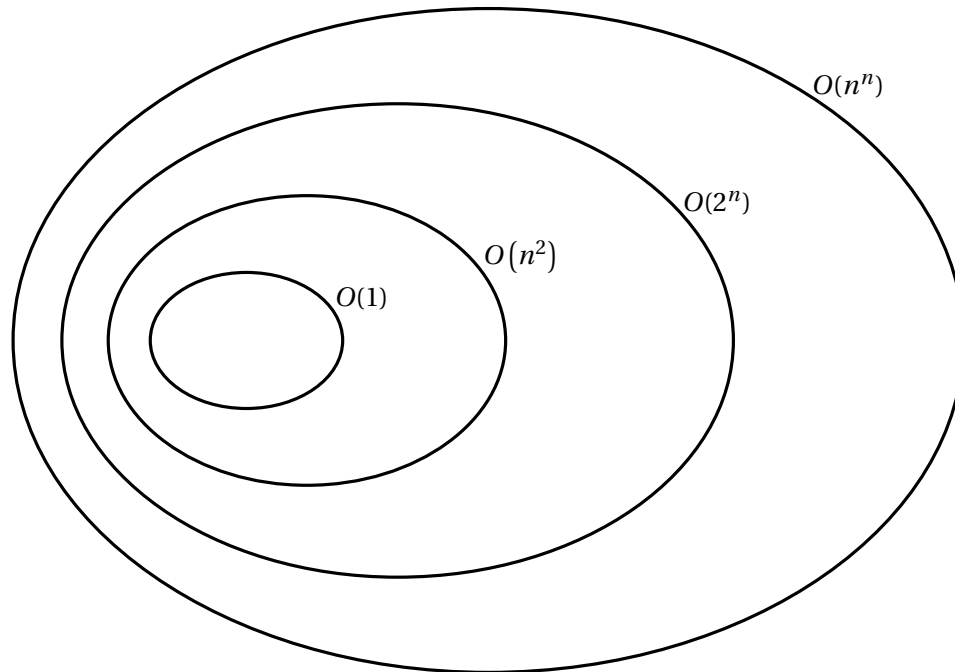


Figure 4.7 Diagramme de Venn de quatre sous-ensembles de fonctions pour la notation grand-O.

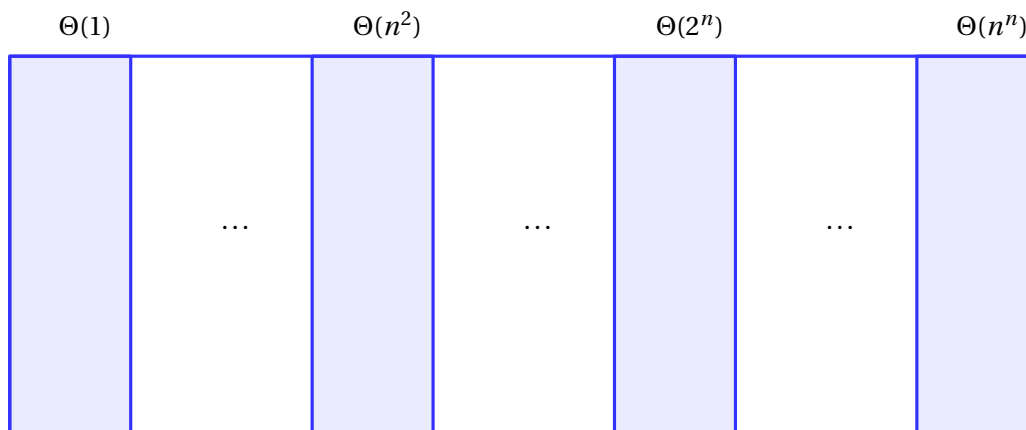


Figure 4.8 Diagramme de Venn illustrant que les sous-ensembles  $\Theta(1)$ ,  $\Theta(n^2)$ ,  $\Theta(2^n)$  et  $\Theta(n^n)$  sont disjoints, contrairement aux sous-ensembles  $O(1)$ ,  $O(n^2)$ ,  $O(2^n)$  et  $O(n^n)$  qui sont emboîtés. Le grand rectangle désigne l'ensemble  $O(n^n)$ .

Les graphes de la figure 4.9 illustrent la croissance des fonctions de la liste du théorème 4.2.

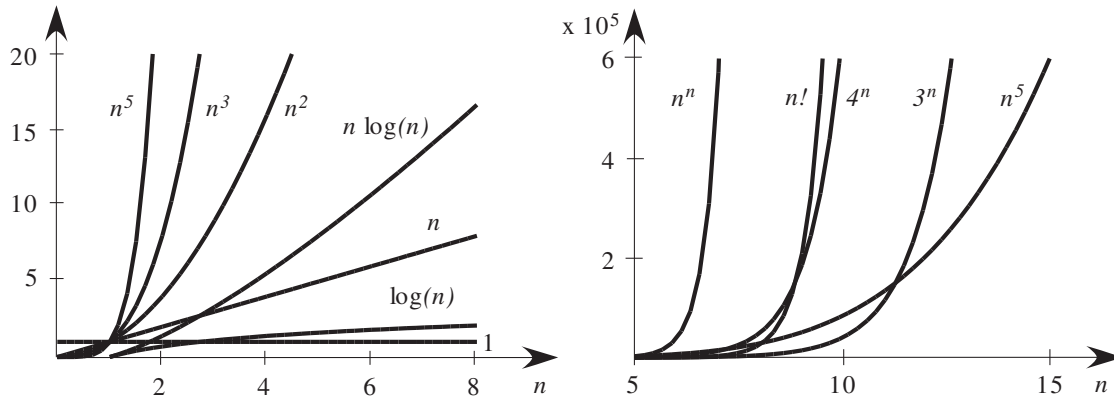


Figure 4.9 Illustration de la liste de fonctions du théorème 4.2.

Par exemple, la liste du théorème 4.2 nous informe que la fonction  $2^n$  croît plus vite que toutes les puissances de  $n$  (comme  $n^3$ ,  $n^4$  ou  $n^{10}$ ) puisqu'elle est située plus à droite. Elle finira donc par les dépasser éventuellement comme l'illustre la figure 4.10.

Une exponentielle de base  $b$  supérieure à 1 arrive toujours à dépasser un polynôme  $p(n)$ , même si la base est très près de 1 et le degré du polynôme est très grand :

$$p(n) \text{ est } O(b^n)$$

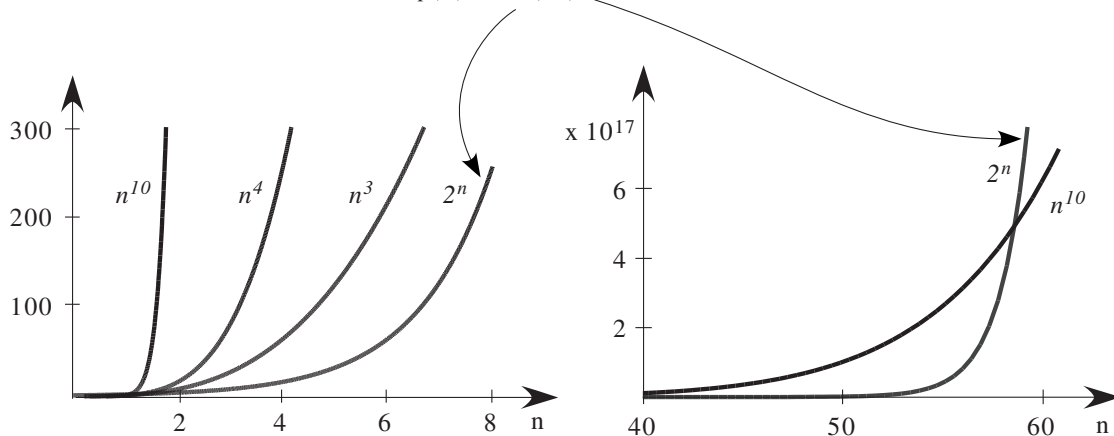


Figure 4.10 Illustration de la liste de fonctions du théorème 4.2.

### 4.2.1 Grand-O d'une fonction composée

#### Théorème 4.3 : Grand-O de la somme

Quand on additionne des fonctions, la somme appartient au plus grand des ensembles grand-O de chacune des fonctions.

Ainsi, si  $f_1 \in O(g_1)$  et  $f_2 \in O(g_2)$  et si  $O(g_1) \subseteq O(g_2)$  alors

$$(f_1 + f_2) \in O(g_2).$$

**Théorème 4.4 : Grand-O du produit**

Quand on multiplie des fonctions, le produit appartient au grand-O du produit des représentants des ensembles grand-O de chacune des fonctions.

Ainsi, si  $f_1 \in O(g_1)$  et  $f_2 \in O(g_2)$  alors

$$f_1 f_2 \in O(g_1 g_2).$$

La figure 4.11 illustre les théorèmes du grand-O de la somme et du produit de fonctions.

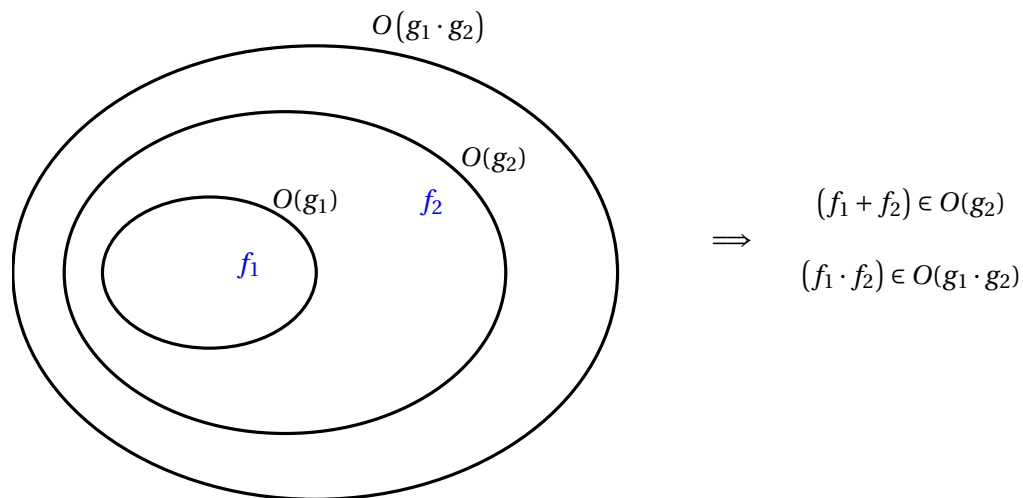


Figure 4.11 Illustration des théorèmes du grand-O de la somme et du produit.

**Théorème 4.5 : Grand-Θ d'un polynôme**

Si  $p(n)$  est un polynôme de degré  $d$  alors  $p(n) \in \Theta(n^d)$ .

**Exemple 4.12**

Soit

$$f(n) = 3\log(n) + 7n^5 + 4n.$$

Trouvez la fonction  $g(n)$  la plus simple possible telle que  $f(n) \in O(g(n))$ .

**Solution :**

En utilisant le théorème 4.1 ou le théorème 4.2, on a que

$$f_1(n) = 3 \log(n) \in O(\log(n)).$$

En utilisant le théorème 4.5, on a que

$$f_2(n) = 7n^5 + 4n \in \Theta(n^5)$$

car  $f_2$  est un polynôme de degré 5. Étant donné que  $\Theta(n^5) \subset O(n^5)$ , on obtient

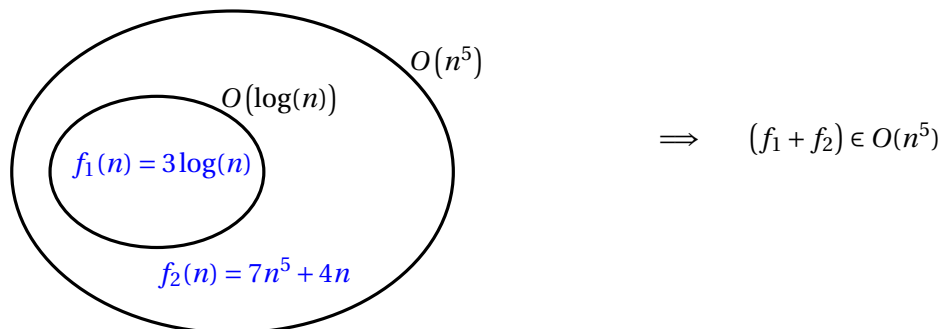
$$f_2(n) = 7n^5 + 4n \in O(n^5).$$

De plus, comme  $O(\log(n)) \subset O(n^5)$ , le théorème de la somme nous dit que

$$f(n) = f_1(n) + f_2(n) \in O(n^5).$$

Autrement dit, la fonction recherchée est

$$g(n) = n^5.$$

**Exemple 4.13**

Soit

$$f(n) = 3 \log(n) \cdot (7n^5 + 4n).$$

Trouvez la fonction  $g(n)$  la plus simple possible telle que  $f(n) \in O(g(n))$ .

**Solution :**

Par l'exemple 4.12, on sait que

$$f_1(n) = 3 \log(n) \in O(\log(n))$$

et

$$f_2(n) = 7n^5 + 4n \in O(n^5).$$

Donc, par le théorème du produit, on a

$$f(n) = f_1(n) \cdot f_2(n) \in O(n^5 \log(n)).$$

Autrement dit, la fonction recherchée est

$$g(n) = n^5 \log(n).$$



**Exemple 4.14**

Soit

$$f(x) = \sqrt{7}x^6 + 7x^5 + \pi x^3 - 194x^2 - 2112$$

Trouvez la fonction  $g(x)$  la plus simple possible telle que  $f(x) \in O(g(x))$ .

**Solution :**

La fonction  $f$  est un polynôme de degré 6. Ainsi, par le théorème 4.5,

$$\sqrt{7}x^6 + 7x^5 + \pi x^3 - 194x^2 - 2112 \in O(x^6).$$

Autrement dit, la fonction recherchée est

$$g(x) = x^6.$$

**Exemple 4.15**

Soit

$$f(x) = x^2 + 5^x$$

Trouvez la fonction  $g(x)$  la plus simple possible telle que  $f(x) \in O(g(x))$ .

**Solution :**

La fonction  $f$  est une somme. On peut traduire le théorème de la somme ainsi: *c'est la fonction dont le grand-O domine les autres qui l'emporte*. Dans le cas de  $f$ , quelle fonction domine l'autre? Allons voir la liste du théorème 4.2: c'est l'exponentielle  $5^x$  qui domine la puissance  $x^2$ , au sens où  $x^2 \in O(5^x)$ . Ainsi,

$$x^2 + 5^x \in O(5^x)$$

La fonction recherchée est donc  $g(x) = 5^x$ .

**Exemple 4.16**

Soit

$$f(n) = (14n + 3)\log(n) + 3n^2$$

Trouvez la fonction  $g(n)$  la plus simple possible telle que  $f(n) \in O(g(n))$ .

**Solution :**

On sait que,

$$14n + 3 \in O(n)$$

car c'est un polynôme de degré 1. Donc, par le théorème du produit,

$$(14n + 3)\log(n) \in O(n \log(n))$$

De plus, on a  $3n^2 \in O(n^2)$  et en regardant la liste du théorème 4.2, on voit que  $n^2$  domine  $n \log(n)$ . Ainsi, d'après le théorème sur la somme,

$$(14n + 3)\log(n) + 3n^2 \in O(n^2)$$

La fonction cherchée est donc  $g(n) = n^2$ .

**Exercices**

**4.1** Pour chaque fonction ci-dessous, déterminez la fonction  $g(n)$  la plus simple possible telle que  $f(n) \in O(g(n))$ .

(a)  $f(n) = 50\sqrt{n} + 600$

(e)  $f(n) = (n^2 + 3n^4)(\log(n) + 5)$

(b)  $f(n) = 2n^2 + 50\sqrt{n} + 600$

(f)  $f(n) = 2^n + 5^5$

(c)  $f(n) = 20n + 3\log(n)$

(g)  $f(n) = 2^n + 5^n$

(d)  $f(n) = (n^2 + 1)(n + 3\log(n))$

(h)  $f(n) = 20! + n!$

**4.2** Pour chaque fonction ci-dessous, déterminez la fonction  $g(n)$  la plus simple possible telle que  $f(n) \in O(g(n))$ .

(a)  $f(n) = 2 + n + \frac{1}{10}n^2$

(f)  $f(n) = \sqrt{n} + \sqrt{n^3} + \sqrt[3]{n} + \sqrt[3]{n^5} + \sqrt[5]{n^4}$

(b)  $f(n) = (n^2 + 1)(n + 3)$

(g)  $f(n) = \sqrt[10]{n} + \log_2(n)$

(c)  $f(n) = (n + 1)(n + 2)(n + 3)(n + 4)$

(h)  $f(n) = 2^n + n^2 + \pi$

(d)  $f(n) = (1 + n^2)(1 + n)^4$

(i)  $f(n) = 100!$

(e)  $f(n) = 4n^2 + 100n + 10$

(j)  $f(n) = 2^{n+2} + 3^{n+3} + 4^{n+4}$

(k)  $f(n) = 2^{3n+1} + 3^{2n+2}$

**4.3** Afin de résoudre un problème, vous avez le choix entre deux algorithmes. Le premier résout un problème de taille  $n$  avec un nombre d'opérations égal à  $f(n) = 2n^{\frac{3}{2}} + 5n + 10$  alors que le deuxième produit le même résultat en  $g(n) = 5n \log_5(n^2)$  opérations.

(a) Exprimez  $f(n)$  et  $g(n)$  en notation grand-O.

(b) Lorsque  $n$  est grand, lequel des deux algorithmes est-il préférable d'utiliser ?

**4.4** Même question que la précédente avec  $f(n) = n(n \log_2(n) + 2)$  et  $g(n) = (n + \sqrt{n})(\sqrt[3]{n^2} + 12)$ .

**4.5** Exprimez les fonctions suivantes en notation grand-O à l'aide d'une fonction aussi simple que possible qui reflète son comportement asymptotique.

(a)  $f(n) = 5n^4 + 10n^2 + n - 100$

(h)  $f(n) = (2^n + \log_3(n))(20 + 2\log_2(n^2))$

(b)  $f(n) = (2n^2 + 1)(\log_2(n) + \sqrt{n})$

(i)  $f(n) = (1 + n + n^2)(10 + 8n^2 + 6n^4 + 4n^6 + 2n^8)$

(c)  $f(n) = (4n^3 + 2n^2 + 15)(25n^4 + 10n - 1)$

(j)  $f(n) = \sqrt{n} + \left(\frac{1000}{1001}\right)^n$

(d)  $f(n) = 2n^2 + 2^{(n+3)}$

(k)  $f(n) = n^{100} + \left(\frac{1001}{1000}\right)^n$

(e)  $f(n) = (n^2 + n)\log(n^3 + 3) + 10!n(n^2 + 4)$

(f)  $f(n) = 10 \cdot 2^n(2^n + \sqrt{n})$

(g)  $f(n) = (25n \log_4(n) + n)(n + 45) + 2^n$

(l)  $f(n) = \log_2(n^3) + \sqrt{n}$

$$(m) f(n) = \frac{3^n}{2^n}$$

$$(n) f(n) = \log_2(n)(n^2 + \log_3(n)) + (1 + n + n^2)(\sqrt{n} + 3)$$

### 4.3 Sommations

Dans un algorithme, les boucles peuvent être vues comme des sommations. Les deux théorèmes suivants nous permettant de faire des calculs directement sur celles-ci, ramenant la détermination de la complexité d'un algorithme parfois moins évidente à un calcul mathématique plus simple.

#### Théorème 4.6 : Manipulation des sommations

Étant donné  $j, m$  et  $n$  des nombres naturels :

$$(a) \sum_{i=m}^n c = \underbrace{c + c + c + \dots + c}_{n-m+1 \text{ fois}} = (n - m + 1)c$$

$$(b) \sum_{i=m}^n (a_i + b_i) = (a_m + b_m) + (a_{m+1} + b_{m+1}) + \dots + (a_n + b_n) = \sum_{i=m}^n a_i + \sum_{i=m}^n b_i$$

$$(c) \sum_{i=m}^n (ca_i) = (ca_m) + (ca_{m+1}) + \dots + (ca_n) = c \cdot \sum_{i=m}^n a_i$$

$$(d) \sum_{i=m}^n a_i = a_m + a_{m+1} + \dots + a_n = \sum_{i=j}^n a_i - \sum_{i=j}^{m-1} a_i, \text{ si } j < m$$

#### Théorème 4.7 : Formes closes de sommation

Étant donné  $n \in \mathbb{N}$  une constante :

$$(a) \sum_{i=1}^n i = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

$$(b) \sum_{i=1}^n i^2 = 1^2 + 2^2 + 3^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

$$(c) \sum_{i=1}^n i^3 = 1^3 + 2^3 + 3^3 + \dots + n^3 = \frac{n^2(n+1)^2}{4}$$

$$(d) \sum_{i=0}^n r^i = r^0 + r^1 + r^2 + \dots + r^n = \frac{r^{n+1} - 1}{r - 1} \text{ si } r \in \mathbb{R} \setminus \{1\}$$

▷ **Démonstration** Prouvons la lettre (a). Posons  $S_n = 1 + 2 + \dots + (n-1) + n$ . Si on additionne deux fois  $S_n$ , on obtient

$$\begin{array}{rcccccccc} S_n & = & 1 & + & 2 & + & \dots & + & (n-1) & + & n \\ + S_n & = & n & + & (n-1) & + & \dots & + & 2 & + & 1 \\ \hline 2S_n & = & (n+1) & + & (n+1) & + & \dots & + & (n+1) & + & (n+1) \end{array}$$

Le membre de droite de la dernière équation contient  $n$  fois  $(n+1)$ , d'où

$$2S_n = n(n+1)$$

et ainsi

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

Nous verrons plus loin, à l'exemple 6.1, comment redémontrer ce résultat à l'aide d'une preuve par récurrence. ◀

### Exemple 4.17

Trouvez le résultat de la sommation suivante en utilisant les théorèmes 4.6 et 4.7 :

$$\sum_{i=4}^{50} (2 + 4i).$$

**Solution :**

$$\begin{aligned} \sum_{i=4}^{50} (2 + 4i) &= \sum_{i=4}^{50} 2 + \sum_{i=4}^{50} 4i && \text{par le théorème 4.6 (b)} \\ &= \sum_{i=4}^{50} 2 + 4 \sum_{i=4}^{50} i && \text{par le théorème 4.6 (c)} \\ &= (50 - 4 + 1)2 + 4 \left( \sum_{i=1}^{50} i - \sum_{i=1}^3 i \right) && \text{par le théorème 4.6 (a) et (d)} \\ &= 94 + 4 \left( \frac{50(50+1)}{2} - \frac{3(3+1)}{2} \right) && \text{par le théorème 4.7 (a)} \\ &= 94 + 5100 - 24 \\ &= 5170 \end{aligned}$$

### Exemple 4.18

Donnez la forme close de la sommation suivante en utilisant les théorèmes 4.6 et 4.7 :

$$\sum_{i=3}^n \sum_{j=1}^{n-2} 2^i.$$

**Solution :**

$$\begin{aligned} \sum_{i=3}^n \sum_{j=1}^{n-2} 2^i &= \sum_{i=3}^n ((n-2) - 1 + 1)2^i && \text{par le théorème 4.6 (a)} \\ &= (n-2) \sum_{i=3}^n 2^i && \text{par le théorème 4.6 (c)} \\ &= (n-2) \left( \sum_{i=0}^n 2^i - \sum_{i=0}^2 2^i \right) && \text{par le théorème 4.6 (d)} \\ &= (n-2) \left( \frac{2^{n+1} - 1}{2 - 1} - \frac{2^{2+1} - 1}{2 - 1} \right) && \text{par le théorème 4.7 (d)} \\ &= (n-2) (2^{n+1} - 1 - (2^3 - 1)) \\ &= (n-2) (2^{n+1} - 8) \end{aligned}$$

### Exercices

**4.6** Soit

$$f(n) = \sum_{i=2}^{n-2} (n + 2i).$$

- (a) Détaillez tous les termes de la sommation ci-dessus pour le cas où  $n = 6$ . Calculez la valeur numérique de la somme.
- (b) •Exprimez  $f(n)$  sous forme close (c'est-à-dire une formule qui ne dépend que de  $n$ , sans utiliser le symbole de sommation).  
 •Citez les théorèmes appropriés.  
 •Simplifiez votre réponse.  
 •Validez votre résultat avec le cas  $n = 6$ .

**4.7** Dans chacun des cas suivants, exprimez  $f(n)$  sous forme close (c'est-à-dire sans utiliser le symbole de sommation ni les « ... »). Tous les calculs doivent être faits **à la main** en utilisant les théorèmes 4.6 et 4.7. Simplifiez votre réponse.

(a)  $f(n) = \sum_{i=1}^n (n + i + 1)$

(e)  $f(n) = \sum_{i=1}^{2n-1} \sum_{j=0}^{n-1} (j + 1)$

(b)  $f(n) = \sum_{i=0}^{n+1} (5n + 3i + 4)$

(f)  $f(n) = \sum_{i=0}^{n-1} \sum_{j=1}^n (n^2 + i + j)$

(c)  $f(n) = \sum_{i=n}^{2n} (i + 1) - n$

(g)  $f(n) = \sum_{k=0}^n \left( \sum_{l=1}^{n^2} (k - l) + k^2 \right)$

(d)  $f(n) = \sum_{i=2}^{n-1} 4^i$

(h)  $f(n) = \sum_{k=10}^n 2 \cdot 5^k$

## 4.4 Établir la fonction de complexité d'un algorithme

Dans le calcul de la complexité d'un algorithme, nous devons d'abord construire une fonction  $f(n)$  représentant le nombre d'opérations effectuées lors de l'appel de l'algorithme avec une donnée de taille  $n$ . Cette fonction peut ensuite être identifiée à l'une des différentes classes de complexité suivantes :

$\Theta(1)$	Constante
$\Theta(\log(n))$	Logarithmique
$\Theta(n)$	Linéaire
$\Theta(n \log(n))$	$n \log(n)$
$\Theta(n^a)$	Polynomiale
$\Theta(a^n)$	Exponentielle
$\Theta(n!)$	Factorielle

**Exemple 4.19**

Comptez le nombre de comparaisons effectuées lors de l'appel de l'algorithme suivant. Quelle est sa complexité?

```

1: algorithme TriBulles( $T$  tableau de 1 à  $n$  d'éléments)
2:   pour  $i = 1$  à  $n - 1$  faire
3:     pour  $j = 1$  à  $n - i$  faire
4:       si  $T[j] > T[j + 1]$  alors
5:         échanger  $T[j]$  et  $T[j + 1]$ 
6:       fin si
7:     fin pour
8:   fin pour
9: fin algorithme

```

**Solution :**

$$\sum_{i=1}^{n-1} \sum_{j=1}^{n-i} 1 = \frac{n^2}{2} - \frac{n}{2} \in \Theta(n^2)$$

**Exemple 4.20**

Comptez le nombre de comparaisons effectuées lors de l'appel de l'algorithme suivant. Quelle est sa complexité?

```

1: algorithme TriSelection( $T$  tableau de 1 à  $n$  d'entiers)
2:   pour  $i = 1$  à  $n - 1$  faire
3:      $m := i$ 
4:     pour  $j = i + 1$  à  $n$  faire
5:       si  $T[j] < T[m]$  alors
6:          $m := j$ 
7:       fin si
8:     fin pour
9:     échanger  $T[i]$  et  $T[m]$ 
10:  fin pour
11: fin algorithme

```

**Solution :**

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n 1 = \frac{n^2}{2} - \frac{n}{2} \in \Theta(n^2)$$

**Exercices**

**4.8** Donnez la complexité de la portion de l'algorithme suivant en comptant le nombre d'additions, en fonction de  $n$ .

```

1:  $i := 2$ 
2: tant que  $i \leq n$  faire
3:    $a[i] := a[i] + 3 + n$ 
4:   pour  $j$  de 1 à  $n - i$  faire
5:      $a[i] := a[i] + 2j$ 
6:   fin pour
7:    $i := i + 1$ 
8: fin tant que

```

**4.9** On suppose que le type *élément* a été défini. Soit  $f(n)$  le nombre de comparaisons d'éléments effectuées par un appel à la fonction  $\mathbb{P}$  *au pire cas*, où  $n$  est la taille du tableau  $T$ .

```

1: fonction  $\mathbb{P}(T$ : tableau de taille  $n$ ,  $x$ : élément)
2:    $trouvé := \text{Faux}$ 
3:   pour  $i$  de 0 à 9 faire
4:     si  $T[i] == x$  alors
5:        $trouvé := \text{vrai}$ 
6:     fin si
7:   fin pour
8:   retourner  $trouvé$ 
9: fin fonction

```

Donnez  $f(n)$  et sa complexité. De plus, expliquez en vos mots ce que retourne la fonction  $\mathbb{P}$ .

**4.10** On suppose que le type *élément* a été défini. Soit  $f(n)$  le nombre de comparaisons d'éléments effectuées par un appel à la fonction  $\mathbb{P}$  *au pire cas*, où  $n$  est la taille du tableau  $T$ .

Considérons que les sous-fonctions `Traitement1` et `Traitement2` ont été définies et qu'elles requièrent respectivement  $f_1(n) = 5n$  et  $f_2(n) = \frac{n(n+1)}{2}$  comparaisons d'éléments.

```

1: fonction  $\mathbb{P}(T$ : tableau de taille  $n$ ,  $x$ : élément)
2:    $trouvé := \text{Faux}$ 
3:   pour  $i$  de 0 à 9 faire
4:     Traitement1(T)
5:   fin pour
6:   Traitement2(T)
7:   retourner  $trouvé$ 
8: fin fonction

```

Donnez  $f(n)$  et sa complexité. Vous *n'avez pas* à expliquer ce que fait la fonction  $\mathbb{P}$ .

**4.11** On suppose que le type *élément* a été défini. Soit  $f(n)$  le nombre de comparaisons d'éléments effectuées par un appel à la fonction `P` *au pire cas*, où  $n$  est la taille du tableau  $T$ .

Considérons que les sous-fonctions `Traitement1` et `Traitement2` ont été définies et qu'elles requièrent respectivement  $f_1(n) = 5n$  et  $f_2(n) = \frac{n(n+1)}{2}$  comparaisons d'éléments.

```

1: fonction P( $T$ : tableau de taille  $n$ ,  $x$ : élément)
2:   trouvé := Faux
3:   pour  $i$  de 0 à  $n - 1$  faire
4:     Traitement2( $T$ )
5:   fin pour
6:   Traitement1( $T$ )
7:   retourner trouvé
8: fin fonction

```

Donnez  $f(n)$  et sa complexité.

**4.12** Dans chacun des cas ci-dessous, donnez  $f(n)$  sous la forme d'une sommation, puis sous forme close et finalement en notation grand-O.

**Remarques :** Pour cet exercice, tous les tableaux et les chaînes de caractères sont indicés de 0 à  $n - 1$ .

- (a) La fonction `estPrésent` détermine si l'élément  $x$  est présent dans le tableau d'éléments  $T$ . On suppose que le type *élément* a été défini. Soit  $f(n)$  le nombre de comparaisons d'éléments effectuées par un appel à `estPrésent` *au pire cas*, où  $n$  est la taille du tableau  $T$ .

```

1: fonction estPrésent( $T$ : tableau de taille  $n$ ,  $x$ : élément)
2:   trouvé := Faux
3:   pour  $i$  de 0 à  $n - 1$  faire
4:     si  $T[i] == x$  alors
5:       trouvé := vrai
6:     fin si
7:   fin pour
8:   retourner trouvé
9: fin fonction

```

- (b) Voici une deuxième version de la fonction `estPrésent`,  $f(n)$  est défini comme en (a).

```

1: fonction estPrésent( $T$ : tableau de taille  $n$ ,  $x$ : élément)
2:   trouvé := Faux
3:    $i$  := 0
4:   tant que  $i < n$  et non trouvé faire
5:     trouvé := ( $T[i] == x$ )
6:      $i$  :=  $i + 1$ 
7:   fin tant que
8:   retourner trouvé
9: fin fonction

```



- (c) La fonction `nbOccurrences` compte le nombre d'occurrences de l'élément  $x$  dans le tableau d'éléments  $T$ . Soit  $f(n)$  le nombre de comparaisons d'éléments effectuées par un appel à `nbOccurrences` *au pire cas*, où  $n$  est la taille du tableau  $T$ .

```
1: fonction nbOccurrences( $T$ : tableau de taille  $n$ ,  $x$ : élément)
2:   compte := 0
3:   pour  $i$  de 0 à  $n - 1$  faire
4:     si  $T[i] == x$  alors
5:       compte := compte + 1
6:     fin si
7:   fin pour
8:   retourner compte
9: fin fonction
```

- (d) La fonction `contientDoublon` détermine si le tableau  $T$  contient deux fois la même valeur. Soit  $f(n)$  le nombre de comparaisons de cases du tableau effectuées par un appel à `contientDoublon` *au pire cas*, où  $n$  est la taille du tableau  $T$ .

```
1: fonction contientDoublon( $T$ : tableau de taille  $n$ )
2:   trouvé := faux
3:   pour  $i$  de 0 à  $n - 2$  faire
4:     pour  $j$  de  $i + 1$  à  $n - 1$  faire
5:       si  $T[i] == T[j]$  alors
6:         trouvé := vrai
7:       fin si
8:     fin pour
9:   fin pour
10:  retourner trouvé
11: fin fonction
```

- (e) ★ La fonction `fusion` prend en entrée deux tableaux triés  $T_1$ ,  $T_2$  et produit un tableau trié contenant tous les éléments de  $T_1$  et  $T_2$ . Soit  $f(n)$  le nombre de comparaisons de cases du tableau effectuées par un appel à `fusion` au pire cas, où  $n$  est la taille du plus grand des deux tableaux passés en paramètre.

(Aide: au pire cas, les deux tableaux sont de taille  $n$ )

```

1: fonction fusion( $T_1$ : Tableau de taille  $n_1$ ,  $T_2$ : Tableau de taille  $n_2$ )
2:    $i, j, k := 0$ 
3:    $T :=$  tableau d'entier de taille  $n_1 + n_2$ 
4:   tant que  $k < n_1 + n_2$  faire
5:     si  $j == n_2$  ou ( $i < n_1$  et  $T_1[i] \leq T_2[j]$ ) alors
6:        $T[k] := T_1[i]$ 
7:        $i := i + 1$ 
8:     sinon
9:        $T[k] := T_2[j]$ 
10:       $j := j + 1$ 
11:    fin si
12:     $k := k + 1$ 
13:  fin tant que
14:  retourner  $T$ 
15: fin fonction

```

- (f) ★ ★ La fonction `recherche` détermine si la chaîne de caractères  $M$  (motif) apparaît dans la chaîne de caractères  $T$  (texte). Soit  $f(n)$  le nombre de comparaisons de caractères effectuées par un appel à `recherche` au pire cas, où  $n$  est la taille du texte  $T$ .

**Aide:** pour analyser cette fonction au pire cas, on suppose que  $n$  est pair, que  $T$  est une lettre répétée  $n$  fois et que  $M$  est cette même lettre répétée  $n/2 - 1$  fois suivies d'une lettre différente. Par exemple,  $T = \underbrace{aaa \cdots a}_{n \text{ fois}}$  avec  $n$  pair et que  $M = \underbrace{aaa \cdots a}_{\frac{n}{2} - 1 \text{ fois}}b$ .

```

1: fonction recherche( $T$ : Chaîne,  $M$ : Chaîne)
2:    $lngT :=$  Taille de  $T$ 
3:    $lngM :=$  Taille de  $M$ 
4:    $trouvé :=$  Faux
5:    $i := 0$ 
6:   tant que  $i \leq (lngT - lngM)$  et non  $trouvé$  faire
7:      $j := 0$ 
8:     tant que ( $j < lngM$  et  $T[i + j] == M[j]$ ) faire
9:        $j := j + 1$ 
10:    fin tant que
11:    si  $j == lngM$  alors
12:       $trouvé :=$  vrai
13:    fin si
14:     $i := i + 1$ 
15:  fin tant que
16:  retourner  $trouvé$ 
17: fin fonction

```

## 4.5 Calculabilité et complexité

Il existe des problèmes qui ne peuvent être résolus par un programme.

En 1936, Alan Turing a fourni un exemple concret en fournissant un exemple de problème qui ne peut pas être résolu par un programme: le fameux problème de l'arrêt. Il serait donc totalement inutile de mettre sur pied une équipe d'ingénieurs en informatique pour tenter de résoudre un tel problème, comme il serait inutile de mettre sur pied une équipe d'ingénieurs en mécanique pour construire une machine à mouvement perpétuel.

**Quels sont les problèmes résolubles? Non résolubles?** Voilà une des questions que traite la théorie de la calculabilité.

De plus, parmi les problèmes résolubles, **quels sont les problèmes résolubles par un algorithme efficace?** Voilà une autre question importante, traitée par la théorie de la complexité. Mais qu'entend-on au juste par algorithme *efficace*? Il est clair qu'un algorithme dont la complexité est exponentielle n'est pas efficace. Il est plus difficile de s'entendre sur le sens du mot *efficace*, mais on convient généralement qu'il s'agit d'un algorithme dont la complexité est au plus polynomiale:  $\Theta(n^d)$  peu importe le degré. Dans les faits, le degré du polynôme devra être petit pour que l'algorithme soit réellement efficace, mais ceci n'est pas important ici.

Il existe toute une gamme de problèmes importants pour lesquels aucun algorithme à complexité polynomiale n'a été trouvé. Et plusieurs de ces problèmes, quoique fort différents, sont équivalents au sens où si l'un de ces problèmes est résoluble en temps polynomial, alors ils le sont tous. Ces problèmes sont qualifiés de **NP-complets**. En voici quelques-uns.

1. En théorie des graphes, le problème du circuit hamiltonien (HC): existe-t-il un circuit fermé permettant de parcourir chaque sommet d'un graphe une et une seule fois?
2. En théorie des graphes, la version décisionnelle du problème du voyageur de commerce (« traveling salesman », TS): pour une distance  $d$ , existe-t-il un chemin plus court que  $d$  permettant au voyageur de visiter toutes villes et de revenir à son point de départ?
3. En logique, le problème de la satisfaisabilité d'une proposition (SAT): étant donnée une proposition sous forme normale conjonctive, par exemple

$$(p \vee \neg q \vee s) \wedge (\neg p \vee q \vee s),$$

existe-t-il une fonction d'interprétation qui la rend vraie? En d'autres mots, y a-t-il un choix de valeurs de vérité pour les propositions simples  $p$ ,  $q$  et  $s$  qui rend la proposition composée vraie?

Il existe un algorithme très simple qui permet de répondre à cette dernière question: vérifier si la table de vérité de la proposition composée contient une ligne Vrai. Mais comme nous l'avons vu au chapitre 1, si la proposition contient  $n$  propositions simples, la table de vérité comporte  $2^n$  lignes. L'algorithme décrit n'est donc malheureusement pas polynomial.

Il en va de même des autres problèmes: on peut les résoudre en énumérant toutes les possibilités puis en vérifiant, mais l'énumération est d'ordre exponentiel ou factoriel alors que l'on recherche un algorithme polynomial.

Une autre caractéristique commune de ces problèmes est que, bien qu'il soit difficile (ou impossible?) de fournir un algorithme qui produit une solution en temps polynomial, il est facile de fournir un algorithme qui vérifie en temps polynomial si une solution donnée est valide.

**Les problèmes NP-complets: très longs à résoudre, très rapides à vérifier.**

**Une question à un million de dollars!**

Le fait qu'à ce jour aucun des problèmes NP-complets n'ait été résolu en temps polynomial nous porte à croire que cela est impossible. Mais l'impossibilité de les résoudre en temps polynomial n'a pas été démontrée. Cette question ouverte constitue d'ailleurs un des sept *problèmes du millénaire* pour chacun desquels un million de dollars est offert en récompense par le Clay Mathematics Institute of Cambridge. Au moment d'écrire ces lignes, seulement un des sept problèmes a été résolu.

Cela ne veut pas dire qu'il faille abandonner toute tentative de résoudre les problèmes NP-complets par des algorithmes efficaces. Par exemple, si le problème en est un d'optimisation, il est peut-être impossible de trouver un algorithme efficace produisant *la* solution optimale, mais possible d'en trouver un produisant une solution *très proche* de la solution optimale. Parfois, il sera possible de trouver un algorithme fonctionnant efficacement pour la grande majorité des cas traités et on s'en contentera.

Si vous êtes confrontés à un problème et que vous établissez qu'il est NP-complet, vous saurez qu'il vaudra mieux chercher une solution approximative. À moins de révolutionner l'informatique et de devenir millionnaire!

## Chapitre 5

# Algorithmes récursifs

En programmation, on dit qu'une fonction est **récursive** si elle s'appelle elle-même. Par exemple, étant donné un ensemble non vide, la fonction suivante génère la liste de toutes les permutations des éléments de cet ensemble.

```
1: fonction Permutations( $A$ : Ensemble non vide)
2:   si  $|A| == 1$  alors
3:     retourner la liste formée de l'unique élément de  $A$ 
4:   sinon
5:      $L =$  liste vide
6:     pour chaque  $x \in A$  faire
7:       pour chaque permutation  $p$  dans Permutations( $A \setminus \{x\}$ ) faire
8:         ajouter  $x$  au début de  $p$ 
9:         ajouter  $p$  à  $L$ 
10:      fin pour
11:    fin pour
12:    retourner  $L$ 
13:  fin si
14: fin fonction
```

On peut vérifier que :

- Permutations( $\{1\}$ ) retourne [1].
- Permutations( $\{1,2\}$ ) retourne [12, 21].
- Permutations( $\{1,2,3\}$ ) retourne [123, 132, 213, 231, 312, 321].
- Permutations( $\{1,2,3,4\}$ ) retourne [1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312, 4321].

De manière plus générale, on dit qu'un algorithme est **récursif** si, pour résoudre un problème d'une certaine taille, il trouve d'abord la solution à une ou plusieurs instances plus petites du même problème. Dans le cas de la fonction Permutations, pour générer la liste des permutations de  $n$  éléments, la fonction effectue  $n$  appels récursifs et chacun de ces appels récursifs génère la liste des permutations de  $n - 1$  éléments. Finalement, on remarque un cas particulier: si  $A$  contient un seul élément alors la solution est obtenue directement, il n'y a pas d'appels récursifs. Il s'agit du **cas de base**.

## 5.1 Définition et exemples d'algorithmes récursifs

### Définition 5.1 : Algorithme récursif

Un algorithme est **récursif** s'il permet de résoudre un problème en le réduisant au même problème avec une ou plusieurs entrées de plus petites tailles (et donc avec un ou plusieurs appels à lui-même).

Pour analyser la complexité d'un algorithme récursif, on définit une relation de récurrence (ou fonction récursive) qui compte le nombre d'opérations effectuées lors de l'appel de celui-ci. On trouve ensuite une forme close pour son terme général, ce qui nous permet de déterminer sa complexité.

### Exemple 5.1

Soit  $a \in \mathbb{R}$  et  $n \in \mathbb{N}$ .

- Implémentez sur *Nspire* une fonction récursive pour calculer  $a^n$ , où  $a \neq 0$  et  $n \in \mathbb{N}$ . Cette fonction retournera `undef` si  $a = 0$  et  $n = 0$ .
- Donnez sa complexité. *Comptez le nombre de multiplications effectuées.*

### Solution :

- Voici une fonction puissance définie avec l'éditeur de programme, puis une fonction  $p$  tout à fait équivalente définie directement dans une fenêtre de calculs avec une fonction par morceaux, suivie de quelques tests.

```
"puissance" enregistr. effectué
Define puissance(a,n)=
Func
© a réel, n naturel; retourne a^n ou undef si a=0 et n=0 ou si n∉N.
If a=0 and n=0 or n<0 or mod(n,1)≠0 Then
Return undef
Else
If n=0 Then
Return 1
Else
Return a * puissance(a,n-1)
EndIf
EndIf
EndFunc
```

```
p(a,n):=
{undef, a=0 and n=0 or n<0 or mod(n,1)≠0
0, a=0 and n≠0
1, a≠0 and n=0
a * p(a,n-1), Else
Terminé
tests:={p(0,0),p(8,-2),p(5,1.3),p(8,0),p(8,1),p(2,4),p(-3,2)}
{undef,undef,undef,1,8,16,9}
p(2,8) 256
p(2/5,3) 8/125
```

- Soit  $f(n)$  la fonction qui compte le nombre de multiplications effectuées lors d'un appel à `puissance(a, n)`. En analysant le code de la fonction `puissance`, on conclut que  $f$  est définie par:

$$f(n) = \begin{cases} 0 & \text{si } n = 0, \\ f(n-1) + 1 & \text{sinon.} \end{cases}$$



**Exemple 5.3**

Donnez une fonction récursive qui déterminera si oui ou non l'élément  $x$  est présent dans la liste  $t$ . Pour ce faire, si la liste possède plus d'un élément, la fonction séparera la liste en 2 sous-listes et vérifiera si  $x$  est présent dans chacune d'elle. Programmez cette fonction sur *Nspire*.

Information utile: en TI-Basic, la commande `left(l,n)` retourne la sous-liste des  $n$  premiers éléments de  $l$ , la commande `right(l,n)` retourne la sous-liste des  $n$  derniers éléments de  $l$  et la commande `dim(t)` retourne la dimension de la liste  $t$ . Rappelons aussi que les listes sont indexées à partir de 1.

**Solution :**

Voici une fonction définie avec l'éditeur de programme, puis une fonction  $p$  tout à fait équivalente définie directement dans une fenêtre de calculs avec une fonction par morceaux, suivie de quelques tests.

```

est_present 10/10
Define est_present(x,t)=
Func
© x élément recherché, t liste: détermine si x est dans t
Local n,gauche,droite
© n est la dimension de la liste t, gauche est la première moitié de la
  liste t et droite est l'autre moitié de la liste t.
n:=dim(t)
If n=1 Then
  Return x=t[1]
Else
  gauche:=left(t,ceiling(n/2)): droite:=right(t,floor(n/2))
  Return est_present(x,gauche) or est_present(x,droite)
EndIf
EndFunc

```

$p(x,t):=$	$\begin{cases} x=t[1], & \dim(t)=1 \text{ Terminé} \\ p\left(x,\text{left}\left(t,\text{ceiling}\left(\frac{\dim(t)}{2}\right)\right)\right) \text{ or } p\left(x,\text{right}\left(t,\text{floor}\left(\frac{\dim(t)}{2}\right)\right)\right), & \text{Else} \end{cases}$
$p(123,\{9,11,17,125,0\})$	false
$p(125,\{9,11,17,125,0\})$	true
$t2:=\{4,4,45,4,75,457,1,8,90,5,3,76\}$	$\{4,4,45,4,75,457,1,8,90,5,3,76\}$
$\dim(t2)$	12
$p(90,t2)$	true
$p(91,t2)$	false



## 5.2 Fonctions récursives et relations de récurrence

Lorsqu'on considère des fonctions au sens mathématique, tel que vu au Chapitre 1, on définit la récursivité de la façon suivante.

### Définition 5.2 : Fonction récursive

Une **fonction récursive** est une fonction dont la définition contient un ou plusieurs appels à elle-même. Si  $f$  est une fonction dont le domaine est  $\mathbb{N}$ , on procède ainsi pour la définir :

**Cas de base :** définir  $f(n)$  pour une ou plusieurs valeurs initiales (comme par exemple 0, 1 et 2 s'il y a 3 cas de base)

**Étape récursive :** donner une règle pour calculer  $f(n)$  à l'aide de  $f(0), f(1), \dots, f(n-2), f(n-1)$ .

Note : on écrit parfois  $f_n$  au lieu de  $f(n)$ .

### Exemple 5.4 (à compléter en classe)

Définir récursivement les fonctions suivantes, où  $n \in \mathbb{N}$  :

(a)  $f(n) = n!$

(b)  $g(n) = a^n$ , où  $a \in \mathbb{R}$ ,  $a \neq 0$ .

**Solution :**

### Définition 5.3 : Relation de récurrence, terme général, solution

Une **relation de récurrence** d'une suite  $\{a_n\}$  est une équation qui exprime le **terme général**  $a_n$  en fonction de  $a_0, a_1, \dots, a_{n-1}$ . On dit qu'une suite est **solution** de la récurrence si ses termes satisfont la relation de récurrence.

### Exemple 5.5

Le suite de Fibonacci 0, 1, 1, 2, 3, 5, 8, 13, 21, ... est définie par la relation de récurrence

**Cas de base :**  $f_0 = 0, f_1 = 1$

**Relation de récurrence (ou étape récursive) :**  $f_n = f_{n-1} + f_{n-2}, n \geq 2$ .

Sous forme de fonction récursive, ceci est noté

$$f(n) = \begin{cases} 0 & \text{si } n = 0, \\ 1 & \text{si } n = 1, \\ f(n-1) + f(n-2) & \text{si } n \geq 2. \end{cases}$$

Une relation de récurrence est donc une définition récursive des termes de la suite  $\{a_n\}$ . Formellement, une suite est une fonction  $f$  dont le domaine est un sous-ensemble des entiers et on note  $f(n) = a_n$  son terme général. Une relation de récurrence peut donc être vue comme la définition récursive d'une fonction.

### 5.2.1 Résolution de relation de récurrence par la méthode itérative

Lorsque l'on considère des relations de récurrences simples, il est parfois possible de *deviner* une forme close pour la solution en itérant directement la relation de récurrence dans sa propre définition. C'est ce qu'on appelle la **méthode itérative**.

#### Exemple 5.6

Résolvez la relation de récurrence  $a_0 = 2; \quad a_n = a_{n-1} + 3, \quad n \geq 1$ .

*C'est-à-dire: trouvez une forme close, en fonction de  $n$ , pour son terme général.*

#### Solution :

On trouve

$$\begin{aligned} a_0 &= 2 \\ a_1 &= a_0 + 3 \\ a_2 &= a_1 + 3 = (a_0 + 3) + 3 = a_0 + 2 \cdot 3 \\ a_3 &= a_2 + 3 = (a_0 + 2 \cdot 3) + 3 = a_0 + 3 \cdot 3 \\ a_4 &= a_3 + 3 = (a_0 + 3 \cdot 3) + 3 = a_0 + 4 \cdot 3 \\ &\vdots \\ a_n &= a_0 + n \cdot 3 = a_0 + 3n \end{aligned}$$

En utilisant  $a_0 = 2$  donné par la définition, on obtient la solution:  $a_n = 3n + 2$ .

**Remarque:** à l'exemple 6.4 nous verrons comment prouver que  $a_n = 3n + 2$  est bien une solution.

#### Exemple 5.7

Résolvez la relation de récurrence  $b_0 = 5; \quad b_n = b_{n-1} + 4n, \quad n \geq 1$ .

*C'est-à-dire: trouvez une forme close, en fonction de  $n$ , pour son terme général.*

#### Solution :

On trouve

$$\begin{aligned} b_0 &= 5 \\ b_1 &= b_0 + 4 \cdot 1 \\ b_2 &= b_1 + 4 \cdot 2 = (b_0 + 4 \cdot 1) + 4 \cdot 2 = b_0 + 4(1 + 2) \\ b_3 &= b_2 + 4 \cdot 3 = (b_0 + 4(1 + 2)) + 4 \cdot 3 = b_0 + 4(1 + 2 + 3) \\ b_4 &= b_3 + 4 \cdot 4 = (b_0 + 4(1 + 2 + 3)) + 4 \cdot 4 = b_0 + 4(1 + 2 + 3 + 4) \\ &\vdots \\ b_n &= b_0 + 4(1 + 2 + 3 + 4 + \cdots + n) = 5 + 4 \frac{n(n+1)}{2} = 2n^2 + 2n + 5 \end{aligned}$$

**Remarque:** à l'exercice 6.3 nous prouverons que la solution obtenue est bien la bonne. Pour l'instant, nous pouvons tester la forme close obtenue avec quelques valeurs de  $n$ .

Expression	Result
$b(n) := \begin{cases} 5, & n=0 \\ b(n-1) + 4 \cdot n, & \text{Else} \end{cases}$ Terminé	
$b(10)$	225
$2 \cdot n^2 + 2 \cdot n + 5   n=10$	225
$b(21)$	929
$2 \cdot n^2 + 2 \cdot n + 5   n=21$	929

### Exercices

**5.1** Déterminez les valeurs de  $f(n)$  pour  $n$  allant de 1 à 7, inclusivement. De plus, décrivez dans vos mots ce que calcule la fonction  $f$ .

(a)  $f(0) = 0$ ,  $f(n) = f(n-1) + 2$ , pour  $n \geq 1$ .

(b)  $f(0) = 1$ ,  $f(n) = f(n-1) + 2$ , pour  $n \geq 1$ .

(c)  $f(n) = \begin{cases} 0 & \text{si } n = 0, \\ 1 & \text{si } n = 1, \\ f(n-2) & \text{si } n \geq 2. \end{cases}$

(d)  $f(0) = 0$ ,  $f(n) = f(n-1) + 2n - 1$ , pour  $n \geq 1$ .

(e)  $f(1) = 1$ ,  $f(n) = f(n-1) + 10^{n-1}$ , pour  $n \geq 1$ .

(f) ★  $f(n) = \begin{cases} 0 & \text{si } n = 1 \\ g(n, n-1) & \text{si } n \geq 2 \end{cases}$  et  $g(a, b) = \begin{cases} 1 & \text{si } b = 1, \\ 0 & \text{si } b \geq 2 \text{ et } a \bmod b = 0, \\ g(a, b-1) & \text{si } b \geq 2 \text{ et } a \bmod b \neq 0. \end{cases}$

**5.2** Utilisez la méthode itérative afin de trouver une solution aux relations de récurrence suivantes, en procédant à tous les calculs à la main :

(a)  $a_0 = 1$ ,  $a_n = 2a_{n-1}$ , pour  $n \geq 1$ .

(b)  $b_0 = 0$ ,  $b_n = b_{n-1} + 2n$ , pour  $n \geq 1$ .

(c)  $c_0 = 1$ ,  $c_n = c_{n-1} + n + 2$ , pour  $n \geq 1$ .

(d)  $f(0) = 2$ ,  $f(n) = 3f(n-1) + 2$ , pour  $n \geq 1$ .

(e)  $f(0) = 5$ ,  $f(n) = 3f(n-1) + 2$ , pour  $n \geq 1$ .

(f)  $f(0) = 5$ ,  $f(n) = nf(n-1)$ , pour  $n \geq 1$ .

(g)  $h_0 = 3$ ,  $h_n = 10h_{n-1}$ , pour  $n \geq 1$ .

(h)  $f(2) = 3$ ,  $f(n) = 4f(n-1) + 6$ , pour  $n \geq 3$ .

### 5.3 Algorithmes de type diviser pour régner

Nous avons vu à l'exemple 5.1 que lorsqu'on analyse un algorithme récursif, on obtient généralement une fonction de complexité qui est elle-même récursive. Par exemple, on considère les deux algorithmes récursifs suivants.

<pre> 1: <b>fonction</b> Fibon(<i>n</i>: Entier) 2:   <b>si</b> <i>n</i> == 0 <b>alors</b> 3:     <b>retourner</b> 0 4:   <b>sinon si</b> <i>n</i> == 1 <b>alors</b> 5:     <b>retourner</b> 1 6:   <b>sinon</b> 7:     <b>retourner</b> Fibon(<i>n</i> - 1) + Fibon(<i>n</i> - 2) 8:   <b>fin si</b> 9: <b>fin fonction</b> </pre>	<pre> 1: <b>fonction</b> Somme(<i>t</i>: Tableau d'entiers de taille <i>n</i>) 2:   <b>si</b> <i>n</i> == 1 <b>alors</b> 3:     <b>retourner</b> <i>t</i>[1] 4:   <b>sinon</b> 5:     <i>S<sub>g</sub></i> = Somme(<i>t</i>[1..<i>n</i>/2]) // moitié gauche 6:     <i>S<sub>d</sub></i> = Somme(<i>t</i>[<i>n</i>/2 + 1..<i>n</i>]) // moitié droite 7:     <b>retourner</b> <i>S<sub>g</sub></i> + <i>S<sub>d</sub></i> 8:   <b>fin si</b> 9: <b>fin fonction</b> </pre>
---	--

Soit  $f(n)$  le nombre d'additions effectuées par l'algorithme `Fibon` pour une entrée  $n$  et  $g(n)$  le nombre d'additions effectuées par l'algorithme `Somme` pour un tableau de taille  $n$ . Alors

$$f(n) = \begin{cases} 0 & \text{si } n = 0 \text{ ou } n = 1, \\ f(n-1) + f(n-2) + 1 & \text{si } n \geq 2, \end{cases}$$

$$g(n) = \begin{cases} 0 & \text{si } n = 1, \\ 2g(n/2) + 1 & \text{si } n \geq 2. \end{cases}$$

Ces deux fonctions de complexité présentent une différence fondamentale :

- pour la fonction  $f$ , le paramètre des termes récursifs est  $n$  auquel on **soustrait une constante**;
- pour la fonction  $g$ , le paramètre des termes récursifs est  $n$  **divisé par une constante**.

Nous avons vu, à la section 5.2.1, comment s'y prendre pour résoudre certaines relations de récurrence du type de la fonction  $f$  par méthode itérative. De façon très similaire, nous verrons à la section 5.3.2 comment résoudre certaines relations de récurrence du type de la fonction  $g$  par la méthode itérative.

### 5.3.1 Algorithmes et relations de récurrence de type diviser pour régner

#### Définition 5.4 : Algorithme de type diviser pour régner (ou algorithme de fractionnement)

Un **algorithme de type diviser pour régner** (ou algorithme de fractionnement) est un algorithme qui procède de la façon suivante. Étant donné un problème à résoudre, il le fractionne en sous-problèmes dont la taille est une fraction de la taille originale. Le fractionnement est appliqué récursivement jusqu'à l'obtention d'un cas de base qu'il résout directement. Finalement, les solutions aux sous-problèmes sont combinées pour former la solution au problème initial.

#### Définition 5.5 : Relation de récurrence de type diviser pour régner

Considérons un algorithme de fractionnement qui procède de la façon suivante pour résoudre un problème de taille  $n$ .

- L'algorithme fractionne le problème de taille  $n$  en  $a$  sous problèmes de taille  $n/b$ .
- Il traite les  $a$  sous problèmes, puis recombine les solutions pour ainsi obtenir la solution au problème de taille  $n$ . On note  $g(n)$  le nombre d'opérations requises pour cette étape de recombinaison.
- On note  $m$  le nombre d'opérations requises pour traiter un problème de taille 1.

Si l'entier  $n$  est une puissance de  $b$ , alors le nombre d'opérations pour résoudre le problème de taille  $n$ , noté  $f(n)$ , satisfait la relation de récurrence

$$f(n) = \begin{cases} m & \text{si } n = 1, \\ af(n/b) + g(n) & \text{si } n \geq b. \end{cases}$$

Pour cette raison, une telle relation est donc appelée **relation de type diviser pour régner**.

Pour le reste de cette section, on supposera toujours que  $n$  est une puissance de  $b$ . Cette hypothèse permet de simplifier grandement les calculs sans affecter le résultat lorsqu'on passe à la notation grand-O.

### 5.3.2 Résolution de relation de type diviser pour régner par la méthode itérative

En reprenant la méthode itérative présentée à la section 5.2.1, nous présentons ici une manière de résoudre les relations de récurrence par fractionnement.

#### Exemple 5.8

Considérons la fonction de complexité de la fonction **Somme** présentée à la page 164.

$$g(n) = \begin{cases} 0 & \text{si } n = 1, \\ 2g(n/2) + 1 & \text{si } n \geq 2. \end{cases}$$

Trouvez une forme close pour  $g(n)$  lorsque  $n$  est une puissance de 2 et donnez sa complexité.

#### Solution :

Bien que  $g(1) = 0$ , nous écrivons explicitement le terme  $g(1)$  jusqu'à la fin pour que l'exemple demeure le plus général possible.

$n$	$g(n)$
$1 = 2^0$	$g(1) = 0$
$2 = 2^1$	$g(2) = 2g(1) + 1$
$4 = 2^2$	$g(4) = 2g(2) + 1 = 2(2g(1) + 1) + 1$ $= 2^2g(1) + 2 \cdot 1 + 1$
$8 = 2^3$	$g(8) = 2g(4) + 1 = 2(2^2g(1) + 2 \cdot 1 + 1) + 1$ $= 2^3g(1) + 2^2 \cdot 1 + 2 \cdot 1 + 1$
$16 = 2^4$	$g(16) = 2g(8) + 1 = 2(2^3g(1) + 2^2 \cdot 1 + 2 \cdot 1 + 1) + 1$ $= 2^4g(1) + 2^3 \cdot 1 + 2^2 \cdot 1 + 2 \cdot 1 + 1$
$32 = 2^5$	$g(32) = 2g(16) + 1 = 2(2^4g(1) + 2^3 \cdot 1 + 2^2 \cdot 1 + 2 \cdot 1 + 1) + 1$ $= 2^5g(1) + 2^4 \cdot 1 + 2^3 \cdot 1 + 2^2 \cdot 1 + 2 \cdot 1 + 1$
$n = 2^k$	$g(n) = 2^k g(1) + 2^{k-1} \cdot 1 + \dots + 2^3 \cdot 1 + 2^2 \cdot 1 + 2 \cdot 1 + 1$ $= 2^k g(1) + \sum_{i=0}^{k-1} 2^i$ $= 2^k g(1) + \frac{2^{k-1+1} - 1}{2 - 1}$ par théorème 4.7(d) $= n \cdot g(1) + n - 1$ $= n \cdot 0 + n - 1$

Ainsi,  $g(n) = n - 1 \in O(n)$ .

**Exemple 5.9**

Lorsqu'elle est implémentée de façon récursive, la fouille dichotomique présentée à l'exemple 4.5 produit la fonction de complexité suivante :

$$f(n) = \begin{cases} 1 & \text{si } n = 1, \\ f(n/2) + 1 & \text{si } n \geq 2. \end{cases}$$

Trouvez une forme close pour  $f(n)$  lorsque  $n$  est une puissance de 2 et donnez sa complexité.

**Solution :**

On trouve

$n$	$f(n)$
$1 = 2^0$	$f(1) = 1$
$2 = 2^1$	$f(2) = f(1) + 1$
$4 = 2^2$	$f(4) = f(2) + 1 = (f(1) + 1) + 1$ $= f(1) + 2$
$8 = 2^3$	$f(8) = f(4) + 1 = (f(1) + 2) + 1$ $= f(1) + 3$
$16 = 2^4$	$f(16) = f(8) + 1 = (f(1) + 3) + 1$ $= f(1) + 4$
$n = 2^k$	$f(n) = f(1) + k$ $= f(1) + \log_2(n) \quad \text{car } n = 2^k \leftrightarrow k = \log_2(n)$ $= 1 + \log_2(n) \quad \text{car } f(1) = 1$

Ainsi,  $f(n) = 1 + \log_2(n) \in O(\log(n))$ .

### Exercices

**5.3** Pour cet exercice, n'utilisez la calculatrice que pour les opérations arithmétiques de base.

- (a) Soit  $f(n) = f(n/2) + 1$ ,  $f(1) = 1$ . Calculez  $f(128)$ .
- (b) Soit  $f(n) = 2f(n/3) + n^2$ ,  $f(1) = 3$ . Calculez  $f(81)$ .
- (c) Soit  $f(n) = 4f(n/2) + n$ ,  $f(1) = 5$ . Calculez  $f(64)$ .
- (d) Soit  $f(n) = 81f(n/9) + 3n^2$ ,  $f(1) = 1$ . Calculez  $f(729)$ .

**5.4** Trouvez une forme close pour  $f(n)$  et donnez sa complexité.

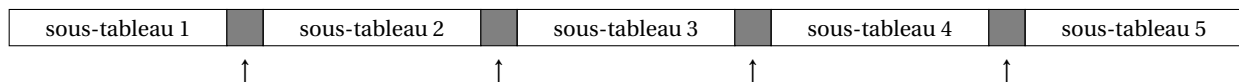
- (a)  $f(n) = 2f(n/2) + 3$ ,  $f(1) = 3$ ,
- (b)  $f(n) = 4f(n/2) + n$ ,  $f(1) = 1$ ,
- (c)  $f(n) = 3f(n/3) + 2n + 1$ ,  $f(1) = 1$ ,
- (d)  $f(n) = 9f(n/3) + 3n^2$ ,  $f(1) = 2$ .

**5.5 Fouille  $d$ -chotomique**

Afin de déterminer si une valeur  $x$  est présente dans un tableau trié de taille  $n$ , on peut utiliser la *fouille dichotomique*. Vulgairement, l'algorithme de la fouille dichotomique se résume à :

- Si  $n = 1$  alors tester si l'unique case est  $x$ .
- Sinon, comparer  $x$  avec la case centrale du tableau. Si  $x$  est plus petit, alors on recommence avec la moitié gauche, sinon avec la moitié droite.

La fouille  $d$ -chotomique est une généralisation qui consiste à diviser le tableau en  $d$  sous-tableaux (où le paramètre  $d$  est un nombre naturel). On compare alors  $x$  avec les cases situées aux extrémités de ces sous-tableaux, puis on effectue un appel récursif sur l'unique sous-tableau où  $x$  peut possiblement se trouver. Au pire cas, il faut effectuer  $d - 1$  comparaisons. Par exemple, pour  $d = 5$ , la figure ci-dessous montre les 4 cases à tester afin de déterminer lequel des 5 sous-tableaux doit être considéré pour l'appel récursif.



La fonction  $f(n)$  qui compte le nombre de comparaisons effectuées lors d'une fouille  $d$ -chotomique est

$$f(n) = f(n/d) + (d - 1), \quad f(1) = 1.$$

Trouvez une forme close pour  $f(n)$  lorsque  $n$  est une puissance de  $d$  et donnez sa complexité.



**5.6 Tri fusion** L'algorithme du *tri fusion* est un algorithme de fractionnement qui effectue un tri de la manière suivante. Soit  $n$  la taille du tableau, si

- $n < 2$  alors il n'y a rien à faire, le tableau est trié.
- $n \geq 2$  alors on coupe le tableau en deux moitiés, la gauche et la droite. On trie récursivement ces deux moitiés puis on les fusionne en un tableau trié.

```

1: fonction TriFusion( $T$ : Tableau)
2:    $n :=$  taille de  $T$ 
3:   si  $n \geq 2$  alors
4:      $gauche := T[1..n/2]$ 
5:      $droite := T[n/2 + 1..n]$ 
6:     TriFusion( $gauche$ )
7:     TriFusion( $droite$ )
8:      $T :=$  fusion( $gauche, droite$ )
9:   fin si
10: fin fonction

```

**Remarque:** Cette fonction utilise la fonction `fusion` vue à l'exercice 4.12(e). Dans le cadre de cet exercice, il a été établi que pour deux tableaux de taille  $n/2$ , le nombre de comparaisons effectuées par `fusion` est  $n - 1$ .

Soit  $f(n)$  le nombre de comparaisons de cases du tableau effectuées par un appel à `TriFusion` au pire cas, où  $n$  est la taille du tableau  $T$ .

- (a) Donnez les valeurs de  $a, b \in \mathbb{N}$  et la fonction  $g(n)$  permettant d'exprimer  $f(n)$  sous la forme

$$f(n) = af(n/b) + g(n).$$

- (b) Trouvez une forme close pour  $f(n)$  lorsque  $n$  est une puissance de  $b$  et donnez sa complexité.



# Chapitre 6

## Preuve par récurrence

Étant donné une fonction propositionnelle  $P(n)$ , si on veut montrer que  $P(n)$  est vraie pour  $n \in \{0, 1, 2, \dots\}$ , alors on vérifie que la proposition  $P(0)$  est vraie, que  $P(1)$  est vraie et, finalement, que  $P(2)$  est vraie. De même, si on veut montrer que la fonction  $P(n)$  est vraie pour tout entier de 0 à 1 000 000, on peut encore procéder une valeur à la fois (préférentiellement avec un ordinateur). Par contre, si on veut montrer que  $P(n)$  est vraie pour tout  $n \in \mathbb{N}$ , on a un problème : il y a une infinité de propositions à vérifier !

Comment vérifier une infinité de propositions en un nombre fini (et idéalement petit) d'étapes ? Le *principe de récurrence* est un raisonnement mathématique rigoureux qui permet justement de faire cela.

### 6.1 Preuve par récurrence simple

Le principe du raisonnement par récurrence peut être illustré simplement par l'analogie des dominos. Soit une suite infinie de dominos alignés de telle sorte que si on fait tomber le premier, tous les dominos tombent par réaction en chaîne.

Pour s'assurer que tous les dominos tombent, il suffit

1. de veiller à faire tomber le premier domino (**cas de base**) ;
2. de s'assurer que la distance entre chacun est telle que si un domino tombe, alors le suivant tombe lui aussi (**étape de récurrence**).



En utilisant une écriture plus mathématique, on peut définir la fonction propositionnelle  $P(n)$  : « le  $n^{\text{e}}$  domino tombe ». Pour démontrer que  $P(n)$  est vraie pour tout  $n \geq 1$ , il suffit de montrer :

#### 1. Cas de base

Montrer que  $P(1)$  : « le 1<sup>er</sup> domino tombe » est vraie.

#### 2. Étape de récurrence

Montrer que si le premier domino tombe, alors le second aussi, et que si le deuxième domino tombe, alors le troisième aussi, etc. Ceci se traduit par  $P(1) \rightarrow P(2) \equiv \mathbf{vrai}$ , et  $P(2) \rightarrow P(3) \equiv \mathbf{vrai}$ , ainsi de suite. De manière plus succincte, on écrit

$$\forall k \geq 1, P(k) \rightarrow P(k+1).$$

Ainsi, en démontrant le cas de base et l'étape de récurrence, on montre que toutes les propositions  $P(1), P(2), P(3), P(4), \dots$  sont vraies. En effet, par la règle d'inférence du *Modus ponens*, on obtient :

$$\begin{array}{ccc} P(1) & P(2) & P(3) \\ \frac{P(1) \rightarrow P(2)}{P(2)} & \frac{P(2) \rightarrow P(3)}{P(3)} & \frac{P(3) \rightarrow P(4)}{P(4)} \quad \dots \end{array}$$

De manière plus générale, soit  $P(n)$  une fonction propositionnelle dont le domaine est  $\mathbb{N}$ . La règle d'inférence suivante permet de montrer que  $P(n)$  est vraie pour tout nombre naturel supérieur ou égal à  $n_0$ .

**Théorème 6.1 : Principe du raisonnement par récurrence.**

Soit  $P(n)$  une fonction propositionnelle portant sur un nombre naturel  $n$  et soit  $n_0 \in \mathbb{N}$ .

$$P(n_0) \wedge [\forall k \geq n_0, P(k) \rightarrow P(k+1)] \longrightarrow \forall n \geq n_0, P(n)$$

Ainsi, pour démontrer  $\forall n \geq n_0, P(n)$  en utilisant le principe du raisonnement par récurrence, on doit procéder en deux étapes, qui sont décrites dans l'encadré qui suit.

**Comment démontrer  $\forall n \geq n_0, P(n)$ , par récurrence simple ?**

On doit :

**1. Cas de base**

Montrer que la fonction propositionnelle est vraie pour la première valeur (on commence habituellement à 0 ou 1, mais nous notons ici  $n_0$  cette première valeur) :

$$P(n_0) \equiv \text{vrai.}$$

**2. Étape de récurrence**

Montrer que si la fonction propositionnelle est vraie pour un entier  $k$ , alors elle est aussi vraie pour l'entier suivant :

$$\forall k \geq n_0, P(k) \rightarrow P(k+1).$$

**Exemple 6.1**

Démontrez la première partie du Théorème 4.7 sur les sommations: pour tout entier  $n$  strictement positif, la somme des entiers de 1 à  $n$  est égale à  $\frac{n(n+1)}{2}$ .

**Solution :**

On définit  $P(n)$ : «  $\sum_{i=1}^n i = \frac{n(n+1)}{2}$  » et on veut montrer que  $\forall n \in \mathbb{N}^*, P(n) \equiv \text{vrai}$ .

**1. Cas de base :**

Pour  $n = 1$ , on a  $P(1) \equiv \text{vrai}$ , car  $\sum_{i=1}^1 i = 1 = \frac{1(1+1)}{2}$ .

**2. Étape de récurrence :**

Soit  $k \geq 1$ , on doit montrer  $P(k) \rightarrow P(k+1)$ , ce qui revient à dire « si on fait l'hypothèse que la proposition  $P(k)$  est vraie alors cela entraîne que  $P(k+1)$  est aussi vraie ».

– **Hypothèse de récurrence :**

$$P(k) : \left\langle \sum_{i=1}^k i = \frac{k(k+1)}{2} \right\rangle \equiv \text{vrai.}$$

– **Objectif :**

$$P(k+1) : \left\langle \sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2} \right\rangle \equiv \text{vrai.}$$

– **Calculs :**

$$\begin{aligned} \sum_{i=1}^{k+1} i &= 1 + 2 + \dots + k + (k+1) \\ &= \sum_{i=1}^k i + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) && \text{(par hyp. de réc.)} \\ &= \frac{k^2 + k}{2} + \frac{2k+2}{2} \\ &= \frac{k^2 + 3k + 2}{2} \\ &= \frac{(k+2)(k+1)}{2} \end{aligned}$$

□

### Exemple 6.2

Démontrez que pour tout entier  $n \geq 1$ , on a  $n < 2^n$ .

#### Solution :

On définit  $P(n) : \langle n < 2^n \rangle$  et on veut montrer que  $\forall n \in \mathbb{N}, P(n) \equiv \text{vrai}$ .

1. **Cas de base :**

Pour  $n = 1$ , on a  $P(1) : \langle 1 < 2^1 \rangle \equiv \langle 1 < 2 \rangle \equiv \text{vrai}$ .

2. **Étape de récurrence :**

Soit  $k \geq 1$ , on doit montrer  $P(k) \rightarrow P(k+1)$ , ce qui revient à dire « si on fait l'hypothèse que la proposition  $P(k)$  est vraie alors cela entraîne que  $P(k+1)$  est aussi vraie ».

– **Hypothèse de récurrence :**

$$P(k) : \langle k < 2^k \rangle \equiv \text{vrai.}$$

– **Objectif :**

$$P(k+1) : \langle k+1 < 2^{k+1} \rangle \equiv \text{vrai.}$$

– **Calculs :**

$$\begin{array}{ll}
 1 & \leq k & \text{(par définition de } k) \\
 \rightarrow k+1 & \leq k+k \\
 \rightarrow k+1 & \leq 2 \cdot k & \text{(car } k+k=2k) \\
 \rightarrow k+1 & < 2 \cdot 2^k & \text{(par hyp. de réc.)} \\
 \rightarrow k+1 & < 2^{k+1} & \text{(comme } 2 \cdot 2^k = 2^{k+1})
 \end{array}$$

□

### Exemple 6.3

Démontrez que pour tout entier  $n$  positif, 6 est un diviseur de  $(7^n - 1)$ .

**Solution :**

**Remarque :** il est difficile d'utiliser directement un énoncé de la forme « 6 est un diviseur de  $(7^n - 1)$  » dans le cadre d'une preuve formelle. On utilise plutôt la définition plus mathématique suivante :

$$\exists c \in \mathbb{Z}, \quad (7^n - 1) = 6c.$$

On définit ainsi  $P(n)$  : «  $\exists c \in \mathbb{Z}, \quad (7^n - 1) = 6c$  » et on veut montrer que  $\forall n \in \mathbb{N}, P(n) \equiv \text{vrai}$ .

1. **Cas de base :**

Pour  $n = 0$ , on a  $P(0)$  : «  $\exists c \in \mathbb{Z}, \quad (7^0 - 1) = 6c$  »  $\equiv$  «  $0 = 6 \cdot 0$  »  $\equiv$  **vrai**.

2. **Étape de récurrence :**

Soit  $k \geq 0$ , on doit montrer  $P(k) \rightarrow P(k+1)$ , ce qui revient à dire « si on fait l'hypothèse que la proposition  $P(k)$  est vraie alors cela entraîne que  $P(k+1)$  est aussi vraie ».

– **Hypothèse de récurrence :**

$$P(k) : \text{« } \exists c \in \mathbb{Z}, \quad (7^k - 1) = 6c \text{ »} \equiv \text{vrai.}$$

– **Objectif :**

$$P(k+1) : \text{« } \exists c' \in \mathbb{Z}, \quad (7^{k+1} - 1) = 6c' \text{ »} \equiv \text{vrai.}$$

– **Calculs :**

$$\begin{aligned}
 7^{k+1} - 1 &= 7 \cdot 7^k - 7 + 6 \\
 &= 7 \cdot (7^k - 1) + 6 \\
 &= 7 \cdot 6c + 6 && \text{(par hyp. de réc.)} \\
 &= 6 \cdot (7c + 1) \\
 &= 6c' && \text{(où } c' = 7c + 1 \in \mathbb{Z})
 \end{aligned}$$

□

### Exemple 6.4

À l'exemple 5.6 on a *trouvé* que la relation de récurrence

$$\begin{cases} a_0 = 2 \\ a_n = a_{n-1} + 3, \quad n \geq 1 \end{cases}$$

à comme solution  $a_n = 2 + 3n$ , sans démontrer le résultat. Prouvez que cette solution est juste.

**Solution :**

Posons  $f(n) = 3n + 2$ . Définissons la fonction propositionnelle  $P(n)$  : «  $a_n = f(n)$  » et montrons que  $\forall n \in \mathbb{N}, P(n) \equiv \text{vrai}$ .

1. **Cas de base:**

$$P(0) : \langle a_0 = f(0) \rangle \equiv \langle a_0 = 3 \cdot 0 + 2 \rangle \equiv \langle a_0 = 2 \rangle \equiv \text{vrai.}$$

2. **Étape de récurrence:**

Soit  $k \geq 0$ , on doit montrer  $P(k) \rightarrow P(k+1)$ , ce qui revient à dire « si on fait l'hypothèse que la proposition  $P(k)$  est vraie alors cela entraîne que  $P(k+1)$  est aussi vraie ».

– **Hypothèse de récurrence:**

$$P(k) : \langle a_k = f(k) \rangle \equiv \text{vrai.}$$

– **Objectif:**

$$P(k+1) : \langle a_{k+1} = f(k+1) \rangle \equiv \text{vrai.}$$

– **Calculs:**

$$\begin{aligned} f(k+1) &= 3(k+1) + 2 && \text{(par définition de } f) \\ &= 3k + 5 && \text{(par développement algébrique)} \end{aligned}$$

et

$$\begin{aligned} a_{k+1} &= a_k + 3 && \text{(par la relation de récurrence qui définit la suite } a) \\ &= 3k + 2 + 3 && \text{(par l'hypothèse de récurrence)} \\ &= 3k + 5. \end{aligned}$$

□

**Exemple 6.5**

**Généralisation de la Loi de De Morgan.** Soit  $p_1, p_2, \dots, p_n$  des propositions. Démontrez que pour tout entier  $n \geq 2$ ,

$$\neg(p_1 \wedge p_2 \wedge \dots \wedge p_n) = \neg p_1 \vee \neg p_2 \vee \dots \vee \neg p_n.$$

**Solution :**

On définit la fonction propositionnelle  $P(n)$  : «  $\neg(p_1 \wedge p_2 \wedge \dots \wedge p_n) = \neg p_1 \vee \neg p_2 \vee \dots \vee \neg p_n$  » et on montre que  $P(n) \equiv \text{vrai}$  pour tout  $n \geq 2$ .

1. **Cas de base:**

$$P(2) \equiv \text{vrai} \text{ car par la loi de De Morgan, } \neg(p_1 \wedge p_2) = \neg p_1 \vee \neg p_2.$$

2. **Étape de récurrence:**

Soit  $k \geq 2$ , on doit montrer  $P(k) \rightarrow P(k+1)$ , ce qui revient à dire « si on fait l'hypothèse que la proposition  $P(k)$  est vraie alors cela entraîne que  $P(k+1)$  est aussi vraie ».

– **Hypothèse de récurrence:**

$$P(k) : \langle \neg(p_1 \wedge p_2 \wedge \dots \wedge p_k) = \neg p_1 \vee \neg p_2 \vee \dots \vee \neg p_k \rangle \equiv \text{vrai.}$$

– **Objectif:**

$$P(k+1) : \langle \neg(p_1 \wedge p_2 \wedge \dots \wedge p_{k+1}) = \neg p_1 \vee \neg p_2 \vee \dots \vee \neg p_{k+1} \rangle \equiv \text{vrai.}$$

– **Calculs :**

$$\begin{aligned}\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_{k+1}) &= \neg((p_1 \wedge p_2 \wedge \cdots \wedge p_k) \wedge p_{k+1}) \\ &= \neg(p_1 \wedge p_2 \wedge \cdots \wedge p_k) \vee \neg p_{k+1} && \text{(par } P(2), \text{ la loi de De Morgan)} \\ &= \neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_k \vee \neg p_{k+1} && \text{(par hyp. de réc.)}\end{aligned}$$

□

## Exercices

**6.1** Utilisez une preuve par récurrence pour démontrer les énoncés suivants.

- (a) Pour tout entier  $n \geq 1$ ,  $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$ .
- (b) Pour tout entier  $n \geq 0$ ,  $\sum_{i=0}^n r^i = \frac{r^{n+1} - 1}{r - 1}$ , si  $r \in \mathbb{R} \setminus \{1\}$ .
- (c) Le nombre  $n^3 - n$  est divisible par 3, pour tout entier  $n \geq 0$ .  
Rappelons que  $n^3 - n$  est divisible par 3 si et seulement si  $n^3 - n = 3 \cdot a$ , pour  $a \in \mathbb{Z}$ .
- (d) Le nombre  $n^2 + 3n$  est divisible par 2, pour tout entier  $n \geq 0$ .  
Rappelons que  $n^2 + 3n$  est divisible par 2 si et seulement si  $n^2 + 3n = 2 \cdot a$ , pour  $a \in \mathbb{Z}$ .
- (e)  $2n + 1 \leq 2^n$  pour tout entier  $n \geq 3$ .
- (f)  $n^2 \leq 2^n$  pour tout entier  $n \geq 4$ .
- (g)  $2^n < n!$  pour tout entier  $n \geq 4$ .
- (h)  $n! < n^n$  pour tout entier  $n \geq 2$ .
- (i) Loi de De Morgan pour les ensembles. Soit  $A_1, A_2, \dots, A_n$  des ensembles. Pour tout entier  $n \geq 2$ ,

$$\overline{A_1 \cup A_2 \cup \cdots \cup A_n} = \overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_n}.$$

- (j) Pour tout entier  $n \geq 0$ , l'échiquier  $2^n \times 2^n$  dont une case est occupée par la reine est pavable par des triominos en forme de L. Voir la figure 6.1 pour une illustration.

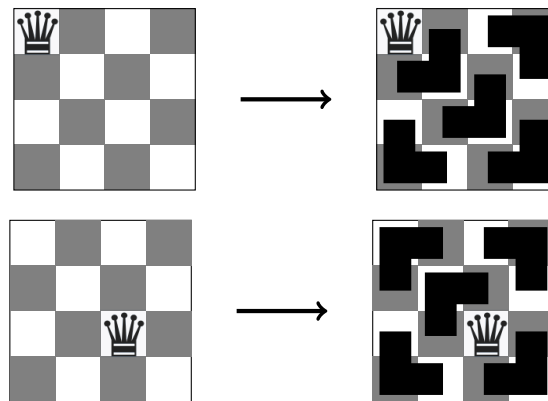


Figure 6.1 Illustration de l'exercice 6.1 (j) avec  $n = 2$ . Dans les deux cas, un échiquier  $2^n \times 2^n$  dont une case est déjà occupée par une reine est pavé à l'aide de triominos en forme de L. Le but de l'exercice est de montrer qu'un tel pavage est possible pour tout entier  $n \geq 0$  et peu importe la position de la reine.



**6.2** On considère la somme des  $n$  premiers entiers impairs  $\sum_{i=1}^n (2i-1) = 1 + 3 + 5 + 7 + \dots + n$ .

- (a) Trouvez une forme close pour cette somme.  
 (b) Montrez, en utilisant une preuve par récurrence, que la forme close trouvée en (a) est juste.

**6.3** À l'exemple 5.7 on a *trouvé* que la relation de récurrence

$$\begin{cases} b_0 = 5 \\ b_n = b_{n-1} + 4n, \quad n \geq 1 \end{cases}$$

a comme solution  $b_n = 2n^2 + 2n + 5$ , sans démontrer le résultat de façon formelle. Montrez maintenant que cette solution est juste.

**6.4** À l'exercice 5.2 on a *trouvé* que la relation de récurrence

$$\begin{cases} f(0) = 2 \\ f(n) = 3f(n-1) + 2, \quad n \geq 1 \end{cases}$$

a comme solution  $f(n) = 3^{n+1} - 1$ , sans démontrer le résultat de façon formelle. Montrez maintenant que cette solution est juste.

**6.5** On considère la fonction propositionnelle:  $P(n)$ : « 3 divise  $(n^3 + 2n + 1)$  ».

- (a) Montrez que pour tout  $k \geq 0$ ,  $P(k) \rightarrow P(k+1)$ .  
 (b) ★ Expliquez pourquoi, malgré le fait que l'*étape de récurrence* a été prouvée en (a), il est faux que  $\forall n \geq 0, P(n)$ .

**6.6** Soit la fonction réelle  $f(x) = x^n$ , avec  $n \geq 1$ . Montrez, à l'aide d'une preuve par récurrence, que la dérivée de  $f$  est  $f'(x) = nx^{n-1}$ . *Indice: utilisez la dérivée d'un produit de fonction  $(uv)' = u'v + uv'$  et  $x' = 1$ .*

**6.7** Montrez, à l'aide d'une preuve par récurrence, que la dérivée  $n^{\text{ième}}$  de  $f(x) = \ln(x)$  est

$$f^{(n)}(x) = \frac{(-1)^{(n-1)}(n-1)!}{x^n}.$$

**6.8** Considérez les nombre de Fibonacci définis par la relation de récurrence

$$\begin{cases} f_0 = 0, \quad f_1 = 1 \\ f_n = f_{n-1} + f_{n-2}, \quad n \geq 2. \end{cases}$$

Montrez par récurrence les deux énoncés suivants:

- (a)  $\forall n \in \mathbb{N}, \sum_{i=0}^n f_i = f_{n+2} - 1$ ;  
 (b)  $\forall n \in \mathbb{N}, \sum_{i=0}^n f_i^2 = f_n \cdot f_{n+1}$ .

## 6.2 Preuve par récurrence forte

Le principe de récurrence forte est une variante plus puissante du principe de récurrence dans la mesure où, à l'étape de récurrence, au lieu de montrer  $P(k) \rightarrow P(k+1)$ , on montre :

$$(P(0) \wedge P(1) \wedge P(2) \wedge \dots \wedge P(k)) \rightarrow P(k+1).$$

Attention, il est important de souligner que même si l'énoncé peut sembler plus complexe, il est *plus facile* à démontrer, car on effectue davantage d'hypothèses. En effet, pour montrer  $P(k+1) \equiv \mathbf{vrai}$ , au lieu de supposer seulement  $P(k) \equiv \mathbf{vrai}$ , on suppose :  $P(0) \equiv \mathbf{vrai}$  et  $P(1) \equiv \mathbf{vrai}$  et  $P(2) \equiv \mathbf{vrai}$  et  $P(3) \equiv \mathbf{vrai}$  et ainsi de suite jusqu'à  $P(k) \equiv \mathbf{vrai}$ .

Ainsi, en démontrant le cas de base  $P(0)$  et l'étape de récurrence, on obtient  $P(0), P(1), P(2), P(3), \dots$ , par les règles d'inférence de la *conjonction* et du *modus ponens* :

		$P(0)$	
	$P(0)$	$P(1)$	
$P(0)$	$P(1)$	$P(2)$	$\dots$
$P(0) \rightarrow P(1)$	$P(0) \wedge P(1) \rightarrow P(2)$	$P(0) \wedge P(1) \wedge P(2) \rightarrow P(3)$	
$P(1)$	$P(2)$	$P(3)$	

De manière plus générale, soit  $P(n)$  une fonction propositionnelle dont le domaine est  $\mathbb{N}$ . La règle d'inférence suivante permet de montrer que  $P(n)$  est vraie pour tout nombre naturel supérieur ou égal à  $n_0$ . Cette règle permet aussi de gérer la situation où il y a plusieurs cas de base

$$n_0, n_0 + 1, \dots, n_0 + j,$$

pour  $j \in \mathbb{N}$ .

### Théorème 6.2 : Principe de raisonnement par récurrence forte.

Soit  $P(n)$  une proposition portant sur un nombre naturel  $n$  et soit  $n_0, j \in \mathbb{N}$ .

$$\begin{aligned} [P(n_0) \wedge P(n_0 + 1) \wedge \dots \wedge P(n_0 + j)] \wedge [\forall k \geq (n_0 + j), P(n_0) \wedge P(n_0 + 1) \wedge \dots \wedge P(k) \rightarrow P(k + 1)] \\ \longrightarrow \forall n \geq n_0, P(n) \end{aligned}$$

Ainsi, pour démontrer  $\forall n \geq n_0, P(n)$  en utilisant le principe du raisonnement par récurrence forte, on doit procéder en deux étapes, qui sont décrites dans l'encadré qui suit.

**Comment démontrer  $\forall n \geq n_0, P(n)$  par récurrence forte?**

On doit :

1. **Cas de base**

Montrer que  $P(n_0), P(n_0 + 1), \dots, P(n_0 + j)$  sont vraies ( $j \in \mathbb{N}$ ).

( $n_0$  est le premier cas de base,  $n_0 + j$  est le dernier cas de base, il y a donc  $j + 1$  cas de base)

2. **Étape de récurrence**

Montrer que

$$\forall k \geq (n_0 + j), (P(n_0) \wedge P(n_0 + 1) \wedge \dots \wedge P(k)) \rightarrow P(k + 1).$$

**Exemple 6.6**

Montrez que tout entier  $n \geq 2$  peut s'écrire comme produit de nombres premiers.

**Solution :**

On définit  $P(n)$  : « il existe des nombres premiers  $p_1, p_2, \dots$ , et  $p_r$  tels que  $n = p_1 p_2 \dots p_r$  » et on montre que  $\forall n \geq 2, P(n) \equiv$  **vrai**.

1. **Cas de base**: (ici  $n_0 = 2$  et  $j = 0$ )

$P(2) \equiv$  **vrai** car 2 est premier. En effet,  $P(2)$  est satisfait avec  $r = 1$  et  $p_1 = 2$ .

2. **Étape de récurrence**:

Soit  $k \geq 2$ , on doit montrer que  $P(2) \wedge P(3) \wedge \dots \wedge P(k) \rightarrow P(k + 1)$ .

– **Hypothèse de récurrence**:

$$P(2) \wedge P(3) \wedge \dots \wedge P(k) \equiv \text{vrai}.$$

On suppose donc que tout nombre entre 2 et  $k$  peut s'écrire comme produit de nombre premiers.

– **Objectif**:

$$P(k + 1) \equiv \text{vrai}.$$

Il faut donc montrer que l'entier  $k + 1$  peut s'écrire comme produit de nombres premiers.

– **Raisonnement**:

On considère deux cas :

- Si  $k + 1$  est un nombre premier alors  $P(k + 1)$  est satisfait avec  $r = 1$  et  $p_1 = k + 1$ .
- Si  $k + 1$  n'est pas premier alors il est composé. Cela signifie qu'il existe deux entiers  $a, b$ ,  $2 \leq a \leq k$ ,  $2 \leq b \leq k$  tels que  $k + 1 = ab$ . Par hypothèse de récurrence,  $P(a)$  est vrai et donc il existe des nombres premiers  $p_1, p_2, \dots, p_r$  tels que  $p_1 p_2 \dots p_r = a$ . De plus, par l'hypothèse de récurrence on a aussi que  $P(b)$  est vrai et donc il existe des nombres premiers  $p'_1, p'_2, \dots, p'_r$  tels que  $b = p'_1 p'_2 \dots p'_r$ .

Pour conclure, on a  $k + 1 = ab = p_1 p_2 \dots p_r p'_1 p'_2 \dots p'_r$ , où tous les nombres du produit de droite sont premiers. □

**Exemple 6.7**

Soit la relation de récurrence

$$\begin{cases} a_1 = 2 \\ a_2 = 9 \\ a_n = 2a_{n-1} + 3a_{n-2}, \quad n \geq 3 \end{cases}$$

Montrez que  $a_n \leq 3^n$ , pour tout entier  $n$  strictement positif.

**Solution :**

On définit  $P(n)$  : «  $a_n \leq 3^n$  » et on montre que  $\forall n \geq 1, P(n) \equiv \mathbf{vrai}$ .

1. **Cas de base:** (ici  $n_0 = 1$  et  $j = 1$ )

$$P(1) : a_1 = 2 \leq 3 = 3^1 \text{ est vrai.}$$

$$P(2) : a_2 = 9 \leq 9 = 3^2 \text{ est vrai.}$$

2. **Étape de récurrence:**

Soit  $k \geq 2$ , on doit montrer que  $P(1) \wedge P(2) \wedge \dots \wedge P(k) \rightarrow P(k+1)$ .

- **Hypothèse de récurrence:**

$$P(1) \wedge P(2) \wedge \dots \wedge P(k) \equiv \mathbf{vrai}.$$

En particulier,  $P(k)$  : «  $a_k \leq 3^k$  »  $\equiv \mathbf{vrai}$  et  $P(k-1)$  : «  $a_{k-1} \leq 3^{k-1}$  »  $\equiv \mathbf{vrai}$ .

- **Objectif:**

$$P(k+1) : \text{« } a_{k+1} \leq 3^{k+1} \text{ »} \equiv \mathbf{vrai}.$$

- **Raisonnement:**

$$\begin{aligned} a_{k+1} &= 2a_k + 3a_{k-1} && \text{(par la définition de la relation de récurrence)} \\ &\leq 2 \cdot 3^k + 3 \cdot 3^{k-1} && \text{(par hypothèse de récurrence)} \\ &= 2 \cdot 3^k + 3^k \\ &= 3^k(2 + 1) \\ &= 3^{k+1}. \end{aligned}$$

□

**Exemple 6.8**

Soit  $n \geq 4$ . Montrez que toute facture de  $n$  \$ peut être payée uniquement par une combinaison de 2 \$ et de 5 \$.

**Solution :**

On voit qu'on peut obtenir des montants de 4\$ = 2 · 2\$ et 5 \$ avec des 2 \$ et 5 \$. En ajoutant un 2 \$ à chacun de ces montants, on obtient des totaux de 6 \$ et 7 \$. En ajoutant à nouveau un 2 \$ à ces totaux, on arrive à payer des factures de 8 \$ et 9 \$. En continuant ainsi, on pourra payer n'importe quelle facture supérieure ou égale à 4 \$, à l'aide de 2 \$ et 5 \$. Formalisons ce raisonnement de façon algébrique.

On définit  $P(n)$  : « Il existe  $a, b \in \mathbb{N}$  tels que  $2a + 5b = n$  » et on veut montrer que  $\forall n \geq 4, P(n) \equiv \mathbf{vrai}$ .

1. **Cas de base:** (ici  $n_0 = 4$  et  $j = 1$ )

$$P(4) \equiv \mathbf{vrai} \text{ avec } a = 2 \text{ et } b = 0 \text{ car } 4 = 2 \cdot 2 + 0 \cdot 5,$$

$$P(5) \equiv \mathbf{vrai} \text{ avec } a = 0 \text{ et } b = 1 \text{ car } 5 = 0 \cdot 2 + 1 \cdot 5,$$

2. **Étape de récurrence:**

Soit  $k \geq 5$ , on doit montrer que  $P(4) \wedge P(5) \wedge \dots \wedge P(k) \rightarrow P(k+1)$ .

- **Hypothèse de récurrence:**

$$P(4) \wedge P(5) \wedge \dots \wedge P(k-1) \wedge P(k) \equiv \mathbf{vrai}.$$

En particulier, on a que  $P(k-1) \equiv \mathbf{vrai}$ . On a donc qu'il existe  $a, b \in \mathbb{N}$  tels que  $2a + 5b = k-1$ .

– **Objectif:**

$$P(k+1) : \ll \text{Il existe } a_2, b_2 \in \mathbb{N} \text{ tels que } 2a_2 + 5b_2 = k+1 \gg \equiv \text{vrai.}$$

– **Calculs:**

$$\begin{aligned} k+1 &= (k-1) + 2 \\ &= 2a + 5b + 5 && \text{(par hyp. de réc.)} \\ &= 2a + 5(b+1) \\ &= 2a_2 + 5b_2. && \text{(avec } a_2 = a \in \mathbb{N} \text{ et } b_2 = b+1 \in \mathbb{N}) \end{aligned}$$

□

### Autre solution:

On voit qu'on peut obtenir des montants de 4 \$ à 8 \$ avec des 2 \$ et 5 \$. En ajoutant un 5 \$ à chacun de ces montants, on obtient des totaux de 9 \$ à 13 \$. En ajoutant à nouveau un 5 \$ à ces totaux, on arrive à payer des factures de 14 \$ et 18 \$. En continuant ainsi, on pourra payer n'importe quelle facture supérieure ou égale à 4 \$, à l'aide de 2 \$ et 5 \$. Formalisons ce raisonnement de façon algébrique.

On définit  $P(n)$  : « il existe  $a, b \in \mathbb{N}$  tels que  $2a + 5b = n$  » et on veut montrer que pour tout  $n \geq 4$ ,  $P(n) \equiv \text{vrai}$ .

1. **Cas de base:** (ici  $n_0 = 4$  et  $j = 3$ )

$$P(4) \equiv \text{vrai} \text{ avec } a = 2 \text{ et } b = 0 \text{ car } 4 = 2 \cdot 2 + 0 \cdot 5,$$

$$P(5) \equiv \text{vrai} \text{ avec } a = 0 \text{ et } b = 1 \text{ car } 5 = 0 \cdot 2 + 1 \cdot 5,$$

$$P(6) \equiv \text{vrai} \text{ avec } a = 3 \text{ et } b = 0 \text{ car } 6 = 3 \cdot 2 + 0 \cdot 5,$$

$$P(7) \equiv \text{vrai} \text{ avec } a = 1 \text{ et } b = 1 \text{ car } 7 = 1 \cdot 2 + 1 \cdot 5,$$

$$P(8) \equiv \text{vrai} \text{ avec } a = 4 \text{ et } b = 0 \text{ car } 8 = 4 \cdot 2 + 0 \cdot 5,$$

2. **Étape de récurrence:** Soit  $k \geq 8$ , on doit montrer que  $P(4) \wedge P(5) \wedge \dots \wedge P(k) \rightarrow P(k+1)$ .

– **Hypothèse de récurrence:**

$$P(4) \wedge P(5) \wedge \dots \wedge P(k-1) \wedge P(k) \equiv \text{vrai.}$$

En particulier, on a que  $P(k-4) \equiv \text{vrai}$ . On a donc qu'il existe  $a, b \in \mathbb{N}$  tels que  $2a + 5b = k-4$ .

– **Objectif:**

$$P(k+1) : \ll \text{Il existe } a_2, b_2 \in \mathbb{N} \text{ tels que } 2a_2 + 5b_2 = k+1 \gg \equiv \text{vrai.}$$

– **Calculs:**

$$\begin{aligned} k+1 &= (k-4) + 5 \\ &= 2a + 5b + 5 && \text{(par hyp. de réc.)} \\ &= 2a + 5(b+1) \\ &= 2a_2 + 5b_2. && \text{(avec } a_2 = a \in \mathbb{N} \text{ et } b_2 = b+1 \in \mathbb{N}) \end{aligned}$$

□

### Exercices

**6.9** On considère la relation de récurrence  $a_0 = 0$ ,  $a_1 = 2$  et  $a_n = 4a_{n-1} - 4a_{n-2}$  pour  $n \geq 2$ . Utilisez le principe de récurrence forte pour montrer que  $a_n = n2^n$  pour tout  $n \geq 0$ .

**6.10** On considère la relation de récurrence  $a_0 = 0$ ,  $a_1 = 0$ ,  $a_2 = 2$  et  $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$  pour  $n \geq 3$ . Utilisez le principe de récurrence forte pour montrer que  $a_n = 3^n - 2^{n+1} + 1$  pour tout  $n \geq 0$ .

**6.11** Soit  $n \geq 12$ . Montrez que tout timbre de  $n$  ¢ peut être remplacé uniquement par une combinaison de timbres de 3 ¢ et de 7 ¢.

**6.12** Soit  $n \geq 24$ . Montrez que tout timbre de  $n$  ¢ peut être remplacé uniquement par une combinaison de 5 ¢ et de 7 ¢.

### 6.3 Preuve de validité d'un algorithme récursif

Les preuves par récurrence sont aussi utilisées pour montrer qu'un algorithme est **correct**, c'est-à-dire qu'il produit toujours la bonne solution.

#### Exemple 6.9

Montrez que l'algorithme **puissance**( $a, n$ ) de l'Exemple 5.1 est correct, pour tout  $a \in \mathbb{R}^*$  et  $n \in \mathbb{N}$ .

#### Solution :

$$p(a,n) := \begin{cases} \text{undef,} & a=0 \text{ and } n=0 \text{ or } n<0 \text{ or } \text{mod}(n,1) \neq 1 \\ 0, & a=0 \text{ and } n \neq 0 \\ 1, & a \neq 0 \text{ and } n=0 \\ a \cdot p(a,n-1), & \text{Else} \end{cases}$$

Soit  $P(n)$  : «  $p(a, n) = a^n$  ». Montrons que  $\forall n \geq 0, P(n) \equiv \text{vrai}$ , par récurrence simple.

#### 1. Cas de base:

$$\begin{aligned} p(a, 0) &= 1 && \text{(par l'algorithme)} \\ &= a^0. && \text{(propriété des exposants)} \end{aligned}$$

Ainsi,  $P(0)$  : «  $p(a, 0) = a^0$  »  $\equiv \text{vrai}$ .

#### 2. Étape de récurrence:

Soit  $k \geq 0$ , on doit montrer que  $P(k) \rightarrow P(k+1)$ .

– **Hypothèse de récurrence :**

$$P(k) : \text{« } p(a, k) = a^k \text{ »} \equiv \text{vrai.}$$

– **Objectif :**

$$P(k+1) : \text{« } p(a, k+1) = a^{k+1} \text{ »} \equiv \text{vrai.}$$

– **Calculs :**

$$\begin{aligned}
 p(a, k+1) &= a \cdot p(a, k) && \text{(par l'algorithme)} \\
 &= a \cdot a^k && \text{(par l'hypothèse de récurrence)} \\
 &= a^{k+1}. && \text{(propriété des exposants)}
 \end{aligned}$$

□

### Exemple 6.10

Problème des tours de Hanoï.



**But :** Déplacer une tour de  $n$  disques ayant des diamètres différents de la première tige à la troisième tige.

**Règles :**

- déplacer un seul disque à la fois;
- déplacer un disque seulement s'il est au sommet d'une pile;
- déplacer un disque seulement sur un disque de diamètre plus grand, ou sur une tige vide.

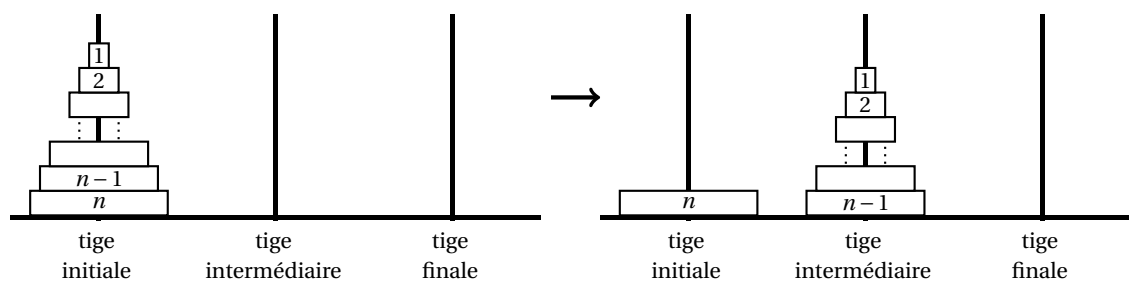
- Donnez un algorithme récursif pour résoudre ce problème.
- Montrez que l'algorithme donné en (a) est correct.
- Donnez sa complexité. *Comptez le nombre de déplacements de disques.*

### Solution :

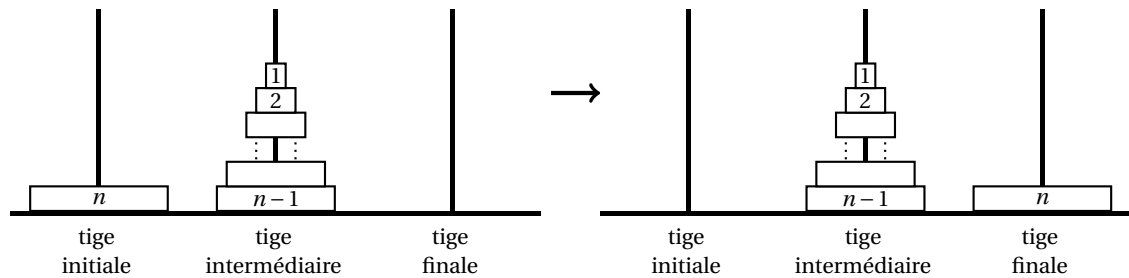
- On veut déplacer tous les disques de la première tige vers la troisième. Pour simplifier, on appellera la première tige la *tige initiale*, la deuxième tige la *tige intermédiaire* et la troisième tige la *tige finale*. Un algorithme récursif peut être déduit des considérations suivantes :
  - À un moment, il faudra déplacer le plus grand disque de la tige initiale vers la tige finale.
  - Pour retirer le plus grand disque de la tige initiale, il faut qu'il n'y ait aucun autre disque sur cette tige.
  - Pour déposer le plus grand disque sur la tige finale, il faut qu'il n'y ait aucun disque sur cette tige.
  - Ainsi, pour déplacer le plus grand disque de la tige initiale à la tige finale, il faut forcément que tous les autres disques soient empilés sur la tige intermédiaire.

L'algorithme général pour  $n$  disques est donc :

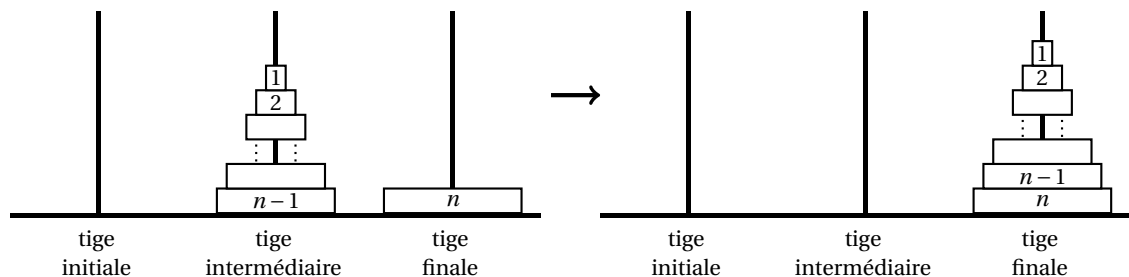
- Déplacer  $n - 1$  disques de la tige initiale vers la tige intermédiaire (possiblement en plusieurs étapes).



2. Déplacer le  $n$ -ième disque de la tige initiale à la tige finale (en une étape).



3. Déplacer  $n - 1$  disques de la tige intermédiaire à la tige finale (possiblement en plusieurs étapes).



```

1: fonction Hanoi( $n, a, b, c$ : Entiers)
2:            $\triangleright n$ : nombre de disques,  $a$  tige initiale,  $b$  tige finale et  $c$  tige intermédiaire.
3:   si  $n > 0$  alors
4:     Hanoi( $n - 1, a, c, b$ )
5:     Déplacer le disque  $n$  de la tige  $a$  à la tige  $b$ 
6:     Hanoi( $n - 1, c, b, a$ )
7:   fin si
8: fin fonction

```

(b) On définit  $P(n)$ : « un appel à  $\text{Hanoi}(n, a, b, c)$  déplace  $n$  disques de la tige  $a$  à la tige  $b$  » et on veut montrer que pour tout  $n \geq 0$ ,  $P(n) \equiv \text{vrai}$ .

### 1. Cas de base :

On montre que l'algorithme est correct pour  $n = 0$ . Avec  $n = 0$ , aucun déplacement de disque n'est effectué.

### 2. Étape de récurrence :

Soit  $k \geq 0$ .

–Hypothèse de récurrence :

$P(k)$  : « un appel à  $\text{Hanoi}(k, a, b, c)$  déplace  $k$  disques de la tige  $a$  à la tige  $b$  »  $\equiv \text{vrai}$ .

–Objectif :

$P(k + 1) \equiv \text{vrai}$ .

On veut montrer que l'algorithme est correct pour déplacer  $k + 1$  disques.



**–Argumentation :**

Tout d'abord, on remarque que  $k + 1 > 0$  et donc la condition du **si** est vraie. Ensuite, par hypothèse de récurrence, l'appel récursif de la ligne 4 déplace les  $k$  premiers disques de la tige initiale vers la tige intermédiaire. Ensuite, la ligne 5 déplace le disque  $k + 1$  de la tige initiale vers la tige finale. Finalement, encore par l'hypothèse de récurrence, on a que l'appel de la ligne 6 déplace les  $k - 1$  premiers disques de la tige intermédiaire vers la tige finale. Tous les disques ont donc été correctement déplacés en respectant les règles des tours de Hanoï.  $\square$

- (c) Soit  $f(n)$  le nombre de déplacements de disques effectués par un appel à  $\text{Hanoï}(n, a, b, c)$ . On observe que :

$$f(n) = \begin{cases} 0 & \text{si } n = 0, \\ 2f(n-1) + 1 & \text{si } n \geq 1. \end{cases}$$

En utilisant la méthode itérative :

$$f(0) = 0$$

$$f(1) = 2f(0) + 1 = 1$$

$$f(2) = 2f(1) + 1 = 2 + 1$$

$$f(3) = 2f(2) + 1 = 2(2 + 1) + 1 = 2^2 + 2 + 1$$

$$f(4) = 2f(3) + 1 = 2(2^2 + 2 + 1) + 1 = 2^3 + 2^2 + 2 + 1$$

$\vdots$

$$f(n) = \sum_{i=0}^{n-1} 2^i = 2^n - 1$$

Donc,  $f(n) = 2^n - 1 \in O(2^n)$ .

**Exercices**

**6.13** Factorielle.

- Écrivez un algorithme récursif pour calculer  $n!$ , où  $n \in \mathbb{N}$ .
- Montrez que cet algorithme est correct.
- Donnez sa complexité. *Compter le nombre de multiplications.*

**6.14**

Considérez l'algorithme récursif suivant, où  $n \in \mathbb{N}$ .

```

1: fonction mystère( $n$ )
2:   si  $n = 0$  alors
3:     retourner 0
4:   sinon
5:     retourner mystère( $n - 1$ ) + 3 ·  $n$  ·  $n - 3$  ·  $n + 1$ 
6:   fin si
7: fin fonction

```

- (a) Devinez ce que la fonction `mystère(n)` retourne.
- (b) Validez le résultat obtenu en (a) en utilisant une preuve par récurrence.
- (c) Soit  $f(n)$  le nombre de multiplications requises lors d'un appel à `mystère(n)`. Trouvez une relation de récurrence pour  $f(n)$ .
- (d) Quelle est la complexité de cet algorithme? *Déduire une forme close pour  $f(n)$ .*

**6.15** Fibonacci.

- (a) Écrire un algorithme récursif pour calculer  $f_n$ , le  $n$ -ième nombre de Fibonacci.
- (b) Montrez que cet algorithme est correct.
- (c) Soit  $g(n)$  le nombre d'additions effectuées lors d'un appel à l'algorithme écrit en (a). Montrer que  $g(n) = f_{(n+1)} - 1$ .
- (d) Écrivez un algorithme itératif pour calculer  $f_n$ , le  $n$ -ième nombre de Fibonacci. Vous ne devriez utiliser que trois variables entières et un indice de boucle.
- (e) Déterminez  $h(n)$ , le nombre d'additions effectuées lors d'un appel à l'algorithme écrit en (d).
- (f) Sachant que  $g(n) \in \Theta(\varphi^n)$ , où  $\varphi = \frac{1+\sqrt{5}}{2}$  est le nombre d'or, déterminez parmi les deux algorithmes écrits en (a) et (d), lequel est le plus efficace?
- (g) ★ Écrivez un algorithme qui calcule le  $n$ -ième nombres de Fibonacci avec un nombre d'opérations arithmétiques dans  $O(\log(n))$ .

# Chapitre 7

## Dénombrement

### 7.1 Notions de base

#### 7.1.1 Principe du produit

##### **Théorème 7.1 : Principe du produit**

Si l'événement  $A_1$  peut se produire de  $n_1$  façons différentes et l'événement  $A_2$  peut se produire de  $n_2$  façons différentes, alors l'événement  $A_1$  suivi de  $A_2$  peut se produire de  $n_1 n_2$  façons différentes.

##### **Exemple 7.1**

Quel est le nombre d'additions effectuées lors de l'exécution du segment d'algorithme suivant?

- 1: **pour**  $i = 1$  à  $n_1$  **faire**
- 2:     **pour**  $j = 1$  à  $n_2$  **faire**
- 3:          $k := k + 1$
- 4:     **fin pour**
- 5: **fin pour**

##### **Solution :**

$$\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} 1 = n_1 n_2$$

##### **Exemple 7.2**

Si  $A_1$  et  $A_2$  sont des ensembles finis, alors

$$|A_1 \times A_2| = |A_1| |A_2|.$$

### 7.1.2 Principe de la somme

#### Théorème 7.2 : Principe de la somme

Si l'événement  $A_1$  peut se produire de  $n_1$  façons différentes et l'événement  $A_2$  peut se produire de  $n_2$  façons différentes et que  $A_1$  et  $A_2$  ne peuvent se produire simultanément, alors l'événement  $A_1$  ou  $A_2$  peut se produire de  $n_1 + n_2$  façons différentes.

#### Exemple 7.3

Quel est le nombre d'additions effectuées lors de l'exécution du segment d'algorithme suivant?

- 1: **pour**  $i = 1$  à  $n_1$  **faire**
- 2:      $k := k + 1$
- 3: **fin pour**
- 4: **pour**  $j = 1$  à  $n_2$  **faire**
- 5:      $k := k + 1$
- 6: **fin pour**

**Solution :**

$$\sum_{i=1}^{n_1} 1 + \sum_{j=1}^{n_2} 1 = n_1 + n_2$$

#### Exemple 7.4

Si  $A_1$  et  $A_2$  sont des ensembles finis et **disjoints**, alors

$$|A_1 \cup A_2| = |A_1| + |A_2|.$$

### 7.1.3 Principe d'inclusion-exclusion

#### Théorème 7.3 : Principe d'inclusion-exclusion

Si  $A_1$  et  $A_2$  sont des ensembles finis, alors

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

### 7.1.4 Principe des tiroirs

« Si  $n$  chaussettes occupent  $m$  tiroirs et  $n > m$ , alors au moins un tiroir doit contenir au moins deux chaussettes. »

#### Théorème 7.4 : Principe des tiroirs

Si  $k + 1$  objets ou plus sont rangés dans  $k$  boîtes alors il y a au moins une boîte qui contient 2 objets ou plus.

#### Exemple 7.5

Dans un groupe de 8 personnes, il y a au moins deux personnes qui sont nées le même jour de la semaine.

#### Théorème 7.5 : Principe des tiroirs généralisé

Si  $n$  objets sont placés dans  $k$  boîtes, alors il y a au moins une boîte qui contient au moins  $\lceil \frac{n}{k} \rceil$  objets.

#### Exemple 7.6

Dans un groupe de 20 personnes, il y a au moins  $\lceil \frac{20}{7} \rceil = 3$  personnes qui sont nées le même jour de la semaine.

## 7.2 Permutations et arrangements

#### Définition 7.1 : Permutation

Une **permutation** d'un ensemble d'objets distincts est un arrangement ordonné de ces objets.

#### Exemple 7.7

Si  $S = \{1, 2, 3\}$ , alors

123, 132, 213, 231, 312, 321

sont les permutations des éléments de  $S$ .

#### Théorème 7.6 : Principe fondamental du dénombrement

Il y a  $n!$  permutations de  $n$  objets distincts.

**Définition 7.2 : Arrangement**

Un **arrangement** de  $k$  éléments (ou une  $k$ -permutation) est une **suite ordonnée** de  $k$  éléments d'un ensemble.

**Exemple 7.8**

Donnez tous les arrangements de 2 éléments de l'ensemble  $S = \{1, 2, 3\}$ .

**Solution :**

12, 13, 21, 23, 31, 32

Remarque. Nous avons utilisé ici une notation allégée, mais on pourrait séparer les éléments par des virgules et présenter l'ensemble des arrangements ainsi :

$\{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)\}$ .

**Théorème 7.7**

Le nombre d'arrangements de  $k$  éléments d'un ensemble de  $n$  éléments distincts est

$$P(n, k) = n(n-1) \cdots (n-k+1) = \frac{n!}{(n-k)!}, \text{ où } 1 \leq k \leq n.$$

Ainsi, lorsque l'**ordre de sélection a de l'importance**, il y a  $P(n, k)$  façons de choisir  $k$  objets parmi  $n$  objets distincts.

**Exemple 7.9**

Si  $S = \{a, b, c\}$ , déterminez le nombre d'arrangements de 2 éléments de  $S$  et énumérez ces arrangements.

**Solution :**

Le nombre d'arrangements de 2 éléments de  $S$  est

$$P(3, 2) = \frac{3!}{(3-2)!} = 6.$$

Les 6 arrangements sont les suivants (notez que l'ordre a de l'importance dans un arrangement, et qu'il ne peut pas y avoir de répétition du même élément) :  $ab, ac, ba, bc, ca, cb$ .

**Exemple 7.10**

Si  $S = \{a, b, c, d\}$ , déterminez le nombre d'arrangements de 2 éléments de  $S$  et énumérez ces arrangements.

**Solution :**

Le nombre d'arrangements de 2 éléments de  $S$  est

$$P(4, 2) = \frac{4!}{(4-2)!} = 12.$$

Les 12 arrangements sont les suivants (notez que l'ordre a de l'importance dans un arrangement, et qu'il ne peut pas y avoir de répétition du même élément) :  $ab, ac, ad, ba, bc, bd, ca, cb, cd, da, db, dc$ .

## 7.3 Combinaisons

### Définition 7.3 : Combinaison

Une **combinaison** de  $k$  éléments (ou une  $k$ -combinaison) est un sous-ensemble de cardinalité  $k$  d'un ensemble.

### Exemple 7.11

Si  $S = \{1, 2, 3\}$ , énumérez les combinaisons de 2 éléments de  $S$ .

#### Solution :

$$\{1, 2\}, \{1, 3\}, \{2, 3\}$$

Notez que l'ordre n'a pas d'importance dans une combinaison, et qu'un élément ne peut être répété.

### Théorème 7.8

Le nombre de combinaisons de  $k$  éléments dans un ensemble de  $n$  éléments distincts est

$$C(n, k) = \frac{n!}{k!(n-k)!} = \binom{n}{k}$$

où  $0 \leq k \leq n$ . Lorsque l'**ordre de sélection n'a pas d'importance**, il y a  $C(n, k)$  façons de choisir  $k$  objets parmi  $n$  objets distincts.

#### Note:

1.  $C(n, k) = \frac{P(n, k)}{k!}$ .
2.  $C(n, k)$  est le nombre de sous-ensembles de cardinalité  $k$  d'un ensemble de  $n$  éléments.
3.  $C(n, k) = C(n, n - k)$ .

### Exemple 7.12

Utilisez le théorème précédent pour calculer le nombre de combinaisons de 2 éléments de l'ensemble  $S = \{1, 2, 3\}$ .

#### Solution :

$$C(3, 2) = \binom{3}{2} = \frac{3!}{2!(3-2)!} = 3.$$

**Exemple 7.13**

Si  $S = \{a, b, c, d\}$ , déterminez le nombre de combinaisons de 2 éléments de  $S$  et énumérez ces combinaisons.

**Solution :**

$$C(4, 2) = \binom{4}{2} = \frac{4!}{2!(4-2)!} = 6.$$

Les 6 combinaisons de 2 éléments de  $S$  sont :

$$\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}.$$

Notez que l'ordre n'a pas d'importance dans une combinaison, et qu'un élément ne peut être répété.

**Théorème 7.9 : Permutations avec objets indistinguables**

Le nombre de permutations différentes de  $n$  objets, où il y a  $n_1$  objets de type 1,  $n_2$  objets de type 2, ...,  $n_k$  objets de type  $k$ , est

$$\frac{n!}{n_1!n_2!\cdots n_k!}.$$

**Exemple 7.14**

Combien peut-on former de mots distincts de 11 lettres en utilisant chacune des lettres du mot MISSISSIPPI?

**Solution :**

$$\frac{11!}{4!1!2!4!} = 34650$$

**Exercices****7.1**

- Combien y a-t-il de trains de bits de longueur 8?
- Combien y a-t-il de train de bits de longueur 8 qui commencent par 01 et terminent par 101?
- Combien de plaques d'immatriculation peut-on former si on utilise 3 chiffres suivis de 3 lettres?

**7.2** Combien y a-t-il de mots de passe comportant 6 à 8 caractères, avec au moins un chiffre et où chaque caractère est une lettre majuscule ou un chiffre?

**7.3**

- Sur un total de 300 étudiants en génie logiciel, 143 sont inscrits au cours MAT210, 122 au cours MAT265, et 21 sont inscrits aux deux. Combien y a-t-il d'étudiants en génie logiciel qui ne sont ni inscrits au cours MAT210 ni inscrits au cours MAT265?
- Combien y a-t-il de train de bits de longueur 8 qui commencent par 01 ou se terminent par 101?



**7.4**

- (a) Cinq femmes et deux hommes attendent à la pharmacie. Combien de façons y a-t-il de les placer dans la file d'attente?
- (b) De combien de façons 5 femmes et 2 hommes peuvent-ils être disposés dans une file d'attente si les 2 hommes doivent se suivre?
- (c) De combien de façons 5 femmes et 2 hommes ils être disposés dans une file d'attente si les 2 hommes doivent être séparés?
- (d) Combien de permutations des lettres A, B, C, D, E, F et G contiennent la chaîne "ABC"?

**7.5** Utilisez le principe des tiroirs ou une autre astuce afin de résoudre les problèmes suivants.

- (a) Combien d'étudiants doit-il y avoir dans une classe pour qu'au moins 6 aient la même note (A, B, C, D ou E)?
- (b) Combien de mots (sur les lettres A à Z) de 6 lettres doit-on engendrer pour qu'au moins quatre mots aient les mêmes 3 premières lettres?
- (c) Déterminez quel est le plus petit nombre d'indicatifs régionaux nécessaires pour couvrir 17 millions d'abonnés téléphoniques si les numéros de téléphone ont 10 chiffres, incluant le code régional de 3 chiffres:  $AXX-AXX-XXXX$ , où  $2 \leq A \leq 9$  et  $0 \leq X \leq 9$ .

**7.6**

- (a) Un programme est conçu pour générer tous les mots de 2 lettres différentes formés à partir des lettres A, B, C, D et E. Combien de mots différents devrait-il générer?
- (b) De combien de façons 5 femmes et 3 hommes peuvent-ils être disposés dans une file d'attente si les 3 hommes doivent être séparés?

**7.7**

- (a) Combien de trains de bits de longueur 8 contiennent exactement trois 1?
- (b) Combien de trains de bits de longueur 8 contiennent au plus trois 1?
- (c) De combien de façons peut-on choisir les 6 numéros du tirage de la 6/49?
- (d) On veut former un comité de 6 étudiants dont 4 sont en génie logiciel et 2 en génie électrique. Sachant qu'il y a 657 étudiants inscrits en génie logiciel et 753 en génie électrique, de combien de façons peut-on former le comité?

**7.8** On compte 40 professeurs et 300 étudiants dans un département.

- (a) Combien y a-t-il de façons différentes de former un comité de 2 professeurs et 5 étudiants de ce département?
- (b) Combien y a-t-il de façons différentes de former un comité de 5 personnes de ce département: soit 2 professeurs et 3 étudiants ou 3 professeurs et 2 étudiants?
- (c) Combien y a-t-il de façons différentes de former un comité de 5 personnes de ce département, sans contrainte sur le fait de choisir des professeurs ou des étudiants?
- (d) Combien y a-t-il de façons différentes de former un comité de 4 à 8 personnes de ce département, sans contrainte sur le fait de choisir des professeurs ou des étudiants?

**7.9** Vous programmez un générateur de mots. À partir des lettres fournies (avec possibilité de doublons), le générateur formera toutes les suites possibles et vérifiera si chacune des suites de lettres constitue un nom commun dans le dictionnaire.

- (a) Combien de suites de 7 lettres formera le générateur si les lettres fournies sont BONJOUR?
- (b) Combien de suites de 8 lettres formera le générateur si les lettres fournies sont RAPLAPLA?
- (c) ★ Combien de suites de 7 ou 8 lettres formera le générateur si les lettres fournies sont RAPLAPLA?

**7.10** Une chaîne ternaire est une suite de symboles choisis parmi 0, 1 ou 2. Procédez aux dénombrements suivants.

- (a) Combien de chaînes ternaires de longueur 7 ne contiennent aucun 0?
- (b) Combien de chaînes ternaires de longueur 7 commencent par 012 et se terminent par 0?
- (c) Combien de chaînes ternaires de longueur 7 commencent par 012 ou se terminent par 0? (Le *ou* est inclusif, comme toujours.)
- (d) Combien de chaînes ternaires de longueur 7 commencent et finissent par le même symbole: 0\*\*\*\*0 ou 1\*\*\*\*1 ou 2\*\*\*\*2?
- (e) Combien de chaînes ternaires de longueur 7 possèdent exactement trois 0?
- (f) Combien de chaînes ternaires de longueur 7 possèdent au moins trois 0?

**7.11** Un cadenas possède une serrure réinitialisable à code de 6 chiffres (parmi les chiffres 0 à 9). Par exemple, 935011 est un code possible.

- (a) Combien peut-on former de codes possibles au total?
- (b) Combien de codes commencent par 45 ou se terminent par un chiffre pair?
- (c) Combien de codes possèdent au moins un 9?
- (d) Combien de codes possèdent exactement un 0 et un 1?
- (e) Combien de codes contiennent deux 0, deux 1, un 2 et un 3, si chaque 1 doit être suivi d'un 0?
- (f) Combien de codes contiennent exactement trois 9 parmi les 5 premières positions?
- (g) Si les chiffres doivent tous être distincts, combien peut-on former de codes au total?
- (h) Si les chiffres doivent tous être distincts, combien de codes possèdent un 9?
- (i) Si les chiffres doivent tous être distincts, combien de codes commencent et se terminent par un chiffre impair?

## 7.4 Relations de récurrence et dénombrement

Rappelons qu'une **relation de récurrence** de la suite  $\{a_n\}$  est une équation qui exprime  $a_n$  en termes de  $a_0, a_1, \dots, a_{n-1}$ . Il est possible de résoudre des problèmes liés au dénombrement en utilisant les relations de récurrence.

### Exemple 7.15

Soit  $a_n$  le nombre de trains de bits de longueur  $n$  qui ne contiennent pas la chaîne "11".

- Trouver une relation de récurrence pour  $a_n$ .
- Combien y a-t-il de trains de bits de longueur 8 qui ne contiennent pas deux 1 consécutifs?

### Solution :

- On suppose que  $n$  est *suffisamment grand* et on considère tous les trains de bits comptés par  $a_n$ . Par le principe de la somme, le nombre de trains de bits de longueur  $n$  qui ne contiennent pas la chaîne "11" est égal au nombre de trains de bits de longueur  $n$  sans une occurrence de "11" qui commencent par 0, plus le nombre de trains de bits de longueur  $n$  sans une occurrence de "11" qui commencent par 1. Ces trains de bits sont donc de l'une des formes suivantes :

Forme des trains de bits	Nombre de trains de bits de cette forme
0   train de bits de lng $n - 1$ sans une occurrence de 11	$a_{n-1}$
1   0   train de bits de lng $n - 2$ sans une occurrence de 11	$a_{n-2}$

On en déduit la relation de récurrence

$$\begin{cases} a_1 = 2 & (0, 1) \\ a_2 = 3 & (00, 01, 10) \\ a_n = a_{n-1} + a_{n-2}, & n \geq 3 \end{cases}$$

- $a_8 = 55$

### Exemple 7.16

Soit  $b_n$  le nombre de trains de bits de longueur  $n$  qui contiennent au moins une occurrence de la chaîne "11".

- Trouver une relation de récurrence pour  $b_n$ .
- Combien y a-t-il de trains de bits de longueur 8 qui contiennent deux 1 consécutifs?

### Solution :

- On suppose que  $n$  est *suffisamment grand* et on considère tous les trains de bits comptés par  $b_n$ . Par le principe de la somme, le nombre de trains de bits de longueur  $n$  qui contiennent la chaîne "11" est égal au nombre de trains de bits de longueur  $n$  avec au moins une occurrence de "11" qui commencent par 0, plus le nombre de trains de bits de longueur  $n$  avec au moins une occurrence de "11" qui commencent par 1. Ces trains de bits sont donc de l'une des formes suivantes :

Forme des trains de bits	Nombre de trains de bits de cette forme
0   train de bits de $\ln g n - 1$ avec au moins une occurrence de 11	$b_{n-1}$
1   0   train de bits de $\ln g n - 2$ avec au moins une occurrence de 11	$b_{n-2}$
1   1   train de bits quelconque de $\ln g n - 2$	$2^{n-2}$

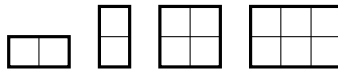
On en déduit la relation de récurrence

$$\begin{cases} b_1 = 0 \\ b_2 = 1 \\ b_n = b_{n-1} + b_{n-2} + 2^{n-2}, \quad n \geq 3 \end{cases} \quad \text{le seul train de bits 11}$$

(b)  $b_8 = 201$ . On vérifie que  $b_8 = 2^8 - a_8 = 2^8 - 55 = 201$ .

### Exemple 7.17

Soit  $a_n$  le nombre de façons de paver un trottoir de dimensions  $2 \times n$  en utilisant des tuiles rectangulaires de dimensions  $1 \times 2$ ,  $2 \times 1$ ,  $2 \times 2$  et  $2 \times 3$ :



- Donnez une relation de récurrence pour  $a_n$ , avec les conditions initiales appropriées.
- Combien de façons y a-t-il de paver un plancher de dimension  $2 \times 18$  en utilisant les tuiles décrites ci-haut?

### Solution :

- On suppose que  $n$  est *suffisamment grand* et on considère tous les pavages comptés par  $a_n$ . Ces pavages sont forcément de l'une des formes suivantes :

Forme des pavages	Nombre de pavages de cette forme
pavage de dimensions $2 \times (n - 1)$	$a_{n-1}$
pavage de dimensions $2 \times (n - 2)$	$a_{n-2}$
pavage de dimensions $2 \times (n - 2)$	$a_{n-2}$
pavage de dimensions $2 \times (n - 3)$	$a_{n-3}$

On en déduit la relation de récurrence

$$\begin{cases} a_1 = 1 \\ a_2 = 3 \\ a_n = a_{n-1} + 2a_{n-2} + a_{n-3}, \quad n \geq 3 \end{cases}$$

(b)  $a_{18} = 578\,949$ .

### Exercices

**7.12** Soit  $b_n$  le nombre de chaînes ternaires (avec des 0, 1, ou 2) de longueur  $n$  qui contiennent au moins deux 1 consécutifs.

- Déterminez  $b_0$  et  $b_1$ .
- Trouvez une relation de récurrence pour  $b_n$ .
- Combien y a-t-il de chaînes ternaires de longueur 8 qui contiennent au moins deux 1 consécutifs?

**7.13** Soit  $a_n$  le nombre de chaînes ternaires de longueur  $n$  ne comportant pas trois 0 consécutifs.

- Établissez une relation de récurrence avec conditions initiales pour  $a_n$ . Expliquez pourquoi la relation de récurrence donnée correspond bien à la situation.
- Donnez la valeur de  $a_{15}$ .
- Quelle proportion des chaînes ternaires de longueur 15 n'ont pas trois 0 consécutifs?
- Combien de chaînes ternaires de longueur 7 possèdent au moins trois 0 consécutifs?

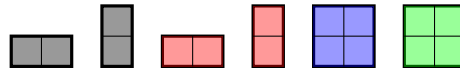
**7.14** Soit  $a_n$  le nombre de chaînes ternaires de longueur  $n$  qui ne contiennent aucune occurrence de la chaîne "000" ni de la chaîne "01".

- Donnez  $a_1$ ,  $a_2$  et  $a_3$  (conditions initiales).
- Trouvez une relation de récurrence satisfaite par  $a_n$ , pour  $n \geq 4$ .
- Combien de chaînes ternaires de longueur 15 ne contiennent aucune occurrence de la chaîne "000" ni de la chaîne "01"?

**7.15** Soit  $b_n$  le nombre de trains de bits de longueur  $n$  qui ne contiennent pas la chaîne "10".

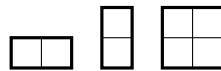
- Établissez une relation de récurrence avec conditions initiales pour  $b_n$ . Expliquez pourquoi la relation de récurrence donnée correspond bien à la situation.
- Donnez la valeur de  $b_{30}$ .


**7.16** Soit  $a_n$  le nombre de façons de paver un trottoir de dimensions  $2 \times n$  en utilisant des tuiles rectangulaires de dimensions  $1 \times 2$  (et  $2 \times 1$ ) de couleur noire et rouge, de dimensions  $2 \times 2$  de couleur bleue et verte :



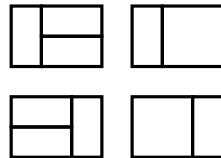
- Donnez une relation de récurrence pour  $a_n$ , avec les conditions initiales appropriées.
- Combien de façons y a-t-il de paver un plancher de dimension  $2 \times 15$  en utilisant les tuiles décrites ci-haut?

**7.17** Considérons les tuiles rectangulaires de dimensions  $1 \times 2$ ,  $2 \times 1$  et  $2 \times 2$  :



Soit  $a_n$  le nombre de pavages d'un trottoir  $2 \times n$  avec ces tuiles, sans le motif .

Par exemple,  $a_3 = 4$  puisque les quatre pavages valides d'un trottoir  $2 \times 3$  sont les suivants :



- Trouvez  $a_1$ ,  $a_2$  et  $a_4$ .
- Donnez une relation de récurrence satisfaite par  $a_n$ , ainsi que les cas de bases appropriés.
- Calculez  $a_{15}$ .

# Chapitre 8

## Théorie des graphes

### 8.1 Terminologie et types de graphes

#### Définition 8.1 : Graphe non orienté

Un **graphe non orienté**  $G = (V, E)$  est un couple d'ensembles, où  $V$  est un ensemble de **sommets** et  $E$  un ensemble d'**arêtes**. Chaque arête est associée à un ensemble d'un ou deux sommets.

Comme l'illustre la figure 8.1, les arêtes représentent les liens qui existent entre les sommets du graphe. Lorsque plusieurs arêtes sont associées au même ensemble de sommets, on les appelle des **arêtes multiples**. Une arête qui relie un sommet à lui-même est appelée une **boucle**.

Deux sommets  $u, v \in V$  sont **adjacents** s'ils sont joints par une arête  $e \in E$ . On dit alors que  $e$  est une **arête incidente** aux sommets  $u$  et  $v$ . Le **degré** d'un sommet  $v$ , noté  $\deg(v)$ , désigne le nombre d'arêtes qui lui sont incidentes.

**Note:** Une boucle compte pour deux dans le degré d'un sommet.

**Notation:** Lorsqu'il n'y a qu'une seule arête reliant une paire de sommets  $u$  et  $v$ , on la dénote parfois par  $\{u, v\}$  ou encore plus simplement par  $u-v$ .

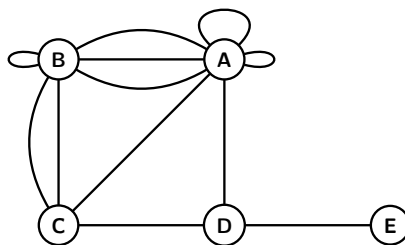


Figure 8.1 Exemple de graphe non orienté comportant 5 sommets et 12 arêtes, dont 3 boucles.

**Exemple 8.1**

Dessinez le graphe non orienté  $G = (V, E)$ , où

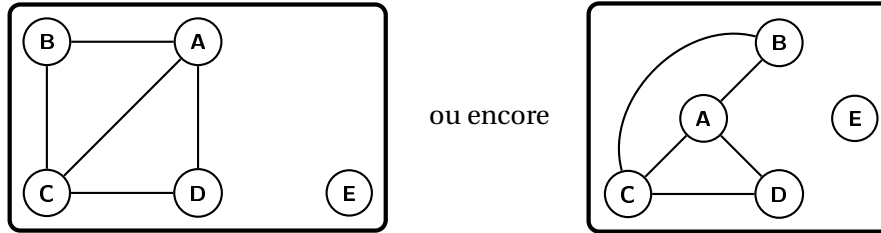
$$V = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}\},$$

$$E = \{\{\mathbf{A}, \mathbf{B}\}, \{\mathbf{A}, \mathbf{C}\}, \{\mathbf{A}, \mathbf{D}\}, \{\mathbf{B}, \mathbf{C}\}, \{\mathbf{C}, \mathbf{D}\}\}.$$

De plus, déterminez le degré de chacun des sommets.

**Solution :**

Il y a plusieurs façons de dessiner un graphe. En effet, on peut illustrer le graphe décrit plus haut comme



Dans les deux cas, on voit que les arêtes respectives entre les sommets sont préservées ainsi que les degrés des sommets :

$$\deg(\mathbf{A}) = 3, \quad \deg(\mathbf{B}) = 2, \quad \deg(\mathbf{C}) = 3, \quad \deg(\mathbf{D}) = 2, \quad \deg(\mathbf{E}) = 0.$$

**Théorème 8.1 : des poignées de mains**

Soit  $G = (V, E)$  un graphe non orienté. Alors

$$\sum_{v \in V} \deg(v) = 2|E|.$$

**Exemple 8.2**

Pour illustrer le théorème des poignées de mains, considérons le graphe non orienté de l'exemple 8.1 qui contient 5 arêtes :

$$\begin{aligned} \sum_{v \in V} \deg(v) &= \deg(\mathbf{A}) + \deg(\mathbf{B}) + \deg(\mathbf{C}) + \deg(\mathbf{D}) + \deg(\mathbf{E}) \\ &= 3 + 2 + 3 + 2 + 0 \\ &= 10 \\ &= 2|E|. \end{aligned}$$



**Définition 8.2 : Graphe orienté**

Un **graphe orienté**  $G = (V, E)$  est un couple d'ensembles, où  $V$  est un ensemble de **sommets** et  $E$  un ensemble d'**arcs**. Chaque arc est associé à une *paire ordonnée* de sommets.

Si plusieurs arcs relient la même paire ordonnée de sommets, on les appelle des **arcs multiples**. Un arc qui relie un sommet à lui-même est appelé une **boucle**.

Soit  $e \in E$  un arc associé à la paire de sommets  $(u, v)$ . Le sommet  $u$  est appelé **sommet initial** et le sommet  $v$  **sommet terminal** de  $e$ . On dit que le sommet  $v$  est **adjacent** au sommet  $u$ . Le **degré sortant** d'un sommet  $v$ , noté  $\deg^+(v)$ , désigne le nombre d'arcs dont  $v$  est le sommet initial. Le **degré entrant** d'un sommet  $v$ , noté  $\deg^-(v)$ , désigne le nombre d'arcs dont  $v$  est le sommet terminal.

**Note:** Une boucle contribue pour 1 degré entrant et 1 degré sortant.

**Notation:** Lorsqu'il n'y a qu'un seul arc du sommet  $u$  au sommet  $v$ , on le dénote parfois par  $(u, v)$  ou encore plus simplement par  $u \rightarrow v$ .

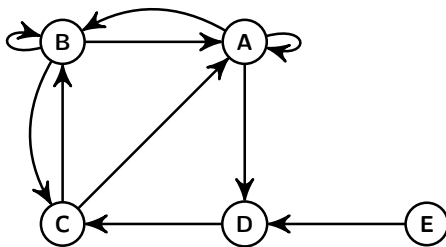
**Exemple 8.3**

Dessinez le graphe orienté  $G = (V, E)$ , où

$$V = \{A, B, C, D, E\},$$

$$E = \{(A, A), (A, B), (A, D), (B, B), (B, A), (B, C), (C, A), (C, B), (D, C), (E, D)\}.$$

De plus, déterminez les degrés entrants et sortants de chacun des sommets.

**Solution :**

$$\deg^+(A) = 3$$

$$\deg^-(A) = 3$$

$$\deg^+(B) = 3$$

$$\deg^-(B) = 3$$

$$\deg^+(C) = 2$$

$$\deg^-(C) = 2$$

$$\deg^+(D) = 1$$

$$\deg^-(D) = 2$$

$$\deg^+(E) = 1$$

$$\deg^-(E) = 0$$

**Théorème 8.2 : des coups de pieds**

Soit  $G = (V, E)$  un graphe orienté. Alors

$$\sum_{v \in V} \deg^+(v) = \sum_{v \in V} \deg^-(v) = |E|$$

**Exemple 8.4**

Pour illustrer le théorème des coups de pied, considérons le graphe orienté de l'exemple 8.3 qui contient 10 arcs :

$$\begin{aligned} \sum_{v \in V} \deg^+(v) &= \deg^+(\mathbf{A}) + \deg^+(\mathbf{B}) + \deg^+(\mathbf{C}) + \deg^+(\mathbf{D}) + \deg^+(\mathbf{E}) \\ &= 3 + 3 + 2 + 1 + 1 \\ &= 10 \\ &= |E| \end{aligned}$$

$$\begin{aligned} \sum_{v \in V} \deg^-(v) &= \deg^-(\mathbf{A}) + \deg^-(\mathbf{B}) + \deg^-(\mathbf{C}) + \deg^-(\mathbf{D}) + \deg^-(\mathbf{E}) \\ &= 3 + 3 + 2 + 2 + 0 \\ &= 10 \\ &= |E|. \end{aligned}$$

Dans certains théorèmes, définitions et exemples, nous utilisons le terme **graphe** pour désigner un graphe orienté ou non. Nous utilisons aussi parfois le terme graphe pour parler de graphe non orienté seulement, quand le contexte ne porte pas à confusion.

**Définition 8.3 : Graphe simple**

Un graphe qui ne contient aucune boucle ni arête (ou arc) multiple est qualifié de **simple**.

Un graphe peut donc être orienté ou non, et simple ou non. Le tableau 8.1 présente chacun des 4 types de graphes avec leurs caractéristiques et un exemple. Attention, certains auteurs utilisent le terme *graphe* pour désigner un *graphe simple*: il faut être vigilant avec les définitions quand on passe d'un livre à un autre.

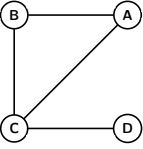
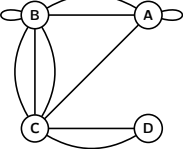
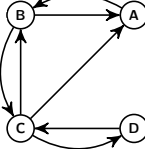
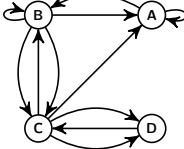
Graphe simple non orienté	Graphe non orienté	Graphe simple orienté	Graphe orienté
			
<b>Arêtes multiples et boucles</b>	<b>Arêtes multiples et boucles</b>	<b>Arcs multiples et boucles</b>	<b>Arcs multiples et boucles</b>
non permises	permises	non permis	permis

Tableau 8.1 Types de graphes

Plusieurs situations de la vie courante peuvent être modélisées par des graphes. Par exemple, le graphe de la figure 8.2 représente le trafic aérien mondial, où les sommets correspondent aux villes et les arêtes aux liens aériens. Le graphe de la figure 8.3 est un graphe partiel d'Internet, où les sommets correspondent à des adresses IP et les arêtes aux liens entre elles.



Figure 8.2 Carte du trafic aérien. Jpatokal (2009)

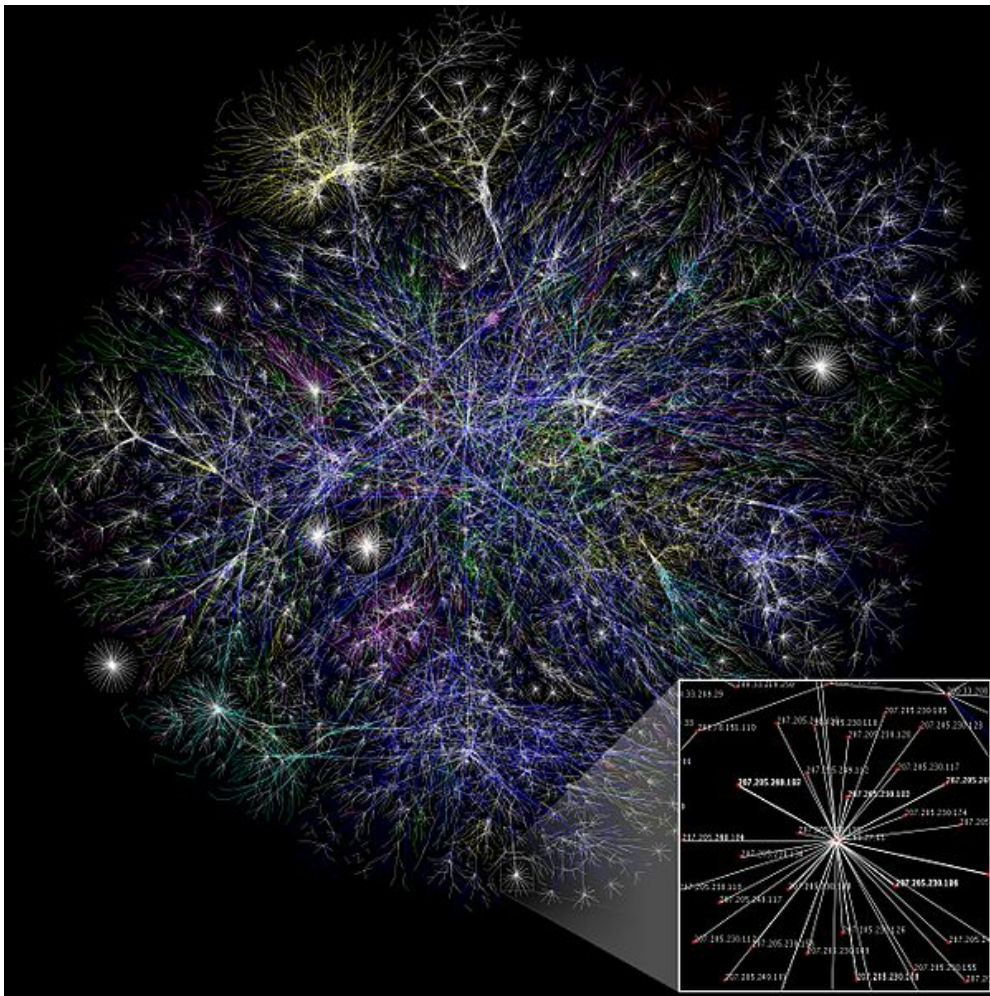


Figure 8.3 Graphe partiel d'Internet. The Opte Project (2006).

## Exercices

**8.1** Prouvez qu'un graphe non orienté possède toujours un nombre pair de sommets de degré impair.

**8.2** Un graphe simple non orienté est dit **complet** si tous ses sommets sont adjacents. Quel est le nombre total d'arêtes dans le graphe complet

- (a)  $K_5$  à 5 sommets?
- (b)  $K_n$  à  $n$  sommets?

Indice: il existe plusieurs façons de résoudre ce problème. Par exemple, on peut utiliser le théorème 8.1 ou encore, le théorème 4.7.

## 8.2 Représentation des graphes

Pour qu'un ordinateur puisse manipuler un graphe  $G = (V, E)$ , il faut une structure de données pour représenter les ensembles  $V$  et  $E$ . Dans cette section, nous donnons deux manières de représenter un graphe, soit par une liste d'adjacence ou une matrice d'adjacence.

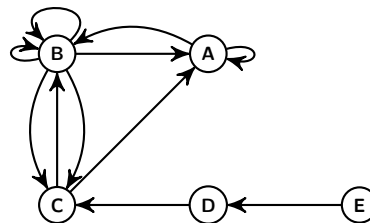
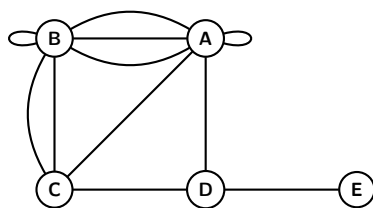
### 8.2.1 Représentation par listes d'adjacence

Un graphe est représenté par un tableau associatif qui, à chaque sommet, associe la liste des sommets auxquels il est adjacent. Autrement dit, la liste associée au sommet  $u$  contient le sommet  $v$  si et seulement s'il existe une arête (ou arc) de  $u$  à  $v$ .

Cette méthode est avantageuse dans le cas où le graphe compte un grand nombre  $n$  de sommets et où le nombre d'arêtes (ou d'arcs) est significativement plus petit que  $n^2$ .

#### Exemple 8.5

Représentez par une liste d'adjacence chacun des graphes suivants.



**Solution :**

sommet	liste d'adjacence	sommet initial	liste d'adjacence
A	A,B,B,B,C,D	A	A,B
B	A,A,A,B,C,C	B	A,B,B,C,C
C	A,B,B,D	C	A,B
D	A,C,E	D	C
E	D	E	D

**8.2.2 Représentation par matrice d'adjacence**

Un graphe est représenté par une matrice dont la case en ligne  $i$ , colonne  $j$  indique le nombre d'arêtes (ou d'arcs) allant du  $i^e$  sommet au  $j^e$  sommet.

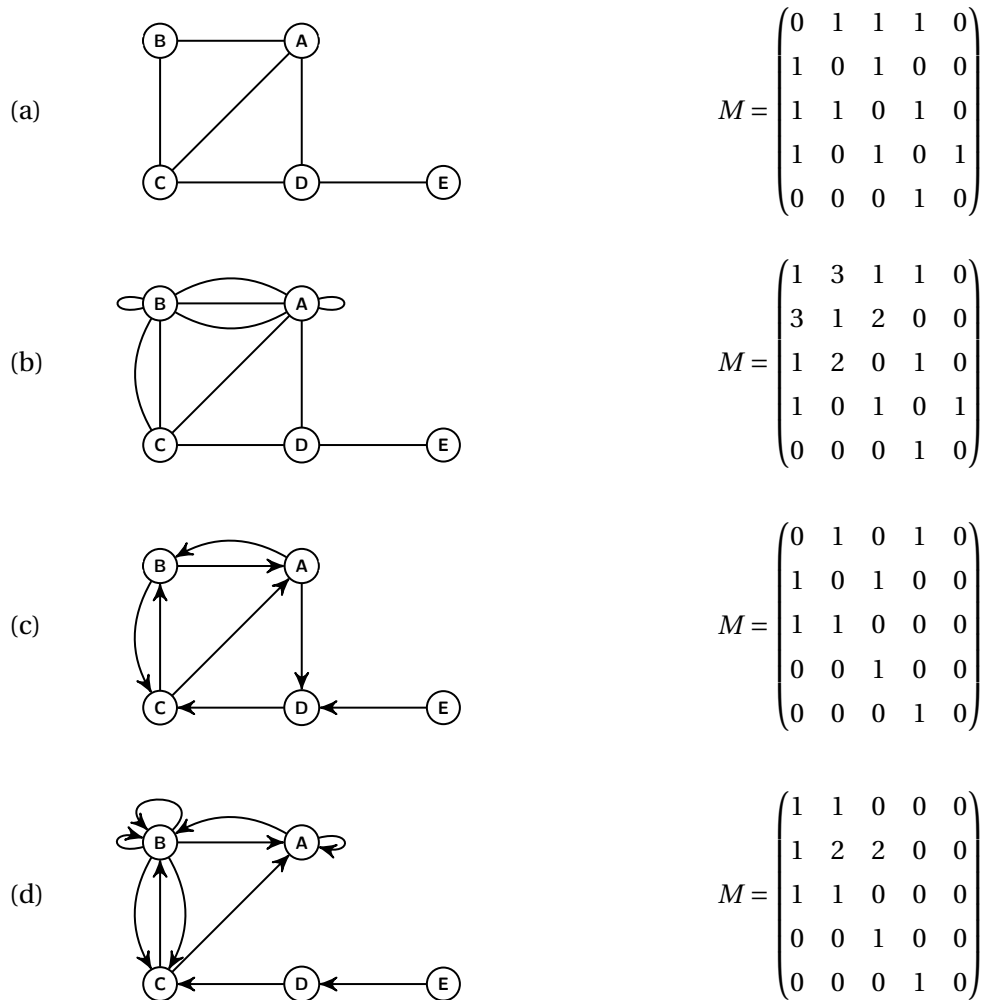
**Définition 8.4 : Matrice d'adjacence**

Soit  $G$  un graphe de  $n$  sommets  $\{v_1, v_2, \dots, v_n\}$ . La **matrice d'adjacence** est la matrice carrée  $M_{n \times n} = [m_{ij}]$  de dimension  $n \times n$ , où l'entrée

$$m_{ij} = \text{nombre d'arêtes (ou d'arcs) du sommet } v_i \text{ au sommet } v_j.$$

**Exemple 8.6**

Considérez les quatre graphes suivants et leur matrice d'adjacence (en prenant l'ordre alphabétique sur les sommets). Pour chacun des graphes, déterminez s'il est simple ou non, orienté ou non.

**Solution :**

- (a) Graphe simple non orienté. Dans un graphe non orienté, on remarque que la matrice d'adjacence est symétrique par rapport à la diagonale. De plus, dans un graphe simple, la diagonale ne contient que des 0.
- (b) Graphe non orienté.
- (c) Graphe simple orienté.
- (d) Graphe orienté.

## 8.3 Chemins dans un graphe

### 8.3.1 Chemins, circuits, cycles

#### Définition 8.5 : Chemin, circuit, cycle, simple ou élémentaire

Soit  $G = (V, E)$  un graphe orienté ou non. Un **chemin** de longueur  $n$  du sommet  $u$  au sommet  $v$  est une suite de  $n$  arêtes (arcs)

$$e_1, e_2, \dots, e_n$$

telle qu'il existe une suite de sommets

$$u = v_0, v_1, v_2, \dots, v_n = v$$

où, pour  $i$  allant de 1 à  $n$ , l'arête  $e_i = \{v_{i-1}, v_i\}$  (ou l'arc  $e_i = (v_{i-1}, v_i)$ ).

- S'il n'y a pas d'arêtes (arcs) multiples, on peut désigner ce chemin par  $v_0 - v_1 - \dots - v_n$ .
- Un **circuit**, aussi appelé **cycle**, est un chemin avec  $v_0 = v_n$ , pour  $n > 0$ . (On réserve parfois le terme *circuit* aux graphes orientés et le terme *cycle* aux graphes non orientés, mais nous les considérerons ici comme synonymes.)
- Un chemin, circuit ou cycle est dit **simple** s'il ne contient pas une même arête (arc) plus d'une fois.
- Un circuit ou cycle est dit **élémentaire** s'il ne contient pas un même sommet plus d'une fois (sauf le premier et le dernier). Un cycle élémentaire ne contient pas d'autre cycle.

Dans le graphe de la figure 8.4, la suite d'arêtes  $e_1, e_2, e_3, e_4, e_5$  est un chemin allant du sommet  $u$  vers le sommet  $v$ . Comme le graphe est simple, on peut aussi désigner ce chemin par  $v_0 - v_1 - v_2 - v_3 - v_4 - v_5$ . La suite d'arêtes  $e_2, e_6, e_7, e_8$  est un circuit simple et élémentaire. La suite d'arêtes  $e_1, e_2, e_2$  (ou encore  $v_0 - v_1 - v_2 - v_1$ ) est un chemin qui n'est pas simple, comme l'arête  $e_2$  se répète.

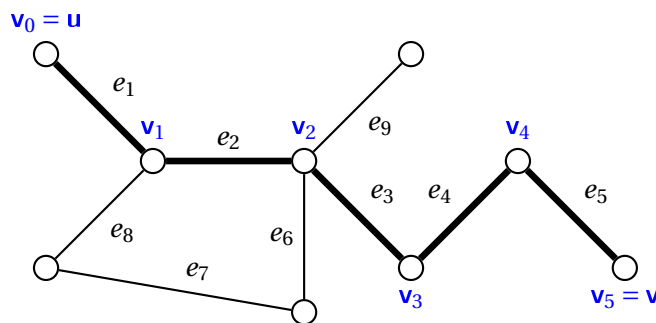
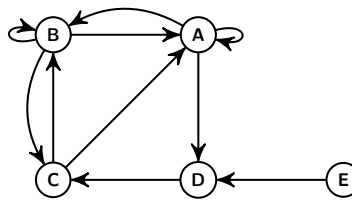


Figure 8.4 La suite d'arêtes  $e_1, e_2, e_3, e_4, e_5$  est un chemin allant du sommet  $u$  vers le sommet  $v$ .

**Exemple 8.7**

Soit le graphe orienté suivant.



Déterminez si les chemins suivants sont des circuits simples ou élémentaires. De plus, donnez leurs longueurs.

- (a) A–B–A–B–C–B–A
- (b) B–C–A–D–C–B
- (c) D–C–B–A–D

**Solution :**

- (a) Circuit de longueur 6 qui n'est ni simple, ni élémentaire.
- (b) Circuit simple, mais pas élémentaire car il passe 2 fois par le sommet **C**, de longueur 5.
- (c) Circuit simple et élémentaire de longueur 4.

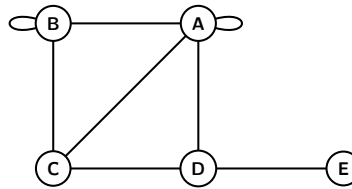
**8.3.2 Dénombrement de chemins****Théorème 8.3**

Soit  $G$  un graphe et  $M$  sa matrice d'adjacence (en respectant l'ordre  $v_1, v_2, \dots, v_n$  des sommets de  $G$ ). Alors le nombre de chemins différents de longueur  $k \in \mathbb{N}^*$  allant du sommet  $v_i$  au sommet  $v_j$  est donné par l'entrée  $(i, j)$  de la matrice  $M^k$ .



**Exemple 8.8**

Déterminez le nombre de chemins de longueur 4 allant du sommet **E** au sommet **B** dans le graphe suivant et dressez une liste de ces chemins.

**Solution :**

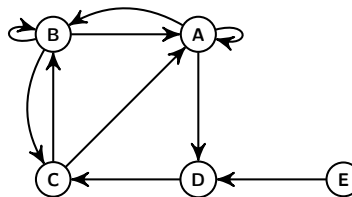
$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad M^4 = \begin{pmatrix} 39 & 31 & 30 & 23 & 8 \\ 31 & 26 & 23 & 20 & 5 \\ 30 & 23 & 24 & 16 & 7 \\ 23 & 20 & 16 & 18 & 3 \\ 8 & 5 & 7 & 3 & 3 \end{pmatrix}$$

Chemins de longueur 4 allant du sommet **E** au sommet **B** :

E-D-A-A-B  
E-D-A-B-B  
E-D-A-C-B  
E-D-C-A-B  
E-D-C-B-B

**Exemple 8.9**

Déterminez le nombre de chemins de longueur 3 allant du sommet **C** au sommet **A** dans le graphe suivant et dressez une liste de ces chemins.

**Solution :**

$$M = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad M^3 = \begin{pmatrix} 6 & 6 & 3 & 2 & 0 \\ 7 & 7 & 4 & 3 & 0 \\ 5 & 5 & 3 & 2 & 0 \\ 2 & 2 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Chemins de longueur 3 allant du sommet **C** au sommet **A** :

C-A-A-A  
C-A-B-A  
C-B-A-A  
C-B-B-A  
C-B-C-A

### Exercices

**8.3** Les randonnées pédestres d'un parc national reliant différents sites (A, B, C, ...) à visiter sont représentées par un graphe. Les sommets correspondent aux différents sites et les arêtes aux randonnées d'une heure entre deux sites. La matrice d'adjacence du graphe, construite selon l'ordre alphabétique des sommets, est:

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Ainsi, l'entrée  $m_{12} = 1$  de la matrice signifie qu'il y a une randonnée d'une heure entre le site A et le site B. Répondez aux questions suivantes en utilisant seulement des opérations matricielle sur  $M$ .

- Quel est le nombre de randonnées de 5 heures partant du site B?
- Quel est le nombre de randonnées de 4 heures ou moins qui se terminent au site C?
- Quel est le nombre de randonnées de 4 heures dont le site de départ est le même que le site d'arrivée?
- Quelle est la durée minimale (en heures) d'une randonnée entre le site A et le site K?

### 8.3.3 Chemins et circuits eulériens

Le problème des **sept ponts de Königsberg** consiste à déterminer si l'on peut prendre une marche dans les rues de la ville (voir figure 8.5) en passant une et une seule fois par chaque pont, en partant d'un point donné et en revenant à ce même point.

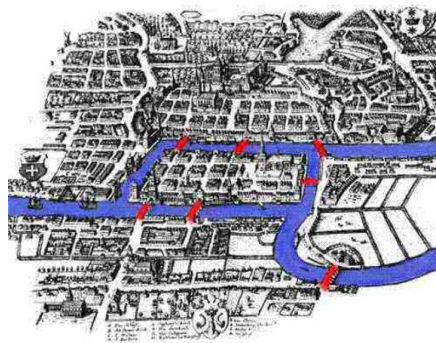


Figure 8.5 Ville de Königsberg.

Le mathématicien Euler a donné une solution à ce problème, qui est considérée comme le fondement même de la théorie des graphes. Dans sa version plus moderne, la solution repose sur l'existence de circuit eulérien dans un graphe. Nous présenterons donc la solution à ce problème après avoir introduit certaines notions importantes.

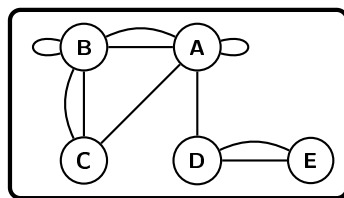
**Définition 8.6 : Chemin et circuit eulériens**

- Un **chemin eulérien** est un chemin simple qui contient toutes les arêtes.
- Un **circuit eulérien** est un circuit simple qui contient toutes les arêtes.

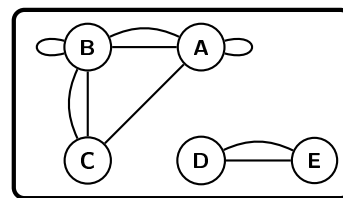
**Définition 8.7 : Graphe connexe**

Un graphe non orienté est **connexe** si pour toute paire de sommets  $(x, y)$ , il existe un chemin allant de  $x$  à  $y$ .

**Exemple 8.10**



graphe connexe



graphe non connexe

**Théorème 8.4**

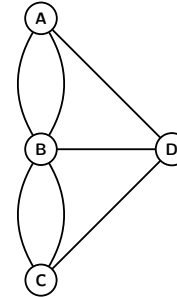
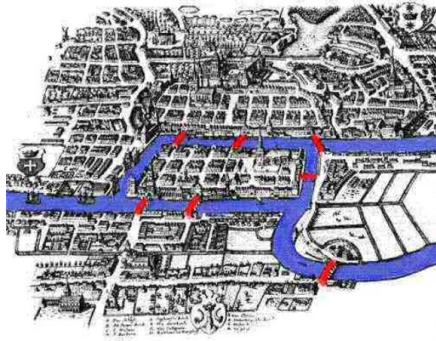
Un graphe connexe non orienté possède un circuit eulérien si et seulement si tous ses sommets sont de degré pair.

**Théorème 8.5**

Un graphe connexe non orienté possède un chemin eulérien, mais pas de circuit eulérien, si et seulement s'il possède exactement 2 sommets de degré impair.

**Exemple 8.11**

Nous pouvons maintenant revenir au problème des **sept ponts de Königsberg**. Est-il possible de prendre une marche dans les rues de la ville en passant une et une seule fois par chaque pont, en partant d'un point donné et en revenant à ce même point? La ville et ses ponts peuvent être modélisés par un graphe, où les ponts correspondent aux arêtes et les sommets aux différentes parties de la ville séparées par le cours d'eau:



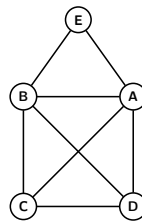
Le problème peut donc être reformulé comme suit: « Existe-t-il un circuit eulérien dans le graphe ci-dessus? »

**Solution :**

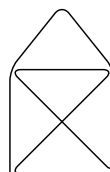
Le Théorème 8.4 nous permet de conclure que ce graphe ne possède aucun circuit eulérien, car il possède des sommets de degré impair. Il est donc impossible de trouver un itinéraire qui parcourt chaque pont de Königsberg une et une seule fois en revenant à son point de départ.

**Exemple 8.12**

Peut-on tracer cette maison sans lever le crayon et sans repasser sur un segment plus d'une fois?

**Solution :**

Comme ce graphe non orienté possède exactement 2 sommets de degré impair, les sommets C et D, le théorème 8.5 permet de conclure que ce graphe possède un chemin eulérien. En voici un: C–B–E–A–D–B–A–C–D. On peut donc tracer la maison sans lever le crayon et sans repasser sur un segment.



**Algorithme de Hierholzer**

L'**algorithme de Hierholzer** permet de construire un circuit eulérien dans un graphe non orienté, s'il existe. Le fait que cet algorithme est correct constitue une démonstration du Théorème 8.4. Étant donné un sommet  $v$ , on effectue le travail suivant :

*À partir de  $v$ , avancer tant qu'il est possible de le faire, en supprimant toutes les arêtes empruntées.*

Il est important de remarquer que si tous les sommets du graphe sont de degré pair, alors un tel chemin termine forcément à son point de départ. La fonction auxiliaire `chemin` formalise cette idée.

---

**Algorithme 1** `chemin`

---

**Entrées:** un graphe non orienté  $G$  et  $v$  un sommet de  $G$

**Sortie:** une liste de sommets décrivant un chemin dans  $G$

- 1:  $C :=$  liste vide
  - 2: Ajouter  $v$  à  $C$
  - 3: **tant que**  $\text{deg}(v) > 0$  **faire**
  - 4: Choisir une arête  $e$  incidente à  $v$  et appeler  $w$  le sommet à son autre extrémité
  - 5: Ajouter  $w$  à la fin de  $C$
  - 6: Retirer l'arête  $e$  du graphe  $G$
  - 7:  $v := w$
  - 8: **fin tant que**
  - 9: **retourner**  $C$
- 

---

**Algorithme 2** `Hierholzer`

---

**Entrées:** un graphe non orienté connexe  $G$  dont tous les sommets sont de degré pair

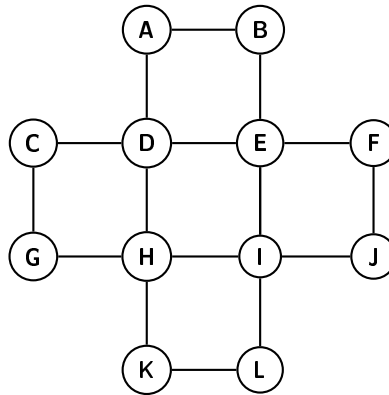
**Sortie:** un circuit eulérien de  $G$

- 1:  $circuit :=$  liste vide
  - 2: Ajouter un sommet de  $G$  à  $circuit$
  - 3:  $i := 0$
  - 4: **tant que**  $G$  possède au moins une arête **faire**
  - 5:  $v := circuit[i]$
  - 6:  $C := \text{chemin}(G, v)$
  - 7: Insérer la liste  $C$  dans  $circuit$ , à la place de  $circuit[i]$
  - 8:  $i := i + 1$
  - 9: **fin tant que**
  - 10: **retourner**  $circuit$
- 

Une utilisation judicieuse des listes chaînées permet d'implémenter l'algorithme de Hierholzer avec une complexité dans  $O(m)$  où  $m$  est le nombre d'arêtes du graphe.

**Exemple 8.13**

Exécuter l'algorithme de Hierholzer sur le graphe suivant en utilisant la convention que lorsque qu'il faut choisir parmi plusieurs sommets, on utilise toujours le premier en ordre alphabétique.

**Solution :**

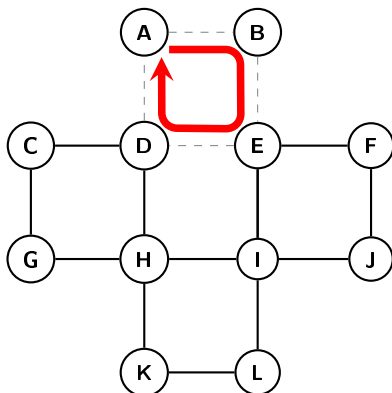
À la ligne 2 de l'algorithme de Hierholzer, on choisit le sommet **A** comme sommet de départ puisqu'il s'agit du plus petit en ordre alphabétique. Ainsi, avant d'entrer dans la boucle **tant que**, la liste *circuit* est **[A]**.

Afin d'effectuer une trace de l'algorithme, on donne l'état du graphe et de chacune des variables à chaque itération de la boucle **tant que**. On remarque que la variable *i* est incrémentée exactement une fois à chaque itération. On l'utilise donc pour identifier chacune des itérations.

- $i = 0$ .

Un chemin est construit à partir du sommet  $circuit[i] = \mathbf{A}$ . Les voisins de **A** étant les sommets **B** et **D**, le chemin débute avec l'arête **A–B** et celle-ci est retirée du graphe afin de s'assurer que le chemin construit ne l'emprunte pas une deuxième fois. Ensuite, à partir de **B** le chemin se poursuit à **E** puis à **D**. Les voisins du sommet **D** étant **A**, **C** et **H** (on rappelle que l'arête **E–D** a été retirée du graphe), le chemin se poursuit avec l'arête **D–A**. Le sommet **A** est alors de degré zéro et la fonction `chemin` termine et retourne la liste **[A,B,E,D,A]**. Finalement, à la ligne 7 de l'algorithme de Hierholzer, la liste *circuit* est modifiée. On remplace le sommet **A** par les sommets de la liste **[A,B,E,D,A]**.

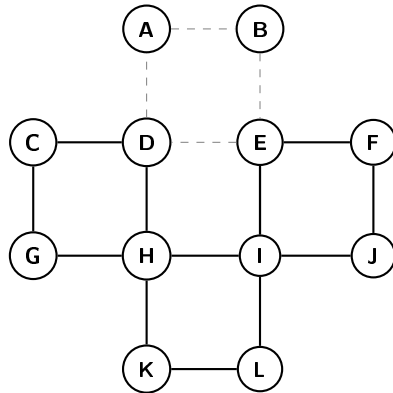
Voici l'état des données après la première itération de la boucle **tant que**.



$v = \mathbf{A}$ ,  
 $circuit = [\mathbf{A}, \mathbf{B}, \mathbf{E}, \mathbf{D}, \mathbf{A}]$ ,

- $i = 1$ .

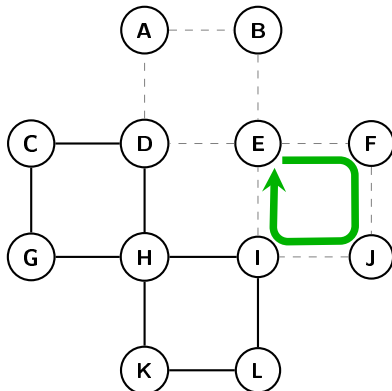
La fonction `chemin` est appelée à partir du sommet  $circuit[1] = B$ . Ce sommet étant déjà de degré 0. La fonction `chemin` termine donc sans modifier le graphe et la liste retournée est simplement  $[B]$ . Ensuite, à la ligne 7, la modification effectuée à la liste  $circuit$  n'a aucun effet puisque  $B$  est remplacé par  $B$ .



$v = A$ ,  
 $circuit = [A, B, E, D, A]$ ,

- $i = 2$ .

La fonction `chemin` est appelée à partir du sommet  $circuit[2] = E$ . En respectant l'ordre alphabétique sur les sommets, le chemin construit est  $[E, F, J, I, E]$ . Ensuite, à la ligne 7, le sommet  $E$  est remplacé par ce chemin.



$v = A$ ,  
 $circuit = [A, B, E, F, J, I, E, D, A]$ ,

- $i = 3$ .

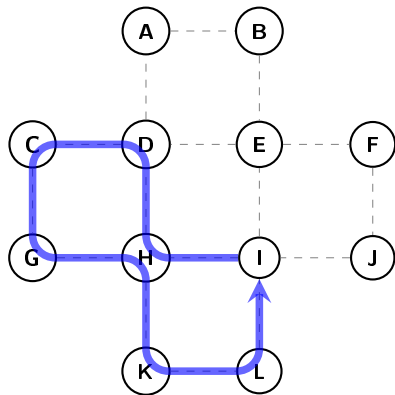
Comme pour le cas  $i = 1$ , le sommet  $circuit[3] = F$  est de degré zéro. Le graphe et la liste  $circuit$  restent inchangés.

- $i = 4$ .

Le sommet  $circuit[4] = J$  est de degré zéro.

- $i = 5$ .

La fonction `chemin` est appelée à partir du sommet `circuit[5] = I` et le chemin construit est  $[I, H, D, C, G, H, K, L, I]$ . Ensuite, à la ligne 7, le sommet `I` est remplacé par ce chemin.

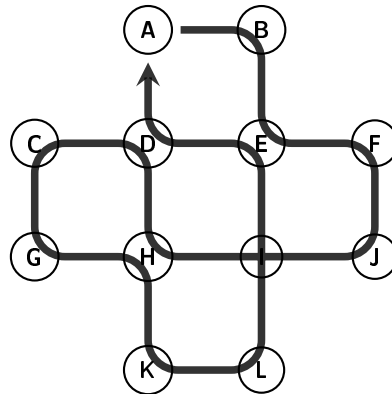
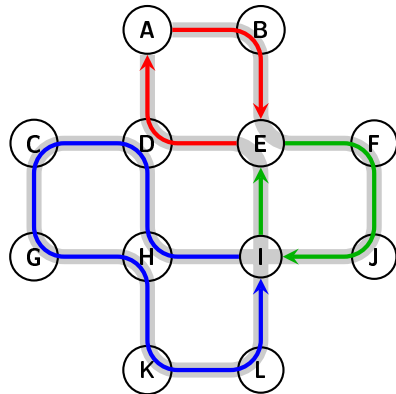


$v = A$ ,  
 $circuit = [A, B, E, F, J, I, H, D, C, G, H, K, L, I, E, D, A]$ .

- $i = 6, 7, \dots, 16$ .

Toutes les arêtes ont été retirées du graphes. Il ne reste plus que des sommets de degré zéro. Aucune modification n'est apportée au données pendant ces itérations.

Le chemin eulérien calculé est donc:  $A-B-E-F-J-I-H-D-C-G-H-K-L-I-E-D-A$ .





### 8.3.4 Chemins et circuits hamiltoniens

#### Définition 8.8 : Chemins et circuits hamiltoniens

- Un **chemin hamiltonien** est un chemin qui passe exactement une fois par tous les sommets.
- Un **circuit hamiltonien** est un circuit formé par un chemin hamiltonien suivi d'une arête menant au sommet de départ.

Attention, les deux théorèmes suivants ne donnent pas des conditions nécessaires et suffisantes pour déterminer si un graphe non orienté possède un circuit hamiltonien. Si les hypothèses sont respectées, on peut affirmer qu'un tel circuit existe, mais dans le cas contraire, on ne peut rien conclure.

#### Théorème 8.6 : Théorème de Ore

Soit  $G = (V, E)$  un graphe simple non orienté avec  $n$  sommets ( $n \geq 3$ ). Si pour tous sommets non adjacents  $u, v \in V$ ,

$$\deg(u) + \deg(v) \geq n$$

alors  $G$  possède un circuit hamiltonien.

#### Théorème 8.7 : Théorème de Dirac

Soit  $G = (V, E)$  un graphe simple non orienté avec  $n$  sommets ( $n \geq 3$ ). Si pour chaque sommet  $v \in V$  on a

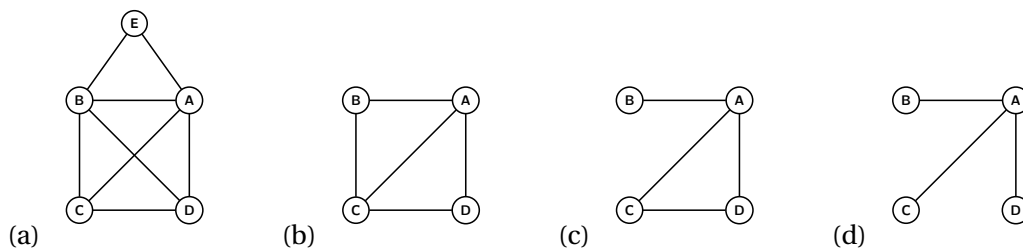
$$\deg(v) \geq \frac{n}{2},$$

alors  $G$  possède un circuit hamiltonien.

Le théorème de Dirac découle du théorème de Ore, bien qu'il ait été prouvé avant! En effet, si tous les sommets du graphe sont de degré plus grand ou égal à  $\frac{n}{2}$ , alors la somme des degrés de deux sommets non adjacents est forcément plus grande ou égale à  $\frac{n}{2} + \frac{n}{2} = n$ . Le théorème de Dirac est plus facile à vérifier, mais moins puissant que celui de Ore (voir exemple 8.14 (a)).

#### Exemple 8.14

Les graphes suivants possèdent-ils des circuits et chemins hamiltoniens? Est-ce que vous pouvez utiliser les Théorèmes 8.7 et 8.6 pour répondre à cette question?



**Solution :**

- (a) **A–D–C–B–E** chemin hamiltonien; **A–D–C–B–E–A** circuit hamiltonien.  
 Théorème 8.7: non; hypothèse n'est pas satisfaite puisque  $\deg(E) = 2 \not\geq \frac{5}{2} = 2,5$ .  
 Théorème 8.6: oui; hypothèse est satisfaite puisque  $\deg(D) + \deg(E) = 5 \geq 5$  et  $\deg(C) + \deg(E) = 5 \geq 5$ .
- (b) **A–B–C–D** chemin hamiltonien; **A–B–C–D–A** circuit hamiltonien.  
 Théorème 8.7: oui; hypothèse est satisfaite puisque pour tout sommet  $v$ ,  $\deg(v) \geq \frac{4}{2} = 2$ .  
 Théorème 8.6: oui; hypothèse est satisfaite puisque  $\deg(B) + \deg(D) = 4 \geq 4$ .
- (c) **B–A–D–C** chemin hamiltonien; pas de circuit hamiltonien.  
 Théorème 8.7: non; hypothèse n'est pas satisfaite puisque  $\deg(B) = 1 \not\geq \frac{4}{2} = 2$ .  
 Théorème 8.6: non; hypothèse n'est pas satisfaite puisque  $\deg(B) + \deg(D) = 3 \not\geq 4$ .
- (d) pas de chemin hamiltonien; pas de circuit hamiltonien.  
 Théorème 8.7: non; hypothèse n'est pas satisfaite puisque  $\deg(B) = 1 \not\geq \frac{4}{2} = 2$ .  
 Théorème 8.6: non; hypothèse n'est pas satisfaite puisque  $\deg(B) + \deg(D) = 2 \not\geq 4$ .

**Exercices**

**8.4** Tracez les graphes non orientés demandés et donner un circuit eulérien et un circuit hamiltonien. S'il un tel circuit n'existe pas, donner un argument comme justification.

**Note:** Tous les graphes de cet exercice sont décrits à la page Wikipédia suivante:

[https://en.wikipedia.org/wiki/Gallery\\_of\\_named\\_graphs](https://en.wikipedia.org/wiki/Gallery_of_named_graphs)

- (a) Les graphes complets  $K_4$ ,  $K_5$  et  $K_6$  (*complete graphs*).
- (b) Les graphes bipartis complets  $K_{3,3}$ ,  $K_{2,4}$  et  $K_{4,4}$  (*complete bipartite graphs*).
- (c) Les graphes d'amitié  $F_2$ ,  $F_3$ ,  $F_4$  (*friendship graphs*).
- (d) ★ Le graphe de Peterson.
- (e) ★ Les solides de Platon.

**8.4 Problème du plus court chemin (algorithme de Dijkstra)**

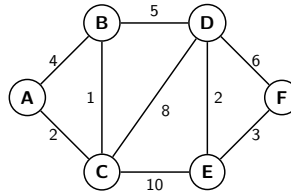
Plusieurs situations peuvent être modélisées à partir de graphes ayant un coût associé aux arêtes. Par exemple, si l'on veut représenter les distances qui relient les villes du Québec, les sommets du graphe sont identifiés par les villes et les arêtes par les routes qui les relient, ainsi que leurs distances. Comment trouver le chemin de distance minimale entre deux villes?

**Définition 8.9 : Graphe pondéré, distance, plus court chemin**

Un **graphe pondéré** est un graphe qui a un poids (ou coût) associé à chacune de ses arêtes. La **distance** (ou **coût** ou **poids total**) d'un chemin dans un graphe pondéré est la somme de chacun des poids des arêtes de ce chemin. Le **plus court chemin** (ou **chemin de coût minimal** ou **chemin de poids minimal**) d'un sommet à un autre est celui de distance minimale.

**Exemple 8.15**

Le graphe suivant est un graphe pondéré.



Le chemin **A–C–D–F** allant du sommet **A** au sommet **F** est de distance  $2 + 8 + 6 = 16$ .

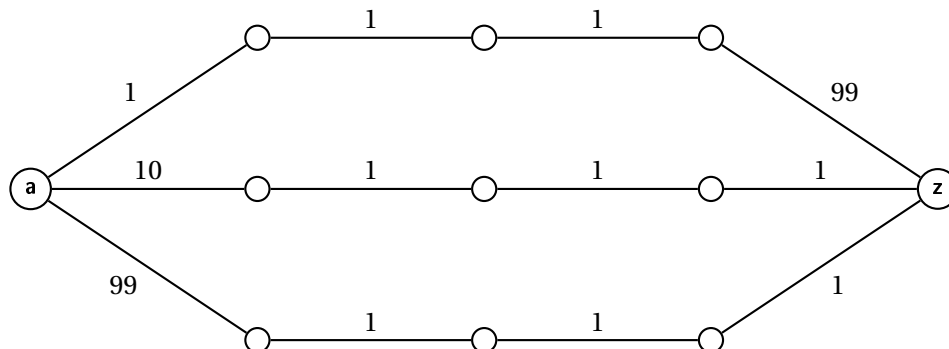
Le chemin **A–C–B–D–E–F** allant du sommet **A** au sommet **F** est de distance minimale  $2 + 1 + 5 + 2 + 3 = 13$ .

L'**algorithme de Dijkstra** donne la distance minimale d'un chemin allant d'un sommet à un autre dans un graphe pondéré de **poids positifs**.

Afin d'obtenir un chemin optimal, quelle que soit la structure du graphe considéré, l'algorithme se doit d'effectuer une exploration du graphe et de considérer plusieurs possibilités. Par contre, afin d'être le plus efficace possible, l'algorithme ne considère pas *toutes* les possibilités.

**Exemple 8.16**

On considère le graphe ci-dessous. Un simple coup d'oeil permet de constater que le chemin le moins coûteux de **a** à **z** est celui du milieu. Le principe de l'algorithme de Dijkstra est de se positionner au sommet de départ puis d'effectuer une exploration *intelligente* du graphe.



Si on considère les arêtes qui sortent du sommet **a**, on constate que le chemin du haut doit être exploré puisqu'il commence avec des arêtes peu coûteuses. Le chemin du milieu doit lui aussi être exploré puisqu'il constitue la solution optimale avec un coût de 13. Par contre, il est inutile d'explorer le chemin du bas plus loin que sa première arête puisque son coût sera forcément supérieur à 13.

**Algorithme 3** Dijkstra

**Entrées:**  $G = (V, E)$  un graphe simple pondéré connexe (orienté ou non orienté),  $\omega : E \rightarrow \mathbb{R}^+$  la fonction de pondération,  $a \in V$  le sommet de départ,  $z \in V$  le sommet d'arrivée.

**Sortie:** coût minimal d'un chemin allant du sommet  $a$  au sommet  $z$

```

1:  $S := \emptyset$                                 ▷ sommets dont le chemin optimal est connu
2:  $L :=$  tableau indexé par les sommets          ▷ coût du plus court chemin connu
3:  $P :=$  tableau indexé par les sommets          ▷ prédécesseur dans le plus court chemin connu
4: pour tout sommet  $v \in V$  faire
5:    $L(v) := \infty$                                ▷ coût du plus court chemin connu de  $a$  et  $v$ 
6:    $P(v) := \text{null}$                                ▷  $v$  n'a pas encore de prédécesseur
7: fin pour
8:  $L(a) := 0$                                        ▷  $a$  est le point de départ, il est à distance 0 de lui-même
9: tant que  $z \notin S$  faire
10:   $u :=$  sommet  $\notin S$  avec  $L(u)$  minimal
11:   $S := S \cup \{u\}$                                ▷ le plus court chemin connu de  $a$  à  $u$  est optimal
12:  pour tout sommets  $v$  tel que  $(u, v) \in E$  et  $v \notin S$  faire   ▷ les voisins de  $u$  qui ne sont pas dans  $S$ 
13:    si  $L(u) + \omega(u, v) < L(v)$  alors ▷ s'il est plus court d'aller à  $u$  puis prendre l'arc (ou l'arête)  $(u, v)$ 
14:       $P(v) := u$                                    ▷ nouveau meilleur chemin: on accède à  $v$  par le sommet  $u$ 
15:       $L(v) := L(u) + \omega(u, v)$                  ▷ coût de ce nouveau meilleur chemin
16:    fin si
17:  fin pour
18: fin tant que
19: retourner  $L(z)$ 

```

Afin de déterminer un chemin de distance minimale allant du sommet  $a$  au sommet  $z$ , le principe de l'algorithme de Dijkstra consiste à séparer les sommets en deux sous-ensembles: ceux pour lesquels on connaît un chemin le plus court et ceux pour lesquels on ne le connaît pas encore. Plus précisément, on définit:

- $S$ : l'ensemble des sommets  $u$  pour lesquels on connaît un chemin de distance minimale allant de  $a$  à  $u$ .

À l'initialisation, l'ensemble  $S$  est vide. À chaque itération de la boucle principale, un sommet est ajouté à l'ensemble  $S$ . Logiquement, si on veut calculer un plus court chemin allant de  $a$  à  $z$ , l'algorithme termine dès que le sommet  $z$  est ajouté à l'ensemble  $S$ .

En plus de cet ensemble, l'algorithme utilise deux tableaux indexés par les sommets:

- $L$ : pour chaque sommet  $u$ ,  $L(u)$  est l'*étiquette* du sommet  $u$ . Il s'agit de la distance du plus court chemin **actuellement connu** allant du sommet  $a$  au sommet  $u$ .
- $P$ : pour chaque sommet  $u$ ,  $P(u)$  est le *prédécesseur* de  $u$ . Il s'agit du sommet à partir duquel on accède à  $u$  dans le plus court chemin **actuellement connu** allant de  $a$  à  $u$ .

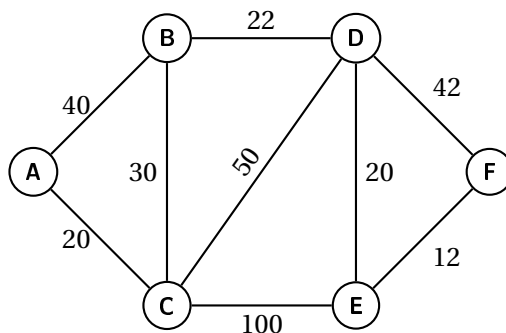
À l'initialisation, le tableau  $L$  associe la valeur  $\infty$  à tous les sommets sauf pour le sommet de départ auquel on attribue la valeur zéro. Cela correspond au fait qu'au moment de l'initialisation, on ne connaît rien du graphe. Pour le tableau  $P$ , on affecte une valeur bidon à tous les sommets. Cette valeur bidon ne doit correspondre à aucun sommet. Dans le pseudo-code, on utilise le mot-clé `null` pour la représenter<sup>1</sup>.

1. Si les sommets sont les entiers de 0 à  $n-1$ , on peut utiliser la valeur  $-1$ . Dans certains langages de programmation, il est possible d'utiliser des mots-clés comme `NULL` (C/C++), `null` (Java), `None` (Python), `Nothing` (VB), etc.

On peut montrer qu'une implémentation de l'algorithme de Dijkstra utilisant un tableau de booléen pour représenter l'ensemble  $S$  nécessite  $O(n^2)$  opérations<sup>2</sup> pour calculer un chemin de poids minimal entre deux sommets d'un graphe à  $n$  sommets.

### Exemple 8.17

Considérez le graphe pondéré ci-dessous, où les coûts sont en \$.



- (a) Trouvez le coût minimal d'un chemin allant du sommet **C** au sommet **E**. Laissez une trace détaillée de l'algorithme, en indiquant le sommet  $u$ , l'ensemble  $S$ , les étiquettes et les prédécesseurs de chacun des sommets à chaque passage dans la boucle **tant que** de la ligne 9.
- (b) Donnez un chemin de coût minimal allant du sommet **C** au sommet **E**.

### Solution :

- (a) L'addition  $L(u) + \omega(u, v)$  est indiquée entre parenthèses. Si le résultat est inférieur à l'étiquette du sommet  $v$ , alors l'étiquette est mise à jour. Pour alléger la trace, on indique ici l'étiquette  $L(v)$  et le prédécesseur  $P(v)$  dans la même colonne en utilisant la convention  $L_P$ .

$u$	A	B	$a = C$	D	$z = E$	F	$S$
	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\emptyset$
<b>C</b>	$(0 + 20 = 20)$ $20_C$	$(0 + 30 = 30)$ $30_C$		$(0 + 50 = 50)$ $50_C$	$(0 + 100 = 100)$ $100_C$	$(0 + \infty = \infty)$ $\infty$	{ <b>C</b> }
<b>A</b>		$(20 + 40 = 60)$ $30_C$		$(20 + \infty = \infty)$ $50_C$	$(20 + \infty = \infty)$ $100_C$	$(20 + \infty = \infty)$ $\infty$	{ <b>A, C</b> }
<b>B</b>				$(30 + 22 = 52)$ $50_C$	$(30 + \infty = \infty)$ $100_C$	$(30 + \infty = \infty)$ $\infty$	{ <b>A, B, C</b> }
<b>D</b>					$(50 + 20 = 70)$ $70_D$	$(50 + 42 = 92)$ $92_D$	{ <b>A, B, C, D</b> }
<b>E</b>							{ <b>A, B, C, D, E</b> }

Le **tant que** prend fin quand  $z \in S$  et retourne l'étiquette du sommet  $z$  (dans notre cas,  $z = E$ ):  $L(z) = 70$ . Ainsi, le coût minimal pour aller de **C** à **E** est 70\$.

- (b) Pour retrouver un chemin de coût minimal, on remonte la chaîne des prédécesseurs à partir du sommet d'arrivée:  $P(E) = D$  et  $P(D) = C$ . Ainsi, **C–D–E** est un chemin de coût minimal allant de **C** à **E**.

2. Cette notation est vue au chapitre sur la complexité des algorithmes.

**Exemple 8.18**

La trace de l'exemple précédent permet-elle de donner aussi le coût minimal pour aller :

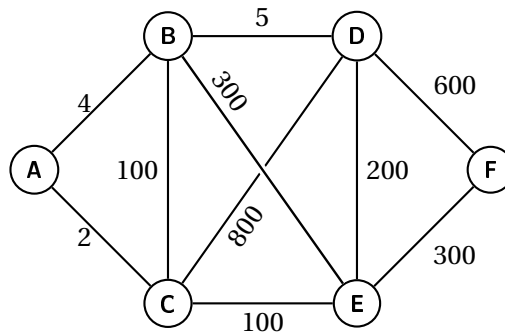
- (a) du sommet **C** au sommet **B**?
- (b) du sommet **C** au sommet **F**?
- (c) du sommet **A** au sommet **E**?

**Solution :**

- (a) Oui. Le sommet **B** a été visité ( $u = B$  à la troisième étape), donc l'étiquette de **B** correspond bien au coût minimal pour aller de **C** à **B** : 30\$.
- (b) Non, car la trace s'est arrêtée sans que le sommet **F** n'ait été visité. L'étiquette de **F** ne correspond donc pas nécessairement au coût minimal pour aller de **C** à **F**. En fait, on remarque que le chemin **C–D–E–F** a un coût de 82\$ tandis que  $L(F) = 92$ .
- (c) Non, car cette trace correspond au sommet initial **C** et non **A**. Toutes les étiquettes correspondent à un coût pour un chemin issu de **C**.

**Exemple 8.19**

Considérez le graphe pondéré ci-dessous, où les poids correspondent à des temps de parcours en millisecondes.



- (a) Trouvez le temps minimal pour aller du sommet **A** au sommet **F**. Laissez une trace de l'algorithme.
- (b) Donnez un chemin de temps minimal allant du sommet **A** au sommet **F**.
- (c) La trace faite en (a) permet-elle aussi de donner le temps minimal pour aller de **A** à **E**?

**Solution :**

(a) Trace de l'algorithme. Le détail des additions  $L(u) + \omega(u, v)$  n'est pas indiqué.

u	a = A	B	C	D	E	z = F	S
	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\emptyset$
A		$4_A$	$2_A$	$\infty$	$\infty$	$\infty$	{A}
C		$4_A$		$802_C$	$102_C$	$\infty$	{A, C}
B				$9_B$	$102_C$	$\infty$	{A, B, C}
D					$102_C$	$609_D$	{A, B, C, D}
E						$402_E$	{A, B, C, D, E}
F							{A, B, C, D, E, F}

Le temps minimal pour aller de A à F est donc 402 millisecondes.

(b) Pour retrouver un chemin minimal, on remonte la chaîne des prédécesseurs à partir du sommet d'arrivée:

$$P(F) = E, \quad P(E) = C, \quad P(C) = A$$

ainsi, A – C – E – F est un chemin de coût minimal allant de A à F.

(c) Oui. Puisque le sommet E a été visité, la trace faite en (a) permet aussi de donner le temps minimal pour aller de A à E : 102 millisecondes.

### 8.4.1 Exemple: plan d'approvisionnement

En décembre, une usine doit planifier son approvisionnement en matière première (pierre) auprès de son fournisseur pour les 4 prochains mois. L'équipe de planification doit tenir compte de plusieurs facteurs. Si elle décide de passer une seule grosse commande pour diminuer le coût d'achat et de livraison, elle devra payer plus cher en frais d'entreposage. Au contraire, si elle décide de passer une petite commande à chaque mois, les coûts d'entreposage seront faibles mais les coûts de livraison seront plus importants. De plus, les prix de la matière première et de la livraison varient dans l'année, tout comme les coûts d'entreposage, mais on suppose ici qu'ils sont connus à l'avance. Le cours *GOL455-Gestion des opérations, des flux et des stocks* aborde en détails la question de l'approvisionnement; nous présentons ici une application de la théorie des graphes en simplifiant le problème.

mois	besoin (en tonnes)
J	20
F	30
M	40
A	30

Figure 8.6 Représentation des besoins en pierre d'une usine pour les prochains mois.

Dans le graphe orienté de la figure 8.7, les sommets correspondent à des mois de l'année: 0 - Décembre, 1 - Janvier, 2 - Février, etc. Le poids d'un arc allant du sommet  $i$  au sommet  $j$  correspond au coût engendré par une commande pour couvrir le besoin des mois  $i + 1$  à  $j$ , cette livraison arrivant au début de la période  $i + 1$ . Le coût tient compte de tous les facteurs: coût unitaire, coût de livraison, coût d'entreposage, etc. Pour clarifier la situation, la quantité de la commande est indiquée entre parenthèses à côté du coût (ce n'est pas nécessaire, car cette information peut-être obtenue à partir du tableau des besoins).

Par exemple, considérons l'arc tireté et bleu allant du sommet 1 au sommet 4:  $c(1, 4) = 5100\$$  est le coût engendré par une commande de 100 tonnes, livrée au début de février, couvrant les besoins de l'usine pour les mois de février à avril.

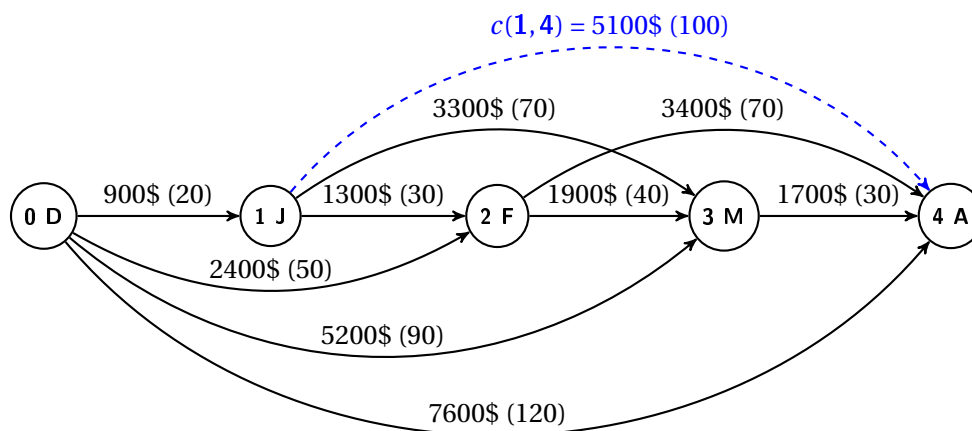


Figure 8.7 Représentation des coûts engendrés par des commandes pour différentes périodes. Exemple adapté des cours GOL455, section 3.2, ÉTS et GSO-6080, ULaval.



Sur le graphe, un chemin allant du sommet **0** au sommet **4** correspond donc à un plan d'approvisionnement. Par exemple, le chemin **0-1-2-3-4** correspond à passer 4 petites commandes, au coût total de  $900\$ + 1300\$ + 1900\$ + 1700\$ = 5800\$$ . Le chemin **0-1-4** correspond à passer une commande de 20 tonnes pour janvier, puis de 100 tonnes pour couvrir les besoins de février à avril, au coût total de

$$900\$ + 5100\$ = 6000\$.$$

**Le chemin de coût minimal allant du sommet 0 au sommet 4 correspond au plan le moins cher.** Pour l'obtenir, on peut utiliser l'algorithme de Dijkstra.

u	a = 0	1	2	3	z = 4
	0	$\infty$	$\infty$	$\infty$	$\infty$
<b>0</b>		(0 + 900 = 900) 900 <sub>0</sub>	(0 + 2400 = 2400) 2400 <sub>0</sub>	(0 + 5200 = 5200) 5200 <sub>0</sub>	(0 + 7600 = 7600) 7600 <sub>0</sub>
<b>1</b>			(900 + 1300 = 2200) 2200 <sub>1</sub>	(900 + 3300 = 4200) 4200 <sub>1</sub>	(900 + 5100 = 6000) 6000 <sub>1</sub>
<b>2</b>				(2200 + 1900 = 4100) 4100 <sub>2</sub>	(2200 + 3400 = 5600) 5600 <sub>2</sub>
<b>3</b>					(4100 + 1700 = 5800) 5800 <sub>2</sub>
<b>4</b>					

Figure 8.8 Trace de l'algorithme de Dijkstra : le coût minimal du chemin allant de **0** à **4** est de 5600\$. Ce chemin est obtenu en remontant la chaîne des prédécesseurs (en indices) : **0-1-2-4**.

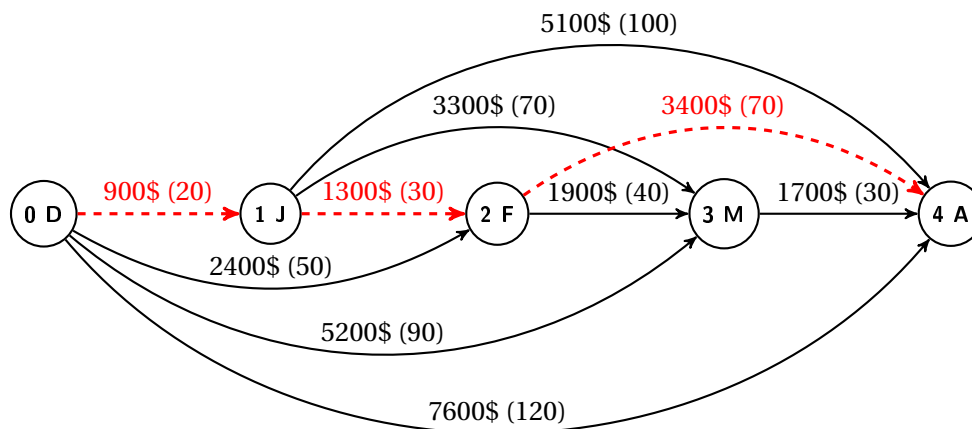
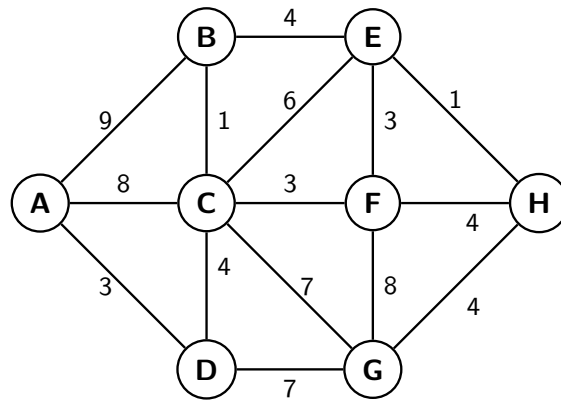


Figure 8.9 Le chemin de coût minimal allant du sommet **0** au sommet **4** est indiqué en tireté : **0-1-2-4**. Il correspond à recevoir 20 tonnes en janvier, 30 en février et 70 en mars (pour couvrir les besoins de mars et avril).

## Exercices

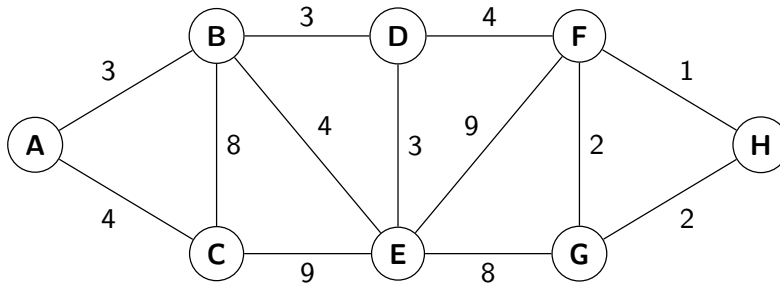
**8.5** Considérez le graphe suivant:



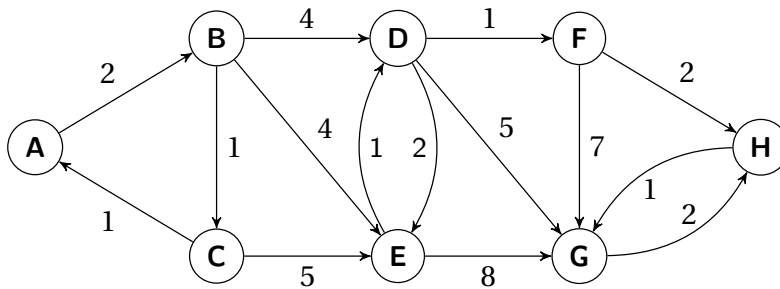
- Trouvez la distance minimale d'un chemin allant de **A** à **H**.
- Donnez explicitement ce chemin.
- Trouvez la distance minimale d'un chemin allant de **A** à **B**.
- Donnez explicitement ce chemin.

**8.6** Pour chacun des graphes suivants, utilisez l'algorithme de Dijkstra afin de calculer la distance minimale d'un chemin de **C** à **G** et donner le chemin obtenu.

(a)

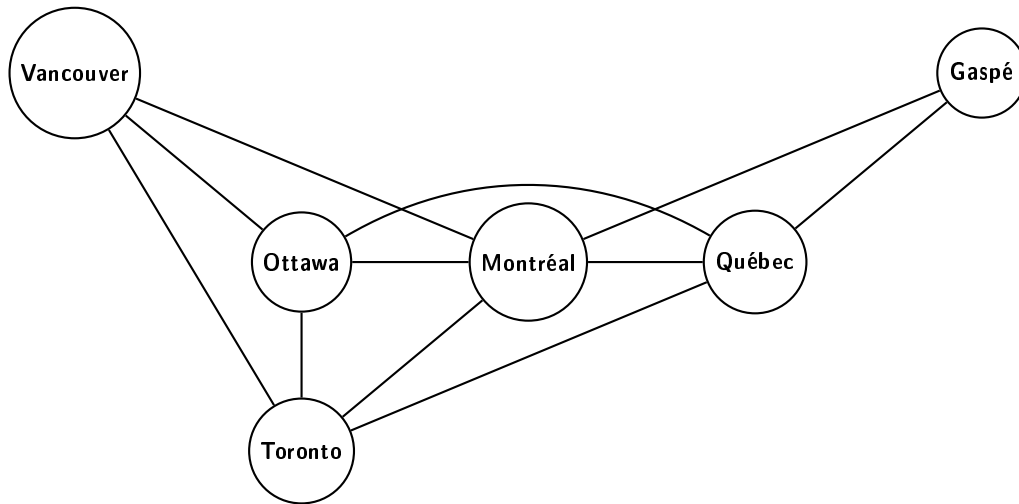


(b)



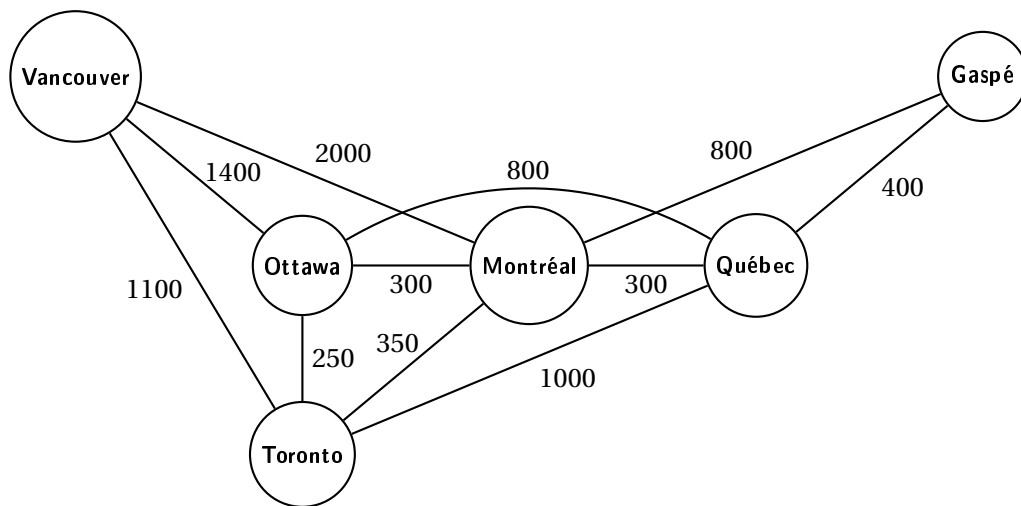
**8.7 Exercice de révision du chapitre.**

Considérez le graphe suivant représentant les liens aériens de la compagnie AéroÉTS entre certaines villes du Canada.



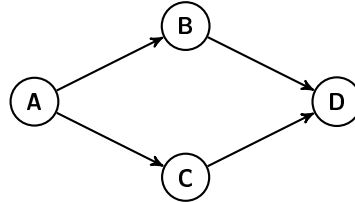
- Combien y a-t-il d'itinéraires possibles d'au plus deux escales de Gaspé à Vancouver?
- Combien y a-t-il d'itinéraires possibles ayant une et une seule escale partant de Montréal?
- Combien y a-t-il d'itinéraires possibles ayant exactement deux escales, avec Toronto comme ville de départ et d'arrivée?
- Combien y a-t-il d'itinéraires possibles ayant exactement sept escales, avec Toronto comme ville de départ et d'arrivée?
- Combien y a-t-il d'itinéraires possibles de quatre escales partant de Gaspé et revenant à Gaspé?
- S'agit-il d'un graphe simple et non orienté?
- Ce graphe vérifie-t-il les conditions du théorème 8.7? Et celles du théorème 8.6?
- Existe-t-il un circuit hamiltonien dans ce graphe? Si oui, donnez-en un.
- Existe-t-il un circuit eulérien dans ce graphe? Si oui, donnez-en un.

- (j) Existe-t-il un chemin eulérien dans ce graphe? Si oui, donnez-en un.
- (k) Comment qualifie-t-on un graphe où chaque arête est associée à un poids?
- (l) Étant donné les prix indiqués (en dollars), utilisez l'algorithme de Dijkstra afin de trouver un trajet de coût minimal entre Gaspé et Vancouver. Puis entre Toronto et Gaspé. Y a-t-il toujours le même nombre de lignes dans le tableau des étiquettes? Autrement dit, l'algorithme de Dijkstra exécute-t-il toujours le même nombre d'étapes pour obtenir le chemin de coût minimal entre deux sommets?

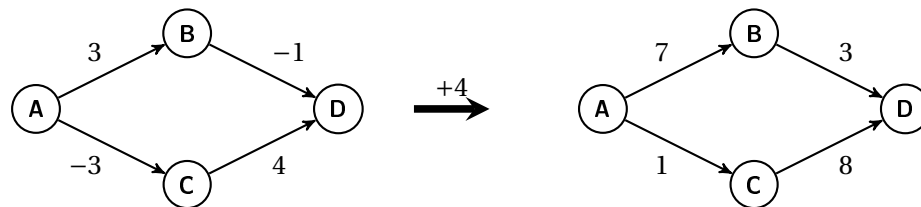


**8.8** En entrée de l'algorithme de Dijkstra, on décrit la fonction de pondération comme étant  $\omega : E \rightarrow \mathbb{R}^+$ . Le but de cet exercice est de comprendre pourquoi on impose que les pondérations ne soient pas négatives.

- (a) Pondérez le graphe ci-dessous avec des valeurs positives et négatives de sorte que l'algorithme de Dijkstra ne produise pas un plus court chemin entre les sommets **A** et **D**.



- (b) Effectuez une trace de l'algorithme de Dijkstra sur le graphe pondéré en (a) afin de montrer que le chemin calculé par l'algorithme n'est pas minimal.
- (c) Afin de calculer un plus court chemin dans un graphe possédant des pondérations négatives, on pourrait être tenté d'ajouter une même valeur à toutes les pondérations de sorte qu'elles soient toutes positives. Par exemple, dans le graphe ci-dessous, la plus petite pondération est  $-3$ . En additionnant 4 sur toutes les arêtes, on obtient un graphe avec des pondérations strictement positives.



Montrez que cette méthode ne fonctionne pas en général en fournissant un contre-exemple.



# Réponses

## Chapitre 1

- Rép. 1.1** (a) Bloc (1) : exécuté lorsque  $p \wedge q \vee r$  est **vrai**.  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{vrai}$ .  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{faux}$ .  
•  $p = \mathbf{faux}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{vrai}$ .  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{vrai}$ .  
•  $p = \mathbf{faux}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{vrai}$ .
- Bloc (2) : exécuté lorsque  $\neg(p \wedge q \vee r)$  est **vrai**.  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{faux}$ .  
•  $p = \mathbf{faux}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{faux}$ .  
•  $p = \mathbf{faux}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{faux}$ .
- (b) Bloc (1) : exécuté lorsque  $p \wedge q \wedge r$  est **vrai**.  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{vrai}$ .
- Bloc (2) : exécuté lorsque  $p \wedge \neg q$  est **vrai**.  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{vrai}$ .  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{faux}$ .
- (c) Bloc (1) : exécuté lorsque  $p \wedge q$  est **vrai**.  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{vrai}$ .  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{faux}$ .
- Bloc (2) : exécuté lorsque  $\neg(p \wedge q) \wedge r$  est **vrai**.  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{vrai}$ .  
•  $p = \mathbf{faux}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{vrai}$ .  
•  $p = \mathbf{faux}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{vrai}$ .
- Bloc (3) : exécuté lorsque  $\neg(p \wedge q) \wedge \neg r$  est **vrai**.  
•  $p = \mathbf{vrai}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{faux}$ .  
•  $p = \mathbf{faux}$ ,  $q = \mathbf{vrai}$  et  $r = \mathbf{faux}$ .  
•  $p = \mathbf{faux}$ ,  $q = \mathbf{faux}$  et  $r = \mathbf{faux}$ .
- Rép. 1.2** (a)  $p \wedge q$  (c)  $p \wedge \neg q$  (e)  $p \rightarrow q$  (g)  $p \leftrightarrow q$   
(b)  $q \rightarrow p$  (d)  $p \vee q$  (f)  $\neg p \wedge \neg q$  (h)  $\neg p \rightarrow \neg q$
- Rép. 1.3** La phrase (b) est la réciproque de (e). La phrase (h) est l'inverse de (e). Aucune des phrases n'est la contraposée de (e). La contraposée de (e) serait, par exemple : « Si l'eau ne gèle pas, alors la température n'est pas au-dessous de 0°C. »
- Rép. 1.4** (a) La condition  $x > 5$  est suffisante pour que le nombre  $x$  soit positif (mais elle n'est pas nécessaire).  
(b) La condition  $x > -5$  est nécessaire pour que  $x$  soit positif (mais elle n'est pas suffisante).  
(c) La condition  $x > -1$  est nécessaire et suffisante pour que l'entier  $x$  soit positif.  
(d) Pour que le nombre  $x$  soit premier, il est nécessaire qu'il soit supérieur ou égal à 2.  
(e) Pour qu'un triangle soit rectangle, il est suffisant que ses côtés mesurent respectivement 3, 4 et 5 centimètres.

- Rép. 1.5** (a) « Il suffit qu'un nombre soit supérieur à 5 pour qu'il soit positif. »  
 (b) « Il est nécessaire qu'un nombre soit supérieur à  $-5$  pour qu'il soit positif. »  
 (c) « Il faut et il suffit qu'un nombre entier soit supérieur à  $-1$  pour qu'il soit positif. »  
 (d) La condition «  $x \geq 2$  » est nécessaire pour que le nombre  $x$  soit premier.  
 (e) La condition « les côtés du triangle mesurent respectivement 3, 4 et 5 centimètres » est suffisante pour que le triangle soit rectangle.

- Rép. 1.6** (a) Vrai. *La condition est nécessaire: si les quatre angles du quadrilatère  $Q$  ne mesurent pas  $90^\circ$ , alors  $Q$  n'est pas un carré.*  
 (b) Faux. *Il est nécessaire que  $Q$  ait deux paires de côtés parallèles pour que  $Q$  soit un rectangle, mais cette condition n'est pas suffisante. Autrement dit, les parallélogrammes ne sont pas tous des rectangles.*

- Rép. 1.7** (a)  $q \rightarrow p$             (c)  $p \rightarrow q$             (e)  $q \wedge \neg p$             (g)  $\neg q \rightarrow \neg p$             (i)  $p \vee \neg q$   
 (b)  $p \wedge \neg q$             (d)  $\neg p \rightarrow \neg q$             (f)  $q \rightarrow p$             (h)  $\neg p \wedge \neg q$             (j)  $q \leftrightarrow p$

- Rép. 1.8** La phrase (g) est la contraposée de (c). Les phrases (a) et (f) sont des réciproques de (c). La phrase (d) est l'inverse de (c).

- Rép. 1.9** (a)  $p$  est nécessaire et suffisant pour  $q$             (e)  $p$  est nécessaire pour  $q$   
 (b)  $p$  est nécessaire pour  $q$             (f)  $p$  est suffisant pour  $q$   
 (c)  $p$  est nécessaire et suffisant pour  $q$   
 (d)  $p$  est suffisant pour  $q$             (g)  $p$  est suffisant pour  $q$

**Rép. 1.10** (a)

$p$	$q$	$p \rightarrow q$	$\neg q$	$(p \rightarrow q) \wedge \neg q$
V	V	V	F	F
V	F	F	V	F
F	V	V	F	F
F	F	V	V	V

(b)

$p$	$q$	$p \rightarrow q$	$\neg p \vee q$	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$
V	V	V	V	V
V	F	F	F	V
F	V	V	V	V
F	F	V	V	V

(c)

$p$	$q$	$r$	$p \rightarrow q$	$(p \rightarrow q) \rightarrow r$
V	V	V	V	V
V	V	F	V	F
V	F	V	F	V
V	F	F	F	V
F	V	V	V	V
F	V	F	V	F
F	F	V	V	V
F	F	F	V	F

(d)

$p$	$q$	$r$	$q \rightarrow r$	$p \rightarrow (q \rightarrow r)$
V	V	V	V	V
V	V	F	F	F
V	F	V	V	V
V	F	F	V	V
F	V	V	V	V
F	V	F	F	V
F	F	V	V	V
F	F	F	V	V



**Rép. 1.11** (a) La proposition est une contradiction, car elle est toujours fausse, comme le montre sa table de vérité.

$p$	$p \vee p$	$\neg(p \wedge p)$	$(p \vee p) \leftrightarrow \neg(p \wedge p)$
V	V	F	F
F	F	V	F

(b) La proposition est une tautologie, car elle est toujours vraie, comme le montre sa table de vérité.

$p$	$q$	$p \oplus q$	$\neg(p \wedge q)$	$(p \oplus q) \rightarrow \neg(p \wedge q)$
V	V	F	F	V
V	F	V	V	V
F	V	V	V	V
F	F	F	V	V

(c) La proposition est une contingence, car elle est parfois vraie et parfois fausse, comme le montrent les deux lignes suivantes de sa table de vérité.

$p$	$q$	$r$	$q \wedge r$	$p \vee q$	$p \vee (q \wedge r)$	$(p \vee q) \wedge r$
V	V	V	V	V	V	V
V	V	F	F	V	V	F

(d) La proposition est une tautologie, car elle est toujours vraie, comme le montre sa table de vérité.

$p$	$q$	$p \rightarrow q$	$\neg(p \rightarrow q)$	$\neg(p \rightarrow q) \rightarrow p$
V	V	V	F	V
V	F	F	V	V
F	V	V	F	V
F	F	V	F	V

**Rép. 1.12** (a)  $p \quad q \quad p \rightarrow q \quad \neg q \rightarrow \neg p \quad (p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$

V	V	V	V	V
V	F	F	F	V
F	V	V	V	V
F	F	V	V	V

(b)  $p \quad q \quad \neg(p \vee q) \quad \neg p \wedge \neg q \quad (\neg(p \vee q)) \leftrightarrow (\neg p \wedge \neg q)$

V	V	F	F	V
V	F	F	F	V
F	V	F	F	V
F	F	V	V	V

**Rép. 1.13** (a) « Ce programme n'est pas rapide ou il n'est pas efficace. » Ou encore, « ce programme n'est pas rapide ou il est inefficace. »

(b) « Le programme ne doit pas être C++ et le programme ne doit pas être en Java. » Ce qui serait mieux de reformuler ainsi: « le programme ne doit être ni en C++, ni en Java. »

**Rép. 1.14** Il suffit d'appliquer la loi de De Morgan  $\neg(p \wedge q) \equiv \neg p \vee \neg q$ .

**Rép. 1.15** (a)  $p \quad q \quad \neg(p \vee (\neg p \wedge q)) \quad \neg p \wedge \neg q$

V	V	F	F
V	F	F	F
F	V	F	F
F	F	V	V

(b) On ne sait pas quel sera le plus court chemin vers la vérité! Voici deux preuves.

$$\begin{aligned}
 \neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{De Morgan} \\
 &\equiv \neg p \wedge (\neg(\neg p) \vee \neg q) && \text{De Morgan} \\
 &\equiv \neg p \wedge (p \vee \neg q) && \text{Double négation} \\
 &\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{Distributivité} \\
 &\equiv \mathbf{F} \vee (\neg p \wedge \neg q) && \text{Négation} \\
 &\equiv \neg p \wedge \neg q && \text{Identité}
 \end{aligned}$$

Ou encore, en commençant par distribuer la disjonction (ou) :

$$\begin{aligned} \neg(p \vee (\neg p \wedge q)) &\equiv \neg((p \vee \neg p) \wedge (p \vee q)) && \text{Distributivité} \\ &\equiv \neg(\mathbf{V} \wedge (p \vee q)) && \text{Négation} \\ &\equiv \neg(p \vee q) && \text{Identité} \\ &\equiv \neg p \wedge \neg q && \text{De Morgan} \end{aligned}$$

(c)  $\text{not } (p \text{ or not } p \text{ and } q)$        $\text{not } p \text{ and not } q$

- Rép. 1.16** (a) Les propositions ne sont pas équivalentes puisque quand  $p$ ,  $q$ , et  $r$  sont toutes les trois fausses,  $(p \rightarrow q) \rightarrow r$  est fausse, mais  $p \rightarrow (q \rightarrow r)$  est vraie.
- (b) Les propositions sont équivalentes. Justification: pour que  $(p \rightarrow r) \vee (q \rightarrow r)$  soit fausse, les deux implications doivent être fausses, ce qui arrive exactement quand  $r$  est fausse et  $p$  et  $q$  sont vraies. Il en va de même pour  $(p \wedge q) \rightarrow r$ .

**Rép. 1.17** (a) Voici une preuve possible :

$$\begin{aligned} (\neg q \rightarrow p) \rightarrow (r \vee p) &\equiv (\neg\neg q \vee p) \rightarrow (r \vee p) && \text{Table 2 équivalence 1} \\ &\equiv (q \vee p) \rightarrow (r \vee p) && \text{Double négation} \\ &\equiv \neg(q \vee p) \vee (r \vee p) && \text{Table 2 équivalence 1} \\ &\equiv (\neg q \wedge \neg p) \vee (r \vee p) && \text{De Morgan} \\ &\equiv (p \vee (\neg q \wedge \neg p)) \vee r && \text{Associativité et Commutativité} \\ &\equiv ((p \vee \neg q) \wedge (p \vee \neg p)) \vee r && \text{Distributivité} \\ &\equiv ((p \vee \neg q) \wedge \mathbf{V}) \vee r && \text{Négation} \\ &\equiv (p \vee \neg q) \vee r && \text{Identité} \\ &\equiv \neg(\neg p \wedge q) \vee r && \text{De Morgan} \\ &\equiv (\neg p \wedge q) \rightarrow r && \text{Table 2 équivalence 1} \end{aligned}$$

(b) Voici une preuve possible :

$$\begin{aligned} ((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r) &\equiv \neg((p \vee q) \wedge (\neg p \vee r)) \vee (q \vee r) && \text{Table 2 équiv. 1} \\ &\equiv \neg(p \vee q) \vee \neg(\neg p \vee r) \vee (q \vee r) && \text{De Morgan} \\ &\equiv (\neg p \wedge \neg q) \vee (p \wedge \neg r) \vee (q \vee r) && \text{De Morgan} \\ &\equiv (q \vee (\neg p \wedge \neg q)) \vee (r \vee (p \wedge \neg r)) && \text{Associativité et Comm.} \\ &\equiv ((q \vee \neg p) \wedge (q \vee \neg q)) \vee ((r \vee p) \wedge (r \vee \neg r)) && \text{Distributivité} \\ &\equiv ((q \vee \neg p) \wedge \mathbf{V}) \vee ((r \vee p) \wedge \mathbf{V}) && \text{Négation} \\ &\equiv (q \vee \neg p) \vee (r \vee p) && \text{Identité} \\ &\equiv (q \vee r) \vee (\neg p \vee p) && \text{Associativité et Comm.} \\ &\equiv (q \vee r) \vee \mathbf{V} && \text{Négation} \\ &\equiv \mathbf{V} && \text{Domination} \end{aligned}$$

**Rép. 1.18** (a) Il y a plusieurs réponses possibles (en raison des équivalences logiques); nous avons choisi celles qui se rapprochent le plus des phrases décrivant les contraintes.

1. Au moins un des postulants numéro 2, 4 ou 5 doit être embauché, car eux seuls ont des connaissances en électricité.

$$p_2 \vee p_4 \vee p_5$$

2. Si le postulant 2 est embauché ou le postulant 4 est embauché, alors le postulant 5 ne doit pas l'être.

$$(p_2 \vee p_4) \rightarrow \neg p_5$$

3. Si les postulants 2 et 4 sont embauchés tous les deux, alors le postulant 5 ne doit pas l'être.

$$(p_2 \wedge p_4) \rightarrow \neg p_5$$

4. Les postulants 3 et 5 forment un couple : ils accepteront l'affectation seulement si les deux sont embauchés.

$$p_3 \leftrightarrow p_5$$

5. Il est impossible d'embaucher à la fois les postulants 2 et 5, et il est impossible d'embaucher à la fois les postulants 4 et 5.

$$\neg(p_2 \wedge p_5) \wedge \neg(p_4 \wedge p_5)$$

6. Il est impossible d'embaucher le trio de postulants 2, 4 et 5 : on peut en choisir un, deux ou aucun, mais pas les trois.

$$\neg(p_2 \wedge p_4 \wedge p_5)$$

7. Il faut absolument embaucher au moins 2 personnes parmi les postulants 1 à 4.

$$p_1 \wedge p_2 \vee p_1 \wedge p_3 \vee p_1 \wedge p_4 \vee p_2 \wedge p_3 \vee p_2 \wedge p_4 \vee p_3 \wedge p_4$$

qui est plus facile à lire si on ajoute des parenthèses :

$$(p_1 \wedge p_2) \vee (p_1 \wedge p_3) \vee (p_1 \wedge p_4) \vee (p_2 \wedge p_3) \vee (p_2 \wedge p_4) \vee (p_3 \wedge p_4)$$

8. Il faut embaucher les postulants 1 et 3, ou embaucher le postulant 2 et au moins un des postulants parmi 1, 3 et 4, ou encore embaucher le postulant 4 et au moins un des postulants parmi 1, 2 et 3.

$$(p_1 \wedge p_3) \vee (p_2 \wedge (p_1 \vee p_3 \vee p_4)) \vee (p_4 \wedge (p_1 \vee p_2 \vee p_3))$$

- (b) Les contraintes C2 et C5 sont équivalentes, en voici la preuve.

$$\begin{aligned} & (p_2 \vee p_4) \rightarrow \neg p_5 \\ \equiv & \neg(p_2 \vee p_4) \vee \neg p_5 && \text{Règle 2.1} \\ \equiv & (\neg p_2 \wedge \neg p_4) \vee \neg p_5 && \text{De Morgan} \\ \equiv & (\neg p_2 \vee \neg p_5) \wedge (\neg p_4 \vee \neg p_5) && \text{Distributivité} \\ \equiv & \neg(p_2 \wedge p_5) \wedge \neg(p_4 \wedge p_5) && \text{De Morgan} \end{aligned}$$

Les contraintes C3 et C6 sont équivalentes, en voici la preuve.

$$\begin{aligned} & (p_2 \wedge p_4) \rightarrow \neg p_5 \\ \equiv & \neg(p_2 \wedge p_4) \vee \neg p_5 && \text{Règle 2.1} \\ \equiv & (\neg p_2 \vee \neg p_4) \vee \neg p_5 && \text{De Morgan} \\ \equiv & \neg p_2 \vee \neg p_4 \vee \neg p_5 && \text{Associativité} \\ \equiv & \neg(p_2 \wedge p_4 \wedge p_5) && \text{De Morgan} \end{aligned}$$

Les contraintes C7 et C8 sont équivalentes, en voici la preuve.

$$\begin{aligned} & (p_1 \wedge p_3) \vee (p_2 \wedge (p_1 \vee p_3 \vee p_4)) \vee (p_4 \wedge (p_1 \vee p_2 \vee p_3)) \\ \equiv & (p_1 \wedge p_3) \vee (p_2 \wedge p_1) \vee (p_2 \wedge p_3) \vee (p_2 \wedge p_4) \vee (p_4 \wedge p_1) \vee (p_4 \wedge p_2) \vee (p_4 \wedge p_3) && \text{Distributivité} \\ \equiv & (p_1 \wedge p_3) \vee (p_1 \wedge p_2) \vee (p_2 \wedge p_3) \vee (p_2 \wedge p_4) \vee (p_1 \wedge p_4) \vee (p_2 \wedge p_4) \vee (p_3 \wedge p_4) && \text{Commutativité} \\ \equiv & (p_1 \wedge p_3) \vee (p_1 \wedge p_2) \vee (p_2 \wedge p_3) \vee (p_2 \wedge p_4) \vee (p_1 \wedge p_4) \vee (p_3 \wedge p_4) && \text{Idempotence} \\ \equiv & (p_1 \wedge p_2) \vee (p_1 \wedge p_3) \vee (p_1 \wedge p_4) \vee (p_2 \wedge p_3) \vee (p_2 \wedge p_4) \vee (p_3 \wedge p_4) && \text{Commutativité} \end{aligned}$$

- Rép. 1.19** (a) Vrai. Un exemple suffit pour démontrer une proposition existentielle :  $P(1)$  est vrai.  
 (b) Faux. Un contre-exemple suffit pour infirmer une proposition universelle. Autrement dit, pour conclure qu'une proposition universelle est fautive, un seul exemple qui la contredit suffit. Ici,  $P(2)$  est faux.  
 (c) Vrai. Un exemple suffit pour démontrer une proposition existentielle :  $P(2)$  est faux donc  $\neg P(2)$  est vrai.  
 (d) Faux. Un contre-exemple suffit, or  $\neg P(1)$  est faux.

- Rép. 1.20** (a)  $P(0) \vee P(1) \vee P(2)$   
 (b)  $P(0) \wedge P(1) \wedge P(2)$   
 (c)  $\neg(P(0) \vee P(1) \vee P(2)) \equiv \neg P(0) \wedge \neg P(1) \wedge \neg P(2)$   
 (d)  $\neg(P(0) \wedge P(1) \wedge P(2)) \equiv \neg P(0) \vee \neg P(1) \vee \neg P(2)$   
 (e)  $\neg P(0) \vee \neg P(1) \vee \neg P(2)$   
 (f)  $\neg P(0) \wedge \neg P(1) \wedge \neg P(2)$

- Rép. 1.21** (a) 1 (d) 7 (g) 3  
 (b) 8 (e) 9 (h) 9  
 (c) 4 (f) 8 (i) 3

- Rép. 1.22** (a) Vrai: par exemple, Baba est ingénieur et sait programmer.  
 (b) Faux: par exemple, Denis n'est pas ingénieur et ne sait pas programmer.  
 (c) Faux.  $\forall x \neg(I(x) \vee P(x)) \equiv \forall x (\neg I(x) \wedge \neg P(x))$  Or cet énoncé est faux, car par exemple, Alain sait programmer.  
 (d) Vrai. Par exemple, pour  $x = France$ .  
 (e) Vrai:  $(P(Alain) \wedge \neg I(Alain))$   
 (f) Faux car  $P(Alain)$ .  
 (g) Vrai, les 3 ingénieurs savent tous programmer.  
 (h) Faux: Alain sait programmer, mais il n'est pas ingénieur.  
 (i) Faux: Alain sait programmer, mais il n'est pas ingénieur.

**Rép. 1.23** Rappel: il peut y avoir plusieurs formulations équivalentes. Si la vôtre est différente de la réponse ci-dessous, n'hésitez pas à consulter votre prof!

- (a)  $\exists x (M(x) \wedge J(x))$   
 (b)  $\exists x (M(x) \wedge \neg J(x))$   
 (c)  $\forall x J(x)$   
 (d)  $\neg \exists x (M(x) \vee J(x)) \equiv \forall x \neg(M(x) \vee J(x)) \equiv \forall x (\neg M(x) \wedge \neg J(x))$   
 (e)  $\forall x (M(x) \rightarrow J(x))$

- Rép. 1.24** (a) Vrai. Un exemple suffit pour démontrer une proposition existentielle: prendre  $n = 0$ .  
 (b) Faux. Un contre-exemple suffit pour infirmer proposition universelle: prendre  $n = -1$ .  
 (c) Faux. Un exemple ne suffit pas, un argument général serait requis pour démontrer que cette proposition existentielle est fausse.  
 (d) Faux. Un contre-exemple suffit pour infirmer proposition universelle: prendre  $n = 0$ .  
 (e) Vrai. Un exemple ne suffit pas, un argument général serait requis.

- Rép. 1.25** (a)  $\exists x A(x) \wedge \neg V(x)$  (e)  $\forall x (A(x) \rightarrow V(x))$   
 (b)  $\forall x (V(x) \rightarrow A(x))$  (f)  $\exists x (A(x) \wedge V(x))$   
 (c)  $\forall x \neg V(x)$  (g)  $\forall x (A(x) \wedge V(x))$  ou encore  $(\forall x A(x)) \wedge (\forall x V(x))$   
 (d)  $\neg \forall x V(x)$  (h)  $(\neg \forall x A(x)) \vee (\exists x \neg V(x))$

- Rép. 1.26** (a) L'énoncé est vrai, car  $2^5 \bmod 10 = 32 \bmod 10 = 2$  et  $4^3 \bmod 10 = 64 \bmod 10 = 4$ .  
 (b) L'énoncé universel est faux. Prenons  $x = 2$  comme contre-exemple:  $x^3 \bmod 10 = 8 \bmod 10 = 8 \neq x$ . Ainsi,  $T(2)$  est faux.  
 (c) L'énoncé existentiel est vrai, car  $T(0)$  est vrai.  
 (d) L'énoncé existentiel est vrai, car  $\neg T(2)$  est vrai.  
 (e) L'énoncé universel est vrai: on peut tester toutes les valeurs possibles pour  $x$  et conclure que  $P(0) \wedge P(1) \wedge P(2) \wedge P(3) \wedge P(4) \wedge P(5) \wedge P(6) \wedge P(7) \wedge P(8) \wedge P(9)$  est vrai. (De façon plus générale, l'énoncé  $x^5 \bmod 10 = x$  est vrai quelque soit la valeur du nombre entier  $x$ , mais ceci ne peut être démontré en testant toutes les possibilités: il faudrait fournir une preuve générale.)  
 (f) Faux, car  $\exists x \neg P(x) \equiv \neg \forall x P(x)$  et  $\forall x P(x)$  est vrai (voir (e)).

**Rép. 1.27** 2 et 3.

**Rép. 1.28** Il y a plusieurs formulations possibles. Tentez d'être le plus clair possible, dans un bon français: reformulez vos traductions initiales pour leur donner une tournure plus naturelle, moins robotisée.

- (a)  $\forall B \in R, \exists x \in X, M(x, B)$  Chaque répertoire peut être modifié par au moins un employé (par nécessairement le même employé). **Vrai.**
- (b)  $\exists x \in X, \forall B \in R, M(x, B)$  Il existe au moins un employé qui peut modifier tous les répertoires (le même employé). **Faux.**
- (c)  $\exists B \in R, \forall x \in X, M(x, B)$  Il existe au moins un répertoire que tous les employés peuvent modifier. **Faux.**
- (d)  $\exists B \in R, \forall x \in X, L(x, B)$  Il existe au moins un répertoire que tous les employés peuvent lire. **Vrai** ( $P$  et  $Z$ ).
- (e)  $\exists B \in R, \forall x \in X, \neg L(x, B)$  Il existe au moins un répertoire qu'aucun employé ne peut lire. **Faux.**
- (f)  $\forall x \in X, \exists B \in R, \neg L(x, B)$  Pour chaque employé, il y a au moins un répertoire qu'il ne peut pas lire. *Dans un langage plus naturel:* aucun employé ne peut lire tous les répertoires. **Vrai** car chaque colonne contient au moins une case sans la lettre  $L$ .
- (g)  $\forall B \in R, (L(\text{Manon}, B) \rightarrow M(\text{Guy}, B))$  Pour chaque répertoire  $B$ , si ce répertoire  $B$  peut être lu par Manon, alors ce répertoire  $B$  peut être modifié par Guy. *Dans un langage plus naturel:* chaque répertoire qui peut être lu par Manon peut être modifié par Guy. **Vrai.**
- (h)  $\forall B \in R, \forall x \in X, (M(x, B) \rightarrow L(x, B))$  Si un employé peut modifier un répertoire, alors il peut le lire. **Vrai.**
- (i)  $\exists B \in R, \exists! x \in X, M(x, B)$  Il existe au moins un répertoire pour lequel un et un seul employé a le droit de modification. **Vrai** (répertoires  $K, J, Z$ ).
- (j)  $\exists! B \in R, \exists x \in X, M(x, B)$  Il existe un et un seul répertoire qui peut être modifié par au moins un employé. **Faux.**
- (k)  $\exists! B \in R, \exists! x \in X, L(x, B)$  Il existe un et un seul répertoire qui peut être lu par un et un seul employé. **Vrai**, le répertoire  $J$ .

- Rép. 1.29**
- |   |  |
|---|--|
| (a) $\forall y C(\text{Ève}, y)$                          | (f) $\forall x C(x, \text{Karim})$     |
| (b) $\exists y \neg C(\text{Julie}, y)$                   | (g) $\forall x \exists y C(x, y)$      |
| (c) $\exists y \forall x \neg C(x, y)$                    | (h) $\exists y \forall x C(x, y)$      |
| (d) $\forall x C(x, x)$                                   | (i) $\forall x \exists y \neg C(x, y)$ |
| (e) $\exists x \forall y (C(x, y) \leftrightarrow x = y)$ |  |

**Rép. 1.30** Il y a plusieurs formulations possibles. Tentez d'être le plus clair possible, dans un bon français: reformulez vos traductions initiales pour leur donner une tournure plus naturelle, moins robotisée.

- (a) Il y a au moins une usine qui fabrique le produit  $c$ . **Vrai**, Amos par exemple.
- (b) L'usine de Montréal et celle de Chicoutimi n'ont aucun produit en commun dans leur fabrication. Aucun produit n'est fabriqué à la fois par l'usine de Montréal et par celle de Chicoutimi. **Faux.** Par exemple, on voit que le produit  $a$  est fabriqué dans les deux usines.
- (c) Si l'usine de Montréal fabrique un produit, alors ce produit n'est pas fabriqué à l'usine de Gatineau. **Vrai.** L'usine de Montréal fabrique les produits  $a, b$  et  $f$  et ceux-ci ne sont pas fabriqués à Gatineau.
- (d) À elles deux, les usines de Montréal et Québec ne couvrent pas l'ensemble de la production: il y a au moins un produit qui n'est fabriqué ni à Montréal, ni à Québec. **Vrai**, car le produit  $c$  n'est fabriqué ni à Montréal, ni à Québec.
- (e) Il y a au moins un produit qui est fabriqué dans chaque usine (il s'agit du même produit). **Faux.**
- (f) Chacune des usines fabrique au moins un produit (pas nécessairement le même produit dans chacune des usines). **Vrai.**
- (g) Chacune des usines fabrique chacun des produits. **Faux**, par exemple, Amos ne fabrique pas le produit  $a$ .
- (h) Si une usine fabrique le produit  $b$ , alors elle ne fabrique ni le produit  $d$ , ni le produit  $e$ . **Vrai.**

- (i) Si une usine fabrique le produit  $a$ , alors elle doit aussi fabriquer le produit  $f$  ou le produit  $g$  (ou les deux, le « ou » étant inclusif). **Faux.** Québec fabrique  $a$ , mais ne fabrique ni  $f$ , ni  $g$ .
- (j) Chaque produit est fabriqué dans au moins une usine. **Vrai.**
- (k) Aucune usine ne fabrique à la fois le produit  $c$  et le produit  $d$ . **Faux.** Amos fabrique les produits  $c$  et  $d$  entre autres.
- (l) Les produits  $a$  et  $b$  doivent être fabriqués ensemble dans la même usine : une usine fabrique  $a$  si et seulement si elle fabrique  $b$ . **Faux,** car Québec fabrique  $a$  sans  $b$ .
- (m) Il y a au moins 2 usines qui fabriquent le produit  $a$ . **Vrai,** Chicoutimi, Montréal et Québec.
- (n) Il y a au moins un produit qui est fabriqué dans une seule usine (aucune autre usine ne le fabrique). **Vrai,** le produit  $g$  est fabriqué uniquement à Gatineau.

**Rép. 1.31** (a)  $\forall i \forall j \exists k C(i, j, k)$

(b)  $\forall i \forall j \forall k \forall l (C(i, j, k) \wedge C(i, j, l) \rightarrow k = l)$

(c)  $\forall j \forall k \exists i C(i, j, k)$

(d)  $\forall i \forall k \exists j C(i, j, k)$

(e)  $\forall i \forall j \forall k \forall l (c(i, j, k) \wedge c(i, l, k) \rightarrow j = l)$

(f)  $\forall i \forall j \forall k \forall l (c(i, j, k) \wedge c(l, j, k) \rightarrow i = l)$

(g)  $\forall k \forall r \in \{0, 1, 2\} \forall s \in \{0, 1, 2\} \exists i \in \{1, 2, 3\} \exists j \in \{1, 2, 3\}, C(3r + i, 3s + j, k)$

(h)  $\forall k \forall s, r \in \{0, 1, 2\} \forall a, b, a', b' \in \{1, 2, 3\}, (c(3r + a, 3s + b, k) \wedge c(3r + a', 3s + b', k)) \rightarrow (a = a' \wedge b = b')$

**Rép. 1.32** (a) Vrai. Il suffit d'un exemple pour prouver un énoncé existentiel. Puisque l'inégalité  $3 < 4$  est vraie, la proposition  $P(x, 4)$  est vraie pour  $x = 3$ . Ainsi, l'énoncé existentiel  $\exists x P(x, 4)$  est vrai.

(b) Faux. Justification: aucun nombre naturel n'est inférieur à 0. Par contre, l'énoncé serait vrai si l'univers du discours était l'ensemble des nombres entiers.

(c) Vrai. Soit  $x$  quelconque. Posons  $y = x + 1$ . Alors  $P(x, y)$  désigne  $x < x + 1$  qui est toujours vrai. Donc l'énoncé est vrai.

(d) Vrai.  $\neg \forall x \forall y P(x, y) \equiv \exists x \exists y \neg P(x, y)$ . Un exemple suffit. Posons  $x = 1$  et  $y = 0$ , alors  $P(x, y)$  est faux,  $\neg P(x, y)$  est vrai et l'énoncé est vrai.

(e) Vrai.  $\neg \exists x \forall y P(x, y) \equiv \forall x \exists y \neg P(x, y)$ . Soit  $x$  quelconque. Posons  $y = x$ . Alors  $\neg P(x, y)$  est vrai. Donc l'énoncé est vrai.

(f) Vrai. Prenons  $x = 0$ . Quelque soit le nombre naturel non nul  $y$ , on a  $x < y$ . Par ailleurs, l'énoncé serait faux si l'univers du discours était l'ensemble des nombres entiers ou l'ensemble des nombres réels : dans ces ensembles, il n'y a aucun nombre qui est inférieur à tous les autres ; il n'y a pas de nombre minimal.

(g) Faux. On considère un contre-exemple, prenons  $x = 5$  et  $y = 5$ . Alors  $P(x, y)$  est faux,  $P(y, x)$  est également faux et donc,  $P(x, y) \vee P(y, x)$  est faux.

**Rép. 1.33** (a)  $x = 1$  et  $y = -1$

(b)  $x = 1$  et  $y = -1$

(c) Pour  $x = 0$  et  $y$  quelconque, la proposition  $xy = 1$  est fautive.

**Rép. 1.34** On pose :

$p$  : « il pleut »,

$m$  : « les trottoirs sont mouillés ».

(a)  $p \rightarrow m$

$$\frac{\neg p}{\neg m}$$

Raisonnement invalide, car si  $p = \mathbf{F}$ ,  $m = \mathbf{V}$  alors,

$$\begin{aligned} ((p \rightarrow m) \wedge \neg p) \rightarrow \neg m &\equiv ((\mathbf{F} \rightarrow \mathbf{V}) \wedge \neg \mathbf{F}) \rightarrow \neg \mathbf{V} \\ &\equiv (\mathbf{V} \wedge \mathbf{V}) \rightarrow \mathbf{F} \\ &\equiv \mathbf{F}. \end{aligned}$$

$$(b) \quad \frac{p \rightarrow m}{p} \\ \frac{\quad}{m}$$

Raisonnement valide par *modus ponens*.

- (c) On pose:  
 $e$ : « tu fais tous les exercices du livre de référence »,  
 $r$ : « tu réussis le cours ».

$$\frac{e \rightarrow r}{r} \\ \frac{\quad}{e}$$

Raisonnement invalide, car si  $e = \mathbf{F}$ ,  $r = \mathbf{V}$  alors,

$$\begin{aligned} ((e \rightarrow r) \wedge r) \rightarrow e &\equiv ((\mathbf{F} \rightarrow \mathbf{V}) \wedge \mathbf{V}) \rightarrow \mathbf{F} \\ &\equiv (\mathbf{V} \wedge \mathbf{V}) \rightarrow \mathbf{F} \\ &\equiv \mathbf{F}. \end{aligned}$$

- (d) On pose:  
 $s$ : « la science peut expliquer ce phénomène »,  
 $m$ : « c'est un miracle ».

$$\frac{s \vee m}{\neg s} \\ \frac{\quad}{m}$$

Raisonnement valide par *sylogisme disjonctif*.

**Remarque:** On aurait également pu interpréter la première affirmation comme étant un « *ou exclusif* ». Dans ce cas, on ne peut pas utiliser directement les règles d'inférence :

$$\frac{s \oplus m}{\neg s} \\ \frac{\quad}{(s \vee m) \wedge \neg(s \wedge m)} \quad \text{logiquement équivalente à la l'hypothèse } s \oplus m. \\ \frac{s \vee m}{m} \quad \text{par simplification de la ligne précédente.} \\ \text{par syllogisme disjonctif.}$$

- (e) On pose:  
 $v$ : « ce sandwich contient de la viande »,  
 $c$ : « ce sandwich contient des champignons »,  
 $m$ : « ce sandwich contient de la mayonnaise ».

$$\frac{\neg v \vee c}{v \vee m} \\ \frac{\quad}{c \vee m}$$

Raisonnement valide par *résolution*.

- Rép. 1.35** (a)  $\exists x \in U \forall y \in U (P(x) \wedge (P(y) \rightarrow x = y))$ .  
 (b)  $\exists x \in U \exists y \in U (P(x) \wedge P(y) \wedge x \neq y)$ .

**Rép. 1.36** On montre que les cinq hypothèses de l'énoncé mènent à la conclusion : *Le téléphone est sur le comptoir de la cuisine*. Afin de simplifier la rédaction, on pose :

- $t$  : « le téléphone est sur la table de chevet »,  
 $v$  : « j'ai vu mon téléphone en me levant »,  
 $r$  : « j'ai été réveillée par la sonnerie de mon téléphone »,  
 $p$  : « j'ai parlé au téléphone dans la cuisine »,  
 $c$  : « le téléphone est sur le comptoir de la cuisine ».

On considère le raisonnement suivant :

1.  $t \rightarrow v$
2.  $r \rightarrow t$
3.  $p \rightarrow c$
4.  $r \vee p$
5.  $\neg v$
6.  $r \rightarrow v$  par *sylogisme hypothétique* des lignes 2 et 1.
7.  $\neg r$  par *modus tollens* des lignes 5 et 6.
8.  $p$  par *sylogisme disjonctif* des lignes 4 et 7.
9.  $c$  par *modus ponens* des lignes 8 et 3.

- Rép. 1.37**
1.  $a \vee \neg b \rightarrow \neg c \wedge d$
  2.  $e \rightarrow c \vee \neg d$
  3.  $\neg f$
  4.  $e \vee f$
  5.  $e$  par *sylogisme disjonctif* des lignes 4 et 3.
  6.  $\neg(\neg c \wedge d) \rightarrow \neg(a \vee \neg b)$  équivalent à la ligne 1 par Table 2, ligne 2 (contraposée).
  7.  $(c \vee \neg d) \rightarrow (\neg a \wedge b)$  équivalent à la ligne 6 par *De Morgan* et *double négation*.
  8.  $e \rightarrow (\neg a \wedge b)$  par *sylogisme hypothétique* des lignes 2 et 7.
  9.  $\neg a \wedge b$  par *modus ponens* des lignes 5 et 8.

**Rép. 1.38** Chaque prénom est abrégé par sa première lettre, en minuscule.

1.  $\exists!x C(x)$  voir 1.35 pour symbole  $\exists!$
2.  $\forall x (C(x) \rightarrow P(x))$
3.  $\exists x (P(x) \wedge \neg C(x))$
4.  $P(a) \wedge P(d) \rightarrow C(c)$
5.  $P(a) \vee P(c) \rightarrow C(b)$
6.  $\neg P(d)$
7.  $\neg C(d) \rightarrow \neg C(a)$
8.  $P(a) \leftrightarrow \neg P(c)$

**Rép. 1.39** On peut déduire que Benoît est coupable. Voici le détail d'un raisonnement possible permettant d'arriver à cette conclusion.



- |     |  |  |
|-----|--|--|
| 9.  | $\neg C(d)$  | <i>inst. univ.</i> de 2 et <i>modus tollens</i> avec 6.      |
| 10. | $\neg C(a)$  | <i>modus ponens</i> de 7 et 9.                               |
| 11. | $\exists!x C(x) \wedge P(x)$   | <i>modus ponens</i> de 1 et 2.                               |
| 12. | $\exists x \exists y, (x \neq y) \wedge P(x) \wedge P(y)$            | par 3 et 11, car $C(x) \wedge \neg C(x) \equiv \mathbf{F}$ . |
| 13. | $(P(a) \wedge P(b)) \vee (P(a) \wedge P(c)) \vee (P(b) \wedge P(c))$ | par à 12 et 6, car $U = \{a, b, c, d\}$                      |
| 14. | $\neg(P(a) \wedge P(c))$   | par 8 et table 3.4.  |
| 15. | $(P(a) \wedge P(b)) \vee (P(b) \wedge P(c))$                         | par <i>sylogisme disjonctif</i> de 13 et 14.                 |
| 16. | $P(b) \wedge (P(a) \vee P(c))$                                       | équivalent à 15 par <i>distributivité</i> .                  |
| 17. | $P(a) \vee P(c)$   | par <i>simplification</i> de 16.                             |
| 18. | $C(b)$   | par <i>modus ponens</i> de 17 et 5.                          |

*Remarque 1.* Voici comment déduire 12 à partir de 3 et 11 sans s'embourber dans les règles logiques: il y a au moins une personne présente et non coupable (3), il y a une personne présente et coupable (11); on peut donc déduire qu'il y a au moins deux personnes présentes.

*Remarque 2.* Il y a d'autres façons de résoudre ce type de problème, comme l'utilisation de tableaux.

**Rép. 1.40** On pose:

$M(x)$ : «  $x$  est inscrit au cours de Mathématiques discrètes »,

$E(x)$ : «  $x$  a fait les exercices suggérés »,

$I(x)$ : «  $x$  a réussi l'intra ».

- |    |                                    |   |
|----|------------------------------------|---|
| 1. | $\exists x(M(x) \wedge \neg E(x))$ |   |
| 2. | $\forall x(M(x) \rightarrow I(x))$ |   |
| 3. | $\overline{M(c) \wedge \neg E(c)}$ | pour un un certain $c$ , par <i>instanciation existentielle</i> de 1. |
| 4. | $M(c)$                             | par <i>simplification</i> de 3.                                       |
| 5. | $M(c) \rightarrow I(c)$            | par <i>instanciation universelle</i> de 2.                            |
| 6. | $I(c)$                             | par <i>modus ponens</i> de 4 et 5.                                    |
| 7. | $\neg E(c)$                        | par <i>simplification</i> de 3.                                       |
| 8. | $I(c) \wedge \neg E(c)$            | par <i>conjonction</i> de 6 et 7.                                     |
| 9. | $\exists x(I(x) \wedge \neg E(x))$ | par <i>généralisation existentielle</i> de 8.                         |

**Rép. 1.41** (a) On pose:

$H(x)$ : «  $x$  est un homme »,

$M(x)$ : «  $x$  est mortel ».

- |    |  |  |
|----|--|--|
| 1. | $\forall x, H(x) \rightarrow M(x)$                           |  |
| 2. | $H(\text{Socrate})$  |  |
| 3. | $\overline{H(\text{Socrate}) \rightarrow M(\text{Socrate})}$ | par <i>instanciation universelle</i> de 1. |
| 4. | $M(\text{Socrate})$  | par <i>modus ponens</i> de 2 et 3.         |

Le raisonnement est donc **valide**.

(b) On pose,  $B(x)$ : «  $x$  est bleu ».

- |    |  |  |
|----|--|--|
| 1. | $\forall x, H(x) \rightarrow B(x)$                           |  |
| 2. | $H(\text{Socrate})$  |  |
| 3. | $\overline{H(\text{Socrate}) \rightarrow M(\text{Socrate})}$ | par <i>instanciation universelle</i> de 1. |
| 4. | $B(\text{Socrate})$  | par <i>modus ponens</i> de 2 et 3.         |

Le raisonnement est donc **valide**.

- (c) 1.  $\exists x, H(x) \wedge B(x)$   
 2.  $H(\text{Socrate})$   
 3.  $\frac{H(\text{Socrate})}{B(\text{Socrate})}$

Ce raisonnement est **invalid**. En effet, supposons que Socrate est un homme, que Socrate n'est pas bleu, mais qu'il existe un autre homme qui n'est pas Socrate et que cet autre homme est bleu. On a alors :

$$\begin{aligned} \exists x, H(x) \wedge B(x) &\equiv \mathbf{V}, \\ H(\text{Socrate}) &\equiv \mathbf{V}, \\ B(\text{Socrate}) &\equiv \mathbf{F}, \\ ((\exists x, H(x) \wedge B(x)) \wedge H(\text{Socrate})) \rightarrow B(\text{Socrate}) &\equiv (\mathbf{V} \wedge \mathbf{V}) \rightarrow \mathbf{F} \\ &\equiv \mathbf{F}. \end{aligned}$$

Remarque, on a traduit la phrase « *certaines hommes sont bleus* » par  $\exists x, H(x) \wedge B(x)$  qui signifie plutôt « *il existe un homme bleu* ». Pour être plus rigoureux, on aurait pu remarquer que la phrase « *certaines hommes sont bleus* » est équivalente à la phrase « *il y a au moins deux hommes bleus* » ce qui peut être traduit par :

$$\exists x, \exists y, x \neq y \wedge (H(x) \wedge B(x)) \wedge (H(y) \wedge B(y)),$$

Cela dit, le raisonnement reste **invalid** et pour la même raison.

- (d) 1.  $\exists x, H(x) \wedge M(x)$   
 2.  $H(\text{Socrate})$   
 3.  $\frac{H(\text{Socrate})}{M(\text{Socrate})}$

Ce raisonnement est **invalid**. En effet, supposons que Socrate est un homme, que Socrate n'est pas mortel, mais qu'il existe un autre homme qui n'est pas Socrate et que cet autre homme est mortel. On a alors :

$$\begin{aligned} \exists x, H(x) \wedge M(x) &\equiv \mathbf{V}, \\ H(\text{Socrate}) &\equiv \mathbf{V}, \\ M(\text{Socrate}) &\equiv \mathbf{F}, \\ ((\exists x, H(x) \wedge M(x)) \wedge H(\text{Socrate})) \rightarrow M(\text{Socrate}) &\equiv (\mathbf{V} \wedge \mathbf{V}) \rightarrow \mathbf{F} \\ &\equiv \mathbf{F}. \end{aligned}$$

**Rép. 1.42** (a) Le raisonnement B est invalide. On montre que l'implication :

$$\left( (a \rightarrow b) \wedge \neg a \right) \rightarrow \neg b$$

n'est pas une tautologie. En effet, avec l'affectation  $a \equiv \mathbf{F}$  et  $b \equiv \mathbf{V}$  :

$$\begin{aligned} \left( (a \rightarrow b) \wedge \neg a \right) \rightarrow \neg b &\equiv \left( (\mathbf{F} \rightarrow \mathbf{V}) \wedge \neg \mathbf{F} \right) \rightarrow \neg \mathbf{V} \\ &\equiv (\mathbf{V} \wedge \mathbf{V}) \rightarrow \mathbf{F} \\ &\equiv \mathbf{F} \end{aligned}$$

(b) Le raisonnement A est valide :

1.  $a \rightarrow (b \wedge c)$   
 2.  $\neg c$   
 3.  $\frac{a \rightarrow (b \wedge c) \quad \neg c}{a \rightarrow b} \quad$   
 4.  $a \rightarrow c$   
 5.  $\neg a$

Table 2.6 et ligne 1

Simplification de 3.

Modus tollens de 2 et 4.

Le raisonnement C est valide :

1.	$(a \vee b) \wedge c$	
2.	$d \rightarrow \neg c$	
3.	$d \vee \neg a$	
4.	$c$	Simplification de 1.
5.	$\neg d$	Modus tollens de 5 et 2.
6.	$\neg a$	Syllogisme disjonctif de 3 et 5.
7.	$a \vee b$	Simplification de 1.
8.	$b$	Syllogisme disjonctif de 7 et 6.

**Rép. 1.43** L'ensemble de spécifications est cohérent.

Démonstration. Soit  $f$  : La communication est fluide,  $c$  : le logiciel Communik est installé,  $s$  : la version 10 ou plus du système d'exploitation est installée. Alors :

p1:  $c \rightarrow f$ , ce qui est équivalent à  $\neg c \vee f$

p2:  $c \rightarrow s$ , ce qui est équivalent à  $\neg c \vee s$

p3:  $\neg s$

p4:  $f$

L'assignation  $c = \mathbf{F}, f = \mathbf{V}, s = \mathbf{F}$  permet de satisfaire chacune des spécifications.

proposition	justification	affectation	prop. satisfaite
1.	$c \rightarrow f$		
2.	$c \rightarrow s$		
3.	$\neg s$	$s = \mathbf{F}$	$p3$
4.	$f$	$f = \mathbf{V}$	$p4$
5.	$\neg c$	$c = \mathbf{F}$	$p2$ (et $p1$ )

**Rép. 1.44** L'ensemble de spécifications est incohérent.

proposition	justification	affectation	prop. satisfaite
1.	$f \rightarrow c$		
2.	$c \rightarrow s$		
3.	$\neg s$	$s = \mathbf{F}$	$p3$
4.	$f$	$f = \mathbf{V}$	$p4$
5.	$\neg c$	$c = \mathbf{F}$	$p2$
6.	$\neg f$	$f = \mathbf{F}$	contradiction avec $p4$

**Rép. 1.45** L'ensemble de spécifications est cohérent.

Démonstration. Soit  $f$  : La communication est fluide,  $c$  : le logiciel Communik est installé,  $s$  : la version 10 ou plus du système d'exploitation est installée. Les propositions 1 à 4 doivent être vraies.

proposition	justification	affectation	prop. satisfaite
1.	$f \rightarrow c$		
2.	$c \rightarrow s$		
3.	$s \rightarrow f$		
4.	$\neg s$	$s = \mathbf{F}$	$p4$ (et $p3$ )
5.	$\neg c$	$c = \mathbf{F}$	$p2$
6.	$\neg f$	$f = \mathbf{F}$	$p1$

Ainsi, l'assignation  $s = \mathbf{F}, c = \mathbf{F}$  et  $f = \mathbf{F}$  satisfait chacune des spécifications.

**Rép. 1.46** L'ensemble de spécifications est cohérent.

**Rép. 1.47** Il y a 2 assignations qui permettent de satisfaire chacune des spécifications:  
 $a = b = \mathbf{V}$  et  $c = d = \mathbf{F}$  ou encore  $a = b = c = \mathbf{F}$  et  $d = \mathbf{V}$ .

**Rép. 1.48** L'ensemble de spécifications est incohérent.

**Rép. 1.49** L'assignation  $p = \mathbf{V}, q = \mathbf{F}, r = \mathbf{V}, s = \mathbf{F}, t = \mathbf{V}$  permet de satisfaire chacune des spécifications.

proposition	justification	affectation	prop. satisfaite
1. $r \wedge (\neg p \rightarrow q)$			
2. $\neg q \vee s$			
3. $t \rightarrow q \vee \neg s$			
4. $\neg(\neg q \wedge s) \rightarrow t$			
5. $s \rightarrow \neg r$			
6. $r$	par <i>Simplification</i> de 1.	$r = \mathbf{V}$	
7. $\neg s$	par <i>Modus Tollens</i> de 6 et 5.	$s = \mathbf{F}$	5.
8. $\neg p \rightarrow q$	par <i>Simplification</i> de 1.		
9. $\neg q$	par <i>Syllogisme disjonctif</i> de 2 et 7.	$q = \mathbf{F}$	2. et 3.
10. $p$	par <i>Modus Tollens</i> de 8 et 9, et <i>Double négation</i> .	$p = \mathbf{V}$	1.
11. $\neg s \vee q$	par <i>Addition</i> sur 7.		
12. $(q \vee \neg s) \rightarrow t$	par <i>De Morgan</i> sur 4.		
13. $t$	par <i>Modus Ponens</i> de 11 et 12.	$t = \mathbf{V}$	4.

**Rép. 1.50** Geneviève est coupable. En effet, les affirmations de Xavier et de Stéphane sont incompatibles: seule l'une d'elles peut-être vraie. Si c'est Xavier qui dit vrai, alors Stéphane est coupable et dans ce cas Geneviève dit vrai aussi (mais c'est impossible, car on sait qu'un seul ami dit vrai). On peut en déduire que Xavier ment. Ainsi, Stéphane n'est pas coupable. Par conséquent, Stéphane dit vrai et on peut en conclure que les autres amis mentent. Puisque Geneviève se dit non coupable est qu'elle ment, elle est coupable!

**Voici une autre façon de rédiger la solution.** Posons  $x$ : Xavier est coupable,  $g$ : Geneviève est coupable, etc. Les quatre déclarations des amis sont donc:

$$p1: x \quad p2: \neg g \quad p3: s \quad p4: \neg s.$$

Nous savons qu'une seule des propositions 1 à 4 est vraie. Analysons les quatre possibilités (il s'agit d'une preuve par cas). Si  $p1$  est vraie, alors on doit avoir

$$x \wedge g \wedge s \wedge \neg s,$$

ce qui est impossible, car  $\neg s \wedge s \equiv \mathbf{F}$ . Si  $p2$  est vraie, on doit avoir

$$\neg x \wedge \neg g \wedge \neg s \wedge s,$$

ce qui, encore une fois, est impossible. Si  $p3$  est vraie, on doit avoir

$$\neg x \wedge g \wedge s \wedge s,$$

ce qui est impossible, car il n'y a qu'un seul coupable donc  $g \wedge s = \mathbf{F}$ . Si  $p4$  est vraie, on doit avoir

$$\neg x \wedge g \wedge \neg s \wedge \neg s,$$

ce qui est possible: aucune incohérence et un seul coupable. Geneviève est donc coupable.

**Rép. 1.51** Xavier est coupable.

**Rép. 1.52** La discussion a lieu un mercredi. Anouk est sincère quand elle dit « hier (mardi) j'ai menti » et Geneviève ment quand elle dit « moi aussi » : elle ne peut avoir menti la veille, car elle est sincère le mardi. (Un tableau est suggéré pour y voir plus clair.)

- Rép. 1.53** (a)  $A$  est sincère et  $B$  est menteur. (d) On ne sait pas.  
 (b)  $A$  est menteur et  $B$  est sincère. (e)  $A$  est menteur et  $B$  est sincère.  
 (c)  $A$  et  $B$  sont sincères.

**Rép. 1.54** La question « Si je demande à l'autre gardien quelle porte mène au cours de Mathématiques discrètes, laquelle me pointerait-il ? » conduira chaque gardien à pointer la porte de l'enfer et il suffira d'ouvrir l'autre porte. En effet, soit  $A$  est sincère et il pointerait alors la porte que son collègue menteur indiquerait (donc la mauvaise), soit  $A$  est menteur et alors  $B$  est sincère :  $B$  indiquerait la bonne porte, mais  $A$  va mentir et indiquera la mauvaise porte.

**Rép. 1.55** (a)  $\forall x \in \mathbb{Z}, P(x^2) \rightarrow P(x) \equiv \forall x \in \mathbb{Z}, \neg P(x) \rightarrow \neg P(x^2)$ .  
 Vrai, preuve par contraposée. Il faut donc fournir une preuve. Voir celle présentée à l'exemple de la page 50.

(b)  $\forall x \in \mathbb{R}, \forall y \in \mathbb{R}, Q(x) \wedge \neg Q(y) \rightarrow \neg Q(x+y)$ .  
 Vrai. Il faut donc fournir une preuve. Voir celle présentée en classe.

(c)  $p \equiv \forall x \in \mathbb{R}, \forall y \in \mathbb{R}, Q(x) \wedge x \neq 0 \wedge \neg Q(y) \rightarrow \neg Q(xy)$ .  
 Vrai. Preuve par contradiction. Supposons que l'énoncé est faux. Par De Morgan et 2.5, on obtient :  
 $\neg p \equiv \exists x \in \mathbb{R}, \exists y \in \mathbb{R}, (Q(x) \wedge x \neq 0 \wedge \neg Q(y)) \wedge Q(xy)$ .  
 Ainsi, il existe un nombre  $x \in \mathbb{Q}$ , avec  $x \neq 0$ , et un nombre  $y \notin \mathbb{Q}$  tels que leur produit est rationnel :  $xy \in \mathbb{Q}$ . Ainsi, il existe des entiers  $a \neq 0, b \neq 0, c$  et  $d \neq 0$  tels que  $x = \frac{a}{b}$  et  $xy = \frac{c}{d}$ . Donc  $y = (xy) \cdot \frac{1}{x} = \frac{c}{d} \cdot \frac{b}{a} = \frac{bc}{ad}$ , et on conclut que  $y$  est rationnel, ce qui contredit  $y \notin \mathbb{Q}$ .  
 On a montré que  $\neg p \rightarrow \text{F}$ , ce qui permet de conclure que  $\neg p$  est faux, et donc que  $p$  est vrai.

(d)  $\neg Q\left(\frac{\sqrt{2}}{3}\right)$ . Vrai, c'est un cas particulier de (c) :  $\frac{\sqrt{2}}{3} = \frac{1}{3} \cdot \sqrt{2}$ ,  $\frac{1}{3} \in \mathbb{Q}$  et  $\sqrt{2} \notin \mathbb{Q}$ .

(e)  $\neg Q\left(\frac{1}{\sqrt{2}}\right)$ . Vrai. Preuve par contradiction. Supposons que l'énoncé est faux. Ainsi,  $x = \frac{1}{\sqrt{2}} \in \mathbb{Q}$ . Alors il existe des entiers  $a$  et  $b$  tels que  $x = \frac{a}{b}$  et  $b \neq 0$ . De plus,  $a \neq 0$  car  $x \neq 0$ . Ainsi,  $\frac{1}{x} = 1 \div \frac{a}{b} = \frac{b}{a} \in \mathbb{Q}$ . Or  $\frac{1}{x} = 1 \div \frac{1}{\sqrt{2}} = \sqrt{2} \notin \mathbb{Q}$ . Contradiction. On conclut donc que  $x = \frac{1}{\sqrt{2}} \notin \mathbb{Q}$ .

(f)  $p \equiv \forall x \in \mathbb{R}, \neg Q(x) \rightarrow \neg Q\left(\frac{1}{x}\right)$ .  
 Vrai. Il faut donc fournir une preuve. Puisque  $\neg Q(x) \equiv x \notin \mathbb{Q}$  et  $Q(x) \equiv x \in \mathbb{Q}$  on a, par contraposée :  
 $p \equiv \forall x \in \mathbb{R}, \frac{1}{x} \in \mathbb{Q} \rightarrow x \in \mathbb{Q}$ .  
 Prenons un nombre réel  $x$  quelconque et supposons que  $\frac{1}{x} \in \mathbb{Q}$ , c'est-à-dire que  $\frac{1}{x}$  peut s'écrire comme une fraction  $\frac{a}{b}$ , avec  $a \in \mathbb{Z}$  et  $b \in \mathbb{Z}$  et  $b \neq 0$ . Il suffit d'inverser la fraction pour obtenir  $x = \frac{b}{a}$ . On sait que  $a \neq 0$ , car  $\frac{a}{b} = \frac{1}{x}$ . Ainsi,  $x$  est rationnel. L'implication est démontrée.

(g)  $p \equiv \forall x \in \mathbb{R}, \forall y \in \mathbb{R}, \neg Q(x) \wedge \neg Q(y) \rightarrow \neg Q(x+y)$ .  
 Faux. Montrons que  $\neg p$  est vrai. Par De Morgan et l'équivalence 2.5, on a :  
 $\neg p \equiv \exists x \in \mathbb{R}, \exists y \in \mathbb{R}, (\neg Q(x) \wedge \neg Q(y)) \wedge Q(x+y)$ .  
 Prenons  $x = \sqrt{2} \notin \mathbb{Q}$  et  $y = 1 - \sqrt{2}$ . On sait que  $y \notin \mathbb{Q}$ , car  $(-1) \cdot \sqrt{2} \notin \mathbb{Q}$  par (c) et donc  $y = 1 + (-1) \cdot \sqrt{2} \notin \mathbb{Q}$  par (b). Or  $x + y = 1 \in \mathbb{Q}$ . Nous avons ainsi trouvé un contre-exemple à l'énoncé universel : il est donc faux.

(h)  $p \equiv \forall x \in \mathbb{R}, \forall y \in \mathbb{R}, \neg Q(x) \wedge \neg Q(y) \rightarrow \neg Q(xy)$ .  
 Prenons  $x = y = \sqrt{2} \notin \mathbb{Q}$ . On a  $xy = 2 \in \mathbb{Q}$ . Nous avons ainsi trouvé un contre-exemple à l'énoncé universel : il est donc faux.

Ou encore, prenons  $x = \sqrt{2} \notin \mathbb{Q}$  et  $y = \frac{1}{\sqrt{2}}$ . On sait que  $y \notin \mathbb{Q}$ , tel que vu en (e). Or  $xy = 1 \in \mathbb{Q}$ . Nous avons ainsi trouvé un contre-exemple à l'énoncé universel : il est donc faux.

(i)  $p \equiv \forall x \in \mathbb{R}, Q(x) \wedge x \neq 0 \rightarrow Q\left(\frac{1}{x}\right) \equiv \forall x \in \mathbb{R}, x \in \mathbb{Q} \wedge x \neq 0 \rightarrow \frac{1}{x} \in \mathbb{Q}$ .  
 Vrai. Il faut donc fournir une preuve. Si  $x$  est rationnel et non nul, il suffit d'inverser la fraction pour obtenir  $\frac{1}{x}$ , donc  $\frac{1}{x} \in \mathbb{Q}$ .

- (j)  $p \equiv \forall x \in \mathbb{R}, \neg Q(x) \rightarrow \neg Q(\frac{1}{x})$ .  
 Vrai. Il s'agit d'une formulation équivalente à (f).

- Rép. 1.56** (a) Faux (d) Faux (g) Vrai (j) Vrai  
 (b) Vrai (e) Vrai (h) Faux  
 (c) Vrai (f) Vrai (i) Vrai

- Rép. 1.57** (a) Vrai (d) Faux (g) Faux en général, mais vrai dans  
 le cas où  $x = \emptyset$ .  
 (b) Faux (e) Faux  
 (c) Faux (f) Vrai

- Rép. 1.58** (a)  $\{1, 2, 3, 4, 7\}$  (g)  $\{1, 8\}$   
 (b)  $\{2\}$  (h)  $\emptyset$   
 (c)  $\{2, 3\}$  (i)  $\{5, 6\}$   
 (d)  $\{7\}$  (j)  $\{6, 7, 8\}$   
 (e)  $\{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$  (k)  $\{2, 3, 5, 6\}$   
 (f)  $\emptyset$  (l)  $\{(6, 1), (6, 2), (7, 1), (7, 2), (8, 1), (8, 2)\}$

- Rép. 1.59** (a)  $A = \{r \in R, |M(\text{Alice}, r)\} = \{J\}$ ; ensemble de cardinalité 1.  
 (b)  $B = \{x \in X, |M(x, P)\} = \{\text{Guy, Julien, Manon, Ugo}\}$ ; ensemble de cardinalité 4.  
 (c)  $C = \{x \in X, |L(x, K)\} = \{\text{Bartez, Guy}\}$ ; ensemble de cardinalité 2.  
 (d)  $D = \{r \in R, |\forall x \in X, L(x, r)\} = \{P, Z\}$ ; ensemble de cardinalité 2.  
 (e)  $E = \{r \in R, |\exists x \in X, \neg L(x, r)\} = \{K, J, S\}$ ; ensemble de cardinalité 3.  
 (f)  $D \cup E = \{K, J, P, S, Z\}$ ; ensemble de cardinalité 5.  
 (g)  $D \cap E = \{\}$ ; ensemble de cardinalité 0.  
 (h)  $B \cap C = \{\text{Guy}\}$ ; ensemble de cardinalité 1.  
 (i)  $X - B = \{\text{Alice, Bartez}\}$ ; ensemble de cardinalité 2.  
 (j)  $\overline{E} = \{P, Z\}$ ; ensemble de cardinalité 2.

- Rép. 1.60** (a)  $L_1 \cup L_2 = \{p_1, p_2, p_3, p_4, p_5, p_6\}$ ; de cardinalité 6.  
 (b)  $L_2 \cup L_4 = \{p_1, p_2, p_3, p_6\}$ ; de cardinalité 4.  
 (c)  $(L_1 \cup L_2) \cap L_4 = \{p_1, p_2, p_6\}$ ; de cardinalité 3.  
 (d)  $L_2 - L_4 = \{p_3\}$ ; de cardinalité 1.  
 (e)  $\overline{L_1 \cup L_2} = \{p_7\}$ ; de cardinalité 1.  
 (f)  $\bigcup_{i=2}^4 L_i = \{p_1, p_2, p_3, p_4, p_6, p_7\}$ ; de cardinalité 6.  
 (g)  $\bigcap_{i=3}^5 L_i = \{\}$ ; de cardinalité 0.

- Rép. 1.61** (a) Vrai  
 (b) Vrai  
 (c) Faux  
 (d) Faux  
 (e) Vrai  
 (f) Vrai  
 (g) Faux  
 (h) Vrai  
 (i) Vrai  
 (j) Vrai

- Rép. 1.62** (a) 1. Faux.  
 (b) 8. Faux.  
 (c) 14. Faux.  
 (d) 13. Faux.  
 (e) 10. Vrai.  
 (f) 9. Vrai.  
 (g) 12. Vrai.

- Rép. 1.63** (a)  $\overline{A \cap B \cap C} \cup C = \overline{A} \cup \overline{B} \cup \overline{C} \cup C$  De Morgan  
 $= \overline{A} \cup \overline{B} \cup U$  Négation  
 $= U$  Domination
- (b)  $A \cap (H \cup D \cup A) = A$  Absorption
- (c)  $\overline{(A \cap B) \cup \overline{B}} = \overline{(A \cap B) \cap \overline{\overline{B}}}$  De Morgan  
 $= \overline{(\overline{A} \cup \overline{B}) \cap B}$  De Morgan et double négation  
 $= \overline{(\overline{A} \cap B) \cup (\overline{B} \cap B)}$  Distributivité  
 $= \overline{(\overline{A} \cap B) \cup \emptyset}$  Négation  
 $= \overline{\overline{A} \cap B}$  Identité
- (d)  $\overline{(\overline{A} \cup \overline{B}) \cup \overline{B}} = \overline{(\overline{A} \cap \overline{B}) \cup \overline{B}}$  De Morgan  
 $= \overline{\overline{B}}$  Absorption
- (e)  $\overline{(\overline{A \cup \overline{B}}) \cup (\overline{C \cup \overline{A}})}$  =  $\overline{(\overline{A \cup \overline{B}}) \cap (\overline{C \cup \overline{A}})}$  De Morgan  
 $= \overline{(\overline{A} \cap \overline{\overline{B}}) \cap (\overline{C} \cap \overline{\overline{A}})}$  De Morgan  
 $= \overline{(\overline{A} \cap B) \cap (\overline{C} \cap A)}$  Double négation  
 $= \overline{A \cap \overline{A} \cap B \cap \overline{C}}$  Associativité et commutativité  
 $= \overline{\emptyset \cap B \cap \overline{C}}$  Négation  
 $= \emptyset$  Domination

- Rép. 1.64** (a)  $(x \geq 2) \wedge (x \leq 4)$   
 (b)  $(x \geq 2)$   
 (c)  $(x \geq 2) \wedge (x \leq 4) \vee (x \geq 7) \wedge (x \leq 9)$   
 (d)  $(x \leq 0) \vee (x \geq 10) \wedge (x \leq 15)$   
 (e)  $(x \geq 2) \wedge (x \leq 8) \wedge \neg(x = 4)$

- Rép. 1.65** (a) **Injective**: non, contre exemple:  $f(11111110) = 1 = f(11111101)$ .  
**Surjective**: non, on considère  $9 \in \mathbb{N}$ , quelque soit  $t \in T_8$ , il est impossible que  $f(t) = 9$  car  $t$  contient au maximum 8 occurrences de 0.  
**Bijective**: non car  $f$  n'est pas injective, ni surjective.  
 (b) **Injective**: oui, soit  $t_1, t_2 \in T_8$ , on montre que si  $f(t_1) = f(t_2)$  alors  $t_1 = t_2$ ,

$$\begin{aligned} f(t_1) &= f(t_2) \\ \sim t_1 &= \sim t_2 && \text{(définition de } f) \\ \sim \sim t_1 &= \sim \sim t_2 && \text{(négation de chaque côté)} \\ t_1 &= t_2 && \text{(double négation).} \end{aligned}$$

**Surjective**: oui, soit  $y \in T_8$ , on montre qu'il existe  $x \in T_8$  tel que  $f(x) = y$ . On choisit  $x = \sim y$ . On a alors:

$$f(x) = f(\sim y) = \sim \sim y = y.$$

**Bijective**: oui car  $f$  est injective et surjective.

(c) **Injective**: non, contre-exemple:

$$\begin{aligned} f(11111111) &= 11111111 \vee 11110000 = 11111111 \\ f(00001111) &= 00001111 \vee 11110000 = 11111111 \end{aligned}$$

**Surjective**: non, on considère  $00000000 \in T_8$ , quelque soit  $t \in T_8$ , il est impossible que  $f(t) = 00000000$ . En effet, si on considère  $b$  le premier bit de  $t$ , on a que  $b \vee 1 = 1$ , ainsi  $f(t)$  commence par le bit 1 et on a  $f(t) \neq 00000000$ .

**Bijective**: non car  $f$  n'est pas injective, ni surjective.

(d) **Injective**: non, contre-exemple:  $f(11111110) = 7 = f(11111101)$ .

**Surjective**: oui, soit  $y \in \{0, 1, \dots, 8\}$ , on considère  $t$  le train de bit:

$$t = \underbrace{111 \cdots 1}_{y \text{ fois}} \cdot \underbrace{000 \cdots 0}_{8-y \text{ fois}}$$

On a bien  $f(t) = y$ .

**Bijective**: non car  $f$  n'est pas injective.

**Rép. 1.66** (a) **Injective**: oui; soit  $x_1, x_2 \in \mathbb{R}$ , on montre que si  $f(x_1) = f(x_2)$  alors  $x_1 = x_2$ :

$$\begin{aligned} f(x_1) &= f(x_2) \\ 2x_1 + 1 &= 2x_2 + 1 \\ 2x_1 &= 2x_2 \\ x_1 &= x_2. \end{aligned}$$

**Surjective**: oui; soit  $y \in \mathbb{R}$ , on montre qu'il existe  $x \in \mathbb{R}$  tel que  $f(x) = y$ :

$$\begin{aligned} f(x) &= y \\ 2x + 1 &= y \\ 2x &= y - 1 \\ x &= \frac{y-1}{2}. \end{aligned}$$

On a bien  $f(x) = f\left(\frac{y-1}{2}\right) = 2\left(\frac{y-1}{2}\right) + 1 = y$ .

**Bijective**: oui;  $f$  est injective et surjective.

(b) **Injective**: non; contre-exemple:

$$f(1) = f(-1) = 2.$$

**Surjective**: non; -1 n'est pas dans l'image puisque:

$$\begin{aligned} f(x) &= -1 \\ x^2 + 1 &= -1 \\ x^2 &= -2 \\ x &= \pm\sqrt{-2} \text{ qui n'est pas dans } \mathbb{R}. \end{aligned}$$

**Bijective**: non;  $f$  n'est pas injective, ni surjective.

(c) **Injective**: oui; soit  $x_1, x_2 \in \mathbb{R}$ , on montre que si  $f(x_1) = f(x_2)$  alors  $x_1 = x_2$ :

$$\begin{aligned} f(x_1) &= f(x_2) \\ \frac{x_1}{x_1} &= \frac{x_2}{x_2} \\ 1 &= 1 \\ x_1 &= x_2. \end{aligned}$$



**Surjective:** non;  $\frac{2}{3}$  n'est pas dans l'image puisque :

$$\begin{aligned} f(x) &= \frac{2}{3} \\ \frac{x}{1} &= \frac{2}{3} \\ x &= \frac{2}{3} \text{ qui n'est pas dans } \mathbb{Z}. \end{aligned}$$

**Bijective:** non;  $f$  n'est pas surjective.

(d) **Injective:** non; contre-exemple :

$$f\left(\frac{2}{3}\right) = f\left(\frac{3}{2}\right) = 5.$$

**Surjective:** oui; soit  $y \in \mathbb{Z}$ , on montre qu'il existe  $x \in \mathbb{Q}$  tel que  $f(x) = y$ . Tout entier  $y$  peut s'écrire comme

$$y = 1 + (y - 1).$$

On choisit donc  $x = \frac{y-1}{1}$ .

On a bien  $f(x) = f\left(\frac{y-1}{1}\right) = (y-1) + 1 = y$ .

**Bijective:** non;  $f$  n'est pas injective.

(e) **Injective:** non; contre-exemple :

$$f\left(\frac{1}{2}\right) = f(1) = 1.$$

**Surjective:** oui; soit  $y \in \mathbb{Z}$ , on montre qu'il existe  $x \in \mathbb{R}$  tel que  $f(x) = y$ . On choisit  $x = y$ .

On a bien  $f(x) = f(y) = \lceil y \rceil = y$ .

**Bijective:** non;  $f$  n'est pas injective.

**Rép. 1.67** (a) Non.

(b) Non.

**Rép. 1.68** (a) Non.

(b) Non car  $f(0) = f(2)$ .

**Rép. 1.69** (a) Oui.

(b) Non.

## Chapitre 2

**Rép. 2.1**

Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
0	1	2	3	4	5	6

$(1 + 30) \bmod 7 = 3$  (Jeudi)

**Rép. 2.2**  $(15 + 60) \bmod 24 = 3$  heures

**Rép. 2.3** (a) Vrai;  $13 = 7 + 1 \cdot 6$

(b) Vrai;  $10 = -2 + 2 \cdot 6$

(c) Faux;  $14 \bmod 6 = 8 \neq 0$

(d) Vrai;  $600 \bmod 6 = 0$  et  $30 \bmod 6 = 0$

(e) Vrai;  $5^3 \bmod 20 = 5$

- Rép. 2.4** (a)  $-2 \equiv -11 + 9 \equiv 0 + 9 \equiv 9 \pmod{11}$ . Donc  $-2 \pmod{11} = 9$ .  
 (b)  $(-2)^3 \equiv -8 \equiv 3 \pmod{11}$ . Donc  $(-2)^3 \pmod{11} = 3$ .  
 (c)  $55 \cdot 13 + 8 \cdot 47 \equiv (5 \cdot 11)(11 + 2) + 8 \cdot (44 + 3) \equiv (0)(0 + 2) + 8(0 + 3) \equiv 0 + 24 \equiv 2 \cdot 11 + 2 \equiv 0 + 2 \equiv 2 \pmod{11}$ .  
 Donc  $55 \cdot 13 + 8 \cdot 47 \pmod{11} = 2$ .  
 (d)  $9^2 \equiv 81 \equiv 7 \cdot 11 + 4 \equiv 0 + 4 \equiv 4 \pmod{11}$ . Donc  $9^2 \pmod{11} = 4$ .  
 (e)  $9^4 \equiv 9^2 \cdot 9^2 \equiv 4 \cdot 4 \equiv 16 \equiv 11 + 5 \equiv 0 + 5 \equiv 5 \pmod{11}$ . Donc  $9^4 \pmod{11} = 5$ .  
 (f)  $9^8 \equiv 9^4 \cdot 9^4 \equiv 5 \cdot 5 \equiv 25 \equiv 2 \cdot 11 + 3 \equiv 0 + 3 \equiv 3 \pmod{11}$ . Donc  $9^8 \pmod{11} = 3$ .  
 (g)  $9^{16} \equiv 9^8 \cdot 9^8 \equiv 3 \cdot 3 \equiv 9 \pmod{11}$ . Donc  $9^{16} \pmod{11} = 9$ .

- Rép. 2.5** (a)  $(52)_{10}$  (c)  $(171)_{10}$   
 (b)  $(333)_{10}$  (d)  $(210)_{10}$

**Rép. 2.6** Tout d'abord, on remarque que pour toute base  $b \geq 3$ , le système de numération possède au moins les chiffres : 0, 1 et 2. Ensuite, par définition de l'écriture dans une base  $b$  quelconque,

- $(121)_b = 1 \cdot b^2 + 2 \cdot b + 1$ ,
- $(11)_b = 1 \cdot b + 1$ .

On vérifie l'égalité demandée :

$$((11)_b)^2 = (b + 1)^2 = b^2 + 2b + 1 = (121)_b.$$

- Rép. 2.7** (a)  $(777)_8$  (d)  $(102101)_3$   
 (b)  $(AAA)_{11}$  (e)  $(1101011101001)_2$   
 (c)  $(ABC)_{16}$

- Rép. 2.8** (a)  $(170235)_8$ . (c)  $(3FC45)_{16}$ .  
 (b)  $(1B6D)_{16}$ . (d)  $(125273)_8$ .

- Rép. 2.9** (a) 1  
 (b) 9  
 (c) 7  
 (d) 3

- Rép. 2.10** (a) 2  
 (b) 239815

- Rép. 2.11** (a) Faux;  $205 = 41 \cdot 5$   
 (b) Vrai.  
 (c) Faux. Un nombre premier doit être supérieur ou égal à 2.  
 (d) Vrai. Preuve par cas :

4	6	8	10	12	14	16	18	20
2 + 2	3 + 3	3 + 5	5 + 5	5 + 7	7 + 7	5 + 11	7 + 11	7 + 13
			3 + 7		3 + 11	3 + 13	5 + 13	3 + 17

- Rép. 2.12** (a)  $3^5 5^2$   
 (b) 24  
 (c) 5

- Rép. 2.13** (a)  $1 = (1)9 + (-2) \cdot 4$   
 (b)  $11 = (-1)121 + (4)33$   
 (c)  $2 = (-3) \cdot 32 + (7) \cdot 14$   
 (d)  $5 = (1) \cdot 35 + (-2) \cdot 15$   
 (e)  $1 = (28)55 + (-19)81$

- Rép. 2.14** (a) Vrai  
 (b) 2  
 (c) 2 ne possède pas d'inverse modulo 4.  
 (d) Modulo 5: l'inverse de 1 est 1, l'inverse de 2 est 3, l'inverse de 3 est 2 et l'inverse de 4 est 4.  
 (e) Faux  
 (f)  $x \equiv 3 \pmod{5}$

- Rép. 2.15** (a) Le nombre 15 n'est pas inversible modulo 55 car  $\text{PGCD}(15, 55) = 5 \neq 1$ .  
 (b) L'inverse de 3 modulo 13 est 9.  
 (c) L'inverse de 11 modulo 12 est 11.  
 (d) Le nombre 21 n'est pas inversible modulo 30 car  $\text{PGCD}(21, 30) = 3 \neq 1$ .  
 (e) L'inverse de 23 modulo 30 est 17.

- Rép. 2.16** (a) 6 (f) Aucune solution;  
 (b) 11 (g) Aucune solution;  
 (c) Aucune solution; (h)  $x \equiv 9 \pmod{90}$ ;  
 (d)  $x \equiv 2 \pmod{12}$ ;  
 $x \equiv 24 \pmod{90}$ ;  
 $x \equiv 6 \pmod{12}$ ;  
 $x \equiv 39 \pmod{90}$ ;  
 $x \equiv 10 \pmod{12}$ .  
 $x \equiv 54 \pmod{90}$ ;  
 (e)  $x \equiv 4 \pmod{10}$ ;  
 $x \equiv 69 \pmod{90}$ ;  
 $x \equiv 9 \pmod{10}$ ;  
 $x \equiv 84 \pmod{90}$ .

- Rép. 2.17** (a)  $f^{-1}(c) = 5c + 2 \pmod{26}$   
 (b)  $f^{-1}(c) = 3c + 15 \pmod{26}$   
 (c) Pas une fonction de chiffrement affine.  
 (d)  $f^{-1}(c) = 19c + 6 \pmod{26}$

- Rép. 2.18** (a) ZORO  
 (b) YOLO

- Rép. 2.19** (a) Le mot STOP correspond au nombre  $M = 18191415$ . La clé de chiffrement est  $n = 2537$  et  $e = 13$ . Le message est trop grand pour être chiffré directement avec la clé. Il faut le couper en deux blocs:  $M_1 = 1819$ ,  $M_2 = 1415$ .

$$C_1 = M_1^e \pmod{n} = 1819^{13} \pmod{2537} = 2081.$$

$$C_2 = M_2^e \pmod{n} = 1415^{13} \pmod{2537} = 2182.$$

Le message chiffré est donc: 2081 2182

- (b) Le mot STOP correspond au nombre  $M = 18191415$ . La clé de chiffrement est  $n = 245363$  et  $e = 323$ . Le message est trop grand pour être chiffré directement avec la clé. Il faut le couper en deux blocs:  $M_1 = 1819$ ,  $M_2 = 1415$ .

$$C_1 = M_1^e \pmod{n} = 1819^{323} \pmod{245363} = 239815.$$

$$C_2 = M_2^e \pmod{n} = 1415^{323} \pmod{245363} = 087533.$$

Le message chiffré est donc: 239815 087533.

- (c) Le message chiffré est  $C_1 = 2081$ ,  $C_2 = 2182$ . La clé de chiffrement utilisée est  $n = 2537$  et  $e = 13$ . Pour déchiffrer il faut calculer la clé de déchiffrement  $d \equiv e^{-1} \pmod{s}$  où  $s = (p-1)(q-1)$  et  $pq = n$ .

- Factorisation de la clé.

$$n = 2537 = 43 \cdot 59.$$

On a donc  $p = 43$ ,  $q = 59$ ,  $s = (p - 1)(q - 1) = 2436$ .

- Calcul de l'inverse modulaire. On fait le Bézout de  $e$  et  $s$ :

$$937 \cdot e + (-5) \cdot s = 1.$$

L'inverse de  $e$  modulo  $s$  est donc  $d = 937$ . On peut maintenant procéder au déchiffrement :

$$M_1 = C_1^d \bmod n = 2081^{937} \bmod 2537 = 1819.$$

$$M_2 = C_2^d \bmod n = 2182^{937} \bmod 2537 = 1415.$$

En convertissant le message  $M = 1819\ 1415$  en lettres, on obtient le mot : STOP.

- (d) Le message chiffré est  $C_1 = 239815$ ,  $C_2 = 087533$ . La clé de chiffrement utilisée est  $n = 245363$  et  $e = 323$ . Pour déchiffrer il faut calculer la clé de déchiffrement  $d \equiv e^{-1} \bmod s$  où  $s = (p - 1)(q - 1)$  et  $pq = n$ .

- Factorisation de la clé.

$$n = 245363 = 131 \cdot 1873.$$

On a donc  $p = 131$ ,  $q = 1873$ ,  $s = (p - 1)(q - 1) = 243360$ .

- Calcul de l'inverse modulaire. On fait le Bézout de  $e$  et  $s$ :

$$-94933 \cdot e + 126 \cdot s = 1.$$

L'inverse de  $e$  modulo  $s$  est donc  $d = -94933 + 243360 = 148427$ . On peut maintenant procéder au déchiffrement :

$$M_1 = C_1^d \bmod n = 239815^{148427} \bmod 245363 = 1819.$$

$$M_2 = C_2^d \bmod n = 87533^{148427} \bmod 245363 = 1415.$$

En convertissant le message  $M = 1819\ 1415$  en lettres, on obtient le mot : STOP.

**Rép. 2.20** (a)  $(n, d) = (73\ 813, 28177)$

(b) OVNI

## Chapitre 3

**Rép. 3.1** (a)  $\sim(0011\ 1100 \oplus 1010\ 0011) = 0110\ 0000$

Expression	Représentation
	0011 1100
$0011\ 1100 \oplus 1010\ 0011$	$\oplus 1010\ 0011$
	<hr/> 1001 1111
$\sim(0011\ 1100 \oplus 1010\ 0011)$	0110 0000

(b)  $1010\ 1010 \vee (0000\ 0001 \ll 5) = 1010\ 1010$ .

Expression	Représentation
$0000\ 0001 \ll 5$	0010 0000
	1010 1010
$1010\ 1010 \vee (0000\ 0001 \ll 5)$	$\vee 0010\ 0000$
	<hr/> 1010 1010

(c)  $\sim(1101\ 1011 \wedge (\sim 0001\ 0111)) = 0011\ 0111$

Expression	Représentation
$\sim 0001\ 0111$	1110 1000
$1101\ 1011 \wedge (\sim 0011\ 0111)$	$\begin{array}{r} 1101\ 1011 \\ \wedge 1110\ 1000 \\ \hline 1100\ 1000 \end{array}$
$\sim(1101\ 1011 \wedge (\sim 0011\ 0111))$	0011 0111

(d)  $(1100\ 1011 \gg 3) \oplus (0101\ 0100 \ll 2) = 1010\ 1001$

Expression	Représentation
$1100\ 1011 \gg 3$	1111 1001
$0101\ 0100 \ll 2$	0101 0000
$(1100\ 1011 \gg 3) \oplus (0101\ 0100 \ll 2)$	$\begin{array}{r} 1111\ 1001 \\ \oplus 0101\ 0000 \\ \hline 1010\ 1001 \end{array}$

**Rép. 3.2** (a)  $((15 \ll 4) \oplus (-93)) \gg 5 = 2.$

Expression	Représentation
15	0000 1111
$15 \ll 4$	1111 0000
-93	1010 0011
$(15 \ll 4) \oplus (-93)$	$\begin{array}{r} 1111\ 0000 \\ \oplus 1010\ 0011 \\ \hline 0101\ 0011 \end{array}$
$((15 \ll 4) \oplus (-93)) \gg 5$	0000 0010

$(0000\ 0010)_{\pm 2} = 2.$

(b)  $((-17) \ll 3) \wedge (\sim 108) = 16.$

Expression	Représentation
-17	1110 1111
$-17 \ll 3$	0111 1000
108	0110 1100
$\sim 108$	1001 0011
$((-17) \ll 3) \wedge (\sim 108)$	$\begin{array}{r} 0111\ 1000 \\ \wedge 1001\ 0011 \\ \hline 0001\ 0000 \end{array}$

$(0001\ 0000)_{\pm 2} = (0001\ 0000)_2 = 16.$

$$(c) (\sim 38) \vee ((-61) \ll 4) = -7.$$

Expression	Représentation
38	0010 0110
$\sim 38$	1101 1001
-61	1100 0011
$-61 \ll 4$	0011 0000
$(\sim 38) \vee ((-61) \ll 4)$	$\begin{array}{r} 1101\ 1001 \\ \vee\ 0011\ 0000 \\ \hline 1111\ 1001 \end{array}$

$$(1111\ 1001)_{\pm 2} = (1111\ 1001)_2 - 256 = -7.$$

$$(d) ((7 \ll 5) \oplus (-96)) \gg 4 = 4.$$

Expression	Représentation
7	0000 0111
$7 \ll 5$	1110 0000
-96	1010 0000
$(7 \ll 5) \oplus (-96)$	$\begin{array}{r} 1110\ 0000 \\ \oplus\ 1010\ 0000 \\ \hline 0100\ 0000 \end{array}$
$((7 \ll 5) \oplus (-96)) \gg 4$	0000 0100

$$(0000\ 0100)_{\pm 2} = 4.$$

$$(e) ((\sim 102) \gg 2) \wedge ((-33) \gg 3) = -30$$

Expression	Représentation
102	0110 0110
$\sim 102$	1001 1001
$(\sim 102) \gg 2$	1110 0110
-33	1101 1111
$(-33) \gg 3$	1111 1011
$((\sim 102) \gg 2) \wedge ((-33) \gg 3)$	$\begin{array}{r} 1110\ 0110 \\ \wedge\ 1111\ 1011 \\ \hline 1110\ 0010 \end{array}$

$$(1110\ 0010)_{\pm 2} = -30.$$

(f)  $((21 \ll 4) \oplus (-123)) \gg 1 = -22$

Expression	Représentation
21	0001 0101
$21 \ll 4$	0101 0000
-123	1000 0101
$((21 \ll 4) \oplus (-123))$	$\begin{array}{r} 0101\ 0000 \\ \oplus 1000\ 0101 \\ \hline 1101\ 0101 \end{array}$
$((21 \ll 4) \oplus (-123)) \gg 1$	1110 1010

$$(1110\ 1010)_{\pm 2} = (1110\ 1010)_2 - 256 = -22.$$

**Rép. 3.3** (a) **Faux.** On considère  $t = 0000\ 0001$ .

$$(t \gg) \ll = 0000\ 0000 \ll = 0000\ 0000 \neq t.$$

(b) **Vrai.** Étant donné un bit  $b$ , l'opération  $b \oplus 0 = b$  puisque «  $\oplus 0$  » ne modifie pas la valeur de  $b$ . Inversement,  $b \oplus 1 = \sim b$  car «  $\oplus 1$  » inverse la valeur du bit. Or, si on inverse deux fois la valeur d'un bit, on retombe sur sa valeur originale:  $(b \oplus 1) \oplus 1 = \sim \sim b = b$ .

En utilisant ce raisonnement sur chacun des bits de  $t_1$ , on conclut que peu importe la valeur de  $t_2$ ,  $(t_1 \oplus t_2) \oplus t_2 = t_1$ .

(c) **Vrai.** Soit  $t_1 \in T_8$ , on pose  $t_2 = t_1$ . Le résultat de  $t_1 \oplus t_2$  est forcément,  $0000\ 0000$  car  $0 \oplus 0 = 0$  et  $1 \oplus 1 = 0$ .

(d) **Vrai.** Soit  $t_1 \in T_8$ , on pose  $t_2 = \sim t_1$ . Le résultat de  $t_1 \oplus t_2$  est forcément,  $1111\ 1111$  car  $0 \oplus 1 = 1$  et  $1 \oplus 0 = 1$ .

(e) **Faux.** On pose  $t_1 = t_2 = 0100\ 0000$ . On a alors:

$$(t_1 + t_2) \gg 1 = 1000\ 0000 \gg 1 = 1100\ 0000.$$

Ce qui est différent de:

$$(t_1 \gg 1) + (t_2 \gg 1) = 0010\ 0000 + 0010\ 0000 = 0100\ 0000.$$

(f) **Faux.** On pose  $t_1 = t_2 = 1000\ 0000$ . On a alors:

$$(t_1 + t_2) \gg 1 = 0000\ 0000 \gg 1 = 0000\ 0000.$$

Ce qui est différent de:

$$(t_1 \gg 1) + (t_2 \gg 1) = 0100\ 0000 + 0100\ 0000 = 1000\ 0000.$$

(g) **Vrai.** Tout d'abord, on rappelle que le décalage à gauche correspond à une multiplication par 2. S'il n'y a pas de dépassement d'entiers, l'égalité correspond tout simplement à la distributivité de la multiplication sur l'addition:

$$2(t_1 + t_2) = 2t_1 + 2t_2.$$

Même s'il y a un dépassement d'entier, cela est équivalent à effectuer l'opération « mod 256 » immédiatement après chacune des opérations arithmétiques. Le théorème 2.4 garantit que la congruence modulaire est conservée lors des opérations d'addition et de multiplication. Ainsi,

$$2((t_1 + t_2) \bmod 256) \bmod 256 = ((2t_1 \bmod 256) + (2t_2 \bmod 256)) \bmod 256.$$

**Rép. 3.4** Voici une des nombreuses solutions possibles ainsi qu'un exemple d'utilisation pour  $r = 200$ ,  $g = 129$  et  $b = 37$ .

$f(r,g,b) := \text{shift}(r,16) \text{ or } \text{shift}(g,8) \text{ or } b$	<i>Done</i>
$f(200,129,37)$	13140261
$f(200,129,37) \blacktriangleright \text{Base2}$	0b110010001000000100100101

**Rép. 3.5** (a) Tout d'abord, on calcule la représentation binaire de l'adresse :

192	168	17	42
1100 0000	1010 1000	0001 0001	0110 0100
Identifiant du sous-réseau (20 bits)		Identifiant de la machine (12 bits)	

• Identifiant du sous-réseau :

$$\begin{array}{r}
 1100\ 0000 \ . \ 1010\ 1000 \ . \ 0001\ 0001 \ . \ 0010\ 1010 \\
 \wedge \ 1111\ 1111 \ . \ 1111\ 1111 \ . \ 1111\ 0000 \ . \ 0000\ 0000 \\
 \hline
 1100\ 0000 \ . \ 1010\ 1000 \ . \ 0001\ 0000 \ . \ 0000\ 0000
 \end{array}$$

L'identifiant du sous-réseau est donc : **192.168.16.0**.

• Identifiant de la machine

$$\begin{array}{r}
 1100\ 0000 \ . \ 1010\ 1000 \ . \ 0001\ 0001 \ . \ 0010\ 1010 \\
 \wedge \ 0000\ 0000 \ . \ 0000\ 0000 \ . \ 0000\ 1111 \ . \ 1111\ 1111 \\
 \hline
 0000\ 0000 \ . \ 0000\ 0000 \ . \ 0000\ 0001 \ . \ 0010\ 1010
 \end{array}$$

L'identifiant de la machine est donc : **0.0.1.42**

(b) L'identifiant du sous-réseau est **192.168.144.0** et l'identifiant de la machine est **0.0.9.205**.

(c) L'identifiant du sous-réseau est **142.137.240.0** et l'identifiant de la machine est **0.0.8.90**.

## Chapitre 4

**Rép. 4.1** (a)  $f \in O(\sqrt{n})$  (e)  $f \in O(n^4 \log(n))$   
 (b)  $f \in O(n^2)$  (f)  $f \in O(2^n)$   
 (c)  $f \in O(n)$  (g)  $f \in O(5^n)$   
 (d)  $f \in O(n^3)$  (h)  $f \in O(n!)$

**Rép. 4.2** (a)  $O(n^2)$  (d)  $O(n^6)$  (g)  $O(n^{1/10})$  (j)  $O(4^n)$   
 (b)  $O(n^3)$  (e)  $O(n^2)$  (h)  $O(2^n)$  (k)  $O(9^n)$   
 (c)  $O(n^4)$  (f)  $O(n^{5/3})$  (i)  $O(1)$

**Rép. 4.3** (a)  $f(n) \in O(n^{3/2})$  et  $g(n) \in O(n \log(n))$   
 (b) Le deuxième, car,  $g(n) \in O(f(n))$  mais  $f(n) \notin O(g(n))$ .

**Rép. 4.4** (a)  $f(n) \in O(n^2 \log(n))$  et  $g(n) \in O(n^{5/3})$   
 (b) Le deuxième, car,  $g(n) \in O(f(n))$  mais  $f(n) \notin O(g(n))$ .



- Rép. 4.5** (a)  $O(n^4)$  (e)  $O(n^3)$  (i)  $O(n^{10})$  (m)  $O\left(\left(\frac{3}{2}\right)^n\right)$   
 (b)  $O(n^{5/2})$  (f)  $O(4^n)$  (j)  $O(n^{1/2})$  (n)  $O(n^{5/2})$   
 (c)  $O(n^7)$  (g)  $O(2^n)$  (k)  $O(1,001^n)$   
 (d)  $O(2^n)$  (h)  $O(2^n \log(n))$  (l)  $O(n^{1/2})$

**Rép. 4.6** (a)  $(6 + 2 \cdot 2) + (6 + 2 \cdot 3) + (6 + 2 \cdot 4) = 36$

(b)  $f(n) = \sum_{i=2}^{n-2} (n + 2i) = \sum_{i=2}^{n-2} n + \sum_{i=2}^{n-2} 2i.$

On traite les sommations séparément :

$$\sum_{i=2}^{n-2} n = ((n-2) - 2 + 1)n = n^2 - 3n,$$

$$\sum_{i=2}^{n-2} 2i = 2 \sum_{i=2}^{n-2} i = 2 \left( \sum_{i=1}^{n-2} i - \sum_{i=1}^1 i \right) = 2 \left( \frac{(n-2)((n-2)+1)}{2} - \frac{(1)(1+1)}{2} \right) = n^2 - 3n.$$

En regroupant :  $f(n) = (n^2 - 3n) + (n^2 - 3n) = 2n^2 - 6n$ . Ce qui donne bien 36 lorsque  $n = 6$ .

**Rép. 4.7** (a)  $f(n) = \sum_{i=1}^n (n + i + 1) = \sum_{i=1}^n n + \sum_{i=1}^n i + \sum_{i=1}^n 1$

On traite les sommations séparément :

$$\sum_{i=1}^n n = n(n-1+1) = n^2, \quad \sum_{i=1}^n i = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n, \quad \sum_{i=1}^n 1 = (n-1+1) = n.$$

En regroupant :  $f(n) = \frac{3}{2}n^2 + \frac{3}{2}n$

(b)  $f(n) = \sum_{i=0}^{n+1} (5n + 3i + 4) = \sum_{i=0}^{n+1} 5n + \sum_{i=0}^{n+1} 3i + \sum_{i=0}^{n+1} 4.$

On traite les sommations séparément :

$$\sum_{i=0}^{n+1} 5n = 5n(n+1-0+1) = 5n^2 + 10n,$$

$$\sum_{i=0}^{n+1} 3i = 3 \sum_{i=0}^{n+1} i = 3 \left( 0 + \sum_{i=1}^{n+1} i \right) = \frac{3(n+1)(n+2)}{2} = \frac{3}{2}n^2 + \frac{9}{2}n + 3,$$

$$\sum_{i=0}^{n+1} 4 = 4(n+1-0+1) = 4n + 8.$$

En regroupant :  $f(n) = \frac{13}{2}n^2 + \frac{37}{2}n + 11.$

(c) Attention, le  $-n$  ne fait pas partie de la sommation :  $f(n) = \left( \sum_{i=n}^{2n} (i+1) \right) - n = \sum_{i=n}^{2n} i + \sum_{i=n}^{2n} 1 - n.$

On traite les sommations séparément :

$$\sum_{i=n}^{2n} i = \sum_{i=1}^{2n} i - \sum_{i=1}^{n-1} i = \frac{2n(2n+1)}{2} - \frac{(n-1)((n-1)+1)}{2} = (2n^2 + n) - \left( \frac{1}{2}n^2 - \frac{1}{2}n \right) = \frac{3}{2}n^2 + \frac{3}{2}n,$$

$$\sum_{i=n}^{2n} 1 = 2n - n + 1 = n + 1.$$

En regroupant :  $f(n) = \left( \frac{3}{2}n^2 + \frac{3}{2}n \right) + (n+1) - n = \frac{3}{2}n^2 + \frac{3}{2}n + 1.$

(d)  $\frac{4^n - 16}{3}$

(e)  $f(n) = \sum_{i=1}^{2n-1} \sum_{j=0}^{n-1} (j+1) = \sum_{i=1}^{2n-1} \left( \sum_{j=0}^{n-1} j + \sum_{j=0}^{n-1} 1 \right) = \sum_{i=1}^{2n-1} \left( \sum_{j=1}^{n-1} j + n \right) = \sum_{i=1}^{2n-1} \left( \frac{n(n-1)}{2} + n \right) = \frac{1}{2} \sum_{i=1}^{2n-1} n^2 +$

$$\frac{1}{2} \sum_{i=1}^{2n-1} n.$$

On traite les sommations séparément :

$$\frac{1}{2} \sum_{i=1}^{2n-1} n^2 = n^2(2n-1-1+1) = n^3 - \frac{1}{2}n^2.$$

$$\frac{1}{2} \sum_{i=1}^{2n-1} n = n(2n-1-1+1) = n^2 - \frac{1}{2}n.$$

$$\text{En regroupant : } f(n) = n^3 + \frac{1}{2}n^2 - \frac{1}{2}n.$$

$$(f) \quad f(n) = \sum_{i=0}^{n-1} \sum_{j=1}^n (n^2 + i + j) = \sum_{i=0}^{n-1} \left( \sum_{j=1}^n (n^2 + i) + \sum_{j=1}^n j \right)$$

On traite les sommations internes séparément :

$$\sum_{j=1}^n (n^2 + i) = (n^2 + i)n = n^3 + in$$

$$\sum_{j=1}^n j = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$$

En regroupant :

$$\begin{aligned} f(n) &= \sum_{i=0}^{n-1} \left( n^3 + in + \frac{1}{2}n^2 + \frac{1}{2}n \right) = \sum_{i=0}^{n-1} \left( n^3 + \frac{1}{2}n^2 + \frac{1}{2}n \right) + \sum_{i=0}^{n-1} in \\ &= n^4 + \frac{1}{2}n^3 + \frac{1}{2}n^2 + n \sum_{i=1}^{n-1} i = n^4 + \frac{1}{2}n^3 + \frac{1}{2}n^2 + n \frac{n(n-1)}{2} \\ &= n^4 + n^3. \end{aligned}$$

$$\begin{aligned} (g) \quad f(n) &= \sum_{k=0}^n \left( \sum_{l=1}^{n^2} (k-l) + k^2 \right) = \sum_{k=0}^n \left( \sum_{l=1}^{n^2} k - \sum_{l=1}^{n^2} l + k^2 \right) \\ &= \sum_{k=0}^n \left( kn^2 - \frac{n^2(n^2+1)}{2} + k^2 \right) = n^2 \sum_{k=1}^n k - \sum_{k=0}^n \frac{n^2(n^2+1)}{2} + \sum_{k=1}^n k^2 \\ &= n^2 \frac{n(n+1)}{2} - \frac{n^2(n^2+1)}{2}(n+1) + \frac{n(n+1)(2n+1)}{6} \\ &= \left( \frac{1}{2}n^4 + \frac{1}{2}n^3 \right) - \left( \frac{1}{2}n^5 + \frac{1}{2}n^4 + \frac{1}{2}n^3 + \frac{1}{2}n^2 \right) + \left( \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n \right) \\ &= -\frac{1}{2}n^5 + \frac{1}{3}n^3 + \frac{1}{6}n. \end{aligned}$$

$$(h) \quad \frac{5^{n+1} - 5^{10}}{2}$$

**Rép. 4.8** Soit  $f(n)$  le nombre d'additions effectuées par l'algorithme en fonction de  $n$ ,

$$\begin{aligned} f(n) &= \sum_{i=2}^n \left( 2 + \left( \sum_{j=1}^{n-i} 1 \right) + 1 \right) \\ &= \sum_{i=2}^n (n-i+3) \\ &= \frac{1}{2}n^2 + \frac{3}{2}n - 2. \end{aligned}$$

On conclut que  $f(n) \in O(n^2)$ .

**Rép. 4.9** La fonction  $\mathbb{P}$  détermine si l'élément  $x$  est présent dans les 10 premières positions du tableau d'éléments  $T$ . Elle requiert  $f(n) = 10$  comparaisons. On a donc  $f(n) \in O(1)$ .

**Rép. 4.10** La fonction  $\mathbb{P}$  requiert  $f(n) = 50n + \frac{n(n+1)}{2}$  comparaisons pour traiter un tableau de taille  $n$ . On a donc  $f(n) \in O(n^2)$ .

**Rép. 4.11** La fonction  $\mathbb{P}$  requiert  $f(n) = \frac{n^2(n+1)}{2} + 5n$  comparaisons pour traiter un tableau de taille  $n$ . On a donc  $f(n) \in O(n^3)$ .

**Rép. 4.12** (a) Compter le nombre de comparaisons d'éléments à la ligne 4;  $f(n) = \sum_{i=0}^{n-1} 1 = n \in O(n)$ .

(b) Compter le nombre de comparaisons d'éléments à la ligne 5;  $f(n) = \sum_{i=0}^{n-1} 1 = n \in O(n)$ .

(c) Compter le nombre de comparaisons d'éléments à la ligne 4;  $f(n) = \sum_{i=0}^{n-1} 1 = n \in O(n)$ .

(d) Compter le nombre de comparaisons de cases à la ligne 5;  $f(n) = \sum_{i=0}^{n-2} \left( \sum_{j=i+1}^{n-1} 1 \right) = \frac{1}{2}n^2 - \frac{1}{2}n \in O(n^2)$ .

(e) Compter le nombre de comparaisons de cases à la ligne 5;  $f(n) = \sum_{k=0}^{2n-2} 1 = 2n - 1 \in O(n)$ .

Note : la borne supérieure est,  $2n - 2$  car il ne peut pas y avoir de comparaison effectuée à la dernière itération de la boucle. En effet, à la dernière itération, on a forcément  $i == n_1$  ou  $j == n_2$ .

(f) Compter le nombre de comparaisons de caractères à la ligne 8;  $f(n) = \sum_{i=0}^{n/2} \left( \sum_{j=0}^{n/2-1} 1 \right) = \frac{1}{4}n^2 + \frac{1}{2}n \in O(n^2)$ .

## Chapitre 5

**Rép. 5.1** (a)  $f(1) = 2, f(2) = 4, f(3) = 6, f(4) = 8, f(5) = 10, f(6) = 12, f(7) = 14$ .

Le terme  $f(n)$  est le  $n$ -ième nombre pair. De manière équivalente,  $f(n) = 2n$ .

(b)  $f(1) = 3, f(2) = 5, f(3) = 7, f(4) = 9, f(5) = 11, f(6) = 13, f(7) = 15$ .

Le terme  $f(n)$  est le  $n$ -ième nombre impair. De manière équivalente,  $f(n) = 2n + 1$ .

(c)  $f(1) = 1, f(2) = 0, f(3) = 1, f(4) = 0, f(5) = 1, f(6) = 0, f(7) = 1$ .

Le terme  $f(n)$  est 0 si  $n$  est pair, 1 si  $n$  est impair. En termes plus techniques, on dit que  $f$  est la *fonction caractéristique* des nombres impairs.

(d)  $f(1) = 1, f(2) = 4, f(3) = 9, f(4) = 16, f(5) = 25, f(6) = 36, f(7) = 49$ .

Le terme  $f(n)$  est le  $n$ -ième carré. De manière équivalente  $f(n) = n^2$ .

(e)  $f(1) = 1, f(2) = 11, f(3) = 111, f(4) = 1111, f(5) = 11111, f(6) = 111111, f(7) = 1111111$ .

Le terme  $f(n)$  est le nombre obtenu en concaténant  $n$  occurrences de 1, en base 10. De manière équivalente,  $f(n) = \sum_{i=0}^{n-1} 10^i = \frac{10^n - 1}{9}$ , ou encore  $f(n) = \underbrace{11111 \dots 111}_{n \text{ fois}}$ .

(f)  $f(1) = 0, f(2) = 1, f(3) = 1, f(4) = 0, f(5) = 1, f(6) = 0, f(7) = 1$ .

La fonction  $f$  est la *fonction caractéristique* des nombres premiers.

**Rép. 5.2** (a) À l'aide de la méthode itérative, on trouve

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 2a_0 \\ a_2 &= 2a_1 = 2(2a_0) = 2^2 a_0 \\ a_3 &= 2a_2 = 2(2^2 a_0) = 2^3 a_0 \\ &\vdots \\ a_n &= 2^n a_0. \end{aligned}$$

En utilisant  $a_0 = 1$  donné par la définition, on obtient la solution:  $a_n = 2^n$ .

(b) À l'aide de la méthode itérative, on trouve

$$\begin{aligned} b_0 &= 0 \\ b_1 &= b_0 + 2 \cdot 1 \\ b_2 &= b_1 + 2 \cdot 2 = (b_0 + 2 \cdot 1) + 2 \cdot 2 \\ b_3 &= b_2 + 2 \cdot 3 = (b_0 + 2 \cdot 1 + 2 \cdot 2) + 2 \cdot 3 \\ &\vdots \\ b_n &= b_0 + \sum_{i=1}^n 2i = b_0 + 2 \sum_{i=1}^n i = b_0 + 2 \left( \frac{n(n+1)}{2} \right) = b_0 + n^2 + n. \end{aligned}$$

En utilisant  $b_0 = 0$  donné par la définition, on obtient la solution:  $b_n = n^2 + n$ .

(c) À l'aide de la méthode itérative, on trouve

$$\begin{aligned} c_0 &= 1 \\ c_1 &= c_0 + 1 + 2 \\ c_2 &= c_1 + 2 + 2 = (c_0 + 1 + 2) + 2 + 2 \\ c_3 &= c_2 + 3 + 2 = (c_0 + 1 + 2 + 2 + 2) + 3 + 2 \\ &\vdots \\ c_n &= c_0 + \sum_{i=1}^n (i + 2) = c_0 + \sum_{i=1}^n i + \sum_{i=1}^n 2 = c_0 + \frac{n(n+1)}{2} + 2n = c_0 + \frac{n^2}{2} + \frac{5n}{2}. \end{aligned}$$

En utilisant  $c_0 = 1$  donné par la définition, on obtient la solution:  $c_n = 1 + \frac{n^2}{2} + \frac{5n}{2}$ .

(d) À l'aide de la méthode itérative, on trouve

$$\begin{aligned} f(0) &= 2 \\ f(1) &= 3f(0) + 2 \\ f(2) &= 3f(1) + 2 = 3(3f(0) + 2) + 2 = 3^2 f(0) + 3 \cdot 2 + 2 \\ f(3) &= 3f(2) + 2 = 3(3^2 f(0) + 3 \cdot 2 + 2) + 2 = 3^3 f(0) + 3^2 \cdot 2 + 3 \cdot 2 + 2 \\ &\vdots \\ f(n) &= 3^n f(0) + \sum_{i=0}^{n-1} 3^i \cdot 2 = 3^n f(0) + 2 \left( \frac{3^n - 1}{2} \right) = 3^n f(0) + 3^n - 1 = 3^n (f(0) + 1) - 1. \end{aligned}$$

En utilisant  $f(0) = 2$  donné par la définition, on obtient la solution:  $f(n) = 3^n(2 + 1) - 1 = 3^{n+1} - 1$ .

(e) À l'aide de la méthode itérative, on trouve

$$f(0) = 5$$

$$f(1) = 3f(0) + 2$$

$$f(2) = 3f(1) + 2 = 3(3f(0) + 2) + 2 = 3^2 f(0) + 3 \cdot 2 + 2$$

$$f(3) = 3f(2) + 2 = 3(3^2 f(0) + 3 \cdot 2 + 2) + 2 = 3^3 f(0) + 3^2 \cdot 2 + 3 \cdot 2 + 2$$

$$\vdots$$

$$f(n) = 3^n f(0) + \sum_{i=0}^{n-1} 3^i \cdot 2 = 3^n f(0) + 2 \left( \frac{3^n - 1}{2} \right) = 3^n f(0) + 3^n - 1 = 3^n (f(0) + 1) - 1.$$

En utilisant  $f(0) = 5$  donné par la définition, on obtient la solution:  $f(n) = 3^n(5 + 1) - 1 = 6 \cdot 3^n - 1$ .

(f) À l'aide de la méthode itérative, on trouve

$$f(0) = 5$$

$$f(1) = 1 \cdot f(0)$$

$$f(2) = 2 \cdot f(1) = 2(1 \cdot f(0)) = 2 \cdot 1 \cdot f(0)$$

$$f(3) = 3 \cdot f(2) = 3(2 \cdot 1 \cdot f(0)) = 3 \cdot 2 \cdot 1 \cdot f(0)$$

$$\vdots$$

$$f(n) = n \cdots 3 \cdot 2 \cdot 1 \cdot f(0).$$

En utilisant  $f(0) = 5$  donné par la définition, on obtient la solution:  $f(n) = n! \cdot 5$ .

(g) À l'aide de la méthode itérative, on trouve

$$h_0 = 3$$

$$h_1 = 10h_0$$

$$h_2 = 10h_1 = 10(10h_0) = 10^2 h_0$$

$$h_3 = 10h_2 = 10(10^2 h_0) = 10^3 h_0$$

$$\vdots$$

$$h_n = 10^n h_0.$$

En utilisant  $h_0 = 3$  donné par la définition, on obtient la solution:  $h_n = 3 \cdot 10^n$ .

(h) À l'aide de la méthode itérative, on trouve

$$f(2) = 3$$

$$f(3) = 4f(2) + 6$$

$$f(4) = 4f(3) + 6 = 4(4f(2) + 6) + 6 = 4^2 f(2) + 4 \cdot 6 + 6$$

$$f(5) = 4f(4) + 6 = 4(4^2 f(2) + 4 \cdot 6 + 6) + 6 = 4^3 f(2) + 4^2 \cdot 6 + 4 \cdot 6 + 6$$

$$\vdots$$

$$f(n) = 4^{n-2} f(2) + \sum_{i=0}^{n-3} 4^i \cdot 6 = 4^{n-2} f(2) + 6 \left( \frac{4^{n-2} - 1}{4 - 1} \right) = 4^{n-2} f(2) + 2 \cdot 4^{n-2} - 2 = 4^{n-2} (f(2) + 2) - 2.$$

En utilisant  $f(2) = 3$ , on obtient la solution:  $f(n) = 4^{n-2} (3 + 2) - 2 = 5 \cdot 4^{n-2} - 2$ .

- Rép. 5.3** (a)  $f(128) = 8$   
 (b)  $f(81) = 8463$   
 (c)  $f(64) = 24512$   
 (d)  $f(729) = 5\,314\,410$

- Rép. 5.4** (a)  $f(n) = 6n - 3 \in O(n)$ .  
 (b)  $f(n) = 2n^2 - n \in O(n^2)$ .  
 (c)  $f(n) = 2n \log_3(n) + \frac{3n}{2} - \frac{1}{2} \in O(n \log(n))$ .  
 (d)  $f(n) = 2n^2 + 3n^2 \log_3(n) \in O(n^2 \log(n))$ .

**Rép. 5.5**  $f(n) = (d-1) \log_d(n) + 1 \in O(\log(n))$

- Rép. 5.6** (a)  $f(n) = 2f(n/2) + n - 1$ ,  $f(1) = 0$ .  
 (b)  $f(n) = n \cdot \log_2(n) - n + 1 \in O(n \log(n))$ .

## Chapitre 6

- Rép. 6.1** (a) On définit  $P(n)$  : «  $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$  » et on veut montrer que  $\forall n \geq 1$ ,  $P(n) \equiv$  **vrai**.

**1. Cas de base :**

Pour  $n = 1$ , on a  $P(1) \equiv$  **vrai**, car  $\sum_{i=1}^1 i^2 = 1$ .

**2. Étape de récurrence :**

Soit  $k \geq 1$ .

– Hypothèse de récurrence :  $P(k)$  : «  $\sum_{i=1}^k i^2 = \frac{k(k+1)(2k+1)}{6}$  »  $\equiv$  **vrai**

– Objectif :  $P(k+1)$  : «  $\sum_{i=1}^{k+1} i^2 = \frac{(k+1)(k+2)(2k+3)}{6}$  »  $\equiv$  **vrai**

– Calculs :

$$\begin{aligned} \sum_{i=1}^{k+1} i^2 &= \sum_{i=1}^k i^2 + (k+1)^2 \\ &= \frac{k(k+1)(2k+1)}{6} + (k+1)^2 && \text{(par hyp. de réc.)} \\ &= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6} \\ &= \frac{(k+1)(k(2k+1) + 6(k+1))}{6} \\ &= \frac{(k+1)(2k^2 + 7k + 6)}{6} \\ &= \frac{(k+1)(k+2)(2k+3)}{6} \end{aligned}$$

□

- (b) On définit  $P(n)$  : «  $\sum_{i=0}^n r^i = \frac{r^{n+1} - 1}{r - 1}$ , si  $r \in \mathbb{R} \setminus \{1\}$  » et on veut montrer que  $\forall n \geq 0$ ,  $P(n) \equiv$  **vrai**.

**1. Cas de base :**

Pour  $n = 0$ , on a  $P(0) \equiv$  **vrai**, car  $\sum_{i=0}^0 r^i = 1 = \frac{r-1}{r-1}$ .

2. **Étape de récurrence:**

Soit  $k \geq 0$ .

- Hypothèse de récurrence:  $P(k)$ : «  $\sum_{i=0}^k r^i = \frac{r^{k+1}-1}{r-1}$ , si  $r \in \mathbb{R} \setminus \{1\}$  »  $\equiv$  **vrai**
- Objectif:  $P(k+1)$ : «  $\sum_{i=0}^{k+1} r^i = \frac{r^{k+2}-1}{r-1}$ , si  $r \in \mathbb{R} \setminus \{1\}$  »  $\equiv$  **vrai**
- Calculs:

$$\begin{aligned} \sum_{i=0}^{k+1} r^i &= \sum_{i=0}^k r^i + r^{k+1} \\ &= \frac{r^{k+1}-1}{r-1} + r^{k+1} && \text{(par hyp. de réc.)} \\ &= \frac{r^{k+1}-1 + r^{k+1}(r-1)}{r-1} \\ &= \frac{r^{k+1}(1+(r-1))-1}{r-1} \\ &= \frac{r^{k+1}r-1}{r-1} \\ &= \frac{r^{k+2}-1}{r-1} \quad \square \end{aligned}$$

- (c) Remarque: il est difficile d'utiliser directement un énoncé de la forme « 3 divise ( $n^3 - n$ ) » dans le cadre d'une preuve formelle. On va plutôt utiliser la définition mathématique:

$$\exists a \in \mathbb{Z}, \quad n^3 - n = 3a.$$

On définit  $P(n)$ : «  $\exists a \in \mathbb{Z}, \quad n^3 - n = 3a$  » et on veut montrer que  $\forall n \geq 0, P(n) \equiv$  **vrai**.

1. **Cas de base:**

Pour  $n = 0$ , on a  $P(0) \equiv$  **V**, car  $0^3 - 0 = 0 = 3 \cdot 0$ .

2. **Étape de récurrence:**

Soit  $k \geq 0$ .

- Hypothèse de récurrence:  $P(k)$ : «  $\exists a \in \mathbb{Z}, \quad k^3 - k = 3a$  »  $\equiv$  **vrai**.
- Objectif:  $P(k+1)$ : «  $\exists a \in \mathbb{Z}, \quad (k+1)^3 - (k+1) = 3a$  »  $\equiv$  **vrai**.
- Calculs:

$$\begin{aligned} (k+1)^3 - (k+1) &= (k^3 + 3k^2 + 3k + 1) - (k+1) \\ &= (k^3 - k) + 3k^2 + 3k \\ &= 3a + 3k^2 + 3k && \text{(par hyp. de réc.)} \\ &= 3(a + k^2 + k) \\ &= 3b && \text{(où } b = a + k^2 + k \in \mathbb{Z}) \end{aligned}$$

□

- (d) Remarque: il est difficile d'utiliser directement un énoncé de la forme « 2 divise ( $n^2 + 3n$ ) » dans le cadre d'une preuve formelle. On va plutôt utiliser la définition mathématique:

$$\exists a \in \mathbb{Z}, \quad n^2 + 3n = 2a.$$

On définit  $P(n)$ : «  $\exists a \in \mathbb{Z}, \quad n^2 + 3n = 2a$  » et on veut montrer que  $\forall n \geq 0, P(n) \equiv$  **vrai**.

1. **Cas de base:**

Pour  $n = 0$ , on a  $P(0) \equiv$  **V**, car  $0^2 + 3 \cdot 0 = 0 = 2 \cdot 0$ .

2. **Étape de récurrence :**Soit  $k \geq 0$ .

- Hypothèse de récurrence:  $P(k)$ : «  $\exists a \in \mathbb{Z}, k^2 + 3k = 2a$  »  $\equiv$  **vrai**.
- Objectif:  $P(k+1)$ : «  $\exists a \in \mathbb{Z}, (k+1)^2 + 3(k+1) = 2a$  »  $\equiv$  **vrai**.
- Calculs:

$$\begin{aligned}
 (k+1)^2 + 3(k+1) &= (k^2 + 2k + 1) + 3(k+1) \\
 &= (k^2 + 3k) + 2k + 4 \\
 &= 2a + 2k + 4 && \text{(par hyp. de réc.)} \\
 &= 2(a + k + 2) \\
 &= 2b && \text{(où } b = a + k + 2 \in \mathbb{Z}\text{)}
 \end{aligned}$$

□

(e) On définit  $P(n)$ : «  $2n + 1 \leq 2^n$  » et on veut montrer que  $\forall n \geq 3, P(n) \equiv$  **vrai**.1. **Cas de base :**Pour  $n = 3$ , on a  $P(3) \equiv$  **vrai**, car  $2 \cdot 3 + 1 = 7, 2^3 = 8$ , et donc «  $2 \cdot 3 + 1 \leq 2^3$  »  $\equiv$  **vrai**.2. **Étape de récurrence :**Soit  $k \geq 3$ .

- Hypothèse de récurrence:  $P(k)$ : «  $2k + 1 \leq 2^k$  »  $\equiv$  **vrai**.
- Objectif:  $P(k+1)$ : «  $2(k+1) + 1 \leq 2^{k+1}$  »  $\equiv$  **vrai**.
- Calculs:

$$\begin{aligned}
 2(k+1) + 1 &= (2k + 1) + 2 \leq 2^k + 2 && \text{(par hyp. de réc.)} \\
 &\leq 2^k + 2^k && \text{(comme } 2 \leq 2^k \text{ pour tout } k \geq 1\text{)} \\
 &= 2 \cdot 2^k \\
 &= 2^{k+1}
 \end{aligned}$$

□

(f) On définit  $P(n)$ : «  $n^2 \leq 2^n$  » et on veut montrer que  $\forall n \geq 4, P(n) \equiv$  **vrai**.1. **Cas de base :**Pour  $n = 4$ , on a  $P(4) \equiv$  **vrai**, car  $4^2 = 16, 2^4 = 16$ , et donc «  $4^2 \leq 2^4$  »  $\equiv$  **vrai**.2. **Étape de récurrence :**Soit  $k \geq 4$ .

- Hypothèse de récurrence:  $P(k)$ : «  $k^2 \leq 2^k$  »  $\equiv$  **vrai**.
- Objectif:  $P(k+1)$ : «  $(k+1)^2 \leq 2^{k+1}$  »  $\equiv$  **vrai**.
- Calculs:

$$\begin{aligned}
 (k+1)^2 &= k^2 + (2k + 1) \leq 2^k + 2^k && \text{(par hyp. de réc. et exercice 6.1 (e))} \\
 &= 2 \cdot 2^k \\
 &= 2^{k+1}
 \end{aligned}$$

□

(g) On définit  $P(n)$ : «  $2^n < n!$  » et on veut montrer que  $\forall n \geq 4, P(n) \equiv$  **vrai**.1. **Cas de base :**Pour  $n = 4$ , on a  $P(4) \equiv$  **vrai**, car  $2^4 = 16, 4! = 24$ , et donc «  $2^4 < 4!$  »  $\equiv$  **vrai**.2. **Étape de récurrence :**Soit  $k \geq 4$ .

- Hypothèse de récurrence:  $P(k)$ : «  $2^k < k!$  »  $\equiv$  **vrai**.
- Objectif:  $P(k+1)$ : «  $2^{k+1} < (k+1)!$  »  $\equiv$  **vrai**.



– Calculs:

$$\begin{aligned} 2^{k+1} &= 2 \cdot 2^k < (k+1) \cdot 2^k && \text{(comme } k \geq 4 \text{ on a que } 2 < k+1) \\ &< (k+1) \cdot k! && \text{(par hyp. de réc.)} \\ &< (k+1)! && \text{(par la déf. de la factorielle)} \end{aligned}$$

□

(h) On définit  $P(n)$  : «  $n! < n^n$  » et on veut montrer que  $\forall n \geq 2, P(n) \equiv \text{vrai}$ .

1. **Cas de base:**

Pour  $n = 2$ , on a  $P(2) \equiv \text{vrai}$ , car  $2! = 2, 2^2 = 4$ , et donc «  $2! < 2^2$  »  $\equiv \text{vrai}$ .

2. **Étape de récurrence:**

Soit  $k \geq 2$ .

– Hypothèse de récurrence:  $P(k)$  : «  $k! < k^k$  »  $\equiv \text{vrai}$ .

– Objectif:  $P(k+1)$  : «  $(k+1)! < (k+1)^{(k+1)}$  »  $\equiv \text{vrai}$ .

– Calculs:

$$\begin{aligned} (k+1)! &= k!(k+1) < k^k(k+1) && \text{(par hyp. de réc.)} \\ &< (k+1)^k(k+1) && \text{(comme } k < (k+1)) \\ &< (k+1)^{(k+1)} && \text{(par la déf. de la factorielle)} \end{aligned}$$

□

(i) On définit la fonction propositionnelle  $P(n)$  :  $\overline{A_1 \cup A_2 \cup \dots \cup A_n} = \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}$  et on montre que  $\forall n \geq 2, P(n) \equiv \text{vrai}$ .

1. **Cas de base:**

$P(2) \equiv \text{vrai}$  car par la loi de De Morgan,  $\overline{A_1 \cup A_2} = \overline{A_1} \cap \overline{A_2}$ .

2. **Étape de récurrence:**

Soit  $k \geq 2$ .

– Hypothèse de récurrence:  $P(k)$  : «  $\overline{A_1 \cup A_2 \cup \dots \cup A_k} = \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_k}$  »  $\equiv \text{vrai}$ .

– Objectif:  $P(k+1)$  : «  $\overline{A_1 \cup A_2 \cup \dots \cup A_{k+1}} = \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_{k+1}}$  »  $\equiv \text{vrai}$ .

– Calculs:

$$\begin{aligned} \overline{A_1 \cup A_2 \cup \dots \cup A_{k+1}} &= \overline{(A_1 \cup A_2 \cup \dots \cup A_k) \cup A_{k+1}} \\ &= \overline{A_1 \cup A_2 \cup \dots \cup A_k} \cap \overline{A_{k+1}} && \text{(par la loi de De Morgan)} \\ &= \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_k} \cap \overline{A_{k+1}} && \text{(par hyp. de réc.)} \end{aligned}$$

□

(j) On définit  $P(n)$  : « l'échiquier  $2^n \times 2^n$  dont une case est occupée par la reine est pavable par des triominos en forme de  $L$  » et on montre que  $\forall n \in \mathbb{N}, P(n) \equiv \text{vrai}$ .

1. **Cas de base:**

Pour  $n = 0$ , on a  $P(0) \equiv \text{vrai}$  car paver un échiquier de taille  $1 \times 1$  dont l'unique case est occupée par la reine est trivial, il n'y a rien à faire:



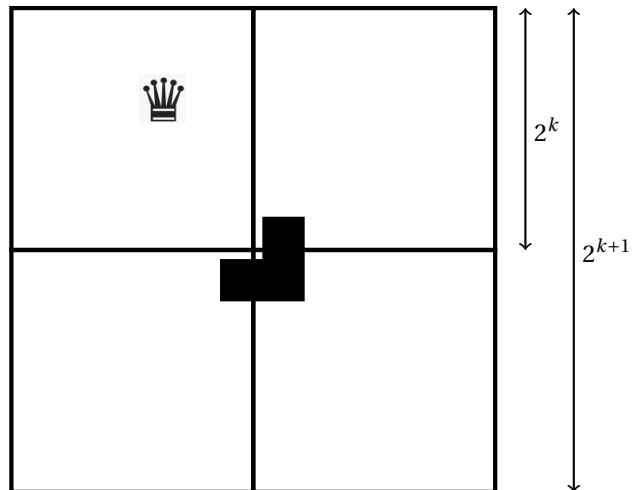
2. **Étape de récurrence:**

Soit  $k \geq 0$ .

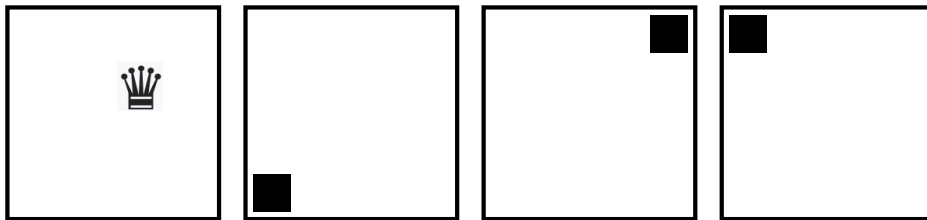
– Hypothèse de récurrence:  $P(k)$  : « l'échiquier  $2^k \times 2^k$  dont une case est occupée par la reine est pavable par des triominos en forme de  $L$  »  $\equiv \text{vrai}$ .

– Objectif:  $P(k+1)$  : « l'échiquier  $2^{k+1} \times 2^{k+1}$  dont une case est occupée par la reine est pavable par des triominos en forme de  $L$  »  $\equiv \text{vrai}$ .

– Argumentation: on considère un échiquier  $2^{k+1} \times 2^{k+1}$  dont une case est occupée par la reine. Comme la hauteur et la largeur sont des nombres pairs, on peut diviser l'échiquier en quatre échiquiers de taille  $2^k \times 2^k$ . Sans perte de généralité, on suppose que la reine est située dans le quart supérieur gauche. On positionne un triomino en forme de  $L$  de manière à ce qu'il couvre exactement une case dans chacun des trois échiquiers restants.



Il reste à montrer qu'il est possible de paver les 4 échiquiers  $2^k \times 2^k$  suivants avec des triominos en forme de  $L$ .



Chacun de ces quatre échiquiers a exactement une case qui est occupée. On a donc que, par **hypothèse de récurrence**, ces quatre échiquiers peuvent être pavés par des triominos en forme de  $L$ , ce qui conclut la preuve.  $\square$

**Rép. 6.2** (a)  $n^2$

(b) On définit  $P(n)$  : «  $\sum_{i=1}^n (2i - 1) = n^2$  » et on montre que  $\forall n \geq 1, P(n) \equiv \text{vrai}$ .

1. **Cas de base :**

Pour  $n = 1$ , on a  $P(1) \equiv \text{vrai}$ , car  $\sum_{i=1}^1 (2i - 1) = 1 = 1^2$ .

2. **Étape de récurrence :**

Soit  $k \geq 1$ .

– Hypothèse de récurrence :  $P(k)$  : «  $\sum_{i=1}^k (2i - 1) = k^2$  »  $\equiv \text{vrai}$ .

– Objectif :  $P(k + 1)$  : «  $\sum_{i=1}^{k+1} (2i - 1) = (k + 1)^2$  »  $\equiv \text{vrai}$ .

– Calculs :

$$\begin{aligned} \sum_{i=1}^{k+1} (2i - 1) &= \sum_{i=1}^k (2i - 1) + (2(k + 1) - 1) \\ &= k^2 + (2(k + 1) - 1) && \text{(par hyp. de réc.)} \\ &= k^2 + 2k + 1 \\ &= (k + 1)^2 \end{aligned}$$

$\square$

**Rép. 6.3** Soit  $f(n) = 2n^2 + 2n + 5$ . On définit la fonction propositionnelle  $P(n) : \langle b_n = f(n) \rangle$  et on montre que  $\forall n \in \mathbb{N}, P(n) \equiv \text{vrai}$ .

1. **Cas de base :**

$P(0)$  est l'affirmation «  $b_0 = f(0)$  ».

Par définition de la suite  $b$ , on a que  $b_0 = 5$ . Par ailleurs,  $f(0) = 2 \cdot 0^2 + 2 \cdot 0 + 5 = 5$ . Donc  $P(0) \equiv \text{vrai}$ .

2. **Étape de récurrence :**

– Hypothèse de récurrence:  $P(k) : \langle b_k = f(k) \rangle \equiv \text{vrai}$ . On suppose donc que  $b_k = 2k^2 + 2k + 5$ .

– Objectif:  $P(k+1) : \langle b_{k+1} = f(k+1) \rangle \equiv \text{vrai}$ .

– Calculs :

$$f(k+1) = 2(k+1)^2 + 2(k+1) + 5 \quad (\text{par définition de } f)$$

$$= 2(k^2 + 2k + 1) + 2(k+1) + 5 \quad (\text{par dév. algébrique})$$

$$= 2k^2 + 6k + 9.$$

et

$$b_{k+1} = b_k + 4(k+1) \quad (\text{par la relation de récurrence qui définit la suite } b)$$

$$= (2k^2 + 2k + 5) + 4(k+1) \quad (\text{par l'hypothèse de récurrence})$$

$$= 2k^2 + 6k + 9. \quad \square$$

**Rép. 6.4** Soit  $g(n) = 3^{n+1} - 1$ . On définit la fonction propositionnelle  $P(n) : \langle f(n) = g(n) \rangle$  et on montre que  $\forall n \in \mathbb{N}, P(n) \equiv \text{vrai}$ .

1. **Cas de base :**

$P(0)$  est l'affirmation «  $f(0) = g(0)$  ».

Par définition de la fonction  $f$ , on a que  $f(0) = 2$ . Par ailleurs,  $g(0) = 3 - 1 = 2$ . Donc  $P(0) \equiv \text{vrai}$ .

2. **Étape de récurrence :**

Soit  $k \geq 0$ .

– Hypothèse de récurrence:  $P(k) : \langle f(k) = g(k) \rangle \equiv \text{vrai}$ . On suppose donc que  $f(k) = 3^{k+1} - 1$ .

– Objectif:  $P(k+1) : \langle f(k+1) = g(k+1) \rangle \equiv \text{vrai}$ . Il faut donc vérifier que  $f(k+1) = 3^{k+2} - 1$ .

– Calculs :

$$f(k+1) = 3f(k) + 2 \quad (\text{par la relation de récurrence qui définit la fonction } f)$$

$$= 3 \cdot (3^{k+1} - 1) + 2 \quad (\text{par l'hypothèse de récurrence})$$

$$= 3 \cdot 3^{k+1} - 3 + 2$$

$$= 3^{k+2} - 1 \quad \square$$

**Rép. 6.5** (a) Soit  $k \geq 0$ .

– Hypothèse: On suppose  $P(k)$ , il existe  $a \in \mathbb{Z}$  tel que  $k^3 + 2k + 1 = 3a$ .

– Objectif:  $P(k+1) \equiv \text{vrai}$ , ce qui revient à montrer qu'il existe  $a' \in \mathbb{Z}$  tel que  $(k+1)^3 + 2(k+1) + 1 = 3a'$ .

– Calculs :

$$(k+1)^3 + 2(k+1) + 1 = k^3 + 2k + 1 + 3k^2 + 3k + 3$$

$$= 3a + 3k^2 + 3k + 3 \quad (\text{par hypothèse})$$

$$= 3(a + k^2 + k + 1)$$

$$= 3a' \quad (\text{avec } a' = a + k^2 + k + 1 \in \mathbb{Z}) \quad \square$$

(b) En (a), on a montré l'étape de récurrence «  $P(k) \rightarrow P(k+1)$  ». Cela veut dire que si, pour un certain entier  $n_0$ ,  $P(n_0) \equiv \text{vrai}$ , alors l'énoncé est vrai pour tous les entiers suivants. Hors, il n'existe aucun entier  $n$  pour lequel  $P(n)$  est vrai, ainsi cette fonction propositionnelle est fautive pour tout  $n \in \mathbb{Z}$ .

**Conclusion : sans cas de base, l'étape récursive n'a aucune valeur.**

**Rép. 6.6** On définit la fonction propositionnelle  $P(n)$  : « Si  $f(x) = x^n$  alors  $f'(x) = nx^{n-1}$  » et on montre que  $\forall n \geq 1, P(n) \equiv \text{vrai}$ .

1. **Cas de base :**

$P(1)$  est l'affirmation « Si  $f(x) = x^1$  alors  $f'(x) = 1x^0 = 1$  ».

Donc  $P(1) \equiv \text{vrai}$ .

2. **Étape de récurrence :**

Soit  $k \geq 1$ .

– Hypothèse:  $P(k)$  : « Si  $f(x) = x^k$  alors  $f'(x) = kx^{k-1}$  »  $\equiv \text{vrai}$

– Objectif:  $P(k+1)$  « Si  $f(x) = x^{k+1}$ , alors  $f'(x) = (k+1)x^k$  »  $\equiv \text{vrai}$ .

– Calculs :

$$\begin{aligned} [x^{k+1}]' &= [x^k x]' \\ &= [x^k]' \cdot x + x^k \cdot [x]' && \text{(par la règle de la dérivée d'un produit de fonctions)} \\ &= kx^{k-1} \cdot x + x^k \cdot 1 && \text{(par hypothèse de récurrence)} \\ &= kx^k + x^k \\ &= (k+1)x^k \end{aligned}$$

□

**Rép. 6.7** On définit  $P(n)$  : « La dérivée  $n^{\text{ième}}$  de  $f(x) = \ln(x)$  est  $f^{(n)}(x) = \frac{(-1)^{(n-1)}(n-1)!}{x^n}$  » et on montre que  $\forall n \geq 1, P(n) \equiv \text{vrai}$ .

1. **Cas de base :**

$P(1)$  est l'affirmation « Si  $f(x) = \ln(x)$  alors  $f'(x) = \frac{(-1)^{(1-1)}(1-1)!}{x^1} = \frac{1}{x}$  ».

Donc  $P(1) \equiv \text{vrai}$ .

2. **Étape de récurrence :**

Soit  $k \geq 1$ .

– Hypothèse:  $P(k)$  : « La dérivée  $k^{\text{ième}}$  de  $f(x) = \ln(x)$  est  $f^{(k)}(x) = \frac{(-1)^{(k-1)}(k-1)!}{x^k}$  »  $\equiv \text{vrai}$ .

– Objectif:  $P(k+1)$  « La dérivée  $(k+1)^{\text{ième}}$  de  $f(x) = \ln(x)$  est  $f^{(k+1)}(x) = \frac{(-1)^k k!}{x^{k+1}}$  »  $\equiv \text{vrai}$ .

– Calculs :

$$\begin{aligned} f^{(k+1)}(x) &= [f^{(k)}(x)]' = \left[ \frac{(-1)^{(k-1)}(k-1)!}{x^k} \right]' && \text{(par hypothèse de récurrence)} \\ &= (-1)^{(k-1)}(k-1)! \left[ \frac{1}{x^k} \right]' && \text{(par les règles de dérivation)} \\ &= (-1)^{(k-1)}(k-1)!(-kx^{-k-1}) \\ &= (-1)^k k! x^{-(k+1)} && \text{(puisque } (k-1)!k = k!) \\ &= \frac{(-1)^k k!}{x^{k+1}} \end{aligned}$$

□

**Rép. 6.8** (a) On définit  $P(n)$  : «  $\sum_{i=0}^n f_i = f_{n+2} - 1$  » et on montre que  $\forall n \in \mathbb{N}, P(n) \equiv \text{vrai}$ .

1. **Cas de base :**

$P(0)$  est l'affirmation «  $f_0 = f_2 - 1$  ». Comme par définition  $f_0 = 0$  et  $f_2 = 1$ , alors  $P(0) \equiv \text{vrai}$ .

2. **Étape de récurrence :**

Soit  $k \geq 0$ .

– Hypothèse:  $P(k)$  : «  $\sum_{i=0}^k f_i = f_{k+2} - 1$  »  $\equiv \text{vrai}$ .

– Objectif:  $P(k+1)$  : «  $\sum_{i=0}^{k+1} f_i = f_{k+3} - 1$  »  $\equiv \text{vrai}$ .

– Calculs:

$$\begin{aligned} \sum_{i=0}^{k+1} f_i &= \sum_{i=0}^k f_i + f_{k+1} \\ &= f_{k+2} - 1 + f_{k+1} && \text{(par hypothèse de récurrence)} \\ &= f_{k+3} - 1 && \text{(par définition des nombres de Fibonacci)} \end{aligned}$$

□

(b) On définit  $P(n)$ : «  $\sum_{i=0}^n f_i^2 = f_n \cdot f_{n+1}$  » et on montre que  $\forall n \in \mathbb{N}, P(n) \equiv \mathbf{vrai}$ .

1. **Cas de base:**

$P(0)$  est l'affirmation «  $f_0^2 = f_0 \cdot f_1$  ». Comme par définition  $f_0 = 0$  et  $f_1 = 1$ , alors  $P(0) \equiv \mathbf{vrai}$ .

2. **Étape de récurrence:**

Soit  $k \geq 0$ .

– Hypothèse:  $P(k)$ : «  $\sum_{i=0}^k f_i^2 = f_k \cdot f_{k+1}$  »  $\equiv \mathbf{vrai}$ .

– Objectif:  $P(k+1)$ : «  $\sum_{i=0}^{k+1} f_i^2 = f_{k+1} \cdot f_{k+2}$  »  $\equiv \mathbf{vrai}$ .

– Calculs:

$$\begin{aligned} \sum_{i=0}^{k+1} f_i^2 &= \sum_{i=0}^k f_i^2 + f_{k+1}^2 \\ &= f_k \cdot f_{k+1} + f_{k+1}^2 && \text{(par hypothèse de récurrence)} \\ &= f_{k+1}(f_k + f_{k+1}) && \text{(par mise en évidence)} \\ &= f_{k+1}f_{k+2} && \text{(par définition des nombres de Fibonacci)} \end{aligned}$$

□

**Rép. 6.9** On définit  $P(n)$ : «  $a_n = n2^n$  » et on veut montrer que  $\forall n \in \mathbb{N}, P(n) \equiv \mathbf{vrai}$ .

1. **Cas de base:**

$P(0) \equiv \mathbf{vrai}$  car  $a_0 = 0$  (par définition) et  $0 \cdot 2^0 = 0$ .

$P(1) \equiv \mathbf{vrai}$  car  $a_1 = 2$  (par définition) et  $1 \cdot 2^1 = 2$ .

2. **Étape de récurrence:**

Soit  $k \geq 1$ .

– Hypothèse de récurrence:  $P(0) \wedge P(1) \wedge \dots \wedge P(k) \equiv \mathbf{vrai}$ . En particulier, on suppose que  $P(k-1) \equiv \mathbf{vrai}$  et donc que  $a_{k-1} = (k-1)2^{k-1}$ ,  $P(k) \equiv \mathbf{vrai}$  et donc que  $a_k = k2^k$ .

– Objectif:  $P(k+1)$  «  $a_{k+1} = (k+1)2^{k+1}$  »  $\equiv \mathbf{vrai}$ .

– Calculs:

$$\begin{aligned} a_{k+1} &= 4a_k - 4a_{k-1} && \text{(par définition)} \\ &= 4k2^k - 4(k-1)2^{k-1} && \text{(par hyp. de réc.)} \\ &= 2k2^{k+1} - (k-1)2^{k+1} \\ &= (2k - k + 1)2^{k+1} \\ &= (k+1)2^{k+1} \end{aligned}$$

□

**Rép. 6.10** On définit  $P(n)$ : «  $a_n = 3^n - 2^{n+1} + 1$  » et on veut montrer que  $\forall n \in \mathbb{N}, P(n) \equiv \mathbf{vrai}$ .

1. **Cas de base:**

$P(0) \equiv \mathbf{vrai}$  car  $a_0 = 0$  (par définition) et  $3^0 - 2^1 + 1 = 0$ .

$P(1) \equiv \mathbf{vrai}$  car  $a_1 = 0$  (par définition) et  $3^1 - 2^2 + 1 = 0$ .

$P(2) \equiv \mathbf{vrai}$  car  $a_2 = 2$  (par définition) et  $3^2 - 2^3 + 1 = 2$ .

2. **Étape de récurrence:**

Soit  $k \geq 2$ .

- Hypothèse de récurrence:  $P(0) \wedge P(1) \wedge \dots \wedge P(k) \equiv \mathbf{vrai}$ . En particulier, on suppose que  $P(k-2) \equiv \mathbf{vrai}$  et donc que  $a_{k-2} = 3^{k-2} - 2^{k-1} + 1$ ,  
 $P(k-1) \equiv \mathbf{vrai}$  et donc que  $a_{k-1} = 3^{k-1} - 2^k + 1$ ,  
 $P(k) \equiv \mathbf{vrai}$  et donc que  $a_k = 3^k - 2^{k+1} + 1$ .
- Objectif:  $P(k+1)$ : «  $a_{k+1} = 3^{k+1} - 2^{k+2} + 1$  »  $\equiv \mathbf{vrai}$ .
- Calculs:

$$\begin{aligned}
 a_{k+1} &= 6a_k - 11a_{k-1} + 6a_{k-2} && \text{(par définition)} \\
 &= 6(3^k - 2^{k+1} + 1) - 11(3^{k-1} - 2^k + 1) + 6(3^{k-2} - 2^{k-1} + 1) && \text{(par hyp. de réc.)} \\
 &= (6 \cdot 3^k - 11 \cdot 3^{k-1} + 6 \cdot 3^{k-2}) - (6 \cdot 2^{k+1} - 11 \cdot 2^k + 6 \cdot 2^{k-1}) + (6 - 11 + 6) \\
 &= (2 \cdot 3^{k+1} - 11 \cdot 3^{k-1} + 2 \cdot 3^{k-1}) - (3 \cdot 2^{k+2} - 11 \cdot 2^k + 3 \cdot 2^k) + 1 \\
 &= (2 \cdot 3^{k+1} - 9 \cdot 3^{k-1}) - (3 \cdot 2^{k+2} - 8 \cdot 2^k) + 1 \\
 &= (2 \cdot 3^{k+1} - 3^{k+1}) - (3 \cdot 2^{k+2} - 2 \cdot 2^{k+2}) + 1 \\
 &= 3^{k+1} - 2^{k+2} + 1
 \end{aligned}$$

□

**Rép. 6.11** On définit  $P(n)$ : « Il existe  $a, b \in \mathbb{N}$  tels que  $3a + 7b = n$  » et on veut montrer que  $\forall n \geq 12, P(n) \equiv \mathbf{vrai}$ .

1. **Cas de base:** (ici  $n_0 = 12$  et  $j = 2$ )

$$P(12) \equiv \mathbf{vrai} \text{ avec } a = 4 \text{ et } b = 0 \text{ car } 12 = 4 \cdot 3 + 0 \cdot 7,$$

$$P(13) \equiv \mathbf{vrai} \text{ avec } a = 2 \text{ et } b = 1 \text{ car } 13 = 2 \cdot 3 + 1 \cdot 7,$$

$$P(14) \equiv \mathbf{vrai} \text{ avec } a = 0 \text{ et } b = 2 \text{ car } 14 = 0 \cdot 3 + 2 \cdot 7.$$

2. **Étape de récurrence:**

Soit  $k \geq 14$ .

- Hypothèse de récurrence:  $P(12) \wedge P(13) \wedge \dots \wedge P(k-1) \wedge P(k) \equiv \mathbf{vrai}$ . En particulier, on a que  $P(k-2) \equiv \mathbf{vrai}$ . On a donc qu'il existe  $a, b \in \mathbb{N}$  tels que  $3a + 7b = k - 2$ .
- Objectif:  $P(k+1)$ : « Il existe  $a_2, b_2 \in \mathbb{N}$  tels que  $k + 1 = 3a_2 + 7b_2$  »  $\equiv \mathbf{vrai}$ .
- Calculs:

$$\begin{aligned}
 k + 1 &= (k - 2) + 3 \\
 &= 3a + 7b + 3 && \text{(par hyp. de réc.)} \\
 &= 3(a + 1) + 7b \\
 &= 3a_2 + 7b_2. && \text{(avec } a_2 = a + 1 \in \mathbb{N} \text{ et } b_2 = b \in \mathbb{N})
 \end{aligned}$$

□

**Rép. 6.12** On définit  $P(n)$ : « Il existe  $a, b \in \mathbb{N}$  tels que  $5a + 7b = n$  » et on veut montrer que  $\forall n \geq 24, P(n) \equiv \mathbf{vrai}$ .

1. **Cas de base:** (ici  $n_0 = 24$  et  $j = 4$ )

$$P(24) \equiv \mathbf{vrai} \text{ avec } a = 2 \text{ et } b = 2 \text{ car } 24 = 2 \cdot 5 + 2 \cdot 7,$$

$$P(25) \equiv \mathbf{vrai} \text{ avec } a = 5 \text{ et } b = 0 \text{ car } 25 = 5 \cdot 5 + 0 \cdot 7,$$

$$P(26) \equiv \mathbf{vrai} \text{ avec } a = 1 \text{ et } b = 3 \text{ car } 26 = 1 \cdot 5 + 3 \cdot 7.$$

$$P(27) \equiv \mathbf{vrai} \text{ avec } a = 4 \text{ et } b = 1 \text{ car } 27 = 4 \cdot 5 + 1 \cdot 7.$$

$$P(28) \equiv \mathbf{vrai} \text{ avec } a = 0 \text{ et } b = 4 \text{ car } 28 = 0 \cdot 5 + 4 \cdot 7.$$

2. **Étape de récurrence:**

Soit  $k \geq 28$ .

- Hypothèse de récurrence:  $P(24) \wedge P(25) \wedge \dots \wedge P(k-1) \wedge P(k) \equiv \mathbf{vrai}$ . En particulier, on a que  $P(k-4) \equiv \mathbf{vrai}$ . On a donc qu'il existe  $a, b \in \mathbb{N}$  tels que  $5a + 7b = k - 4$ .
- Objectif:  $P(k+1)$ : « Il existe  $a_2, b_2 \in \mathbb{N}$  tels que  $k + 1 = 5a_2 + 7b_2$  »  $\equiv \mathbf{vrai}$ .

– Calculs:

$$\begin{aligned}
 k+1 &= (k-4)+5 \\
 &= 5a+7b+5 && \text{(par hyp. de réc.)} \\
 &= 5(a+1)+7b \\
 &= 5a_2+7b_2. && \text{(avec } a_2 = a+1 \in \mathbb{N} \text{ et } b_2 = b \in \mathbb{N})
 \end{aligned}$$

□

**Rép. 6.13** (a) 1: **fonction** factorielle( $n$ : Entier non négatif)

```

2:   si n == 0 alors
3:     retourner 1
4:   sinon
5:     retourner n · factorielle(n-1)
6:   fin si
7: fin fonction
    
```

(b) On définit  $P(n)$ : « factorielle( $n$ ) retourne  $n!$  » et on montre que  $\forall n \in \mathbb{N}, P(n) \equiv \text{vrai}$ .

1. **Cas de base:**

$P(0) \equiv \text{vrai}$  comme en analysant le code de la fonction on vérifie que pour  $n = 0$ , la fonction retourne 1.

2. **Étape de récurrence:**

Soit  $k \geq 0$ .

– Hypothèse de récurrence:  $P(k)$ : « un appel à factorielle( $k$ ) retourne  $k!$  »  $\equiv \text{vrai}$ .

– Objectif:  $P(k+1)$ : « un appel à factorielle( $k+1$ ) retourne  $(k+1)!$  »  $\equiv \text{vrai}$ .

– Argumentation: comme  $k \geq 0$ , on a que  $k+1 > 0$  et donc la condition du **si** est fausse et on passe à la ligne suivante: **sinon**. Par hypothèse de récurrence, l'appel récursif retourne  $k!$  et l'appel à factorielle( $k+1$ ) retourne  $(k+1) \cdot k! = (k+1)!$ . □

(c) Soit  $f(n)$  le nombre de multiplications effectuées par un appel à factorielle( $n$ ), on a

$$f(n) = \begin{cases} 0 & \text{si } n = 0, \\ f(n-1) + 1 & \text{si } n \geq 1. \end{cases}$$

En itérant,  $f(n) = f(n-1) + 1 = f(n-2) + 2 = f(n-3) + 3 = \dots = f(0) + n$ . On conclut que  $f(n) = n$ .

**Rép. 6.14** (a) mystère( $n$ ) =  $n^3$ .

(b) On définit  $P(n)$ : « mystère( $n$ ) retourne  $n^3$  » et on veut montrer que  $\forall n \geq 0, P(n) \equiv \text{vrai}$ .

1. **Cas de base:**

$P(0) \equiv \text{vrai}$  comme en analysant le code de la fonction on vérifie que pour  $n = 0$ , la fonction retourne 0.

2. **Étape de récurrence:**

Soit  $k \geq 0$ ,

– Hypothèse de récurrence:  $P(k)$ : « un appel à mystère( $k$ ) retourne  $k^3$  »  $\equiv \text{vrai}$ .

– Objectif:  $P(k+1)$ : « un appel à mystère( $k+1$ ) retourne  $(k+1)^3$  »  $\equiv \text{vrai}$ .

– Argumentation: comme  $k \geq 0$ , on a que  $k+1 > 0$  et donc la condition du **si** est fausse et on passe à la ligne suivante: **sinon**. Par hypothèse de récurrence, l'appel récursif retourne  $k^3$  et l'appel à mystère( $k+1$ ) retourne  $k^3 + 3(k+1)^2 - 3(k+1) + 1 = (k+1)^3$ . □

(c)  $f(0) = 0, f(n) = f(n-1) + 3, n \geq 1$ .

(d)  $f(n) = 3n \in O(n)$ , linéaire.

**Rép. 6.15** (a) 1: **fonction** Fibon( $n$ : Entier non négatif)

```

2:   si n == 0 alors
3:     retourner 0
4:   sinon si n == 1 alors
5:     retourner 1
6:   sinon
    
```

7:       **retourner**  $\text{Fib}_0(n-1) + \text{Fib}_0(n-2)$   
 8:       **fin si**  
 9:       **fin fonction**

(b) On définit  $P(n)$ : «  $\text{Fib}_0(n)$  retourne  $f_n$  » et on veut montrer que  $\forall n \geq 0, P(n) \equiv \text{vrai}$ .

1. **Cas de base:**

$P(0) \equiv \text{vrai}$ , en analysant le code de la fonction on vérifie que pour  $n = 0$ , la fonction retourne 0.  
 $P(1) \equiv \text{vrai}$ , en analysant le code de la fonction on vérifie que pour  $n = 1$ , la fonction retourne 1.

2. **Étape de récurrence:**

Soit  $k \geq 1$ .

– Hypothèse de récurrence:  $P(0) \wedge P(1) \wedge \dots \wedge P(k) \equiv \text{vrai}$ , c'est-à-dire qu'un appel à  $\text{Fib}_0(n)$  avec  $n \in \{0, 1, \dots, k\}$  retourne  $f_n$ .

– Objectif:  $P(k+1)$ : « un appel à  $\text{Fib}_0(k+1)$  retourne  $f_{k+1}$  »  $\equiv \text{vrai}$ .

– Argumentation: comme  $k \geq 1$ , on a que  $k+1 \geq 2$  et donc la fonction effectue d'abord un appel à  $\text{Fib}_0(k-1)$ . Par hypothèse de récurrence, cet appel retourne  $f_{k-1}$ . Ensuite, on effectue un appel à  $\text{Fib}_0(k)$  qui, par hypothèse de récurrence, retourne  $f_k$ . Finalement, la fonction retourne  $f_{k-1} + f_k$  hors, par la définition de la suite de Fibonacci,  $f_{k-1} + f_k = f_{k+1}$ .  $\square$

(c) On définit  $P(n)$ : «  $g(n) = f_{n+1} - 1$  » et on montre que  $\forall n \geq 0, P(n) \equiv \text{vrai}$ .

1. **Cas de base:**

$P(0) \equiv \text{vrai}$ , par l'algorithme il est évident que  $g(0) = 0$  ce qui est bien égal à  $f_1 - 1 = 0$ .

$P(1) \equiv \text{vrai}$ , encore une fois, par l'algorithme il est évident que  $g(1) = 0$ , et on a  $f_2 - 1 = 0$ .

2. **Étape de récurrence:**

Soit  $k \geq 1$ .

– Hypothèse de récurrence:  $P(0) \wedge P(1) \wedge \dots \wedge P(k) \equiv \text{vrai}$  et donc, en particulier, on accepte les hypothèses  $g(k-1) = f_k - 1$  et  $g(k) = f_{k+1} - 1$ .

– Objectif:  $P(k+1)$ : «  $g(k+1) = f_{k+2} - 1$  »  $\equiv \text{vrai}$ .

– Argumentation: Comme  $k \geq 1$ , on a que  $k+1 \geq 2$  et donc lors d'un appel à  $\text{Fib}_0(k+1)$ , les conditions des deux **si** sont fausses et le **retourner** de la ligne 7 est effectué. Ainsi, le nombre d'additions est donné par:

$$\begin{aligned} g(k+1) &= g(k) + g(k-1) + 1 && \text{(par l'algorithme)} \\ &= (f_{k+1} - 1) + (f_k - 1) + 1 && \text{(par hypothèse de récurrence)} \\ &= f_{k+2} - 1. && \text{(par déf. des nombres de Fibonacci)} \end{aligned}$$

$\square$

(d) 1: **fonction**  $\text{Fib}_0(n$ : Entier non négatif)

2:     $a := 0$

3:     $b := 1$

4:    **pour**  $i$  de 0 à  $n - 1$  **faire**

5:        $c := a + b$

6:        $a := b$

7:        $b := c$

8:    **fin pour**

9:    **retourner**  $a$

10: **fin fonction**

(e)  $h(n) = n$ .

(f) La version itérative est beaucoup plus efficace. En effet, on a que,  $h(n) \in O(g(n))$  mais  $g(n) \notin O(h(n))$ .

(g) Malheureusement, nous ne vous donnerons pas la réponse ici. Voici quand même deux indices:

– Considérez le produit matriciel suivant:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} 0 & f_n \\ 0 & f_{n+1} \end{bmatrix} = \begin{bmatrix} 0 & af_n + bf_{n+1} \\ 0 & cf_n + df_{n+1} \end{bmatrix}$$

– Inspirez-vous de l'algorithme de l'exponentiation modulaire vu au Cours 5 afin de calculer efficacement de grandes puissances.



## Chapitre 7

**Rép. 7.1** (a) Il a  $2^8 = 256$  trains de bits de longueur 8.

Il y a deux choix pour le premier bit **ET** deux choix pour le deuxième bit **ET** deux choix pour le troisième bit, et ainsi de suite jusqu'au huitième. On conclut en utilisant le *principe du produit*. Visuellement :

Bit #1	Bit #2	Bit #3	Bit #4	Bit #5	Bit #6	Bit #7	Bit #8
0 ou 1	0 ou 1	0 ou 1	0 ou 1	0 ou 1	0 ou 1	0 ou 1	0 ou 1
2	2	2	2	2	2	2	2
choix	choix	choix	choix	choix	choix	choix	choix

(b) Il y a  $2^3 = 8$  trains de bits de longueur 8 qui commencent par 01 et terminent par 101. Par le *principe du produit* :

#1	#2	#3	#4	#5	#6	#7	#8
0	1	0 ou 1	0 ou 1	0 ou 1	1	0	1
1	1	2	2	2	1	1	1
choix	choix	choix	choix	choix	choix	choix	choix

(c) Il y a  $10^3 \cdot 26^3 = 17\,576\,000$  plaques d'immatriculation de 3 chiffres suivis de 3 lettres. Par le *principe du produit* :

#1	#2	#3	#4	#5	#6
0 à 9	0 à 9	0 à 9	A à Z	A à Z	A à Z
10	10	10	26	26	26
choix	choix	choix	choix	choix	choix

**Rép. 7.2** Il y a 2 684 483 063 360 mots de passe possibles.

On commence par évaluer le nombre de mots de passe valides de longueur  $n$  puis on utilise le *principe de la somme* pour les cas  $n = 6$ ,  $n = 7$  et  $n = 8$ .

Comme chaque caractère est soit un chiffre, soit une lettre majuscule, par le *principe de la somme* il y a  $10+26 = 36$  possibilités pour chaque caractère.

Ensuite, il faut tenir compte de la contrainte voulant que chaque mot de passe contienne au moins un chiffre. Le plus simple est de compter tous les mots de passe sans tenir compte de la contrainte et ensuite soustraire tout ce qui a été compté en trop.

#1	#2	#3		#n
0 à 9 ou A à Z	0 à 9 ou A à Z	0 à 9 ou A à Z	...	0 à 9 ou A à Z
36	36	36		36
choix	choix	choix		choix

Sans tenir compte de la contrainte, il y a  $36^n$  mots de passe possibles. À ce nombre, il faut soustraire tous les mots de passe qui ne contiennent aucun chiffre.

#1	#2	#3		#n
A à Z	A à Z	A à Z	...	A à Z
26	26	26		26
choix	choix	choix		choix

Il y a donc  $26^n$  mots de passe invalides à soustraire. Le nombre de mots de passe valides de longueur  $n$  est donc  $36^n - 26^n$ . En sommant pour  $n = 6, 7$  et  $8$ , on obtient :  $\sum_{i=6}^8 (36^i - 26^i)$ .

**Rép. 7.3** (a) Réponse: 56 étudiants.

On définit les ensembles suivants :

- $U$  est l'ensemble des étudiants inscrits en génie logiciel,
- $A_1$  est l'ensemble des étudiants en génie logiciel inscrits en MAT210,
- $A_2$  est l'ensemble des étudiants en génie logiciel inscrits en MAT265.

On cherche à déterminer  $|U| - |A_1 \cup A_2|$ . Par le *principe d'inclusion-exclusion*,

$$|U| - |A_1 \cup A_2| = |U| - (|A_1| + |A_2| - |A_1 \cap A_2|) = 300 - (143 + 122 - 21) = 56.$$

(b) Réponse: 88 trains de bits.

On pose :

- $A_1$  l'ensemble des trains de bits de longueur 8 qui commencent par 01.
- $A_2$  l'ensemble des trains de bits de longueur 8 qui terminent par 101.

On a alors que

- $|A_1| = 2^6$  car:  $01 \underbrace{\hspace{4em}}_{6 \text{ bits}}$ ,
- $|A_2| = 2^5$  car:  $\underbrace{\hspace{4em}}_{5 \text{ bits}} 101$ ,
- $|A_1 \cap A_2| = 2^3$  car:  $01 \underbrace{\hspace{2em}}_{3 \text{ bits}} 101$ .

Par le *principe d'inclusion-exclusion*:  $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2| = 2^6 + 2^5 - 2^3$ .

**Rép. 7.4** (a) Il y a 5040 dispositions.

Les individus étant considérés comme distincts, le nombre de dispositions est donné par le nombre de permutations de 7 objets:  $7! = 5040$ .

(b) Il y a 1440 dispositions.

On peut considérer d'un bloc les deux hommes placés consécutivement dans la file. On compte ainsi les permutations formées par ce bloc et les 5 femmes. Il s'agit du nombre de permutations de 6 objets  $6! = 720$ .

**Attention:** pour chacune de ces permutations, les deux hommes peuvent être disposés différemment à l'intérieur du bloc. Il y a  $2!$  ordres possibles à l'intérieur du bloc. Ainsi, le nombre total de dispositions est  $6! \cdot 2! = 1440$ .

(c) Il y a 3600 dispositions.

Il suffit de considérer le nombre total de dispositions (calculé en (a)) et y soustraire le nombre de dispositions où les deux hommes se suivent (calculé en (b)).

(d) Il y a 120 permutations.

On compte les permutations des 5 objets: "ABC", "D", "E", "F" et "G". Il y en a  $5! = 120$ .

**Rép. 7.5** (a) Il faut au moins 26 étudiants.

Soit  $n$  le nombre d'étudiants et  $k = 5$  le nombre de notes possibles. On veut ranger  $n$  étudiants dans  $k$  tiroirs et avoir au moins six étudiants dans le même tiroir.

Par le *principe des tiroirs*, on cherche le plus petit entier  $n$  tel que :

$$\left\lceil \frac{n}{5} \right\rceil = 6 \quad \leftrightarrow \quad 5 < \frac{n}{5} \leq 6 \quad \leftrightarrow \quad 25 < n \leq 30.$$

Il doit y avoir au minimum 26 étudiants garantir qu'au moins 6 ont la même note.

(b) Il faut 52729 mots.

Soit  $n$  le nombre de mots et  $k$  le nombre de suites de trois lettres. On veut ranger  $n$  mots dans  $k$  tiroirs et avoir au moins quatre mots dans le même tiroir. Par le *principe des tiroirs*, on cherche le plus petit entier  $n$  tel que :

$$\left\lceil \frac{n}{k} \right\rceil = 4 \quad \leftrightarrow \quad 3 < \frac{n}{k} \leq 4 \quad \leftrightarrow \quad 3k < n \leq 4k.$$

Par le *principe du produit*, le nombre de tiroirs est  $k = 26^3$ . Il faut donc  $n = 3k + 1 = 3 \cdot 26^3 + 1 = 52729$  mots.

- (c) Il faut au minimum 3 indicatifs régionaux.

Un numéro de téléphone est composé d'un indicatif régional (les trois premiers chiffres) et d'un suffixe (les 7 chiffres restants). Soit  $n = 17 \cdot 10^6$  le nombre d'abonnés et  $k$  le nombre de suffixes possibles. On veut *ranger*  $n$  abonnés dans  $k$  tiroirs.

Le nombre d'indicatifs régionaux nécessaires est ainsi donné par le nombre d'abonnés *rangés* dans le même tiroir.

Par le *principe du produit*, le nombre de suffixes est  $k = 8 \cdot 10^6$ .

$$\begin{array}{|c|c|c|} \hline A & X & X \\ \hline \end{array} - \begin{array}{|c|c|c|c|} \hline X & X & X & X \\ \hline \end{array}$$

$\begin{array}{ccc} 8 & 10 & 10 \\ \text{choix} & \text{choix} & \text{choix} \end{array}$ 
 $\quad$ 
 $\begin{array}{cccc} 10 & 10 & 10 & 10 \\ \text{choix} & \text{choix} & \text{choix} & \text{choix} \end{array}$

Par le *principe des tiroirs*, le nombre minimum d'indicatifs régionaux nécessaires est :

$$\left\lceil \frac{17 \cdot 10^6}{8 \cdot 10^6} \right\rceil = 3.$$

- Rép. 7.6** (a) Il y a 20 mots.

On compte le nombre d'arrangements de 2 lettres parmi 5,  $P(5, 2) = 20$ .

- (b) Il y a 14400 dispositions.

On commence par considérer toutes les permutations possibles des 5 femmes (il y a en 5!). Pour chacune de ces permutations, on ajoute les trois hommes en s'assurant qu'ils soient séparés. On a donc la situation suivante :

$$\bullet F_1 \bullet F_2 \bullet F_3 \bullet F_4 \bullet F_5 \bullet$$

Parmi les six  $\bullet$ , il faut choisir celui où on place le premier homme, puis celui où on place le deuxième et finalement celui où on place le troisième. On construit donc un arrangement de trois  $\bullet$  parmi six et il y a  $P(6, 3)$  façons de faire.

En multipliant, on obtient  $5! \cdot P(6, 3) = 14400$ .

- Rép. 7.7** (a) Il y en a 56.

Parmi les 8 bits, on en *choisit* trois et on leur assigne la valeur 1. L'ordre dans lequel ces trois bits sont choisis n'a pas d'importance puisque les 1 ne sont pas distinguables. On compte donc le nombre de combinaisons de 3 parmi 8; il y en a  $C(8, 3)$ . Visuellement, pour construire un tel train de bit, on considère 8 positions et on en choisit 3 où on affecte la valeur 1.

$$\begin{array}{cccccccc} - & - & \frac{1}{1} & - & \frac{1}{1} & - & - & \frac{1}{1} \\ & & \uparrow & & \uparrow & & & \uparrow \end{array}$$

Ensuite, on affecte la valeur 0 à tous les bits qui n'ont pas été choisis. Dans l'exemple ci-dessus, on obtient : 00101001. Cette dernière étape ne change rien au comptage, car il n'y a qu'une seule façon de faire. Il y a donc  $C(8, 3) = 56$  trains de bits avec exactement 3 occurrences de 1.

- (b) Il y en a 93.

Par le raisonnement expliqué en (a), on conclut que le nombre de trains de bits de longueur  $n$  possédant exactement  $k$  occurrences de 1 est donné par  $C(n, k)$ . Ainsi,

- il y a exactement  $C(8, 0)$  trains de bits de longueur 8 avec zéro 1,
- il y a exactement  $C(8, 1)$  trains de bits de longueur 8 avec un 1,
- il y a exactement  $C(8, 2)$  trains de bits de longueur 8 avec deux 1,
- il y a exactement  $C(8, 3)$  trains de bits de longueur 8 avec trois 1.

Par le *principe de la somme*, le nombre de trains de bits avec au plus trois 1 est :

$$\sum_{i=0}^3 C(8, i) = 93.$$

- (c) Il a 13 983 816 façons de choisir 6 numéros parmi 49.  
L'ordre des numéros n'a pas d'importance, il s'agit donc du nombre de combinaisons de 6 parmi 49, soit  $C(49, 6)$ .
- (d) Il y en a  $C(657, 4) \cdot C(753, 2) = 2\,178\,009\,835\,742\,880$ .  
On choisit 4 étudiants parmi les 657 en génie logiciel ET on choisit 2 étudiants parmi les 753 en génie électrique. Comme l'ordre dans lequel les étudiants sont choisis n'a pas d'importance,  
- il y a  $C(657, 4)$  façon de choisir 4 étudiants parmi 657,  
- il y a  $C(753, 2)$  façon de choisir 2 étudiants parmi 753.  
Il y a donc  $C(657, 4) \cdot C(753, 2)$  façons de former un tel comité.

- Rép. 7.8** (a)  $C(40, 2) \cdot C(300, 5) = 15\,274\,613\,296\,800$   
 (b)  $C(40, 2) \cdot C(300, 3) + C(40, 3) \cdot C(300, 2) = 3\,918\,096\,000$   
 (c)  $C(340, 5) = 36\,760\,655\,568$   
 (d)  $C(340, 4) + C(340, 5) + C(340, 6) + C(340, 7) + C(340, 8) = 4\,176\,447\,114\,716\,383$

- Rép. 7.9** (a) 2520  
 (b) 1680  
 (c) 3360

- Rép. 7.10** (a) 128  
 (b) 27  
 (c) 783  
 (d) 729  
 (e) 560  
 (f) 939

- Rép. 7.11** (a)  $10^6$   
 (b)  $10^4 + 5 \cdot 10^5 - 10^3 \cdot 5 = 505\,000$   
 (c)  $10^6 - 9^6 = \sum_{i=1}^6 C(6, i) 9^{6-i} = 468\,559$   
 (d)  $C(6, 1) \cdot C(5, 1) \cdot 8^4 = 122\,880$   
 (e) Permutation de 4 objets (10, 10, 2 et 3), dont deux sont indistinguables:  $\frac{4!}{2!1!1!} = 12$   
 (f)  $C(5, 3) \cdot 9^2 \cdot 10 = 8\,100$   
 (g)  $P(10, 6) = 151\,200$   
 (h) 90720  
 (i) 33600

- Rép. 7.12** (a)  $b_0 = 0$  et  $b_1 = 0$ .  
 (b) On suppose que  $n$  est suffisamment grand et on considère toutes les chaînes comptées par  $b_n$ . Ces chaînes sont forcément de l'une des formes suivantes:

Forme des chaînes		Nombre de chaînes de cette forme
0	chaîne de $\ln g n - 1$ avec au moins une occurrence de 11	$b_{n-1}$
2	chaîne de $\ln g n - 1$ avec au moins une occurrence de 11	$b_{n-1}$
1 0	chaîne de $\ln g n - 2$ avec au moins une occurrence de 11	$b_{n-2}$
1 2	chaîne de $\ln g n - 2$ avec au moins une occurrence de 11	$b_{n-2}$
1 1	chaîne ternaire quelconque de $\ln g n - 2$	$3^{n-2}$

On en déduit la relation de récurrence  $b_0 = 0$ ;  $b_1 = 0$ ;  $b_n = 2b_{n-1} + 2b_{n-2} + 3^{n-2}$  pour  $n \geq 2$ .

- (c) En utilisant la commande `seqgen` de Nspire:

$$\text{seqGen}\left(2 \cdot b(n-1) + 2 \cdot b(n-2) + 3^{n-2}, n, b, \{0, 8\}, \{0, 0\}\right) \\ \{0, 0, 1, 5, 21, 79, 281, 963, 3217\}$$

On a donc que  $b_8 = 3217$ .

- Rép. 7.13** (a)  $a_0 = 1, a_1 = 3, a_2 = 9, a_n = 2a_{n-1} + 2a_{n-2} + 2a_{n-3}$ . Remarque  $a_0 = 1$  est obtenu en considérant la chaîne vide. Pour éviter de se casser la tête avec la chaîne vide, on peut commencer avec  $a_1 = 3, a_2 = 9$  et ajouter le troisième cas de base :  $a_3 = 26$ .
- (b) 9989792
- (c) 69.6%
- (d) 295

- Rép. 7.14** (a)  $a_1 = 3, a_2 = 3^2 - 1 = 8$  et  $a_3 = 3^3 - 7 = 20$ .
- (b) On suppose que  $n$  est *suffisamment grand* et on considère toutes les chaînes comptées par  $a_n$ . Ces chaînes sont forcément de l'une des formes suivantes :

	Forme des chaînes	Nombre de chaînes de cette forme
1	chaîne de lng $n - 1$ sans 000 ni 01	$a_{n-1}$
2	chaîne de lng $n - 1$ sans 000 ni 01	$a_{n-1}$
0 2	chaîne de lng $n - 2$ sans 000 ni 01	$a_{n-2}$
0 0 2	chaîne de lng $n - 3$ sans 000 ni 01	$a_{n-3}$

On en déduit la relation de récurrence  $a_1 = 3; a_2 = 8; a_3 = 20; a_n = 2a_{n-1} + a_{n-2} + a_{n-3}$  pour  $n \geq 4$ .

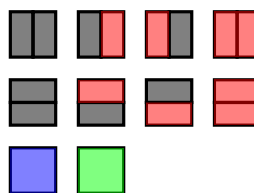
- (c)  $a_{15} = 1\,492\,163$ .

- Rép. 7.15** (a)  $b_1 = 2, b_n = b_{n-1} + 1$ .
- (b) 31









- Rép. 7.16** (a) On trouve  $a_1 = 2$ , comme les pavages possibles d'un trottoir  $2 \times 1$  sont :



On trouve  $a_2 = 10$ , comme les pavages possibles d'un trottoir  $2 \times 1$  sont :



On suppose que  $n$  est *suffisamment grand* et on considère tous les pavages comptés par  $a_n$ . Ces pavages sont forcément de l'une des formes suivantes :

Forme des pavages	Nombre de pavages de cette forme
 pavage de dimensions $2 \times (n-1)$	$a_{n-1}$
 pavage de dimensions $2 \times (n-1)$	$a_{n-1}$
 pavage de dimensions $2 \times (n-2)$	$a_{n-2}$
 pavage de dimensions $2 \times (n-2)$	$a_{n-2}$
 pavage de dimensions $2 \times (n-2)$	$a_{n-2}$
 pavage de dimensions $2 \times (n-2)$	$a_{n-2}$
 pavage de dimensions $2 \times (n-2)$	$a_{n-2}$
 pavage de dimensions $2 \times (n-2)$	$a_{n-2}$

On en déduit la relation de récurrence

$$\begin{cases} a_1 = 2 \\ a_2 = 10 \\ a_n = 2a_{n-1} + 6a_{n-2}, \quad n \geq 3 \end{cases}$$

(b)  $a_{15} = 184\,082\,432$ .

- Rép. 7.17**
- (a)  $a_1 = 1, a_2 = 2, a_4 = 6$ .  
 (b)  $a_1 = 1, a_2 = 2, a_3 = 4, a_n = 2a_{n-2} + 2a_{n-3}$  pour  $n \geq 4$ .  
 (c)  $a_{15} = 3456$

## Chapitre 8

**Rép. 8.1** Rappel:

- La somme de deux nombres pairs est paire.
- La somme de deux nombres impairs est paire.
- La somme d'un nombre pair et d'un nombre impair est impaire.

Ainsi, la somme d'un nombre impair de nombres impairs est impaire.

**Preuve:** Par contradiction, on suppose qu'il existe un graphe  $G = (V, E)$  à  $n$  sommets avec un nombre impair de sommets de degré impair. On pose:

$$S = \sum_{v \in V} \deg(v).$$

Soit  $V_p$  l'ensemble des sommets de  $G$  de degré pair et  $V_i$  l'ensemble des sommets de  $G$  de degré impair. On a que  $V_p \cup V_i = V$  et  $V_p \cap V_i = \emptyset$  et donc,

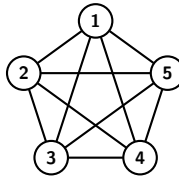
$$S = \sum_{v \in V_p} \deg(v) + \sum_{v \in V_i} \deg(v).$$

D'un côté,  $\sum_{v \in V_p} \deg(v)$  est pair car c'est une somme de nombres pairs.

D'un autre côté,  $\sum_{v \in V_i} \deg(v)$  est impair car c'est la somme d'un nombre impair de nombres impairs.

Ainsi,  $S$  est impair car c'est la somme d'un nombre pair et d'un nombre impair. Or, le Théorème 8.1 stipule que  $S = 2|E|$ , c'est donc un nombre pair. Contradiction.

Rép. 8.2 (a) Le graphe  $K_5$  possède 10 arêtes.



(b) Numérotons les sommets du graphe  $K_n$  de 1 à  $n$ . Le sommet 1 est relié par une arête à ses  $n - 1$  sommets adjacents, le sommet 2 aux  $n - 2$  sommets adjacents restants, ainsi de suite jusqu'au sommet  $n$ . En additionnant toutes ces arêtes on obtient la sommation suivante, dont on trouve la somme avec le théorème 4.7:

$$(n-1) + (n-2) + (n-3) + \dots + (n-(n-1)) + (n-(n-0)) = 0+1+2+3+\dots+(n-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}.$$

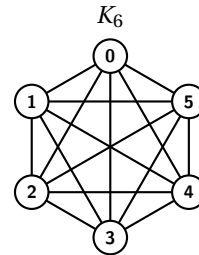
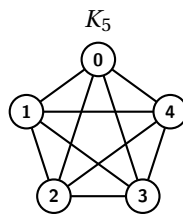
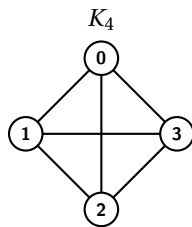
Une autre façon d'obtenir le résultat est d'utiliser le théorème 8.1 des poignées de mains. Soit  $G = (V, E)$  un graphe non orienté. Alors

$$\sum_{v \in V} \deg(v) = 2|E|.$$

Puisque le graphe  $K_n$  est complet, chacun des  $n$  sommets de l'ensemble  $V$  est reliés aux  $n - 1$  autres. Ainsi, la somme des degrés des sommets est de  $n$  fois  $(n - 1)$ . En isolant  $|E|$ , on trouve que le nombre d'arêtes est de  $\frac{n(n-1)}{2}$ .

- Rép. 8.3 (a) 2 147 randonnées  
 (b) 785 randonnées  
 (c) 606 randonnées  
 (d) 3 heures

Rép. 8.4 (a)



$K_4$

- circuit eulérien: impossible, les 4 sommets sont de degré impair.
- circuit hamiltonien:  $0 - 1 - 2 - 3 - 0$ .

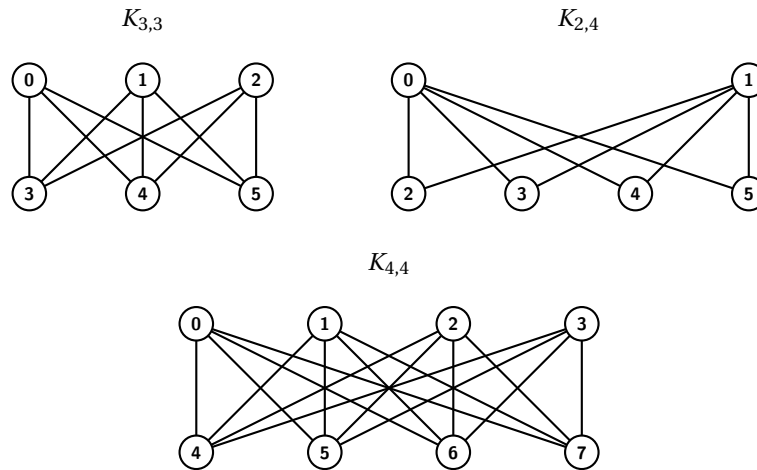
$K_5$

- circuit eulérien:  $0 - 1 - 2 - 0 - 3 - 1 - 4 - 2 - 3 - 4 - 0$ ;
- circuit hamiltonien:  $0 - 1 - 2 - 3 - 4 - 0$ .

$K_6$

- circuit eulérien: impossible, les 6 sommets sont de degré impair.
- circuit hamiltonien:  $0 - 1 - 2 - 3 - 4 - 5 - 0$ .

(b)

 $K_{3,3}$ 

- circuit eulérien : impossible, les 6 sommets sont de degré impair.
- circuit hamiltonien :  $0 - 3 - 1 - 4 - 2 - 5 - 0$ .

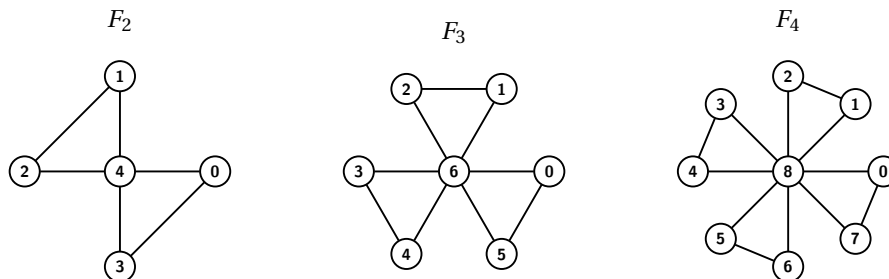
 $K_{2,4}$ 

- circuit eulérien :  $0 - 2 - 1 - 3 - 0 - 4 - 1 - 5 - 0$ ;
- circuit hamiltonien : impossible. Les ensembles  $B_0 = \{0, 1\}$  et  $B_1 = \{2, 3, 4, 5\}$  forment une bipartition du graphe. Un chemin dans ce graphe doit forcément visiter en alternance un sommet de  $B_0$  puis un sommet de  $B_1$  puis un sommet de  $B_0$  et ainsi de suite. Hors, comme  $|B_0| = 2$  et  $|B_1| = 4$  il est impossible qu'un circuit passe un et une seule fois par chacun des sommets.

 $K_{4,4}$ 

- circuit eulérien :  $0 - 7 - 3 - 6 - 2 - 5 - 3 - 4 - 2 - 7 - 1 - 6 - 0 - 5 - 1 - 4 - 0$ ;
- circuit hamiltonien :  $0 - 4 - 1 - 5 - 2 - 6 - 3 - 7 - 0$ .

(c)

 $F_2$ 

- circuit eulérien :  $0 - 4 - 2 - 1 - 4 - 3 - 0$ ;
- circuit hamiltonien : impossible. Tout d'abord, on remarque que lorsqu'un sommet est de degré 2, un circuit hamiltonien passe forcément par ses deux arêtes. Ainsi, si un sommet est adjacent à plus de deux sommets de degré 2 alors on peut affirmer qu'il n'existe pas de circuit hamiltonien. Ici, le sommet 4 est adjacent à quatre sommets de degré 2.

**Remarque :** Bien qu'il n'existe aucun circuit hamiltonien dans ce graphe, il existe des chemins hamiltoniens. Par exemple :  $0 - 3 - 4 - 1 - 2$ .

 $F_3$ 

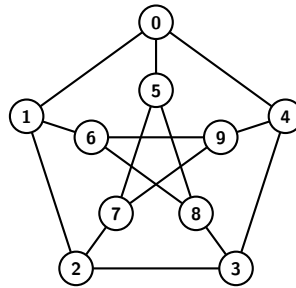
- circuit eulérien :  $0 - 5 - 6 - 4 - 3 - 6 - 2 - 1 - 6 - 0$
- circuit hamiltonien : impossible. De manière similaire au cas du graphe  $F_2$ , on remarque ici que le sommet 6 est adjacent à six sommets de degré 2.

 $F_4$ 

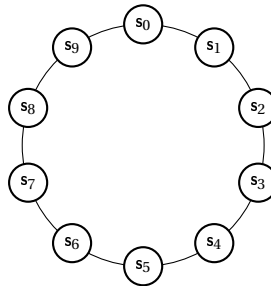
- circuit eulérien :  $0 - 7 - 8 - 6 - 5 - 8 - 4 - 3 - 8 - 2 - 1 - 8 - 0$ .
- circuit hamiltonien : impossible. De manière similaire au cas du graphe  $F_2$ , on remarque ici que le sommet 8 est adjacent à huit sommets de degré 2.



(d) Graphe de Peterson



- circuit eulérien : impossible. Les 10 sommets sont de degré impair.
- circuit hamiltonien : impossible. Par contradiction, on suppose qu'il existe un circuit hamiltonien et que ce circuit est  $s_0 - s_1 - \dots - s_9 - s_0$ . Maintenant, on trace le graphe de Peterson, en deux étapes. Première étape : on dispose les 10 sommets le long d'un cercle, dans l'ordre donné par le circuit hamiltonien :

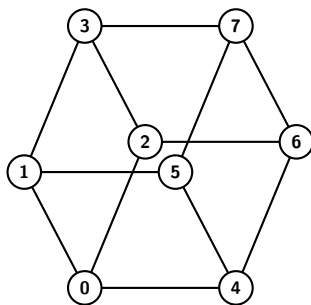


Deuxième étape : tracer les 5 arêtes restantes (le graphe de Peterson compte 15 arêtes). On remarque alors que lorsqu'on ajoute 5 arêtes à un tel graphe, on forme forcément un circuit de taille 3 ou 4 or les plus petits circuits du graphe de Peterson sont de longueur 5. Contradiction.

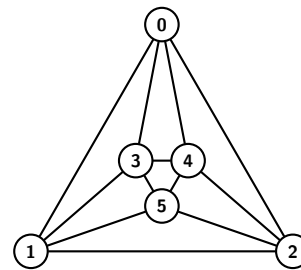
**Remarque :** Bien qu'il n'existe aucun circuit hamiltonien dans ce graphe, il existe des chemins hamiltoniens. Par exemple :  $0-1-6-8-5-7-9-4-3-2$ .

(e) Solides de Platon. Il existe cinq solides de Platon : le tétraèdre, le cube, l'octaèdre, le dodécaèdre et l'icosaèdre. Le graphe du tétraèdre est  $K_4$ , qui a déjà été traité en (a).

Cube



Octaèdre

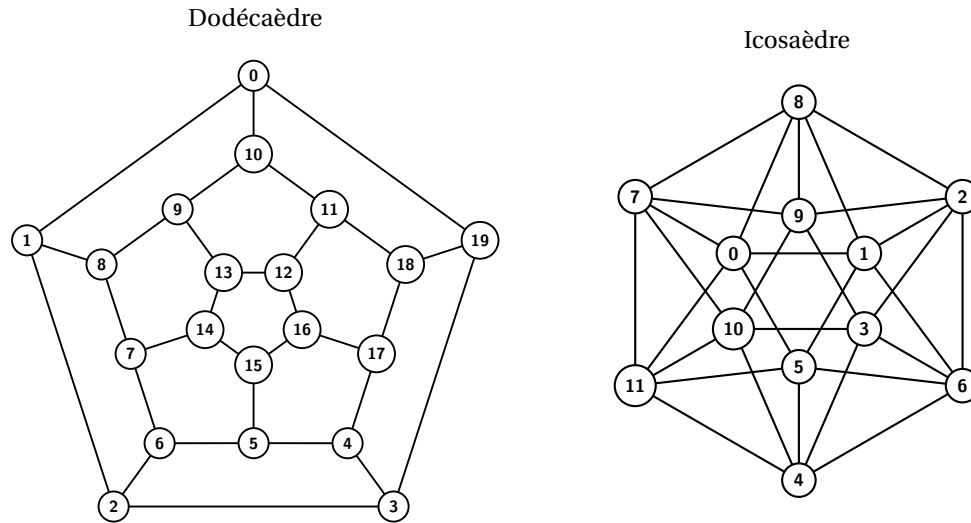


**Cube**

- circuit eulérien : impossible, tous les sommets sont de degré impair.
- circuit hamiltonien :  $0 - 1 - 3 - 2 - 6 - 7 - 5 - 4 - 0$ .

**Octaèdre**

- circuit eulérien :  $0 - 4 - 5 - 3 - 4 - 2 - 5 - 1 - 3 - 0 - 2 - 1 - 0$
- circuit hamiltonien :  $0 - 1 - 2 - 4 - 5 - 3 - 0$ .

**Dodécaèdre**

- circuit eulérien : impossible, tous les sommets sont de degré impair.
- circuit hamiltonien :  $0-1-2-3-19-18-17-4-6-5-7-8-9-13-14-15-16-12-11-10-0$ .

**Icosaèdre**

- circuit eulérien : impossible, tous les sommets sont de degrés impair.
- circuit hamiltonien :  $0-1-8-9-2-3-10-4-6-5-11-7-0$ .

**Rép. 8.5**

$u$	A	B	C	D	E	F	G	H	S
	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\emptyset$
A		$9_A$	$8_A$	$3_A$	$\infty$	$\infty$	$\infty$	$\infty$	{A}
D		$9_A$	$7_D$		$\infty$	$\infty$	$10_D$	$\infty$	{A, D}
C		$8_C$			$13_C$	$10_C$	$10_D$	$\infty$	{A, C, D}
B					$12_B$	$10_C$	$10_D$	$\infty$	{A, B, C, D}
F					$12_B$		$10_D$	$14_F$	{A, B, C, D, F}
G					$12_B$			$14_F$	{A, B, C, D, F, G}
E								$13_E$	{A, B, C, D, E, F, G}
H									{A, B, C, D, E, F, G, H}

- (a) Distance minimale de **A** à **H**: 13.  
 (b) Chemin minimal: **A** – **D** – **C** – **B** – **E** – **H**.  
 (c) Distance minimale de **A** à **B**: 8.  
 (d) Chemin minimal: **A** – **D** – **C** – **B**.

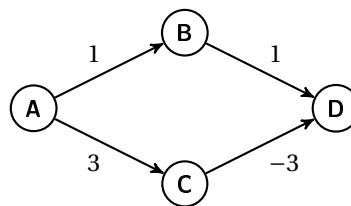
- Rép. 8.6** (a) La distance minimale est 16 via le chemin **C** – **A** – **B** – **D** – **F** – **G**.  
 (b) La distance minimale est 10 via le chemin **C** – **E** – **D** – **F** – **H** – **G**.

- Rép. 8.7** (a) 6 itinéraires.  
 (b) 17 itinéraires.  
 (c) 10 itinéraires.  
 (d) 9747 itinéraires.  
 (e) 44 itinéraires.  
 (f) Oui.  
 (g) Non. Non.

- (h) Oui. Par exemple, le circuit hamiltonien: Gaspé - Québec- Toronto - Vancouver - Ottawa - Montréal - Gaspé.
- (i) Non. Le sommet Vancouver est de degré impair, le théorème 8.4 permet donc de conclure que non.
- (j) Oui, car le graphe possède 2 sommets de degrés impairs (théorème 8.5). Par exemple, le chemin eulérien: Vancouver - Toronto - Québec - Gaspé - Montréal - Québec- Ottawa - Montréal - Vancouver - Ottawa - Toronto - Montréal.
- (k) Graphe pondéré.
- (l) Chemin: Gaspé - Q - M - T - Vancouver, pour un coût de 2150 dollars. Chemin T- M-Q - G, pour un coût de 1050 dollars. Non, pas toujours le même nombre, l'algorithme s'arrête quand le sommet de destination a été ajouté à l'ensemble S des sommets visités.

**Rép. 8.8**

- (a) Voici un exemple de graphe pour lequel l'algorithme de Dijkstra ne produit pas un chemin de pondération minimale.

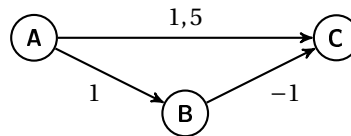


- (b)

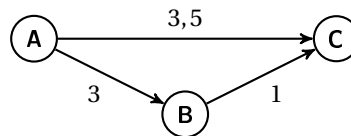
$u$	A	B	C	D	S
	0	$\infty$	$\infty$	$\infty$	$\emptyset$
A		$1_A$	$3_A$		{A}
B			$3_A$	$2_B$	{A,B}
D			$3_A$		{A,B,D}

Le chemin calculé est donc **A – B – D** avec un coût de 2 alors que le chemin **A – C – D** a un coût de 0.

- (c) On considère le graphe suivant :



La plus petite pondération est  $-1$ . On peut donc obtenir des pondérations positives en additionnant 2 à chaque arc. On obtient alors ce graphe:



On effectue une trace de l'algorithme de Dijkstra pour le calcul d'un chemin minimum de **A** à **C**.

$u$	A	B	C	S
	0	$\infty$	$\infty$	$\emptyset$
A		$3_A$	$3,5_A$	{A}
B			$3,5_A$	{A,B}
C				{A,B,C}

Le chemin calculé est **A – C**, or dans le graphe original ce chemin à un coût de 1,5 alors que le chemin **A – B – C** à un coût de 0.

## Bibliographie

- [1] Rosen, Kenneth H.: *Discrete Mathematics and its applications*. McGraw-Hill, seventh édition, 2012.
- [2] Rosen, Kenneth H.: *Mathématiques discrètes*. Chenelière/McGraw-Hill, 1998.
- [3] Johnsonbaugh, Richard: *Discrete Mathematics*. Pearson, 2009. seventh edition.
- [4] Yves Norbert, Roch Ouellet, Régis Parent: *Méthodes d'optimisation pour la gestion*. Chenelière Éducation, 2016. 2<sup>e</sup>édition.

Rédigé par Geneviève Savard, Anouk Bergeron-Brlek et Xavier Provençal,  
révisé en août 2021,  
Service des enseignements généraux,  
École de technologie supérieure.

Ce document est mis à disposition selon les termes de la licence Creative  
Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification  
4.0 International.

