MATHÉMATIQUES DISCRÈTES

Mathieu SABLIK

Table des matières

Chapitre I

Introduction à la théorie des ensembles

I.1 Notions sur les ensembles

I.1.1 Construction par extension et compréhension

Intuitivement, un *ensemble* est une collection d'objets deux à deux distincts appelés *éléments*. On peut définir un ensemble de deux manières :

— en *extension* : on donne la liste exhaustive des éléments qui y figurent ;
— en *compréhension* : on donne les propriétés que doivent posséder les éléments de l'ensemble.

Exemple I.1. Voilà quelques exemples d'ensembles d'élèves :

— {Pierre ; Paul ; Marie}, on donne les trois éléments qui définissent l'ensemble ;
— {élèves de la classe qui ont les yeux bleus} ;
— {élèves qui viennent en cours en pyjama}, mais cet ensemble est certainement vide !

Exemple I.2. Dans votre scolarité vous avez rencontré certains ensembles classiques de nombres :

— $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ est l'ensemble des nombres naturels ;
— $\mathbb{N}^* = \{1, 2, 3, \ldots\}$ est l'ensemble des nombres naturels non nul ;
— $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$ est l'ensemble des nombres entiers ;
— $\mathbb{Q} = \{p/q : p \in \mathbb{Z} \text{ et } q \in \mathbb{N} \text{ avec } q \neq 0\}$;
— $\mathbb{R}$ l'ensemble des nombres réels ;
— $\mathbb{C}$ l'ensemble des nombres complexes.

Exemple I.3. Les langages de programmation actuels exigent que certaines variables soient déclarées avec un certain *type de données*. Un type de données est un ensemble d'objets associés à une liste d'opérations standards effectuées sur ces objets. Définir le type d'une variable équivaut à déclarer l'ensemble des valeurs possibles et autorisées pour cette variable.

Dans la sémantique de Python vous avez dû rencontrer :

— le type `bool` s'interprète comme l'ensemble $\{Vrai, Faux\}$,
— le type `int` s'interprète comme l'ensemble des entiers
— le type `float` s'interprète comme l'ensemble des nombres à virgule flottante
— le type `str` s'interprète comme l'ensemble des chaînes de caractères
— le type `list` s'interprète comme l'ensemble des listes de longueur variable.

I.1.2 Principales règles de fonctionnement

On admettra l'existence d'ensembles. Sans rentrer dans l'axiomatique, la notion d'ensemble satisfait un certain nombre de règles de fonctionnement, en voici les principales :

Relation d'appartenance Il faut pouvoir dire si un objet est dans l'ensemble. On note $x \in A$ l'élément x est dans l'ensemble A.

Objets distincts On peut distinguer deux éléments entre eux et un ensemble ne peut pas contenir deux fois le même objet.

Ensemble vide Il existe un ensemble qui ne contient aucun élément, c'est l'ensemble vide et on le note $\emptyset$ ou $\{\}$.

Paradoxe de Russell Un ensemble peut être élément d'un autre ensemble mais pas de lui même.

Remarque I.1. Cette dernière règle peut ne pas sembler naturelle. A la naissance de la théorie des ensembles, les mathématiciens ne voyaient pas d'objection à envisager un ensemble dont les éléments seraient tous les ensembles : l'ensemble des ensembles. Russell leur opposa le paradoxe suivant :

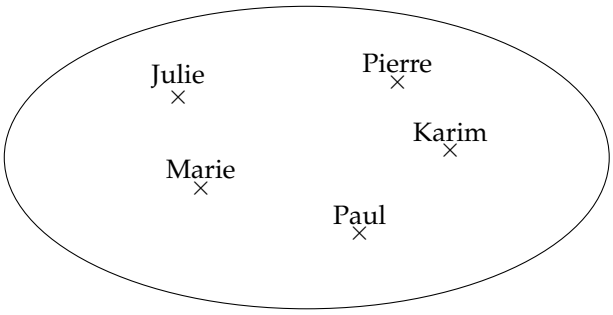
Supposons que l'ensemble de tous les ensembles existe, et notons-le E. On considère l'ensemble $A = \{x \in E : x \notin x\}$. Comme E contient tous les ensembles, A appartient à E. Est-ce que A appartient à A ?

— si $A \in A$ alors par définition de A, on a $A \notin A$,
— si $A \notin A$ alors par définition de A, on a $A \in A$.

I.1.3 Représentation

On peut représenter les ensembles à l'aide d'un diagramme de Venn, ce sont les fameux diagrammes "patates".

Exemple I.4. L'ensembe $\{$Pierre ; Paul ; Marie ; Julie ; Karim$\}$ se représente par :



I.2 Sous-ensembles

I.2.1 Inclusion

Définition I.1 (Sous-ensembles)**.** L'ensemble A est un *sous-ensemble* de B si tous les éléments de A sont des éléments de B (autrement dit $x \in A \implies x \in B$). On dit aussi que A est *inclus* dans B, on le note $A \subseteq B$.

Remarque I.2. Pour tout ensemble A on a $\emptyset \subseteq A$ et $A \subseteq A$.

Exemple I.5. On a $\{1,2\} \subseteq \{1,2,3\}$.
Bien sûr on a $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$.

Définition I.2 (Egalité d'ensembles)**.** Deux ensembles sont *égaux* si et seulement si ils ont les mêmes éléments, autrement dit si $A \subseteq B$ et $B \subseteq A$.

I.2.2 Ensemble des parties

Définition I.3 (Ensemble des parties)**.** Soit A un ensemble, l'*ensemble des parties de* A, noté $\mathcal{P}(A)$, est l'ensemble des sous-ensembles de A.

On remarque que l'on a toujours $\emptyset \in \mathcal{P}(A)$ car $\emptyset \subseteq A$ et $A \in \mathcal{P}(A)$ car $A \subseteq A$.

Exemple I.6. Si $A = \{1,2,3\}$ alors $\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$.

Remarque I.3. On a $\mathcal{P}(\emptyset) = \{\emptyset\}$ et $\mathcal{P}(\mathcal{P}(\emptyset)) = \{\emptyset, \{\emptyset\}\}$. La notation $\emptyset$ décrit un ensemble qui ne contient rien alors que $\{\emptyset\}$ décrit un ensemble contenant un élément, l'ensemble vide. Un tiroir contenant un sac vide ($\{\emptyset\}$) n'est pas vide et contient bien un objet.

I.3 Opérations sur les ensembles

On présente ici des opérations sur les ensembles qui permettent de construire de nouveaux ensembles.

I.3.1 Union et Intersection

Définition I.4 (Union)**.** L'*union* des ensembles A et B est l'ensemble des éléments qui sont éléments de A ou de B. On le note $A \cup B$.

Proposition I.1 Propriétés de la réunion

La réunion admet certaines propriétés :

Idempotence : $A \cup A = A$

Commutativité : $A \cup B = B \cup A$

Associativité : $A \cup (B \cup C) = (A \cup B) \cup C$

Elément neutre : $A \cup \emptyset = A$

Définition I.5 (Intersection)**.** L'*intersection* des ensembles A et B est l'ensemble des éléments communs à A et à B. On le note $A \cap B$.

On dit que deux ensembles sont *disjoints* (ou *incompatibles*) si $A \cap B = \emptyset$.

Proposition I.2 Propriétés de l'intersection

L'intersection admet certaines propriétés :

Idempotence : $A \cap A = A$

Commutativité : $A \cap B = B \cap A$

Associativité : $A \cap (B \cap C) = (A \cap B) \cap C$

Elément neutre : *si l'on se place dans un ensemble Ω appelé univers et que A est un sous-ensemble de Ω alors* $A \cap \Omega = A$

Proposition I.3 Propriétés de distributivité

On a les distributivités suivantes entre l'union et l'intersection :

de $\cup$ sur $\cap$: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

de $\cap$ sur $\cup$: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

I.3.2 Différence et complémentaire

Définition I.6 (Différence)**.** La *différence* de l'ensemble A par l'ensemble B est l'ensemble des éléments qui sont dans A mais pas dans B, on le note $A \setminus B$.

Définition I.7 (Différence symétrique)**.** La *différence symétrique* entre les ensembles A et B est l'ensemble des éléments qui sont dans A ou B mais pas dans les deux, on le note

$$A \Delta B = (A \cup B) \setminus (A \cap B).$$

Définition I.8 (Complémentaire)**.** On se fixe un ensemble Ω appelé *univers*. Pour $A \subseteq \Omega$, on définit le *complémentaire* de A par rapport à Ω comme l'ensemble des éléments de Ω qui ne sont pas éléments de A, on le note $\overline{A} = \Omega \setminus A$ lorsqu'il n'y a pas d'ambiguïtés.

Remarque I.4. Il faut obligatoirement se placer dans un ensemble de référence pour définir la complémentation.

Proposition I.4 Propriétés de la complémentation

La complémentation a plusieurs propriétés :

Involution : $\overline{\overline{A}} = A$

Loi de Morgan : $\overline{A \cap B} = \overline{A} \cup \overline{B}$ *et* $\overline{A \cup B} = \overline{A} \cap \overline{B}$

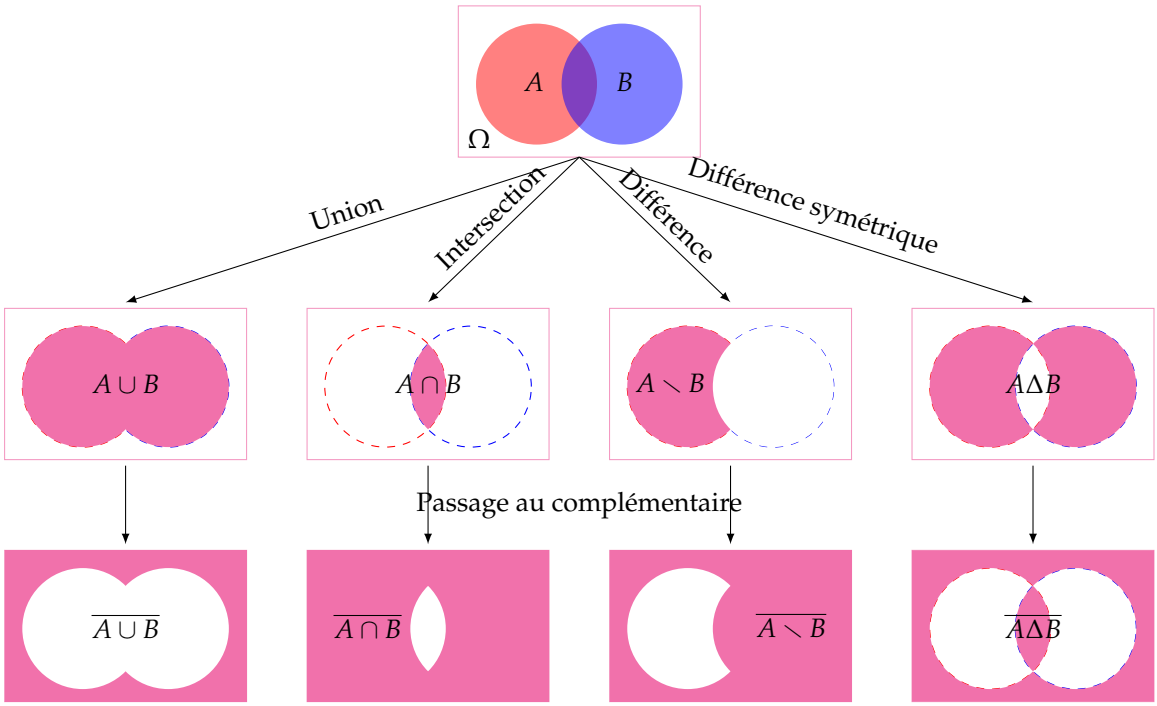


FIGURE I.1 – Exemples de constructions d'ensembles à partir des ensembles A et B contenus dans l'univers Ω

I.3.3 Produit cartésien

Définition I.9 (Produit cartésien)**.** Le *produit cartésien* des ensembles A et B (dans cet ordre) est l'ensemble des *couples* (a, b) où $a \in A$ et $b \in B$, on le note $A \times B$.

Le *produit cartésien* des ensembles $A_1, A_2, \ldots, A_k$ (dans cet ordre) est l'ensemble des *k-uplets* $(a_1, \ldots, a_k)$ où $a_i \in A_i$ pour tout $i \in \{1, \ldots, k\}$, on le note $A_1 \times \cdots \times A_k$.

Si $A_1 = \cdots = A_k$ on note A^k l'ensemble des k-uplets formés par les éléments de A.

Remarque I.5. Le couple (a, b) n'est pas un ensemble.

Si $a \neq b$ alors (a, b) est distinct de (b, a).

Exemple I.7. Le système de codage informatique des couleurs RGB, (de l'anglais "Red, Green, Blue") reconstitue une couleur par synthèse additive à partir de trois couleurs primaires (rouge, vert et bleu), formant sur l'écran une mosaïque trop petite pour être aperçue. Ainsi pour chacune des trois couleurs primaires, on donne une valeur s'exprimant dans un intervalle entre 0 et 255. D'un point de vue informatique, une couleur est donc un élément de $[0;255] \times [0;255] \times [0;255] = [0;255]^3$.

Chapitre II

Notions sur les langages

II.1 Exemples de problèmes

La notion de langage est utilisée pour modéliser différents problèmes où l'information est stockée sous une forme de chaîne de caractères. Voici quelques exemples :

- Langage naturel : chaque mot est formé par un ensemble de lettres concaténées. L'ensemble des mots forme un dictionnaire. Puis ces mots sont organisés pour former des phrases. Dans ce cas, une structure apparaît qui est régie par la grammaire de la langue utilisé.
- Stocker de l'information sur un disque dur : toute information stockée sur un disque dur est codée par une succession de bits (0 ou 1), que ce soit du texte, image, musique... On peut se demander s'il est possible de compresser cette information, c'est-à-dire trouver une fonction qui a un chaîne de $\{0,1\}$ renvoie de manière bijective une chaîne plus courte.
- Recherche de chaîne de caractères dans un texte.
- Compilation : un programme est une suite de caractères. Un compilateur s'intéresse essentiellement aux deux choses suivantes :
  - Analyse lexicale : on cherche les éléments de bases qui structurent le programme (If, For, While, affectation...).
  - Analyse syntaxique : on vérifie que les expressions sont correctes (ex : $var + var * var$ va être interprété comme $var + (var * var)$).
- Bio-informatique : l'ADN code l'information génétique à l'aide de 4 bases azotées : adénine (A), cytosine (C), guanine (G) ou thymine (T).

II.2 Mots sur un alphabet fini

II.2.1 Un peu de vocabulaire

Alphabet Un *alphabet* $\mathcal{A}$ est un ensemble fini dont les éléments sont appelés des lettres.

Exemple II.1. $\mathbb{B} = \{0,1\}$ est l'alphabet binaire, $\mathcal{A} = \{a,b,c\}$ est un alphabet à trois lettres, $\mathcal{B} = \{a,\dots,z\}$ un alphabet à 26 lettres . On peut considérer n'importe quel ensemble fini, par exemple $\mathcal{C} = \{hello, word\}$ est un alphabet à deux lettres.

Mots sur un alphabet fini Un *mot* est une suite finie d'éléments de $\mathcal{A}$ on le note $u = u_1u_2\dots u_n$ et n est la longueur du mot u, notée $|u|$. Le mot vide est noté ε.

On note $\mathcal{A}^*$ l'ensemble des mots sur $\mathcal{A}$ et $\mathcal{A}^+$ l'ensemble des mots sans le mot vide.

Opérations sur les mots Soient u et v deux mots de $\mathcal{A}^*$, on définit la concaténation $w = u.v$ comme le mot de longueur $|u| + |v|$ tel que $w = u_1u_2\dots u_{|u|}v_1v_2\dots v_{|v|}$.

Pour $n \in \mathbb{N}$ on définit par récurrence la puissance d'un mot par $u^0 = \varepsilon$ et $u^{n+1} = u.u^n$ pour $n \in \mathbb{N}$.

On dit que v est un *préfixe* de u s'il existe un mot w tel que $u = v.w$ et v est un *suffixe* de u s'il existe un mot w tel que $u = w.v$.

Distance sur les mots On peut définir différentes distances sur les mots. On s'intéressera ici à la distance édition définie comme étant le plus petit nombre d'opérations d'édition élémentaires nécessaires pour transformer le mot u en le mot v. Les opérations d'édition élémentaires sont la suppression ou l'insertion d'un symbole.

De multiples variantes de cette notion de distance ont été proposées, qui utilisent des ensembles d'opérations différents et/ou considèrent des poids variables pour les différentes opérations. Pour prendre un exemple réel, si l'on souhaite réaliser une application qui « corrige » les fautes de frappe au clavier, il est utile de considérer des poids qui rendent d'autant plus proches des séquences qu'elles ne diffèrent que par des touches voisines sur le clavier, permettant d'intégrer une modélisation des confusions de touches les plus probables.

L'utilitaire Unix `diff` implante une forme de calcul de distances. Cet utilitaire permet de comparer deux fichiers et d'imprimer sur la sortie standard toutes les différences entre leurs contenus respectifs.

II.2.2 Propriété d'équidivisibilité

Lemme II.1 Lemme de Levi

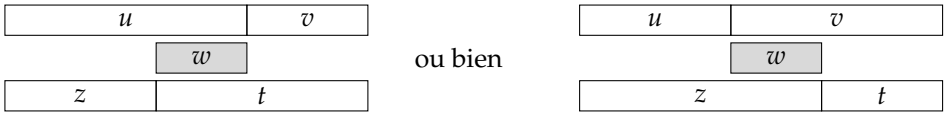
Soient u, v, z, t des mots sur l'alphabet $\mathcal{A}$ tels que $uv = zt$. Alors il existe un mot $w \in \mathcal{A}^*$ tel qu'un des deux cas suivant est vérifié :
— *ou bien $u = zw$ et $t = wy$ si $|u| \geq |z|$,*
— *ou bien $z = uw$ et $v = wt$ si $|u| \leq |z|$.*

Démonstration : On considère que $uv = zt = a_1 a_2 \dots a_n$ où les a_k sont des lettres de $\mathcal{A}$. Soit i l'entier tel que $u = a_1 a_2 \dots a_i$ et $v = a_{i+1} \dots a_n$. De même soit j l'entier tel que $z = a_1 a_2 \dots a_j$ et $t = a_{j+1} \dots a_n$.

Si $|u| \geq |z|$, alors $i \geq j$ et on a $u = zw$ et $t = wv$ avec $w = a_{j+1} \dots a_i$.

Si au contraire $|u| \leq |z|$, alors $i \leq j$ et on a $z = uw$ et $v = wt$ avec $w = a_{i+1} \dots a_j$. ■

Graphiquement cela signifie que l'on a une des deux décompositions suivantes :



Exemple II.2. Soient $u = anti$, $v = constitutionnellement$, $z = anticonstitutionnel$, $t = lement$ quatre mots. Comme $anti.constitutionnellement = anticonstitutionnel.lement$, et que de plus $|anti| \leq |anticonstitutionnel|$ alors il existe le mot $w = constitutionnel$ tel que $anticonstitutionnel = anti.constitutionnel$ et $constitutionnellement = constitutionnel.lement$.

Une conséquence importante : si on applique le Lemme de Levi avec $u = z$, on a $w = \epsilon$ et donc $v = t$. On en déduit que si $uv = ut$ alors $v = t$, autrement dit, on peut simplifier à gauche. De même, on peut simplifier à droite une équation sur les mots.

Proposition II.2

Soient u, v, z et $t \in \mathcal{A}^*$. Si $uv = ut$ alors $v = t$.
De même si $uv = zv$ alors $u = z$.

II.3 Langage

II.3.1 Définition et exemples de langages

Un *langage* $\mathcal{L}$ sur un alphabet fini $\mathcal{A}$ est un ensemble de mots définis sur $\mathcal{A}$ autrement dit $\mathcal{L} \subseteq \mathcal{A}^*$.

Exemple II.3. Exemples de langages sur $\mathbb{B} = \{0,1\}$:
- $\emptyset \neq \{\varepsilon\}$;
- $\mathbb{B}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$;
- $\mathbb{B}^+ = \{0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$;
- tout ensemble fini de mots ;
- $\{0^n : n \in \mathbb{N}\}$;
- $\{0^n 1^m : n, m \in \mathbb{N}\}$;
- $\{0^n 1^n : n \in \mathbb{N}\}$;
- $\{0^p : p \in \mathbb{N}$ nombre premier$\}$;
- $\{u \in \mathbb{B}^* : u$ est le codage en binaire d'un nombre premier$\}$;
- $\{u \in \mathbb{B}^* : u$ est un palindrome$\}$;
- $\{u \in \mathbb{B}^* : u$ est un code html certifié$\} \neq \{u \in \mathcal{A}^* : u$ est un code html bien interprété par Firefox$\}$;
- $\{u \in \mathbb{B}^* : u$ est le codage en MP3 de votre chanson préférée$\}$;
- $\{u \in \mathbb{B}^* : u$ est le codage en assembleur d'un programme qui s'arrête sur l'entrée vide$\}$;
- ...

II.3.2 Opérations sur les langages

Soient $\mathcal{L}$, $\mathcal{L}_1$ et $\mathcal{L}_2$ des langages sur un alphabet fini $\mathcal{A}$. On peut définir différentes opérations sur les langages :
- *Union* : $\mathcal{L}_1 \cup \mathcal{L}_2$ langage comportant des mots de $\mathcal{L}_1$ ou de $\mathcal{L}_2$;
- *Intersection* : $\mathcal{L}_1 \cap \mathcal{L}_2$ langage comportant des mots de $\mathcal{L}_1$ et de $\mathcal{L}_2$;
- *Compémentaire* : $\overline{\mathcal{L}}$ langage comportant des mots de $\mathcal{A}^*$ qui ne sont pas dans $\mathcal{L}$;
- *Concaténation* : $\mathcal{L}_1.\mathcal{L}_2$ langage comportant les mots formés en concaténant un mot $\mathcal{L}_1$ à un mot de $\mathcal{L}_2$
$$\mathcal{L}_1.\mathcal{L}_2 = \{u_1.u_2 : u_1 \in \mathcal{L}_1 \text{ et } u_2 \in \mathcal{L}_2\};$$
- *Puissance* : On définit par récurrence la puissance $n^{\text{ème}}$ de $\mathcal{L}$, notée $\mathcal{L}^n$ par
$$\mathcal{L}^0 = \{\varepsilon\} \text{ et } \mathcal{L}^{n+1} = \mathcal{L}.\mathcal{L}^n.$$
Attention, il ne faut pas confondre, on a $\mathcal{L}^n = \{u \in \mathcal{A}^* :$ il existe $u_1, u_2, \ldots, u_n \in \mathcal{L}$ tel que $u = u_1.u_2.\cdots.u_n\}$ qui en général est différent de $\{u^n : n \in \mathbb{N}\}$.
- *Fermeture de Kleene* : On définit
$$\mathcal{L}^* = \bigcup_{n \geq 0} \mathcal{L}^n \qquad \text{et} \qquad \mathcal{L}^+ = \bigcup_{n > 0} \mathcal{L}^n.$$

Exemple II.4. Considérons les langages $L_1 = \{a\}$, $L_2 = \{ab, ba\}$ et $L_3 = \{\epsilon, a\}$ on a :
- $L_1 \cup L_2 = \{ab, ba, a\}$, $L_2 \cup L_3 = \{ab, ba, a, \epsilon\}$ et $L_1 \cup L_3 = \{a, \epsilon\}$;
- $L_1 \cap L_2 = \emptyset$, $L_2 \cap L_3 = \emptyset$ et $L_1 \cap L_3 = \{a\}$;
- $L_1.L_2 = \{aab, aba\}$, $L_1.L_3 = \{a, aa\}$ et $L_2.L_3 = \{ab, aba, ba, baa\}$;
- pour $n \in \mathbb{N}$, on a $L_1^n = \{a^n$ tel que $n \in \mathbb{N}\}$;
- on a $abbaab \in L_2^*$ car $abbaab = ab.ba.ab$ et $ab \in L_2$ et $ba \in L_2$;
- on a $abbab \notin L_2^*$ car un mot de L_2^* est de longueur paire.

II.3.3 Equations sur les langages

Proposition II.3 Le lemme d'Arden

Soient $\mathcal{M}$ et $\mathcal{N}$ deux langages sur $\mathcal{A}$ tels que $\varepsilon \notin \mathcal{M}$. L'équation sur les langages $X = \mathcal{M}.X \cup \mathcal{N}$ où X est le langage inconnu, admet pour unique solution $\mathcal{M}^*.\mathcal{N}$.

Chapitre III

Fonctions et applications

La notion d'application permet d'associer à chaque élément d'un ensemble un élément d'un autre ensemble. En informatique, on a souvent de cette notion pour passer d'un objet à sa représentation, pour traduire une représentation vers une autre, comme support de preuves de propriétés de programmes, etc...

III.1 Premières notions

III.1.1 Définition

Définition III.1 (Fonction)**.** Soient E et F deux ensembles. Une *fonction* $f : E \to F$ (de E dans F) est définie par un sous-ensemble de $G_f \subseteq E \times F$ tel que pour tout $x \in E$, il existe au plus un $y \in F$ tel que $(x, y) \in G_f$. Quand il existe, on note cet élément par $f(x)$. L'élément x est alors l'*antécedent* de y et y est l'*image* de x. On note aussi $f : x \longmapsto y$ le fait que f associe l'élément y comme image de x.

Exemple III.1. Soit $E = \{1, 2, 3, 4\}$ et $F = \{a, b, c\}$.
On définit la fonction f par le graphe :

$$G_f = \{(1, a), (2, c), (4, a)\} \subseteq E \times F$$

Autrement dit

$$\begin{array}{rcl} f : E & \longrightarrow & F \\ 1 & \longmapsto & a \\ 2 & \longmapsto & c \\ 4 & \longmapsto & a \end{array}$$

On dit que a est l'image de 1 par f ou bien que 1 est un antécédent de a par f.
Par contre l'ensemble $H = \{(1, a), (2, c), (4, a)\} \subseteq E \times F$ n'est pas le graphe d'une fonction.

Définition III.2 (Ensemble image et préimage)**.** Etant donné $A \subseteq E$ et $B \subseteq F$, on définit
— l'image de A par f :

$$\begin{array}{rcl} f(A) & = & \{y \in F : \text{ il existe } x \in E \text{ tel que } f(x) = y\} \\ & = & \{y \in F : \text{ il existe } x \in E \text{ tel que } (x, y) \in G_f\}; \end{array}$$

— la préimage de B par f :

$$\begin{array}{rcl} f^{-1}(B) & = & \{x \in E : \text{ il existe } y \in F \text{ tel que } f(x) = y\} \\ & = & \{x \in E : \text{ il existe } y \in F \text{ tel que } (x, y) \in G_f\} \end{array}$$

Définition III.3 (Domaine de définition, Image)**.** Le *domaine de définition* d'une fonction $f : E \to F$, noté $\mathrm{Dom}(f)$ est l'ensemble des éléments de $x \in E$ qui ont une image par f. Autrement dit :

$$\mathrm{Dom}(f) = f^{-1}(F) = \{x \in E : \text{ il existe } y \in F \text{ tel que } f(x) = y\}$$

L'*ensemble image* d'une fonction $f : E \to F$, noté $\mathrm{Im}(f)$ est l'ensemble des éléments de $y \in E$ qui ont un antécédent par f. Autrement dit :

$$\mathrm{Im}(f) = f(E) = \{y \in F : \text{ il existe } x \in E \text{ tel que } f(x) = y\}$$

Exemple III.2. Soit $E = \{1, 2, 3, 4\}$ et $F = \{a, b, c\}$. On considère la fonction f définit par le graphe $G_f = \{(1, a), (2, c), (4, a)\} \subseteq E \times F$.

On a :

— $f(\{1\}) = \{a\}$, $f(\{1, 4\}) = \{a\}$, $f(\{3\}) = \emptyset$ et $f(\{1, 2, 3\}) = \{a, c\}$;
— $f^{-1}(\{a\}) = \{1, 4\}$, $f^{-1}(\{a, c\}) = \{1, 2, 4\}$, $f^{-1}(\emptyset) = \emptyset$ et $f^{-1}(\{b\}) = \emptyset$;
— le domaine de la fonction f est $\mathrm{Dom}(f) = \{a, c\}$;
— l'image de la fonction f est $\mathrm{Im}(f) = \{1, 2, 4\}$.

Définition III.4 (Egalité)**.** Deux fonctions $f : E \to F$ et $g : E \to F$ sont égales si $G_f = G_g$ ou de manière équivalente si $\mathrm{Dom}(f) = \mathrm{Dom}(g)$ et $f(x) = g(x)$ pour tout $x \in \mathrm{Dom}(f)$.

Définition III.5 (Application)**.** Une *application* de E dans F est une fonction de E dans F telle que $\mathrm{Dom}(f) = E$. On note F^E l'ensemble des applications de E dans F.

Exemple III.3. Soit $E = \{1, 2, 3, 4\}$, $E' = \{1, 2, 4\}$ et $F = \{a, b, c\}$.

Le graphe $G = \{(1, a), (2, c), (4, a)\} \subseteq E \times F$ définit une fonction de E dans F mais pas une application.

Par contre G définit une application de E' dans F.

III.1.2 Modes de représentation

On donne ici différents moyens pour représenter une fonction $f : E \to F$.

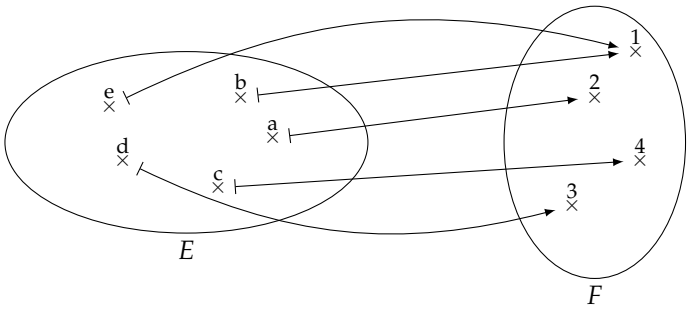
Table de valeur Pour chaque élément de E on donne l'élément de F associé. Si E est fini on peut représenter cela sous forme de tableau.

x	0	1	2	3	4	5	6	7	8	9
f(x)	aa	nj	zj	nk	za	az	aa	aa	zz	ju

Un autre exemple est le code ASCII qui permet d'associer à chaque entier entre 0 et 127 un caractère :

Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char
0	0000 0000	00	[NUL]	32	0010 0000	20	space	64	0100 0000	40	@	96	0110 0000	60	`
1	0000 0001	01	[SOH]	33	0010 0001	21	!	65	0100 0001	41	A	97	0110 0001	61	a
2	0000 0010	02	[STX]	34	0010 0010	22	"	66	0100 0010	42	B	98	0110 0010	62	b
3	0000 0011	03	[ETX]	35	0010 0011	23	#	67	0100 0011	43	C	99	0110 0011	63	c
4	0000 0100	04	[EOT]	36	0010 0100	24	$	68	0100 0100	44	D	100	0110 0100	64	d
5	0000 0101	05	[ENQ]	37	0010 0101	25	%	69	0100 0101	45	E	101	0110 0101	65	e
6	0000 0110	06	[ACK]	38	0010 0110	26	&	70	0100 0110	46	F	102	0110 0110	66	f
7	0000 0111	07	[BEL]	39	0010 0111	27	'	71	0100 0111	47	G	103	0110 0111	67	g
8	0000 1000	08	[BS]	40	0010 1000	28	(	72	0100 1000	48	H	104	0110 1000	68	h
9	0000 1001	09	[TAB]	41	0010 1001	29	)	73	0100 1001	49	I	105	0110 1001	69	i
10	0000 1010	0A	[LF]	42	0010 1010	2A	*	74	0100 1010	4A	J	106	0110 1010	6A	j
11	0000 1011	0B	[VT]	43	0010 1011	2B	+	75	0100 1011	4B	K	107	0110 1011	6B	k
12	0000 1100	0C	[FF]	44	0010 1100	2C	,	76	0100 1100	4C	L	108	0110 1100	6C	l
13	0000 1101	0D	[CR]	45	0010 1101	2D	-	77	0100 1101	4D	M	109	0110 1101	6D	m
14	0000 1110	0E	[SO]	46	0010 1110	2E	.	78	0100 1110	4E	N	110	0110 1110	6E	n
15	0000 1111	0F	[SI]	47	0010 1111	2F	/	79	0100 1111	4F	O	111	0110 1111	6F	o
16	0001 0000	10	[DLE]	48	0011 0000	30	0	80	0101 0000	50	P	112	0111 0000	70	p
17	0001 0001	11	[DC1]	49	0011 0001	31	1	81	0101 0001	51	Q	113	0111 0001	71	q
18	0001 0010	12	[DC2]	50	0011 0010	32	2	82	0101 0010	52	R	114	0111 0010	72	r
19	0001 0011	13	[DC3]	51	0011 0011	33	3	83	0101 0011	53	S	115	0111 0011	73	s
20	0001 0100	14	[DC4]	52	0011 0100	34	4	84	0101 0100	54	T	116	0111 0100	74	t
21	0001 0101	15	[NAK]	53	0011 0101	35	5	85	0101 0101	55	U	117	0111 0101	75	u
22	0001 0110	16	[SYN]	54	0011 0110	36	6	86	0101 0110	56	V	118	0111 0110	76	v
23	0001 0111	17	[ETB]	55	0011 0111	37	7	87	0101 0111	57	W	119	0111 0111	77	w
24	0001 1000	18	[CAN]	56	0011 1000	38	8	88	0101 1000	58	X	120	0111 1000	78	x
25	0001 1001	19	[EM]	57	0011 1001	39	9	89	0101 1001	59	Y	121	0111 1001	79	y
26	0001 1010	1A	[SUB]	58	0011 1010	3A	:	90	0101 1010	5A	Z	122	0111 1010	7A	z
27	0001 1011	1B	[ESC]	59	0011 1011	3B	;	91	0101 1011	5B	[	123	0111 1011	7B	{
28	0001 1100	1C	[FS]	60	0011 1100	3C	<	92	0101 1100	5C	\	124	0111 1100	7C	|
29	0001 1101	1D	[GS]	61	0011 1101	3D	=	93	0101 1101	5D	]	125	0111 1101	7D	}
30	0001 1110	1E	[RS]	62	0011 1110	3E	>	94	0101 1110	5E	^	126	0111 1110	7E	~
31	0001 1111	1F	[US]	63	0011 1111	3F	?	95	0101 1111	5F	_	127	0111 1111	7F	[DEL]

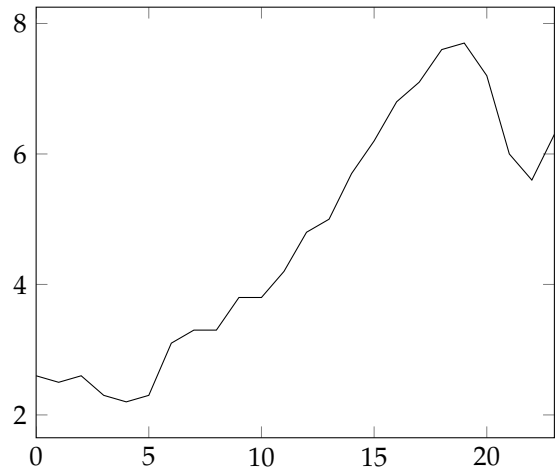
Diagramme de Venn Parfois il est plus visuel de montrer les différents liens sur un diagramme de Venn.



Formule algébrique Lorsque l'ensemble E est infini, on ne peut pas stocker toutes les valeurs, on peut définir la fonction par une formule :

$$\begin{array}{rccc} f: & \mathbb{R} & \longrightarrow & \mathbb{R} \\ & x & \longmapsto & 3x^2+2x-5 \end{array}$$

Courbe On peut aussi représenter la fonction sous forme de courbe qui à chaque point de l'abscisse fait correspondre un élément de l'ordonnée.



Algorithme On peut aussi définir une fonction $f : E \to F$ par un algorithme qui prend en argument un élément de E et lorsqu'il s'arrête, il renvoie un élément de F. Le domaine de définition est l'ensemble des valeurs pour lesquelles l'algorithme s'arrête.

III.1.3 Composition de fonction et d'applications

Définition III.6 (Composition)**.** On considère les fonctions $f : E \to F$ et $g : F \to G$. On définit la *fonction composée* de f par g, notée $g \circ f : E \to G$, définie par $g \circ f(x) = g(f(x))$. On applique f à l'argument x, puis on applique g au résultat s'il existe.

Le domaine de définition de $g \circ f$ est donné par :

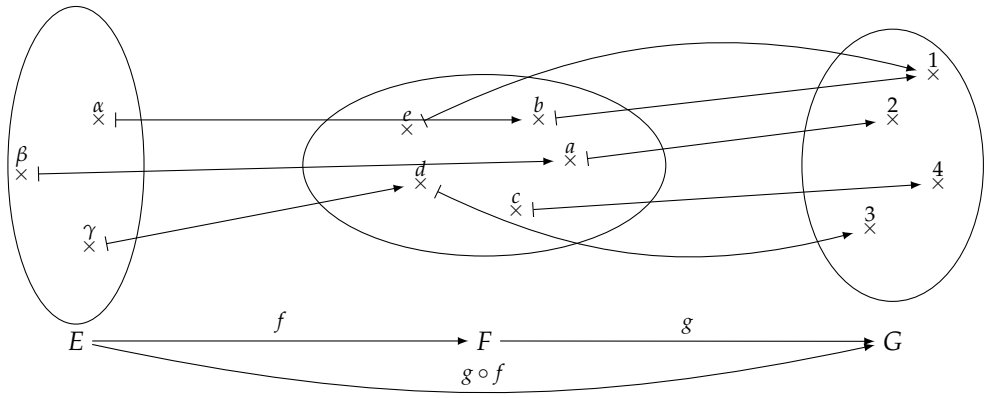
$$\mathrm{Dom}(g \circ f) = \{x \in \mathrm{Dom}(f) : f(x) \in \mathrm{Dom}(g)\}$$

Attention, en général, on considère la composée d'applications pour s'abstraire des contraintes de domaines de définition.

Exemple III.4. Soient $E = \{\alpha, \beta, \gamma\}$, $F = \{a, b, c, d, e\}$ et $G = \{1, 2, 3, 4\}$, on définit $f : E \to F$ et $g : F \to G$ par :

$$\begin{array}{rccc} f: & E & \longrightarrow & F \\ & \alpha & \longmapsto & b \\ & \beta & \longmapsto & a \\ & \gamma & \longmapsto & d \end{array} \qquad \begin{array}{rccc} g: & F & \longrightarrow & G \\ & a & \longmapsto & 2 \\ & b & \longmapsto & 1 \\ & c & \longmapsto & 4 \\ & d & \longmapsto & 3 \\ & e & \longmapsto & 1 \end{array}$$

La représentation avec le diagramme de Venn donne :



Ainsi la fonction $f \circ g : E \longrightarrow G$ donne les associations suivantes :

$$\alpha \longmapsto 1 \qquad \beta \longmapsto 2 \qquad \gamma \longmapsto 3$$

Remarque III.1. Au milieu du XX$^{\text{ème}}$ siècle, quelques mathématiciens trouvèrent que la notation $g \circ f$ portait à confusion et décidèrent d'utiliser une notation post-fixée : xf pour $f(x)$ et xfg pour $g \circ f(x)$.

Remarque III.2. Attention en général $f \circ g \neq g \circ f$.

Proposition III.1 Associativité de la composée

Soient $f : E \to F$, $g : F \to G$ et $h : G \to H$ trois applications. Alors on a $h \circ (g \circ f) = (h \circ g) \circ f$ et cette application se note $h \circ g \circ f$.

Démonstration : Par définition de la composition d'applications, il vient $h \circ (g \circ f)(x) = h(g \circ f(x)) = h(g(f(x)))$ et $(h \circ g) \circ f(x) = h \circ g(f(x)) = h(g(f(x)))$ pour tout $x \in E$ d'où l'égalité recherchée. ■

III.1.4 Applications singulières

Définition III.7 (Identité)**.** Etant donné un ensemble A, la fonction *identité* est l'application définie par

$$\begin{array}{rccc} \mathrm{Id}_A : & A & \longrightarrow & A \\ & x & \longmapsto & x \end{array}$$

Définition III.8 (Injection canonique)**.** Soit $A \subseteq B$, l'*injection canonique* est l'application définie par

$$\begin{array}{rccc} f : & A & \longrightarrow & B \\ & x & \longmapsto & x \end{array}$$

Définition III.9 (Projection canonique)**.** Soit $A_1 \times \cdots \times A_k$ le produit cartésien de k ensemble, la *projection canonique suivant la $i^{\text{ème}}$ coordonnée* pour $i \in \{1, \ldots, k\}$ est l'application définie par

$$\begin{array}{rccc} \pi_i : & A_1 \times \cdots \times A_k & \longrightarrow & A_i \\ & (a_1, \ldots, a_k) & \longmapsto & a_i \end{array}$$

III.2 Propriétés sur les fonctions

III.2.1 Injection et surjection

Définition III.10 (Fonction injective)**.** Une fonction est *injective* si tout élément de l'espace d'arrivée admet au plus un antécédent.

Définition III.11 (Fonction surjective)**.** Une fonction est *surjective* si chaque élément de l'espace d'arrivée admet au moins un antécédent.

Autrement dit, si $f : E \to F$ est une fonction alors $\mathrm{Im}(f) = f(E) = F$.

III.2.2 Bijection et application réciproque

Définition III.12 (Application bijective)**.** Une application qui est à la fois injective et surjective est bijective.

Attention, en général, la notion de bijection est utilisé pour les applications.

Proposition III.2

L'application $f : E \to F$ est bijective si et seulement si il existe une application $g : F \to E$ telle que $f \circ g = Id_F$ et $g \circ f = Id_E$.

*Si f est bijective, l'application g est unique, c'est l'*application réciproque *de l'application f, notée f^{-1}.*

Démonstration : Si $f : E \to F$ est bijective alors à chaque élément $x \in E$ correspond un et un seul élément $y \in F$, c'est la définition de l'application. De même, à chaque élément $y \in F$ correspond un élément $x \in E$ (c'est la surjectivité), et cet élément est unique (c'est l'injectivité). Cela permet de définir une fonction $g : F \to E$ qui vérifie

$$x = g(y) \text{ si et seulement si } f(x) = y.$$

On en déduit que $f \circ g = \mathrm{Id}_F$ et $g \circ f = \mathrm{Id}_E$.

Montrons qu'une telle application est unique. Soit $h : F \to E$ telle que $f \circ h = \mathrm{Id}_F$ et $h \circ f = \mathrm{Id}_E$. On a $f \circ h(y) = f \circ g(y) = \mathrm{Id}_F(y)$ pour tout $y \in F$, par injectivité de f, on en déduit que $h(y) = g(y)$. Les fonctions g et h sont donc égales.

Supposons maintenant qu'il existe une fonction $g : F \to E$ telle que $f \circ g = \mathrm{Id}_F$ et $g \circ f = \mathrm{Id}_E$ et montrons que f est bijective :

— Soient $x_1, x_2 \in E$ tels que $f(x_1) = f(x_2)$, on a donc $g(f(x_1)) = g(f(x_2))$ et comme $g \circ f = \mathrm{Id}_E$ on en déduit que $x_1 = x_2$ et donc que f est injective.

— Soient $y \in F$, on a $y = \mathrm{Id}_F(y) = f(g(y))$. Ainsi, $g(y)$ est une préimage de y par f, on en déduit que f est surjective.

f est injective et surjective, on en déduit que f est bijective. ■

Exemple III.5. L'application de $\mathbb{Q}$ dans $\mathbb{Q}$ définie par $f : x \longmapsto 2x$ est :

— surjective car pour tout $y \in \mathbb{Q}$ on a $\frac{y}{2} \in \mathbb{Q}$ et $f(\frac{y}{2}) = y$;

— injective car si $x, y \in \mathbb{Q}$ vérifient $f(x) = f(y)$ alors $2x = 2y$ autrement dit $x = y$.

La fonction f est donc bijective et son application réciproque est $f^{-1} : x \longmapsto \frac{x}{2}$.

Proposition III.3

Soient $f : E \to F$ et $g : F \to G$ deux bijections. La composée $g \circ f$ est bijective et

$$(g \circ f)^{-1} = f^{-1} \circ g^{-1}.$$

Démonstration : On a :

$$(g \circ f) \circ (f^{-1} \circ g^{-1}) = g \circ f \circ f^{-1} \circ g^{-1} = g \circ \mathrm{Id}_F \circ g^{-1} = g \circ g^{-1} = \mathrm{Id}_G.$$

De même, on a :

$$(f^{-1} \circ g^{-1}) \circ (g \circ f) = f^{-1} \circ g^{-1} \circ g \circ f = f^{-1} \circ \mathrm{Id}_F \circ f = f^{-1} \circ f = \mathrm{Id}_E.$$

On en déduit que $g \circ f$ est bijective et par unicité de l'application réciproque, on a $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$. ■

III.3 Quelques classes importantes de fonctions

III.3.1 Fonction caractéristique d'un ensemble

Définition III.13. Soit Ω un ensemble pour tout $A \subseteq \Omega$ on définit la *fonction caractéristique* de l'ensemble A par

$$\begin{array}{rccl} \mathbf{1}_A : & \Omega & \longrightarrow & \{0,1\} \\ & x & \longmapsto & \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases} \end{array}$$

La fonction caractéristique d'un ensemble permet de définir un ensemble de manière fonctionnelle. Ainsi, on a :

$$A = \{x \in \Omega : \mathbf{1}_A(x) = 1\}$$

Proposition III.4 Propriétés des fonctions caractéristiques

Soient $A, B \in \mathcal{P}(\Omega)$, pour tout $x \in \Omega$, on a :

1. $\mathbf{1}_{A\cap B}(x) = \mathbf{1}_A(x) \times \mathbf{1}_B(x)$
2. $\mathbf{1}_{A\cup B}(x) = \mathbf{1}_A(x) + \mathbf{1}_B(x) - \mathbf{1}_{A\cap B}(x)$
3. $\mathbf{1}_{\overline{A}}(x) = 1 - \mathbf{1}_A(x)$

Exemple III.6. A l'aide des fonctions caractéristiques on peut retrouver les formules classiques, par exemple pour tout $x \in \Omega$, on a :

$$\begin{aligned}
\mathbf{1}_{\overline{A\cup B}}(x) &= 1 - (\mathbf{1}_A(x) + \mathbf{1}_B(x) - \mathbf{1}_{A\cap B}(x)) \\
&= 1 - \mathbf{1}_A(x) - \mathbf{1}_B(x) + \mathbf{1}_A(x)\mathbf{1}_B(x) \\
&= (1 - \mathbf{1}_A(x))(1 - \mathbf{1}_B(x)) \\
&= \mathbf{1}_{\overline{A}}(x)\mathbf{1}_{\overline{B}}(x) \\
&= \mathbf{1}_{\overline{A}\cap\overline{B}}(x)
\end{aligned}$$

Comme l'égalité est vraie pour tout $x \in \Omega$, on en déduit que $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

III.3.2 Suites

Soit $\mathbb{K}$ un ensemble, une *suite à valeurs dans* $\mathbb{K}$ est une application de $\mathbb{N}$ dans $\mathbb{K}$. On note $\mathbb{K}^{\mathbb{N}}$ l'ensemble des suite à valeurs dans $\mathbb{K}$. Etant donnée une suite $u \in \mathbb{K}^{\mathbb{N}}$, on note souvent u_n le $n^{\text{ème}}$ élément de la suite et $u = (u_n)_{n\in\mathbb{N}}$.

Chapitre IV

Cardinalité

En informatique, la résolution algorithmique de nombreux problèmes consiste en l'énumération exhaustive de ses possibilités pour ensuite décider pour chacune si elle est solution ou non au problème. Il est parfois bon d'évaluer le nombre de possibilités afin d'évaluer le temps d'exécution de l'algorithme.

IV.1 Cardinalité des ensembles finis

IV.1.1 Ensembles de même cardinalité

Considérons les ensembles $E = \{a, b, c, d\}$ et $F = \{1, 2, 3\}$. Il est possible de définir une application surjective de E sur F, mais pas d'application injective. Il est possible de définir une application injective de F sur E, mais pas d'application surjective. En fait, il n'y a pas assez d'éléments dans F (ou trop peu dans E). Le cardinal d'un ensemble précise la notion de nombre d'éléments.

Définition IV.1 (Ensemble de même cardinal)**.** Deux ensembles (fini ou non) sont *équipotents* ou de *même cardinal* s'il existe une bijection entre eux.

Remarque IV.1. Par abus de langage on dit qu'il existe une surjection (respectivement une injection, une bijection) s'il existe une application surjective (respectivement une application injective, une application bijective).

IV.1.2 Cardinal d'un ensemble fini

Lemme IV.1

Soient n et k deux entiers naturels.
- *S'il existe une application injective de $\{1, \dots, n\}$ dans $\{1, \dots, k\}$ alors $n \leq k$.*
- *S'il existe une application surjective de $\{1, \dots, n\}$ dans $\{1, \dots, k\}$ alors $n \geq k$.*
- *S'il existe une application bijection de $\{1, \dots, n\}$ dans $\{1, \dots, k\}$ alors $n = k$.*

Démonstration : Montrons par récurrence sur $n \in \mathbb{N}^*$ la propriété P_n : pour tout $k \in \mathbb{N}$, s'il existe une application injective de $\{1, \dots, n\}$ dans $\{1, \dots, k\}$ alors $n \leq k$.

Clairement cette propriété est vraie pour P_1 car il n'y a pas d'application d'un ensemble non vide dans un ensemble vide.

Supposons que P_n est vérifiée et montrons que P_{n+1} est aussi vérifiée. Soit $f : \{1, \dots, n+1\} \to \{1, \dots, k\}$ une injection. Pour tout $x \in \{1, \dots, n\}$, on a $f(x) \neq f(n+1)$ car f est une injection. On a deux cas possibles :
- Si $f(x) < f(n+1)$ pour tout $x \in \{1, \dots, n\}$ alors $f(x) \in \{1, \dots, k-1\}$ car $f(x) \leq f(n+1) - 1 \leq k - 1$. En utilisant l'hypothèse de récurrence, on en déduit que $n \leq k - 1$ et donc $n + 1 \leq k$.

— Sinon, on considère $x_0 \in \{1,\dots,n\}$ tel que $f(x_0) > f(x)$ pour tout $x \in \{1,\dots,n+1\} \setminus \{x_0\}$. On définit la fonction $g : \{1,\dots,n+1\} \to \{1,\dots,k\}$ telle que $g(x) = f(x)$ pour tout $x \in \{1,\dots,n+1\} \setminus \{x_0, n+1\}$, $g(x_0) = f(n+1)$ et $g(n+1) = f(x_0)$. Comme f est une injection, les éléments de $\{1,\dots,n+1\}$ n'ont pas deux images identiques par g donc g est injective. De plus $g(x) < g(n+1)$ pour tout $x \in \{1,\dots,n\}$, en réalisant le même raisonnement que le point précédent, on en déduit que $n+1 \leq k$.

Par récurrence la propriété P_n est vérifiée pour tout $n \in \mathbb{N}$ ce qui montre le premier point du lemme.

Le deuxième point se montre de la même manière.

Si $f : \{1,\dots,n\} \to \{1,\dots,k\}$ est bijective, elle est injective d'où $n \leq k$, et elle est surjective d'où $n \geq k$. On en déduit que $n = k$. ■

Si on a une bijection $f : E \to \{1,\dots,n\}$ et une bijection $g : E \to \{1,\dots,m\}$ alors $g \circ f^{-1}$ réalise une bijection de $\{1,\dots,n\}$ dans $\{1,\dots,m\}$. On en déduit que $n = m$ ce qui nous permet de définir le cardinal d'un ensemble fini.

Définition IV.2 (Cardinal d'un ensemble fini)**.** Un ensemble E est *fini* s'il est vide ou s'il existe $n \in \mathbb{N}^*$ tel que E est en bijection avec $\{1,\dots,n\}$. Cet entier est unique, il est appelé le *cardinal* de E. On le note $\mathrm{Card}\,(E)$. Si E est vide, on pose $\mathrm{Card}\,(E) = 0$.

Remarque IV.2. Le cardinal d'un ensemble fini correspond à l'idée naturelle du nombre d'éléments d'un ensemble.

Proposition IV.2

Soient E et F deux ensembles finis. On a :
— *Il existe une application injective de E dans F si et seulement si $\mathrm{Card}\,(E) \leq \mathrm{Card}\,(F)$.*
— *Il existe une application surjective de E dans F si et seulement si $\mathrm{Card}\,(E) \geq \mathrm{Card}\,(F)$.*
— *Il existe une application bijective de E dans F si et seulement si $\mathrm{Card}\,(E) = \mathrm{Card}\,(F)$.*

Démonstration : Soient $g : E \to \{1,\dots,\mathrm{Card}\,(E)\}$ et $h : F \to \{1,\dots,\mathrm{Card}\,(F)\}$ deux bijections définissant le cardinal de E et F.

Montrons le premier point. Soit $f : E \to F$ une injection, $h \circ f \circ g^{-1}$ est une injection de $\{1,\dots,\mathrm{Card}\,(E)\}$ dans $\{1,\dots,\mathrm{Card}\,(F)\}$, donc $\mathrm{Card}\,(E) \leq \mathrm{Card}\,(F)$. Réciproquement, supposons que $\mathrm{Card}\,(E) \leq \mathrm{Card}\,(F)$. On définit $\varphi = h^{-1} \circ i \circ f$ où $i : x \mapsto x$ est l'injection canonique de $\{1,\dots,\mathrm{Card}\,(E)\}$ dans $\{1,\dots,\mathrm{Card}\,(F)\}$. L'application φ est une injection de E dans F.

Montrons le deuxième point. Soit $f : E \to F$ une surjection, $h \circ f \circ g^{-1}$ est une surjection de $\{1,\dots,\mathrm{Card}\,(E)\}$ dans $\{1,\dots,\mathrm{Card}\,(F)\}$, donc $\mathrm{Card}\,(E) \geq \mathrm{Card}\,(F)$. Réciproquement, supposons que $\mathrm{Card}\,(E) \geq \mathrm{Card}\,(F)$. On définit $\varphi = h^{-1} \circ s \circ f$ où $s : \{1,\dots,\mathrm{Card}\,(E)\} \to \{1,\dots,\mathrm{Card}\,(F)\}$ est définie par $s(x) = x$ si $x \in \{1,\dots,\mathrm{Card}\,(F)\}$ et $s(x) = 1$ si $x \in \{\mathrm{Card}\,(F)+1,\dots,\mathrm{Card}\,(E)\}$. L'application s est surjective et donc φ est une surjection de E dans F.

Pour le dernier point, on considère une bijection $f : E \to F$. Alors $h \circ f \circ g^{-1}$ est une bijection de $\{1,\dots,\mathrm{Card}\,(E)\}$ dans $\{1,\dots,\mathrm{Card}\,(F)\}$, on en déduit que $\mathrm{Card}\,(E) = \mathrm{Card}\,(F)$. Réciproquement, supposons que $\mathrm{Card}\,(E) = \mathrm{Card}\,(F) = n$ et considérons deux bijections $g : E \to \{1,\dots,n\}$ et $h : F \to \{1,\dots,n\}$. On en déduit que $h^{-1} \circ g$ est une bijection de E dans F. ■

IV.1.3 Principe des tiroirs

La contraposée du premier point de la proposition IV.2 établit un principe naturel très utilisé en combinatoire appelé "principe des tiroirs" : si on doit ranger plus de paires de chaussettes qu'on a de tiroirs, alors forcément un tiroir recevra au moins deux paires. Formellement cela se traduit par la proposition suivante.

Proposition IV.3 Principe des tiroirs

Soient E et F deux ensembles finis non vides et $f : E \to F$ une application. Si $\mathrm{Card}\,(E) > \mathrm{Card}\,(F)$ alors il existe $x_1, x_2 \in E$ tels que $f(x_1) = f(x_2)$.

Démonstration : On peut faire une preuve directe de ce résultat. Comme E est fini on note $E = \{x_1, \dots, x_{\text{Card}(E)}\}$. Soit $f : E \to F$ et supposons que $f(x_i) \neq f(x_j)$ si $i \neq j$. On en déduit que $f(x_1), \dots, f(x_{\text{Card}(E)})$ sont tous distincts et sont des éléments de F ainsi $\text{Card}(E) \leq \text{Card}(F)$. On en déduit que si $\text{Card}(E) > \text{Card}(F)$ alors au moins deux éléments de E ont la même image. ■

Exemple IV.1. Il y a au moins deux personnes à Paris qui ont exactement le même nombre de cheveux. Le nombre moyen de cheveux chez un humain est de 150000 donc raisonnablement, personne n'en a plus d'un million. Or il y a plus d'un million d'habitants à Paris, donc au moins deux habitants ont le même nombre de cheveux.

Il est possible d'étendre le principe des tiroirs.

Proposition IV.4

Soient E et F deux ensembles finis non vides et $f : E \to F$ une application. Si $\text{Card}(E) > k\,\text{Card}(F)$ avec $k \in \mathbb{N}^*$ alors il existe une valeur de f qui est répétée au moins $k+1$ fois.

Démonstration : Si chaque valeur de f est répétée au plus k fois alors E contient au plus $k\,\text{Card}(F)$ éléments. Ainsi si $\text{Card}(E) > k\,\text{Card}(F)$, une valeur de f est répétée au moins $k+1$ fois. ■

Exemple IV.2. Voyons combien de noms différents doivent apparaître dans l'annuaire pour qu'au moins cinq noms commencent et terminent par la même lettre. Soient E l'ensemble des noms et F l'ensemble des paires de lettres qu'il est possible de constituer avec un alphabet de 26 lettres. Clairement $\text{Card}(F) = 26 \times 26 = 676$. On considère la fonction $f : E \to F$ qui renvoie pour chaque nom la paire de lettre correspondant aux premières et dernières lettre du nom (par exemple $f(\text{maurice}) = (m, e)$). Pour avoir cinq noms qui commence par la même lettre et termine par la même lettre il faut que $\text{Card}(E) > 4\,\text{Card}(F) = 4 \times 676 = 2704$. Ainsi il faut que l'annuaire contienne au moins 2705 noms.

Exemple IV.3. Montrons que dans tout groupe de six personnes, trois se connaissent mutuellement ou trois ne se sont jamais vues. Considérons x une personne, E l'ensemble des cinq autres personnes et $\mathbb{B} = \{0, 1\}$. Prenons la fonction $f : E \to \mathbb{B}$ tel que $f(a) = 1$ si $a \in E$ connaît x et 0 sinon. Comme $\text{Card}(E) > 2\,\text{Card}(\mathbb{B})$, soit trois personnes connaissent x, soit il ne connaissent pas x.

Supposons que a, b et c connaissent x. Si deux d'entre eux se connaissent, avec x on a trouvé trois personnes qui se connaissent mutuellement. Sinon a, b et c ne se connaissent pas.

Le même type de raisonnement fonctionne si a, b et c ne connaissent pas x.

IV.2 Dénombrement

On cherche le cardinal d'un ensemble fini, c'est à dire à dénombrer le nombre d'éléments de cet ensemble. Dans votre scolarité en informatique vous utiliserez la notion de dénombrement au moins dans les deux cas de figures suivants :

— dénombrer le nombre de cas à analyser par un algorithme en vu d'étudier sa complexité ;

— lorsqu'on tire au hasard un élément dans un univers finis Ω de manière équiprobable (c'est à dire que chaque élément à la même probabilité d'être tiré), la probabilité que cet élément soit dans l'ensemble $A \subseteq \Omega$ est

$$\mathbf{P}(A) = \frac{\text{Card}(A)}{\text{Card}(\Omega)}.$$

IV.2.1 Dénombrement et opération sur les ensembles

On répertorie ici les principes de bases pour dénombrer un ensemble. Cela revient à décomposer l'ensemble considéré en ensembles plus simples à l'aide des opérations sur les ensembles.

Union et intersection

Si A et B sont deux ensembles disjoints, c'est à dire $A \cap B = \emptyset$, on a :

$$\text{Card}(A \cup B) = \text{Card}(A) + \text{Card}(B).$$

Si $(A_i)_{i\in I}$ est une famille d'ensembles disjoints indexée par I, on a :

$$\mathrm{Card}\left(\bigcup_{i\in I} A_i\right) = \sum_{i\in I} \mathrm{Card}\,(A_i)\,.$$

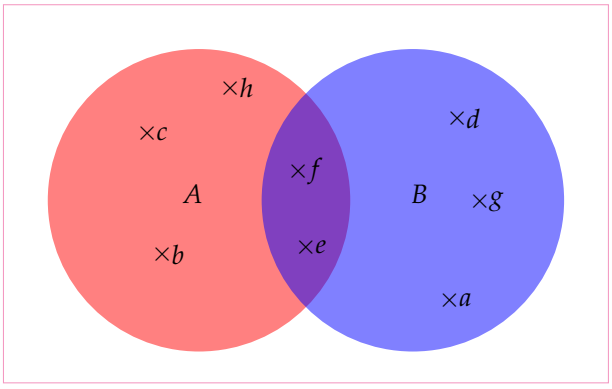
Lorsque A et B sont deux ensembles quelconques, on cherche à écrire l'union comme une union disjointe d'ensembles. Ainsi $A \cup B = (A \setminus B) \cup (B \setminus A) \cup (A \cap B)$ et l'union est disjointe ainsi

$$\mathrm{Card}\,(A \cup B) = \mathrm{Card}\,(A \setminus B) + \mathrm{Card}\,(B \setminus A) + \mathrm{Card}\,(A \cap B)$$

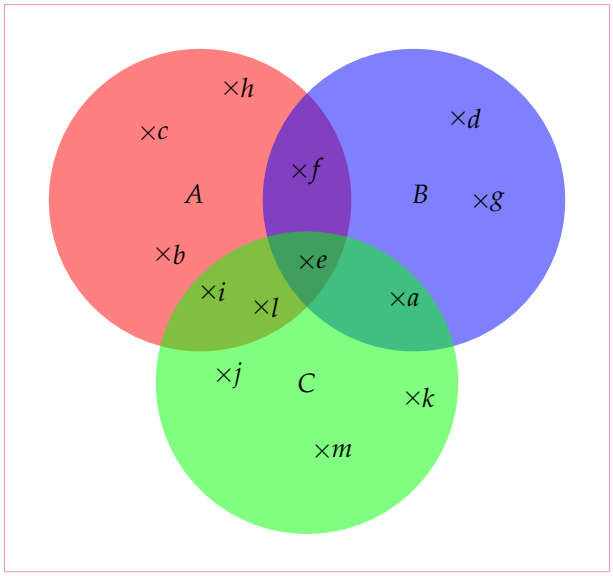
On a aussi $A = (A \setminus B) \cup (A \cap B)$ et $B = (B \setminus A) \cup (A \cap B)$ et les deux unions sont disjointes. On en déduit que $\mathrm{Card}\,(A) = \mathrm{Card}\,(A \setminus B) + \mathrm{Card}\,(A \cap B)$ et $\mathrm{Card}\,(A \cup B) = \mathrm{Card}\,(B \setminus A) + \mathrm{Card}\,(A \cap B)$. On en déduit que

$$\mathrm{Card}\,(A \cup B) = \mathrm{Card}\,(A) + \mathrm{Card}\,(B) - \mathrm{Card}\,(A \cap B)$$

Sur le diagramme de Venn cela donne :



Lorsqu'on fait l'opération $\mathrm{Card}\,(A) + \mathrm{Card}\,(B)$, les éléments e et f sont comptés deux fois.
Pour compter le nombre d'éléments dans $A \cup B$, on doit donc retrancher $\mathrm{Card}\,(A \cap B)$ à $\mathrm{Card}\,(A) + \mathrm{Card}\,(B)$.



Lorsqu'on fait l'opération $\mathrm{Card}\,(A) + \mathrm{Card}\,(B) + \mathrm{Card}\,(C)$, les éléments a, f, i et l sont comptés deux fois et l'élément e est compté trois fois.
Lorsqu'on fait l'opération $\mathrm{Card}\,(A \cap B) + \mathrm{Card}\,(B \cap C) + \mathrm{Card}\,(C \cap A)$, les éléments a, f, i et l sont comptés une fois et l'élément e est compté trois fois.
Ainsi dans l'expression

$$\big(\mathrm{Card}\,(A) + \mathrm{Card}\,(B) + \mathrm{Card}\,(C)\big) - \big(\mathrm{Card}\,(A \cap B) + \mathrm{Card}\,(B \cap C) + \mathrm{Card}\,(C \cap A)\big)$$

l'élément e n'est pas comptabilisé, il faut donc rajouter à cet somme la quantité $\mathrm{Card}\,(A \cap B \cap C)$ pour obtenir $\mathrm{Card}\,(A \cup B \cup C)$.

On a donc la proposition suivante.

Proposition IV.5

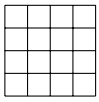
$$\begin{aligned} \mathrm{Card}\,(A \cup B) &= \mathrm{Card}\,(A) + \mathrm{Card}\,(B) - \mathrm{Card}\,(A \cap B) \\ \mathrm{Card}\,(A \cup B \cup C) &= \mathrm{Card}\,(A) + \mathrm{Card}\,(B) + \mathrm{Card}\,(C) - \mathrm{Card}\,(A \cap B) - \mathrm{Card}\,(A \cap C) \\ &\quad - \mathrm{Card}\,(B \cap C) + \mathrm{Card}\,(A \cap B \cap C) \end{aligned}$$

Remarque IV.3. Cela correspond au **principe de l'addition** : si on cherche à dénombrer un événement où l'on s'est ramené à considérer un cas **ou bien** un autre **ou bien** un autre, etc..., cela revient à dénombrer une **union** de sous-ensembles, ce qui revient à effectuer la **somme** des cardinaux de chaque sous-ensemble éventuellement en réajustant les intersections.

Remarque IV.4. Il existe une formule générale (non exigible) pour une union finie d'ensembles $(A_i)_{i\in\{1,\dots,n\}}$:

$$\text{Card}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n}(-1)^{i+1} \sum_{I\subseteq\{1,\dots,n\} \text{ avec Card}(I)=i} \text{Card}\left(\bigcap_{i\in I} A_i\right)$$

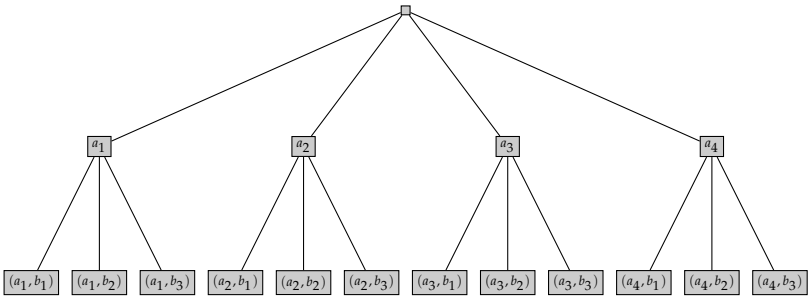
Exemple IV.4 (Nombre de carrés). On cherche à compter le nombre de carrés dans la figure ci dessous :



On note A_1, A_2, A_3 et A_4 les ensembles de carrés dont les côtés sont respectivement 1, 2, 3 et 4. L'ensemble des carrés peut donc s'écrire comme une union disjointe : $A = A_1 \cup A_2 \cup A_3 \cup A_4$. Donc Card $(A) = 16 + 9 + 4 + 1 = 30$.

Produit cartésien

Soient A et B deux ensembles finis. L'ensemble $A \times B$ correspond aux couples d'éléments (a, b) où $a \in A$ et $b \in B$. Ainsi on a Card (A) possibilités pour choisir le premier éléments et une fois que l'on a choisi un élément de A, il y a Card (B) possibilités pour choisir $b \in B$. On en déduit que $A \times B$ contient Card $(A) \times$ Card (B) éléments. On peut représenter les différents choix à l'aide d'un arbre comme sur la figure IV.1.



$A = \{a_1, a_2, a_3, a_4\}$

$B = \{b_1, b_2, b_3\}$

Card $(A \times B) = 4 \times 3 = 12$

FIGURE IV.1 – Représentation sous forme d'arbre du produit cartésien de deux ensembles finis.

Ce raisonnement se généralise facilement au cas où on fait un produit cartésien d'une famille finie d'ensembles finis.

Proposition IV.6

Soient A et B deux ensembles finis, on a :

$$\textit{Card}\,(A \times B) = \textit{Card}\,(A) \times \textit{Card}\,(B)$$

Et plus généralement, si $(A_i)_{i\in\{1,\dots,n\}}$ est une famille finie d'ensembles finis alors :

$$\textit{Card}\,(A_1 \times \cdots \times A_n) = \textit{Card}\,(A_1) \times \cdots \times \textit{Card}\,(A_n)$$

Remarque IV.5. Cela correspond au **principe de la multiplication** : si on cherche à dénombrer un événement où l'on s'est ramené à considérer un cas pris dans un ensemble, **puis** un autre dans un autre ensemble, **puis** un autre, etc..., on effectue le **produit** des cardinaux de chaque ensemble.

Exemple IV.5 (Nombre de mots). On considère un alphabet $\mathcal{A}$ de cardinal Card $(\mathcal{A}) = m$, on souhaite compter le nombre de mots de longueur n noté $\mathcal{A}^n$ (par exemple pour connaitre le nombre de mots de passe possibles).

Si $\mathcal{A} = \{a, b\}$ et $n = 4$, le nombre de mots de longueur 4 qui peuvent être composés à partir des lettres a et b est $2^4 = 16$, on peut les énumérer :

$$aaaa, aaab, aaba, aabb, abaa, abab, abba, abbb, baaa, baab, baba, babb, bbaa, bbab, bbba, bbbb$$

Exemple IV.6 (Nombre de noms de variables BASIC). Les noms des variables du langage BASIC se définissent ainsi :

<Lettre>::= A | B | ... | Z
<Digit> ::= 0 | 1 | ... | 9
<Var> ::= <Lettre> | <Lettre><Digit>

Ainsi une variable est une lettre ou bien une lettre composée d'un digit. Le nombre de variables est donc :

$$\text{Card}\,(\texttt{<Var>}) = \text{Card}\,(\texttt{<Lettre>}\,|\,) + \text{Card}\,(\texttt{<Lettre>}).\text{Card}\,(\texttt{<Digit>}) = 26 + (26 * 10) = 286$$

Passage au complémentaire

Parfois il est plus facile de dénombrer le complémentaire d'un ensemble. Par exemple, si $A \subseteq \Omega$ et que l'on connaît Card (Ω) et Card $\left(\overline{A}\right)$, alors Ω est une union disjointe de A et $\overline{A}$, on en déduit que Card $(\Omega) =$ Card $(A) +$ Card $\left(\overline{A}\right)$. Ainsi on a

$$\text{Card}\,(A) = \text{Card}\,(\Omega) - \text{Card}\,\left(\overline{A}\right)$$

IV.2.2 Arrangements et combinaisons

On peut aussi utiliser les formules classiques d'arrangements et de combinaisons. Avant d'utiliser les différentes formules, il est commode de se poser les questions suivantes :

— Quel est le nombre n d'objets de référence ?
— Quel est le nombre p d'objets concernés ($p \leq n$) par la situation considérée ?
— Les p objets sont-ils considérés en "vrac" (sans ordre, tirage simultané), ou sont ils classés d'une certaine façon (avec ordre, tirage successif) ?

Arrangement

Proposition IV.7

Permutation de n éléments : *c'est le nombre de façon de ranger n objets dans l'ordre.*

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1$$

Démonstration : Le nombre de permutations de n éléments peut être calculé de la façon suivante : il y a n places possibles pour un premier élément, $n-1$ pour un deuxième élément, ..., et il ne restera qu'une place pour le dernier élément restant. On remarque facilement alors qu'il y a $n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1$ permutations possibles. On note $n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1 = n!$ et par convention, $0! = 1$. ■

Exemple IV.7. Voici les $4! = 24$ permutations de quatre éléments distinct a, b, c et d :

abcd	*abdc*	*acbd*	*acdb*	*adbc*	*adcb*
bacd	*badc*	*bcad*	*bcda*	*bdac*	*bdca*
cabd	*cadb*	*cbad*	*cdba*	*cdab*	*cdba*
dabc	*dacb*	*dbac*	*dbca*	*dcab*	*dcba*

Exemple IV.8. De combien de façons pouvez-vous ranger 10 livres sur une étagère ?

$$10! = 3628800$$

De combien de façons peut-on mélanger un jeu de 36 cartes ?

$$36! = 3,72.10^{41}$$

Proposition IV.8

Arrangements de p éléments parmi n sans répétition : *c'est le nombre de listes de p éléments parmi n, les éléments sont donc ordonnés dans la liste.*

$$A_n^p = n \times (n-1) \times (n-2) \times \cdots \times (n-p+1) = \frac{n!}{(n-p)!}$$

Démonstration : Le premier élément peut être choisi de n façons différents, le deuxième peut prendre $n-1$ valeurs, le troisième $n-2$ valeurs et le p-ième élément $n-p+1$ valeurs. D'où :

$$A_n^p = n \times (n-1) \times (n-2) \times \cdots \times (n-p+1) = \frac{n!}{(n-p)!}$$ ■

Exemple IV.9. Les $A_4^3 = 4 \times 3 \times 32 = 24$ arrangements de 3 éléments choisis parmi a, b, c, d :

abc	*abd*	*acb*	*acd*	*adb*	*adc*
bac	*bad*	*bca*	*bcd*	*bda*	*bdc*
cab	*cad*	*cba*	*cdb*	*cda*	*cdb*
dab	*dac*	*dba*	*dbc*	*dca*	*dcb*

Exemple IV.10. Après les prolongations d'un match de football, l'entraîneur doit choisir les 5 tireurs de penaltys parmi les onze joueurs et l'ordre de passage. Combien de choix a-t-il ?

$$A_{11}^5 = 11 \times 10 \times 9 \times 8 \times 7 = 55440$$

Exemple IV.11. Le bingo est un jeu où les nombres tirés sont annoncés les uns à la suite des autres. S'il y a 90 numéros en tout dans un sac, combien de suites différentes peut-on former avec les 10 premiers numéros tirés ?

$$A_{90}^{10} \simeq 2,076.10^{19}$$

Proposition IV.9

Arrangement de p éléments parmi n avec répétition : *c'est le nombre de listes de p éléments parmi n, mais on s'autorise des répétitions éventuelles des éléments. Les éléments sont ordonnés dans la liste.*

$$n^p$$

Démonstration : Le premier élément peut être choisi de n façons différents, le deuxième d n façons, le troisième n valeurs... Il y a donc n^p possibilités. ■

Exemple IV.12. Les $3^2 = 9$ arrangements avec répétitions de 2 éléments choisis parmi a, b, c :

aa *ab* *ac* *ba* *bb* *bc* *ca* *cb* *cc*

Exemple IV.13. Combien de numéros de téléphone composés de 7 chiffres existe-t-il ?

$$10^7$$

On a 6 clochettes produisant chacune un son différent des autres. On veut faire une mélodie de 10 notes avec ces clochettes. Combien de possibilités a-t-on ?

$$6^{10} = 60466176$$

Proposition IV.10

Soient E et F deux ensembles finis, le cardinal de l'ensemble des applications de E dans F, noté F^E, est :

$$Card\left(F^E\right) = Card\,(F)^{Card(E)}$$

Démonstration : Pour définir une fonction de E dans F, il faut associé à chaque élément de E un élément de F. Pour chaque élément de E on a donc Card (F) possibilités. ■

Combinaison

Proposition IV.11

Combinaisons de p éléments parmi n sans répétition : *c'est le nombre de sous-ensembles de p éléments dans un ensemble contenant n éléments, les éléments ne sont donc pas ordonnés.*

$$C_n^p = \frac{n!}{p!(n-p)!}$$

Démonstration : Le nombre de combinaisons de p éléments choisis parmi n est noté C_n^p. Si l'on permute les éléments de chaque combinaison simple, on obtient tous les arrangements simples. Il y a donc $p!$ fois plus d'arrangements que de combinaisons, ce qui s'écrit $A_n^p = p!C_n^p$. Le nombre de combinaisons de p éléments choisis parmi n est donc :

$$C_n^p = \frac{A_n^p}{p!} = \frac{n!}{p!(n-p)!}$$

■

Exemple IV.14. Les $C_3^2 = \frac{3!}{2!1!} = 3$ combinaisons de 2 éléments choisis parmi a, b, c :

$$ab \qquad ac \qquad bc$$

Proposition IV.12

Quelques formules :

$$C_n^{n-p} = C_n^p$$

$$C_{n+1}^{p+1} = C_n^p + C_n^{p+1}$$

$$(a+b)^n = \sum_{i=0}^{n} C_n^k a^k b^{n-k} \text{ (formule du binôme)}$$

Proposition IV.13

Combinaisons de p éléments parmi n avec répétition : *c'est le nombre de listes non ordonnées, avec répétition éventuelle, de p éléments dans un ensemble contenant n éléments, les éléments ne sont donc pas ordonnés.*

$$K_n^p = C_{n+p-1}^p = \frac{(n+p-1)!}{p!(n-1)!}$$

Exemple IV.15. Les $K_4^2 = C_{4+2-1}^2 = \frac{(4+2-1)!}{(4-1)!2!} = \frac{5\times4}{2} = 10$ combinaisons avec répétitions de 2 lettres choisies parmi a, b, c, d sont :

$$aa \qquad ab \qquad ac \qquad ad \qquad bb \qquad bc \qquad bd \qquad cc \qquad cd \qquad dd$$

Exemple IV.16. Combien y a-t-il de dominos avec 10 symboles différents ?

$$K_{10}^2 = C_{10+2-1}^2 = \frac{11!}{9!.2!} = \frac{11 \times 10}{2} = 55$$

IV.3 Cas des ensembles infinis

IV.3.1 Définition et premiers exemples d'ensembles dénombrables

Définition IV.3 (Ensemble dénombrable)**.** Un ensemble est *dénombrable* s'il est fini ou s'il est en bijection $\mathbb{N}$.

Exemple IV.17. Voilà quelques exemples d'ensembles dénombrables :

— $\mathbb{N} \setminus \{0\}$ est dénombrable par la bijection

$$\begin{array}{rccc} f: & \mathbb{N} & \longrightarrow & \mathbb{N} \setminus \{0\} \\ & n & \longmapsto & n+1 \end{array}.$$

— l'ensemble des nombres pairs, noté $2\mathbb{N}$, est dénombrable par la bijection

$$\begin{array}{rccc} f: & \mathbb{N} & \longrightarrow & 2\mathbb{N} \\ & n & \longmapsto & 2n \end{array}.$$

— l'ensemble des nombres impairs, noté $2\mathbb{N}+1$, est dénombrable par la bijection

$$\begin{array}{rccc} f: & \mathbb{N} & \longrightarrow & 2\mathbb{N}+1 \\ & n & \longmapsto & 2n+1 \end{array}.$$

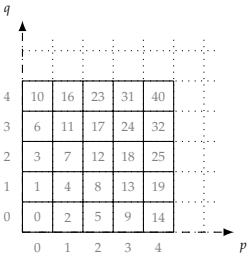
— l'ensemble des entiers relatifs $\mathbb{Z}$ est dénombrable par la bijection

$$\begin{array}{rccc} f: & \mathbb{N} & \longrightarrow & \mathbb{Z} \\ & 2n & \longmapsto & -n \\ & 2n+1 & \longmapsto & n+1 \end{array}.$$

— l'ensemble $\mathbb{N}^2$ est dénombrable par la bijection

$$\begin{array}{rccc} f: & \mathbb{N}^2 & \longrightarrow & \mathbb{N} \\ & (p,q) & \longmapsto & 2^p(2q+1)-1 \end{array}.$$

On peut aussi considérer la bijection dite fonction de couplage de Cantor :



$$\begin{array}{rccc} f: & \mathbb{N}^2 & \longrightarrow & \mathbb{N} \\ & (p,q) & \longmapsto & p + \sum_{i=0}^{p+q-1}(i+1) = p + \frac{(p+q)(p+q+1)}{2} \end{array}$$

— Par récurrence sur l'entier $k \in \mathbb{N}^*$, on montre que $\mathbb{N}^k$ est dénombrable.

Proposition IV.14

Tout sous-ensemble $X \subseteq \mathbb{N}$ est dénombrable.

Démonstration : Si X est fini, c'est terminé. Supposons que X est infini. On définit par récurrence une application $\varphi : \mathbb{N} \to X$ de la manière suivante :

$$\varphi(0) = \min\{x \in X\} \text{ et } \varphi(n+1) = \min\{x \in X : x > \varphi(n)\} \text{ pour } n \geq 1.$$

On vérifie que φ est une bijection de $\mathbb{N}$ sur X ■

IV.3.2 Critères de dénombrabilité

Proposition IV.15

Il existe une application $f : X \to \mathbb{N}$ qui est injective si et seulement si X est dénombrable.

Démonstration : Supposons que X est infini, autrement c'est terminé.

Si $f : X \to \mathbb{N}$ est injective alors X est en bijection avec $f(X)$, qui est un sous-ensemble infini de $\mathbb{N}$. D'après la proposition IV.14, $f(X)$ est dénombrable, comme $f(X)$ est infini, il existe une bijection $h : f(X) \to \mathbb{N}$. Dans ce cas, $h \circ f$ réalise une bijection de X sur $\mathbb{N}$.

La réciproque découle de la définition de la dénombrabilité. ■

Exemple IV.18 (Applications). Cette proposition est très commode pour montrer qu'un ensemble est dénombrable :

— Un sous-ensemble d'un ensemble dénombrable est dénombrable.

— $\mathbb{N}^2$ est dénombrable car l'application suivante est injective :

$$\begin{array}{rccc} f : & \mathbb{N}^2 & \longrightarrow & \mathbb{N} \\ & (p,q) & \longmapsto & 2^p 3^q \end{array} .$$

— $\mathbb{N}^k$ est dénombrable, on considère k nombres premiers distincts $p_1, \dots, p_k$ et on vérifie que l'application suivante est injective :

$$\begin{array}{rccc} f : & \mathbb{N}^k & \longrightarrow & \mathbb{N} \\ & (a_1, \dots, a_k) & \longmapsto & p_1^{a_1} p_2^{a_2} \dots p_k^{a_k} \end{array} .$$

Proposition IV.16

Un produit fini d'ensembles dénombrables est dénombrable.

Démonstration : Soient $X_1, \dots, X_k$ des ensembles dénombrables et pour tout $i \in \{1, \dots, k\}$ on considère une application injective $f_i : X_i \to \mathbb{N}$.

Comme $\mathbb{N}^k$ est infini et dénombrable, il existe donc une bijection $h : \mathbb{N}^k \to \mathbb{N}$. On définit :

$$\begin{array}{rccc} \varphi : & X_1 \times \cdots \times X_k & \longrightarrow & \mathbb{N} \\ & (x_1, \dots, x_k) & \longmapsto & h(f_1(x_1), \dots, f_k(x_k)) \end{array}$$

Si $h(f_1(x_1), \dots, f_k(x_k)) = h(f_1(x'_1), \dots, f_k(x'_k))$ pour $(x_1, \dots, x_k), (x'_1, \dots, x'_k) \in X_1 \times \cdots \times X_k$ alors $(f_1(x_1), \dots, f_k(x_k)) = (f_1(x'_1), \dots, f_k(x'_k))$ par injectivité de h et $(x_1, \dots, x_k) = (x'_1, \dots, x'_k)$ par injectivité de chaque f_i. On en déduit que φ est injective donc $X_1 \times \cdots \times X_k$ est dénombrable. ■

Proposition IV.17

Il existe une application $f : \mathbb{N} \to X$ qui est surjective si et seulement si X est dénombrable.

Démonstration : Supposons X infini autrement c'est terminé. Par hypothèse, il existe une application $f : \mathbb{N} \to X$ qui est surjective. Pour tout $x \in X$, définissons $g(x) = \min\{y \in \mathbb{N} : f(y) = x\}$. On vérifie que $g : X \to \mathbb{N}$ est injective et on en déduit que X est dénombrable. ■

Exemple IV.19 (Dénombrabilité de $\mathbb{Q}$). L'ensemble $\mathbb{Q}$ est dénombrable. On considère l'application :

$$\begin{array}{rccc} f : & \mathbb{Z} \times \mathbb{N} & \longrightarrow & \mathbb{Q} \\ & (p,q) & \longmapsto & \frac{p}{q+1} \end{array} .$$

Cette application est surjective et en composant avec une bijection de $\mathbb{N}$ sur $\mathbb{Z} \times \mathbb{N}$ on obtient une surjection de $\mathbb{N}$ sur $\mathbb{Q}$.

Proposition IV.18

Une réunion dénombrable d'ensembles dénombrables est dénombrable.

Démonstration : Soit I un ensemble dénombrable et $(X_i)_{i\in I}$ une famille d'ensembles dénombrables indexées par I. Pour chaque $i \in I$, il existe une application injective $f_i : X_i \to \mathbb{N}$. On définit l'ensemble

$$Y = \bigcup_{i\in I} \{(i, f_i(x)) : x \in X_i\} \subseteq I \times \mathbb{N}$$

L'ensemble $I \times \mathbb{N}$ est dénombrable, donc Y est dénombrable. Pour tout $(i, y) \in Y$, on définit $\varphi(i, y)$ la préimage de y par f_i qui est défini de manière unique par injectivité de f_i.

L'application $\varphi : Y \to \bigcup_{i\in I} X_i$ est surjective et Y est dénombrable, donc $\bigcup_{i\in I} X_i$ est dénombrable. ■

IV.3.3 Ensembles non dénombrables

Il existe des ensembles qui ne sont pas dénombrables. L'exemple le plus simple est l'ensemble des parties de $\mathbb{N}$ noté $\mathcal{P}(\mathbb{N})$. En fait, on peut démontrer le théorème suivant qui est un peu plus général :

Théorème IV.19 (Cantor 1981)

Soient E un ensemble. Il n'existe pas d'application bijective de E dans $\mathcal{P}(E)$.

Démonstration : Supposons qu'il existe une bijection $f : E \longrightarrow \mathcal{P}(E)$ et posons

$$A = \{x \in E : x \notin f(x)\}$$

Puisque $A \subseteq E$, il existe x_0 tel que $f(x_0) = A$, on a donc deux cas possible :
— si $x_0 \in A$, par définition $x_0 \notin f(x_0) = A$ ce qui est impossible ;
— si $x_0 \notin A$, par définition $x_0 \in f(x_0) = A$, on aboutit aussi à une contradiction.
L'hypothèse de départ était absurde, il n'existe donc pas de telle bijection. ■

Un autre argument utilisé en informatique théorique permet de montrer que $[0, 1[$ n'est pas dénombrable. C'est l'argument diagonal de Cantor.

Théorème IV.20 (Argument diagonal de Cantor)

L'ensemble $[0, 1[$ n'est pas dénombrable.

Démonstration : Supposons que $[0, 1[$ soit dénombrable alors il existe une suite de réels $(x_i)_{i\in\mathbb{N}}$ telle que $[0, 1[= \{x_i : i \in \mathbb{N}\}$. Chaque réel x_i admet une écriture décimale $x_i = 0, x_i^0 x_i^1 x_i^2 x_i^3 \dots$ avec $x_i^j \in \{0, 1, \dots, 9\}$ pour $j \in \mathbb{N}$.

Considérons une suite d'entiers $y^i \in \{0, 1, \dots, 8\}$ telle que $y^i \neq x_i^i$ pour tout $i \in \mathbb{N}$. Alors le réel $y = 0, y^1 y^2 y^3 \cdots \in [0, 1[$ est différent du réel x_i pour tout $i \in \mathbb{N}$ car il diffère pour la $i^{\text{ème}}$ décimal. On en déduit une contradiction. ■

On en déduit que $\mathbb{R}$ n'est pas dénombrable. De plus l'ensemble des nombres réels irrationnels, c'est à dire $\mathbb{R} \setminus \mathbb{Q}$, n'est pas dénombrable. En effet, si tel n'était pas le cas, $\mathbb{R} = (\mathbb{R} \setminus \mathbb{Q}) \cup \mathbb{Q}$ serait dénombrable.

IV.3.4 Théorème de Cantor-Schröder-Bernstein

Pour deux ensembles finis E et F, s'il existe une application injective de E dans F et une application injective de F dans E alors on a respectivement $\text{Card}(E) \leq \text{Card}(F)$ et $\text{Card}(F) \leq \text{Card}(E)$. On a donc $\text{Card}(F) = \text{Card}(E)$, c'est à dire que E et F sont en bijection. En fait ce phénomène se généralise à des ensembles infinis, c'est le théorème de Cantor-Schröder-Bernstein et on verra la démonstration plus tard.

Théorème IV.21 (Théorème de Cantor-Schröder-Bernstein)

S'il existe une application injective de A vers B et une application injective de B vers A, alors il existe une application bijection de A vers B.

Exemple IV.20. Ce théorème est très utile pour montrer que deux ensembles sont équipotent. On peut montrer que $[0,1]$, $]0,1[$, $]0,1[^2$, $\mathbb{R}$ et $\mathcal{P}(\mathbb{N})$ sont équipotents deux à deux. Par exemple pour montrer que $\mathcal{P}(\mathbb{N})$ et $[0,1]$ sont en bijection, on montre que les deux fonctions suivantes sont des injections :

$$\begin{array}{rccc} f: & \mathcal{P}(\mathbb{N}) & \longrightarrow & [0,1] \\ & A & \longmapsto & \sum_{n\in A} 3^{-n} \end{array} \qquad \begin{array}{rccc} g: & [0,1] & \longrightarrow & \mathcal{P}(\mathbb{N}) \\ & x = 0, x_0x_1x_2x_3x_4\ldots & \longmapsto & \{i \in \mathbb{N} \text{ si } x_i = 1\} \end{array}$$

Chapitre V

Relations sur les ensembles

V.1 Vocabulaire des relations

V.1.1 Définition

Définition V.1 (Relation binaire)**.** Une *relation binaire* $\mathcal{R}$ d'un ensemble E vers un ensemble F est définie par une partie $G_\mathcal{R} \subseteq E \times F$. Si $(x, y) \in G_\mathcal{R}$, on dit que x est *en relation avec* y et l'on note $x\mathcal{R}y$ (notation infixe) ou $\mathcal{R}(x, y)$ ou $\mathcal{R}\ x\ y$ (notations préfixes). L'ensemble E est appelé *ensemble de départ*, l'ensemble F est l'*ensemble d'arrivée* et la partie $G_\mathcal{R}$ de $E \times F$ est appelé le *graphe de la relation*.

Quand une relation binaire est définie d'un ensemble E vers lui-même (autrement dit $E = F$ dans la définition précédente, donc définie par une partie $G_\mathcal{R}$ de E^2), on dit que c'est une *relation interne* sur E, ou simplement relation sur E.

Exemple V.1. Soient $A = \{a, b, c, d, e\}$ l'ensemble des élèves et $B = \{Math, Info, Ang, Phys\}$ l'ensemble des cours. L'ensemble $A \times B$ correspond aux couples possibles d'étudiants et cours. On peut définir les relations suivantes :

— la relation $\mathcal{R}$ qui décrit si un étudiant suit le cours régulièrement défini par

$$G_\mathcal{R} = \{(a, Math), (a, Phys), (b, Info), (c, Ang), (d, Ang), (e, Math), (e, Ang)\}$$

— la relation $\mathcal{S}$ définit par $G_\mathcal{S} = \{(a, Math), (c, Math), (d, Ang)\}$ qui décrit si un étudiant présent à un intérêt personnel pour la matière (éventuellement il peut présenter un intérêt pour la matière mais ne va pas souvent en cours).

V.1.2 Modes de représentations

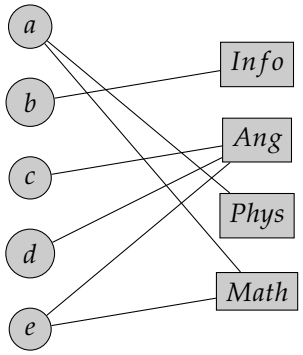
Il existe différentes manières de représenter une relation d'un ensemble E vers un ensemble F.

Diagramme cartésien et matrice de relation Un diagramme cartésien est un tableau où les lignes désignent les éléments de E et les colonnes les éléments de F. Les couples en relations sont marqués par le symbole V. On peut aussi représenter la relation par une matrice en remplaçant les espaces vides par 0 et les espaces marqués par 1.

$\mathcal{R}$	$Math$	$Phys$	Ang	$Info$
a	V	V		
b				V
c			V	
d			V	
e	V		V	

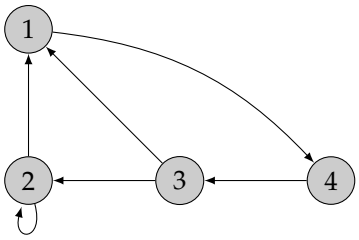
$\mathcal{S}$	$Math$	$Phys$	Ang	$Info$
a	V			
b				
c				
d	V		V	
e				

Diagramme sagittal Par un graphique où les éléments de E sont situés à gauche, les éléments de F sont situé à droite et les éléments en relation sont reliés par une arête. On appelle cela un *graphe biparti*.



Graphe orienté Lorsque on a une relation interne sur un ensemble fini, on dessine un *graphe orienté* où les sommets sont les éléments et on a un arc allant d'un sommet a à un sommet b si a est en relation avec b.

Par exemple la relation interne sur $\{1,2,3,4\}$ définit par $\{(2;1);(2;2);(3;1);(1;4);(4;3)\}$ est représentée par :



Remarques sur ces représentations Lorsqu'on représente une relation entre l'ensemble E et l'ensemble F, tous les liens entre les éléments de E et F sont représentés et peuvent être reconstitués. Cependant, dans les représentations précédentes les choix sont arbitraires :

— Diagramme sagittal : où placer les sommets et quelle forme donner aux flèches ?

— Diagramme cartésien et matrice : quel ordre de parcours faut il donner aux éléments ?

De manière générale établir l'égalité entre deux relations est un problème difficile et il n'y a pas d'algorithme performant pour décider ce problème.

V.1.3 Quelques notions proches

Relation fonctionnelle Une fonction $f : E \to F$ associe a chaque élément de E au plus un élément de F. On peut alors définir la relation $\mathcal{R}_f$ définie par le graphe

$$G_{\mathcal{R}_f} = \{(x, f(x)) : x \in E\} \subseteq E \times F.$$

Réciproquement, pour une relation $\mathcal{R}$ telle que pour tout $x \in E$ il y a au plus un $y \in F$ vérifiant $x\mathcal{R}y$ alors on peut lui associer une fonction f telle que $f(x) = y$ si et seulement si $x\mathcal{R}y$. On dit que $\mathcal{R}$ est une *relation fonctionnelle*.

Relation n-aire Etant donné n ensembles $E_1, E_2, \ldots, E_n$, une *relation n-aire* $\mathcal{R}$ est définie par un sous ensemble $G_{\mathcal{R}} \subseteq E_1 \times E_2 \times \cdots \times E_n$.

Si $n = 2$, on retrouve la notion de relation binaire, si $n = 3$ on dit que l'on a une relation ternaire... La notion de relation n-aire est au centre des bases de données qui cherchent à mettre en relation différentes données.

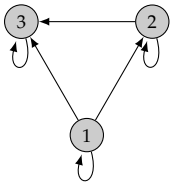
V.2 Propriétés sur les relations

On s'intéresse principalement aux relations internes, c'est à dire définies sur un seul ensemble. On représentera donc cette relation soit avec une matrice carrée, soit avec un graphe orienté. Dans cette section on définit des propriétés sur les relations.

Reflexivité Une relation $\mathcal{R}$ est *réflexive* si pour tout $x \in E$ on a

$$x\mathcal{R}x.$$

Il est possible de repérer la réflexivité sur les modes de représentations :
— Diagramme cartésien : la diagonale doit être notée.
— Diagramme sagittal : chaque sommet admet une boucle.



	1	2	3
1	V	V	V
2		V	V
3			V

Exemple V.2. Quel que soit l'ensemble, la relation d'égalité $=$ est réflexive. Sur $\mathbb{N}$, la relation $\leq$ est réflexive, mais $<$ n'est pas réflexive.

Exemple V.3. Sur l'ensemble des mots $\mathcal{A}^*$, on considère la relation $\overset{l}{\equiv}$ définit par

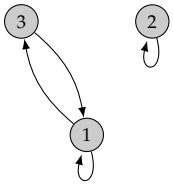
$$u \overset{l}{\equiv} v \text{ si et seulement si } u \text{ et } v \text{ ont même longueur.}$$

Par exemple `grand` $\overset{l}{\equiv}$ `petit` et `grand` $\overset{l}{\equiv}$ `grand` mais `grand` $\overset{l}{\not\equiv}$ `grande`.
La relation $\overset{l}{\equiv}$ est réflexive.

Symétrie Une relation $\mathcal{R}$ est *symétrique* si pour tout $x, y \in E$ on a

$$x\mathcal{R}y \text{ si et seulement si } y\mathcal{R}x.$$

Il est possible de repérer la réflexivité sur les modes de représentations :
— Diagramme cartésien : symétrie par rapport à la diagonale.
— Diagramme sagittal : quand une flèche va de a vers b, il y a aussi une flèche de b vers a.



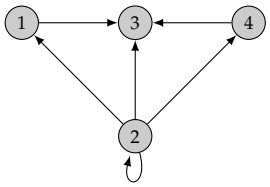
	1	2	3
1	V		V
2		V	
3	V		

Exemple V.4. Quel que soit l'ensemble, la relation d'égalité $=$ est symétrique. Sur $\mathbb{N}$, la relation $\leq$ est n'est pas symétrique.
La relation $\overset{l}{\equiv}$ sur $\mathcal{A}^*$ est symétrique.

Transitivité Une relation $\mathcal{R}$ est *transitive* si pour tout $x, y, z \in E$ on a

$$x\mathcal{R}y \text{ et } y\mathcal{R}z \text{ implique que } x\mathcal{R}z.$$

Il est possible de repérer la réflexivité sur les modes de représentations :
— Diagramme sagittal : tout chemin qui part d'un sommet s et va à un sommet s' en suivant la direction des flèches admet un raccourci, c'est à dire un chemin de longueur un.



	1	2	3	4
1			V	
2	V	V	V	V
3				
4			V	

Example : Quel que soit l'ensemble, la relation d'égalité $=$ est transitive.

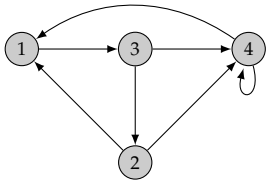
Sur $\mathbb{N}$, la relation $\leq$ est transitive.

La relation $\overset{l}{\equiv}$ sur $\mathcal{A}^*$ est transitive.

La relation "est le père de" n'est pas transitive : on ne peut pas à la fois être grand père et père.

Antisymétie Une relation $\mathcal{R}$ est *antisymétrique* si pour tout $x, y \in E$ on a

$$x\mathcal{R}y \text{ et } y\mathcal{R}x \text{ implique que } x = y.$$



	1	2	3	4
1			V	
2	V			V
3		V		V
4	V			V

Example : Sur $\mathbb{N}$, la relation $\leq$ est antisymétrique.

La relation $\overset{l}{\equiv}$ sur $\mathcal{A}^*$ n'est pas antisymétrique.

V.3 Relations d'équivalence

L'égalité est réflexive, symétrique et transitive. Dans ce chapitre, on veut généraliser la notion d'égalité en considérant que deux éléments sont identiques s'ils vérifient une propriété donnée : c'est la notion d'équivalence.

V.3.1 Définition et exemples

La relation d'équivalence est une abstraction de la notion d'égalité, elle permet de lier entre eux des éléments qui partagent un ou plusieurs attributs communs.

Définition V.2. Une relation binaire définie sur un unique ensemble E est une *relation d'équivalence* si elle est réflexive, symétrique et transitive.

Exemple V.5. Par définition, pour $x, y \in \mathbb{Z}$, on note $x \equiv y[\text{mod } n]$, lire x est congru à y modulo n, si et seulement s'il existe $k \in \mathbb{Z}$ tel que $x - y = kn$. On a défini une relation d'équivalence sur $\mathbb{Z}$ car on peut vérifier :

- **Réflexivité :** $x \equiv x[\text{mod } n]$ car $x - x = 0.n$ et $0 \in \mathbb{Z}$.
- **Symétrie :** si $x \equiv y[\text{mod } n]$ alors il existe $k \in \mathbb{Z}$ tel que $x - y = k.n$, on a donc $y - x = -k.n$ et $-k \in \mathbb{Z}$ d'où $y \equiv x[\text{mod } n]$.
- **Transitivité :** si $x \equiv y[\text{mod } n]$ et $y \equiv z[\text{mod } n]$ alors il existe $k, k' \in \mathbb{Z}$ tels que $x - y = k.n$ et $y - z = k'.n$. Ainsi $x - z = x - y + y - z = (k + k').n$. On en déduit que $x \equiv z[\text{mod } n]$

Exemple V.6. Voici quelques exemples de relations d'équivalence :

- Sur l'ensemble des personnes, la relation "a le même âge que" est une relation d'équivalence. Des personnes liées appartiennent à la même tranche d'âge.
- Sur l'ensemble des triangles, la relation "a les mêmes angles que" est une relation d'équivalence. Des triangles liés par cette relation sont dits semblables.

— La relation $\mathcal{R}$ définie sur $\mathbb{R} \setminus \{0\}$ par $x\mathcal{R}y$ si et seulement si $xy > 0$ est une relation d'équivalence. Deux réels liés par cette relation ont le même signe.

On pourra vérifier que ce sont bien des relations d'équivalence.

Remarque V.1. Si $\mathcal{R}$ est une relation d'équivalence, on dit que x est équivalent à y si $a\mathcal{R}b$. La relation étant symétrique, on a aussi b est équivalent à a. On dit que a et b sont équivalents.

V.3.2 Classes d'équivalence et partition

Définition V.3. Soit $\mathcal{R}$ une relation d'équivalence sur un ensemble E. La *classe d'équivalence* d'un élément x, noté $\mathbf{Cl}(x)$, est l'ensemble des éléments de E qui sont en relation avec x. Autrement dit

$$\mathbf{Cl}(x) = \{y \in E : x\mathcal{R}y\}.$$

Parfois la classe d'équivalence de x est aussi notée $\dot{x}$.

Proposition V.1

Une classe d'équivalence n'est jamais vide.

Démonstration : La classe d'un élément contient toujours au moins cet élément par réflexivité de la relation d'équivalence. ■

Proposition V.2

L'intersection de deux classes d'équivalence distinctes est vide.

Démonstration : Soit $\mathcal{R}$ une relation d'équivalence sur un ensemble E, on considère deux éléments $x, y \in E$ et leurs classes d'équivalence $\mathbf{Cl}(x)$ et $\mathbf{Cl}(y)$.

Supposons qu'il existe $z \in \mathbf{Cl}(x) \cap \mathbf{Cl}(y)$, on a donc $x\mathcal{R}z$ et $y\mathcal{R}z$. Par symétrie, on a $z\mathcal{R}y$. Ainsi $x\mathcal{R}z$ et $z\mathcal{R}y$ par transitivité on en déduit que $x\mathcal{R}y$ et par réflexivité $y\mathcal{R}x$.

Pour tout $t \in \mathbf{Cl}(y)$, on a $y\mathcal{R}t$ donc par transitivité $x\mathcal{R}t$ d'ou $t \in \mathbf{Cl}(x)$. Autrement dit $\mathbf{Cl}(y) \subseteq \mathbf{Cl}(x)$.

De même, pour tout $t \in \mathbf{Cl}(x)$, on a $x\mathcal{R}t$ donc par transitivité $y\mathcal{R}t$ d'ou $t \in \mathbf{Cl}(y)$. Autrement dit $\mathbf{Cl}(x) \subseteq \mathbf{Cl}(y)$.

On en déduit que $\mathbf{Cl}(x) = \mathbf{Cl}(y)$. Ainsi si deux classes sont distinctes alors l'intersection est vide. ■

Définition V.4. Soit E un ensemble, la famille d'ensembles $(A_i)_{i\in I}$ indexée par I est une *partition* si :

— l'union des $(A_i)_{i\in I}$ est égale à E, c'est à dire $E = \cup_{i\in I} A_i$,

— deux éléments de $(A_i)_{i\in I}$ distincts sont disjoints, c'est à dire que si $i \neq j$ alors $A_i \cap A_j = \emptyset$.

Théorème V.3

Etant donné une relation d'équivalence sur un ensemble, les classes d'équivalences forment une partition.

Démonstration : Les classes sont des parties de E. De plus la classe d'un élément contient cet élément. L'union des classes d'équivalence est donc E.

Si l'on choisi deux classes distinctes (i.e. $\mathbf{Cl}(x) \neq \mathbf{Cl}(y)$) alors leur intersection est vide d'après la proposition précédente. ■

Exemple V.7. Considérons la relation d'équivalence correspondant à la congruence modulo 3. On a trois classes d'équivalence :

— $\mathbf{Cl}(0) = \{\ldots, -6, -3, 0, 3, 6, 9, 12, \ldots\}$;

— $\mathbf{Cl}(1) = \{\ldots, -5, -2, 1, 4, 7, 10, 13, \ldots\}$;

— $\mathbf{Cl}(2) = \{\ldots, -4, -1, 2, 5, 8, 11, 14, \ldots\}$.

V.3.3 Ensemble quotient

Définition V.5. Soit E un ensemble munit d'une relation d'équivalence $\mathcal{R}$. L'ensemble quotient est l'ensemble des classes d'équivalence de tous les éléments de E. On le note $E/\mathcal{R}$.

Proposition V.4

Etant donné une relation d'équivalence $\mathcal{R}$ sur E, la fonction suivante est surjective :

$$\begin{array}{rccc} f: & E & \longrightarrow & E/\mathcal{R} \\ & x & \longmapsto & \mathbf{Cl}(x) \end{array}$$

Parfois, pour parler aisément d'une classe, on choisit un de ses éléments qui représente cette classe. Cet élément, surmonté d'un point, sert à représenter la classe en question.

Exemple V.8 (Congruence modulo 4). On considère $\mathbb{Z}$ muni de la relation d'équivalence $x \equiv y [\text{mod } 4]$. On choisit pour représentants les entiers 0, 1, 2 et 3. L'ensemble-quotient est $\mathbb{Z}/4\mathbb{Z} = \{\dot{0}, \dot{1}, \dot{2}, \dot{3}\}$.

Chapitre VI

Relations d'ordre

Dans ce chapitre on va définir la notion de relation d'ordre sur un ensemble qui permet de mettre en place une hiérarchie sur E par une relation de précédence : l'élément a est en relation avec b si a précède b dans la hiérarchie.

VI.1 Premières notions

VI.1.1 Définition

Définition VI.1. Une relation binaire $\preceq$ sur un ensemble E est une *relation d'ordre* si elle est réflexive, transitive et antisymétrique. Autrement dit :

$\preceq$ **réflexive :** Pour tout $x \in E$ on a $x \preceq x$.

$\preceq$ **transitive :** Pour tout $x, y, z \in E$, si $x \preceq y$ et $y \preceq z$ alors $x \preceq z$.

$\preceq$ **antisymétrique :** Pour tout $x, y \in E$, si $x \preceq y$ et $y \preceq x$ alors $x = y$.

Définition VI.2. Un ordre est *total* si pour tous éléments $x, y \in E$ on a $x \preceq y$ ou $y \preceq x$. Un ordre est dit *partiel* pour souligner qu'on n'a pas forcément cette propriété.

Si $x \preceq y$, on dit que x est un minorant de y et que y est un majorant de x.

VI.1.2 Exemples de relations d'ordre classiques

Ordres sur les nombres

- Les relations $\leq$ et $\geq$ sont des relations d'ordre total sur $\mathbb{N}$ qui s'étendent à $\mathbb{Z}$, $\mathbb{Q}$ ou $\mathbb{R}$.
- Les relations $<$ et $>$ ne sont pas des relations d'ordre sur $\mathbb{N}$ (respectivement $\mathbb{Z}$, $\mathbb{Q}$ ou $\mathbb{R}$), ce sont des relations d'ordre strictes.
- Sur $\mathbb{N}^*$ la relation a divise b, notée $a|b$, est une relation d'ordre mais n'est pas total. On rappelle que a divise b s'il existe $k \in \mathbb{N}^*$ tel que $b = a\,k$, par exemple $3|57$.

Ordres sur les parties d'un ensemble Soit E un ensemble l'inclusion, notée $\subseteq$, est une relation d'ordre sur l'ensemble des parties $\mathcal{P}(E)$ qui n'est pas totale.

Ordres sur les mots Soit $\mathcal{A}$ un alphabet. Il existe différentes notions pour ordonner l'ensemble des mots $\mathcal{A}^*$:

- La relation u est préfixe de v, notée $u \sqsubset_{\text{perd}} v$ et définit par il existe un mot $w \in \mathcal{A}^*$ tel que $v = u.w$, est une relation d'ordre qui n'est pas total

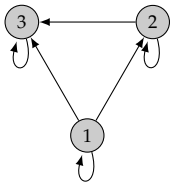
— Soit $\preceq$ un ordre total sur $\mathcal{A}$, on définit l'*ordre lexicographique* sur $\mathcal{A}^*$ par

$$u \leq_{lex} v \iff (u \text{ préfixe de } v) \text{ ou } (\exists m \in \mathbb{N} \text{ tel que } u_1 \dots u_m = v_1 \dots v_m \text{ et } u_{m+1} \leq v_{m+1})$$

C'est une relation d'ordre total sur $\mathcal{A}^*$. On a par exemple `a` $\leq_{lex}$ `fa`, `poule` $\leq_{lex}$ `poulet`, `avion` $\leq_{lex}$ `train`, `livraison` $\leq_{lex}$ `livre`, `foot` $\leq_{lex}$ `fort`.

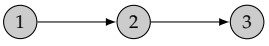
VI.1.3 Mode de représentation

Voilà le diagramme sagittal d'un ordre sur un ensemble à trois éléments :



	1	2	3
1	V	V	V
2		V	V
3			V

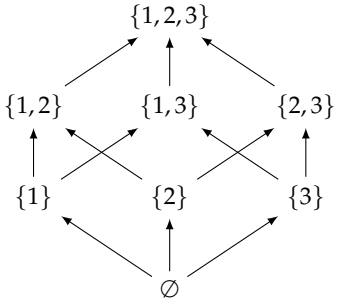
Pour simplifier la lecture du diagramme, on supprime les boucles dues à la réflexivité et les flèches déductibles par transitivité :



L'idée est de représenter les sommets du diagramme et tracer seulement les flèches correspondant aux successeurs immédiats. On dit que y est un successeur immédiat de x si $x \preceq y$, $x \neq y$ et il n'existe pas de z tel que $x \preceq z \preceq y$.

L'élément x minore y, ($x \preceq y$) si et seulement si on peut passer du point qui représente x au point qui représente y en suivant les flèches du diagramme de Hasse. Le diagramme de Hasse d'un ordre total est une chaîne.

Example : Le diagramme de Hasse pour l'ordre $\subseteq$ sur $E = \{1, 2, 3\}$ est :



VI.1.4 Fonctions croissantes et décroissantes

Définition VI.3. Soient A et B deux ensembles munis respectivement des relations d'ordre $\preceq_A$ et $\preceq_B$ et $f : A \longrightarrow B$ une application. On dit que

— f est *croissante* si $x \preceq_A y$ alors $f(x) \preceq_B f(y)$.
— f est *décroissante* si $x \preceq_A y$ alors $f(y) \preceq_B f(x)$.
— f est *strictement croissante* si $x \preceq_A y$ et $x \neq y$ alors $f(x) \preceq_B f(y)$ et $f(x) \neq f(y)$.
— f est *strictement décroissante* si $x \preceq_A y$ et $x \neq y$ alors $f(y) \preceq_B f(x)$ et $f(x) \neq f(y)$.

Proposition VI.1

Soit A et B deux ensembles munis respectivement des relations d'ordre $\preceq_A$ et $\preceq_B$ tel que $\preceq_A$ est total. Une application $f : A \to B$ strictement croissante ou strictement décroissante est injective.

VI.2 Bornes d'un ensemble

Soit A une partie d'un ensemble ordonné E par la relation d'ordre $\preceq$.

Définition VI.4. Un *élément minimal* de A est un élément de A qui n'admet pas d'élément plus petit dans A.

On appelle *minorant* de A tout élément de E qui est plus petit que n'importe lequel des éléments de A. Autrement dit :

$$x \text{ minorant de } A \iff \text{pour tout } y \in A \text{ on a } x \preceq y$$

Par la propriété d'antisymétrie, A admet au plus un seul minorant dans A, c'est *le plus petit élément* de A, s'il existe on le note $\min(A)$.

On appelle *borne inférieure* le plus grand des minorants, on le note $\inf(A)$. Autrement dit

$$x \text{ borne inférieure de } A \iff \begin{cases} \text{pour tout } y \in A \text{ on a } x \preceq y & (x \text{ est un minorant}) \\ \text{et} & \\ \text{pour tout } z \text{ minorant } A \text{ on a } z \preceq x & (x \text{ est le plus grand des minorants}) \end{cases}$$

Définition VI.5. Un *élément maximal* de A est un élément de A qui n'admet pas d'élément plus grand dans A.

On appelle *majorant* de A tout élément de E qui est plus grand que n'importe lequel des éléments de A. Autrement dit :

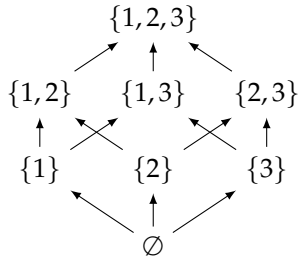
$$x \text{ majorant de } A \iff \text{pour tout } y \in A \text{ on a } y \preceq x$$

Par la propriété d'antisymétrie, A admet au plus un seul majorant dans A, c'est *le plus grand élément* de A.

On appelle *borne supérieur* le plus petit des majorant, on le note $\sup(A)$. Autrement dit

$$x \text{ borne supérieur de } A \iff \begin{cases} \text{pour tout } y \in A \text{ on a } y \preceq x & (x \text{ est un majorant}) \\ \text{et} & \\ \text{pour tout } z \text{ majorant de } A \text{ on a } x \preceq y & (x \text{ est le plus petit des majorants}) \end{cases}$$

Exemple VI.1. Le diagramme de Hasse pour l'ordre $\subseteq$ sur $E = \{1,2,3\}$ est :



Considérons les ensembles suivants :

$$\begin{aligned} A &= \{\{1\},\{1,2\},\{1,3\}\} \\ B &= \{\{2\},\{3\},\{2,3\}\} \\ C = A \cup B &= \{\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\}\} \end{aligned}$$

On a

	A	B	C
Eléments minimaux	$\{1\}$	$\{2\},\{3\}$	$\{1\},\{2\},\{3\}$
Eléments maximaux	$\{1,2\},\{1,3\}$	$\{2,3\}$	$\{1,2\},\{1,3\},\{2,3\}$
Plus petit élément	$\{1\}$	non existant	non existant
Plus grand élément	non existant	$\{2,3\}$	non existant
Borne inférieure	$\{1\}$	$\emptyset$	$\emptyset$
Borne supérieure	$\{1,2,3\}$	non existant	$\{1,2,3\}$

Exemple VI.2. L'intervalle $I = [0,1[$ sous ensemble de $\mathbb{R}$ muni de la relation d'ordre classique $\leq$ n'admet pas d'élément maximaux, admet un plus petit élément $\min(I) = 0$, admet une borne supérieur $\sup(I) = 1$ et admet une borne inférieure $\inf(I) = 0$.

VI.3 Induction

VI.3.1 Ordre bien fondé

Définition VI.6. Un ensemble ordonné $(E, \preceq)$ est *bien fondé* s'il n'existe pas de suite infinie strictement décroissante d'éléments de E.

De manière équivalente, un ensemble bien fondé peut se définir de la manière suivante :

Théorème VI.2

Un ensemble ordonné $(E, \preceq)$ est bien fondé si et seulement si toute partie non vide admet au moins un élément minimal.

Exemple VI.3. L'ordre usuel $\leq$ sur $\mathbb{N}$ est bien fondé mais il ne l'est pas sur $\mathbb{Z}$, $\mathbb{R}$, $[0,1]$.
L'ordre $|$ sur $\mathbb{N} \setminus \{0,1\}$ défini par "$a|b \iff a$ divise b" est bien fondé.
L'ordre lexicographique sur $\mathbb{N}^2$ est bien fondé. Il est défini par

$$(a,b) \preceq (c,d) \iff ; (a < c \text{ ou } (a = c \text{ et } b \leq d))$$

Soit $\mathcal{A}$ un alphabet contenant au moins deux lettres, $\sqsubset_{\text{perd}}$ est bien fondé mais pas $\leq_{lex}$.

VI.3.2 Application à l'étude de la terminaison d'algorithme

On dit qu'un programme P termine sur l'entrée x si le calcul de $P(x)$ nécessite un nombre fini d'étapes. Il est indécidable de savoir si un programme termine sur une entrée donnée, c'est à dire il n'existe pas de programme qui prend en entrée le code d'un programme P et le code d'une entrée x et répond oui si $P(x)$ s'arrête. Autrement dit il n'existe pas de méthodes qui fonctionne à tout les coup pour prouver la terminaison d'un algorithme. La notion d'ordre bien fondé fonctionne assez souvent.

Cas des algorithmes itératifs : les variants de boucles

Définition VI.7. Etant donné $(E, \preceq)$ un ordre bien fondé, un *variant* de boucle est une fonction de l'ensemble des états du programme dans E strictement décroissant à chaque passage dans la boucle.

Proposition VI.3

Si une boucle admet un variant alors elle termine.

Exemple VI.4. Considérons l'algorithme d'Euclide (algorithme 1), la fonction $(a,b) \to b$ est un variant de boucle donc l'algorithme termine.

Algorithm 1: Algorithme d'euclide

Data: $(x, y) \in \mathbb{N}^2$
Result: le pgcd de x et y

$a \leftarrow x$;
$b \leftarrow y$;
while $b \neq 0$ **do**
  $tmp \leftarrow a$;
  $a \leftarrow b$;
  $b \leftarrow tmp[\text{mod } b]$;

Cas des algorithmes récursifs

On considère une fonction f définit de manière récursive. Si les appels successifs de f ne s'arrête pas, on pourrait construire une suite infinie strictement décroissante.

Proposition VI.4

Soit f une fonction récursive définit sur un ensemble ordonné $(E, \preceq)$ bien fondé. Si f est défini sur les éléments minimaux et si pour tout $x \in E$ non minimal, le définition de $f(x)$ ne fait appel à des valeurs $f(y)$ pour $y \preceq x$ avec $x \neq y$ alors f est bien définit.

Exemple VI.5. On considère la fonction `fact` définie par :
— $\texttt{fact}(0) = 1$;
— $\texttt{fact}(n+1) = (n+1) * \texttt{fact}(n)$.
Elle est bien définie car $(\mathbb{N}, \leq)$ est bien fondé, `fact` est définie sur l'élément minimal 0 et l'appel de `fact` sur un élément non minimal fait appel à des éléments plus petit.

Exemple VI.6. On considère la fonction f définie sur $\mathbb{N} \setminus \{0, 1\}$ par :
— $f(p) = 1$ si p premier ;
— $f(n) = f(a) + f(b)$ si $n = ab$ et $a \neq 1$ et $b \neq 1$.
Elle est bien définie car $(\mathbb{N} \setminus \{0, 1\}, |)$ est bien fondé, f est définie sur les éléments minimaux (les nombres premiers) et l'appel de `fact` sur un élément non minimal fait appel à des éléments plus petit.

VI.3.3 $\mathbb{N}$ et le principe de récurrence

L'ensemble $(\mathbb{N}, \leq)$ est bien fondé.

Théorème VI.5 Principe de récurrence

Soit P une propriété dépendant d'un élément n de $\mathbb{N}$. Si les deux hypothèses suivantes sont vérifiées

Initialisation : *$P(0)$ est vraie,*

Héridité : *si pour tout $n \in \mathbb{N}$ on a la propriété suivante :*

$$\text{"}P(n) \text{ est vraie} \Longrightarrow P(n+1) \text{ est vraie"}$$

alors pour tout $n \in \mathbb{N}$, la propriété $P(n)$ est vraie.

Démonstration : On raisonne par l'absurde : supposons que les hypothèses du théorème sont vraies mais que la conclusion est fausse.

Soit $X = \{n \in \mathbb{N}, P(n) \text{ est fausse}\}$. L'ensemble X est une partie non vide de $\mathbb{N}$, comme $(\mathbb{N}, \leq)$ est bien fondé, X admet un plus petit élément noté n_0.

Comme $P(0)$ est vraie, on a $n_0 > 0$ donc $n_0 - 1$ est un entier positif ou nul, autrement dit $n_0 - 1 \in \mathbb{N}$. $P(n_0 - 1)$ est vraie car $n_0 - 1 \notin X$. Par hypothèse $P(n_0 - 1) \Longrightarrow P(n_0)$ donc $P(n_0)$ est vraie ce qui est contradictoire avec le fait que $n_0 \in X$.

■

Exemple VI.7. On cherche à démontrer par récurrence la propriété $P(n)$ définie par :

$$\sum_{i=0}^{n}(2i+1) = (n+1)^2$$

Initialisation : Pour $n = 0$, on a $\sum_{i=0}^{0}(2i+1) = 1 = (0+1)^2$ donc $P(0)$ est vérifiée.

Héridité : On suppose que pour $n \in \mathbb{N}$ la propriété $P(n)$ est vraie, c'est à dire $\sum_{i=0}^{n}(2i+1) = (n+1)^2$. Montrons que $P(n+1)$ est vérifiée :

$$\begin{aligned}
\sum_{i=0}^{n+1}(2i+1) &= 2(n+1)+1+\sum_{i=0}^{n}(2i+1) \\
&= 2(n+1)+1+(n+1)^2 \qquad \text{(par hypothèse de récurrence)} \\
&= (n+1)^2+2(n+1)+1^2 \\
&= ((n+1)+1)^2 \\
\sum_{i=0}^{n+1}(2i+1) &= (n+2)^2
\end{aligned}$$

Par le principe de récurrence on en déduit que pour tout $n \in \mathbb{N}$ on a $\sum_{i=0}^{n}(2i+1) = (n+1)^2$.

Le principe de récurrence peut être généralisé en considérant que dans la deuxième hypothèse la propriété est vraie pour tout $k \leq n$.

Corollaire VI.6 Principe de récurrence généralisée

Soit P une propriété dépendant d'un élément n de $\mathbb{N}$. Si les deux hypothèses suivantes sont vérifiées

Initialisation : *$P(0)$ est vraie.*

Héridité : *si pour tout $n \in \mathbb{N}$ on a la propriété suivante :*

$$\text{"}P(k) \text{ est vraie pour tout } k \leq n \Longrightarrow P(n+1) \text{ est vraie"}$$

Alors pour tout $n \in \mathbb{N}$, la propriété $P(n)$ est vraie.

Démonstration : On applique le principe de récurrence du théorème VI.5 à la propriété Q tel que pour $n \in \mathbb{N}$, $Q(n)$ est vraie si $P(k)$ est vraie pour tout $k \leq n$. ■

Exemple VI.8. Démontrons que pour deux mots $u, v \in \mathcal{A}^*$ tels que $uv = vu$ alors u et v sont des puissances d'un même mot w. On note $P(n)$ la propriété suivante :

$$(|uv| = n \text{ et } uv = vu) \Longleftrightarrow \exists w \in \mathcal{A}^*, \exists p, q \in \mathbb{N} \text{ tels que } u = w^p \text{ et } v = w^q)$$

On va montrer que $P(n)$ est vraie par récurrence généralisée :

Initialisation : Pour $n = 0$, on a $u = v = \varepsilon$ donc $P(0)$ est vraie.

Héridité : On suppose que $P(k)$ est vraie pour tout $k \leq n$, c'est à dire que si $|uv| = k \leq n$ et $uv = vu$ alors il existe $w \in \mathcal{A}^*$ et $p, q \in \mathbb{N}$ tels que $u = w^p$ et $v = w^q$.

Prenons u et v dans $\mathcal{A}^*$ tels que $|uv| = n+1$ et $uv = vu$. Sans perte de généralité, on suppose que $|u| \geq |v|$. Le mot v est un préfixe de u donc il existe un mot $t \in \mathcal{A}^*$ tel que $u = vt$. L'égalité $u.v = v.u$ s'écrit $vt.v = v.vt$ donc en simplifiant par v on a $tv = vt$. On a deux cas :

— Si $|v| = 0$ alors $v = \varepsilon = u^0$ et $u = u^1$.

— Si $|v| \geq 1$ alors $|vt| = |u| < |uv| = n+1$, par l'hypothèse de récurrence il existe $w \in \mathcal{A}^*$ et $p, q \in \mathbb{N}$ tels que $v = w^p$ et $t = w^q$. On a alors $v = w^p$ et $u = vt = w^p w^q = w^{p+q}$.

Par le principe de récurrence généralisée, la propriété $P(n)$ est vraie pour tout $n \in \mathbb{N}$ donc pour deux mots $u, v \in \mathcal{A}^*$ tels que $uv = vu$ alors u et v sont des puissances d'un même mot w.

VI.3.4 Principe d'induction

On souhaite généraliser le principe de récurrence à des ensembles ordonnés bien fondés.

Théorème VI.7

Soit $\preceq$ un ordre bien fondé sur un ensemble E. Soit P une propriété dépendante d'un élément x de E. Si les deux hypothèses suivantes sont vérifiées

Initialisation : *$P(x)$ est vraie pour tout élément minimal x de E.*

Héridité : *Si pour tout $x \in E$ qui n'est pas minimal on a la propriété suivante :*

$$\text{"}P(y) \text{ est vraie pour tout } y \preceq x \text{ avec } y \neq x \Longrightarrow P(x) \text{ est vraie"}$$

Alors pour tout $x \in E$, la propriété $P(x)$ est vraie.

Démonstration : On raisonne par l'absurde : supposons que les hypothèse du théorème sont vraies mais que la conclusion est fausse.

Soit $X = \{x \in E, P(x) \text{ est fausse}\}$. L'ensemble X est une partie non vide de E, comme $(E, \preceq)$ est bien fondé, X admet un plus petit élément noté x_0.

Comme P est vraie pour tout élément minimal de E, l'élément x_0 n'est pas minimal. Pour tout $y \in E$ tel que $y \preceq x_0$ et $y \neq x_0$, la propriété $P(y)$ est vraie car x_0 minimal dans X et donc $y \notin X$. Par hypothèse d'hérédité $P(x_0)$ est vraie ce qui est contradictoire avec le fait que $x_0 \in X$. ■

Exemple VI.9. Montrons par induction que tout entier $n \in \mathbb{N} \setminus \{0, 1\}$ la propriété $P(n)$ suivante est vérifiée : "n s'écrit comme un produit de nombre premier".

L'ensemble $E = \mathbb{N} \setminus \{0, 1\}$ muni de l'ordre divise, noté $|$, est bien fondé et les éléments minimaux sont les nombres premiers.

Initialisation : Pour tout nombre premier $x \in E$ la propriété $P(x)$ est vraie.

Héridité : Soit $x \in E$ qui n'est pas minimal on suppose que pour tout $y \in E$ tel que tel que y divise x et $y \neq x$ la propriété $P(y)$ est vraie. Montrons que $P(x)$ est vraie.

Comme x n'est pas minimal, x n'est pas premier donc il existe $x_1, x_2 \in E$ tel que $x = x_1 \times x_2$. On a $x \neq x_1$ et $x \neq x_2$, et x_1 divise x et x_2 divise x. Par hypothèse d'hérédité x_1 et x_2 s'écrivent comme produit de nombre premiers donc $x = x_1 \times x_2$ s'écrit aussi comme produit de nombres premiers.

VI.3.5 Définition inductive

Définition inductive d'un ensemble

Définition VI.8. Soit E un ensemble. Une définition inductive d'un sous-ensemble X de E consiste à la donnée :

— d'un sous-ensemble B de E appelé *base*,

— d'un ensemble K d'opérations $\varphi : E^{r_\varphi} \longrightarrow E$ où r_φ est l'arité de φ.

L'ensemble X est alors défini comme le plus petit (pour l'inclusion) ensemble vérifiant les assertions suivantes :

Base : $B \subseteq X$,

Induction : pour tout $\varphi \in K$ et pour tous $x_1, x_2, \ldots, x_{r_\varphi} \in X$ on a $\varphi(x_1, x_2, \ldots, x_{r_\varphi}) \in X$.

On dit que X est la *fermeture inductive de B par K*.

Exemple VI.10. Quelques ensembles définis inductivement :

— L'ensemble des entiers naturels est défini par :
Base : $B = \{0\}$,
Induction : $\begin{array}{rccc} \text{succ} : & \mathbb{N} & \longrightarrow & \mathbb{N} \\ & n & \longmapsto & n+1 \end{array}$.

— L'ensemble des entiers pairs est défini par :
Base : $B = \{0\}$,

Induction : $\begin{array}{rccc}\varphi : & \mathbb{N} & \longrightarrow & \mathbb{N} \\ & n & \longmapsto & n+2\end{array}$.

— L'ensemble des entiers impairs est défini par :
Base : $B = \{1\}$,
Induction : $\begin{array}{rccc}\varphi : & \mathbb{N} & \longrightarrow & \mathbb{N} \\ & n & \longmapsto & n+2\end{array}$.

— L'ensemble des mots binaires est défini par :
Base : $B = \{\varepsilon\}$,
Induction : $\begin{array}{rccc}\varphi_0 : & \mathcal{A}^* & \longrightarrow & \mathcal{A}^* \\ & u & \longmapsto & 0u\end{array}$ et $\begin{array}{rccc}\varphi_1 : & \mathcal{A}^* & \longrightarrow & \mathcal{A}^* \\ & u & \longmapsto & 1u\end{array}$.

— L'ensemble des mots de $\{0,1\}^*$ qui contiennent autant de 0 que de 1 est défini par :
Base : $B = \{\varepsilon\}$,
Induction : $\begin{array}{rccc}\psi_1 : & \mathcal{A}^* \times \mathcal{A}^* & \longrightarrow & \mathcal{A}^* \\ & (u,v) & \longmapsto & 0u1v\end{array}$ et $\begin{array}{rccc}\psi_2 : & \mathcal{A}^* \times \mathcal{A}^* & \longrightarrow & \mathcal{A}^* \\ & (u,v) & \longmapsto & 1u0v\end{array}$.

— L'ensemble des mots de Dyck $\Delta \subseteq \{0,1\}^*$ est défini par :
Base : $B = \{\varepsilon\}$,
Induction : $\begin{array}{rccc}\psi : & \mathcal{A}^* \times \mathcal{A}^* & \longrightarrow & \mathcal{A}^* \\ & (u,v) & \longmapsto & 0u1v\end{array}$.

Preuve par induction

Le principe d'induction permet de démontrer des propriétés sur des ensembles définis inductivement.

Théorème VI.8

Soit $X \subseteq E$ la fermeture inductive de B par K. Soit P une propriété définie sur X. Pour montrer que pour tout $x \in X$ la propriété $P(x)$ est vraie, il suffit de montrer que :

Base : *Pour tout $x \in B$, on a $P(x)$ vraie.*

Induction *Pour tout $\varphi \in K$ d'arité r_φ et tous $x_1, x_2, \ldots, x_{r_\varphi}$ alors on a*

$$P(x_1), P(x_2), \ldots, P(x_{r_\varphi}) \text{ vraies} \implies P(\varphi(x_1, x_2, \ldots, x_{r_\varphi})) \text{ vraie}$$

Exemple VI.11. On considère l'ensemble des mots de Dyck $\Delta \subseteq \{0,1\}^*$ défini par :
Base : $B = \{\varepsilon\}$,
Induction : $\begin{array}{rccc}\psi : & \mathcal{A}^* \times \mathcal{A}^* & \longrightarrow & \mathcal{A}^* \\ & (u,v) & \longmapsto & 0u1v\end{array}$.

On veut montrer par induction que tout mot $w \in \Delta$ vérifie la propriété $P(w)$: "w a autant de 0 que de 1 et tout préfixe de w a plus de 0 que de 1".
Base : ε vérifie la propriété demandée,
Induction : Soient $u, v \in \Delta$ tels que $P(u)$ et $P(v)$ soient vérifiées. On note $w = \psi(u,v) = 0u1v$. Comme u et v ont autant de 0 et de 1, il en est de même pour w. Soit t un préfixe de w. Il y a deux cas :
— Si $|t| \leq 1 + |u|$ alors t est un préfixe de $0u$. Il s'écrit $t = 0t'$ où t' est un préfixe de u. Comme les préfixes de u ont plus de 0 que de 1, on en déduit que t a plus de 0 que de 1.
— Si $|t| > 1 + |u|$ alors t s'écrit $t = 0u1t'$ où t' est un préfixe de v. Comme les préfixes de v ont plus de 0 que de 1, on en déduit que t a plus de 0 que de 1.

Définition inductive d'une fonction

Lorsqu'un ensemble X est défini inductivement de telle sorte que chaque élément se décompose de manière unique, il est possible de définir une fonction sur cette ensemble. Cela est très utile pour programmer récursivement des fonctions.

Définition VI.9. Soit $X \subseteq E$ la fermeture inductive de B par K. On dit que X est *librement engendré* si pour tout $x \in X$ on a $x \in B$ ou bien il existe une unique règle $\varphi \in K$ d'arisé r_φ et un unique k-uplet $(x_1, \ldots, x_{r_\varphi})$ tels que $x = \varphi(x_1, \ldots, x_{r_\varphi})$.

Définition VI.10. Soient Y un ensemble et X un sous-ensemble de E défini par :
— une base $B \subseteq E$,
— un ensemble K d'opérations $\varphi : E^{r_\varphi} \longrightarrow E$ où r_φ est l'arité de φ.
Une fonction $f : X \to Y$ est définie *inductivement* par la donnée de :

Base : $f(b)$ pour tout $b \in B$,

Induction : Pour tout $\varphi \in K$ et pour tous $x_1, x_2, \ldots, x_{r_\varphi} \in X$ la valeur de $f(\varphi(x_1, x_2, \ldots, x_{r_\varphi}))$ se définit à partir de $f(x_1), f(x_2), \ldots, f(x_{r_\varphi})$.

Exemple VI.12. Quelques fonctions définies inductivement :
— La factorielle d'un entier $n \in \mathbb{N}$ se définit par :
Base : $0! = 1$,
Induction : $(n+1)! = (n+1) \times n!$.
— L'exposant d'un réel a^n se définit pour $n \in \mathbb{N}$ par :
Base : $a^0 = 1$,
Induction : $a^{n+1} = a \times a^n$.
— La longueur $l : \mathcal{A}^* \to \mathbb{N}$ d'un mot binaire $u \in \mathcal{A}^*$ est défini inductivement par
Base : $l(\varepsilon) = 0$;
Induction : Pour $u \in \mathcal{A}^*$, on a $l(\varphi_0(u)) = l(0u) = 1 + l(u)$ et $l(\varphi_1(u)) = l(1u) = 1 + l(u)$.

Langages rationnels

Définition VI.11. On définit inductivement les langages rationnels $\mathcal{R}at \subseteq \mathcal{P}(A^*)$ par :
Base : $\emptyset \in \mathcal{R}at$, $\{\varepsilon\} \in \mathcal{R}at$ et $\{a\} \in \mathcal{R}at$ pour tout $a \in \mathcal{A}$;
Induction : Si $\mathcal{L}$ et $\mathcal{L}'$ sont des langages rationnels alors :
— $\mathcal{L} \cup \mathcal{L}' \in \mathcal{R}at$,
— $\mathcal{L}.\mathcal{L}' \in \mathcal{R}at$,
— $\mathcal{L}^* \in \mathcal{R}at$.

Exemple VI.13. Voilà quelques exemples de langages rationnels :
— tous les langages finis sont rationnels et en particulier $\mathcal{A}$;
— $\mathcal{A}^*$ est rationnel ;
— $\mathcal{A}^+ = \mathcal{A}.\mathcal{A}^*$;
— le langage des mots sur $\mathcal{A} = \{0,1\}$ qui contient au moins une fois le mot 111 est rationnel car il s'écrit $\mathcal{A}^*.\{111\}.\mathcal{A}^*$;
— le langage des mots sur $\mathcal{A} = \{0,1\}$ qui contient un nombre pair de fois la lettre 1 est rationnel car il s'écrit $(\{b\}^*.\{a\}.\{b\}^*.\{a\}.\{b\}^*)^*$;

Les langages rationnels constituent un outil pour décrire des langages simples (rationnels). Ces formules sont par exemple utilisées pour effectuer des recherche des occurrences d'un motif. On notera deux applications concrètes notables :
— Sous UNIX, il existe par exemple un utilitaire `grep` qui permet de rechercher les occurrences d'un motif dans un fichier texte. La commande suivante imprime sur la sortie standard toutes les lignes du fichier 'cours.pdf' contenant au moins une occurrence du mot graphe :
```
> grep 'graphe' cours.pdf
```
Il est possible de faire des recherches de répétition (`e+`) ...
— L'analyse lexicale se trouve tout au début de la chaîne de compilation. C'est la tâche consistant à décomposer une chaîne de caractères en lexèmes, qui vont ensuite être analysés par l'analyseur syntaxique qui va ensuite les interpréter.

Chapitre VII

Quelques problèmes sur les graphes

VII.1 Différents problèmes à modéliser

On peut considérer que l'article fondateur de la théorie des graphe fut publié par le mathématicien suisse Leonhard Euler en 1741. Il traitait du problème des sept ponts de Königsberg : est il possible de réaliser une promenade dans la ville de Königsberg partant d'un point donné et revenant à ce point en passant une et une seule fois par chacun des sept ponts de la ville ?

Cette théorie va connaitre un essor au cours du $XIX^{\text{ème}}$ par l'intermédiaire du problème suivant : quel est le nombre minimal de couleurs nécessaires pour colorier une carte géographique de telle sorte que deux régions limitrophe n'ont pas la même couleur ? Le théorème des quatre couleurs affirme que seulement quatre sont nécessaires. Le résultat fut conjecturé en 1852 par Francis Guthrie, intéressé par la coloration de la carte des régions d'Angleterre, mais ne fût démontré qu'en 1976 par deux Américains Kenneth Appel et Wolfgang Haken. Ce fut la première fois que l'utilisation d'un ordinateur a permis de conclure leur démonstration en étudiant les 1478 cas particulier auxquels ils ont ramené le problème.

Au $XX^{\text{ème}}$ siècle, la théorie des graphes va connaître un essor croissant avec le développement des réseaux dont il faut optimiser l'utilisation. On peut citer quelques exemples de manière non exhaustive :

- réseaux de transports routier, d'eau, d'électricité : les sommets représentent les carrefours et les arêtes les rues ;
- réseaux informatiques : les sommets représentent les ordinateurs et les arêtes les connexions physiques ;
- réseaux sociaux : les sommets représentent les membres du groupe, deux personnes sont reliées par une arête si elles se connaissent (Facebook : graphe non orienté, twiter : graphe orienté, combien de poignées de main on est du président ?...) ;
- graphe du web : les sommets représentent les pages web et chaque arc correspond a un hyperliens d'une page vers une autre ;
- réseau de transports de données (téléphonie, wifi, réseaux informatique...) ;
- représentation d'un algorithme, du déroulement d'un jeu ;
- réseaux de régulation génétique ;
- organisation logistique : les sommets représentent des évènements, deux évènements sont reliées par une arête s'ils ne peuvent pas avoir lieu en même temps ;
- ordonnancement de projet : les sommets représentent les différentes tâches composant un projet, deux tâches sont reliés par une flèche si la deuxième ne peut pas commencer avant que la première soit terminée ;
- et beaucoup d'autres encore...

L'étude des graphes se réalise sous deux point de vues complémentaire. L'étude de propriétés structurelles de graphes ou de familles de graphes et l'étude algorithmique de certaines propriétés.

VII.2 Premières propriétés

VII.2.1 Graphe orienté ou non

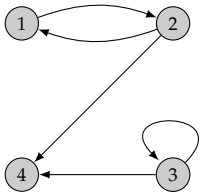
Dans les exemples que l'on a vus, un graphe est un ensemble fini de sommets reliés par des arêtes. Ces arêtes peuvent être orientées ou non, de plus une valeur peut être associée à chaque arête ou aux sommets.

Définition VII.1. Un *graphe orienté* $G = (S, A)$ est la donnée :
- d'un ensemble S dont les éléments sont des sommets ;
- d'un ensemble $A \subseteq S \times S$ dont les éléments sont les arcs.

Un arc $a = (s, s')$ est aussi noté $s \to s'$, s est l'*origine* de a et s' l'*extrémité*. On dit aussi que s' est le *successeur* de s et s le *prédécesseur* de s'.

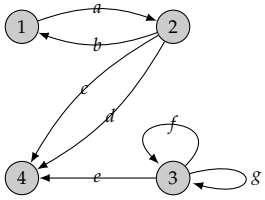
On peut souhaiter qu'il y ait plusieurs arcs entre deux mêmes sommets. On parle alors de graphe orienté *multi-arcs*. Formellement, $G = (S, A, \mathbf{i}, \mathbf{f})$ c'est la donnée :
- d'un ensemble S dont les éléments sont des sommets ;
- d'un ensemble A dont les éléments sont les arcs ;
- de deux fonctions $\mathbf{i} : A \to S$ et $\mathbf{f} : A \to S$ qui à chaque arcs $a \in A$ associe son prédécesseur $\mathbf{i}(a)$ et son successeur $\mathbf{f}(a)$.

Exemple VII.1. Exemple de graphe orienté :



$G = (S, A)$ où
- $S = \{1, 2, 3, 4\}$,
- $A = \{(1,2), (2,1), (2,4), (3,4), (3,3)\}$.

Exemple de graphe orienté multi-arcs :



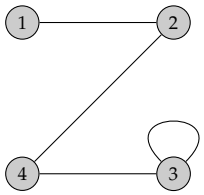
$G = (S, A, \mathbf{i}, \mathbf{f})$ où
- $S = \{1, 2, 3, 4\}$,
- $A = \{a, b, c, d, e, f, g, h\}$,
- $\mathbf{i} : \begin{array}{ccc} a & \mapsto & 1 \\ b & \mapsto & 2 \\ c & \mapsto & 2 \\ d & \mapsto & 2 \\ e & \mapsto & 3 \\ f & \mapsto & 3 \\ g & \mapsto & 3 \end{array}$ et $\mathbf{f} : \begin{array}{ccc} a & \mapsto & 2 \\ b & \mapsto & 1 \\ c & \mapsto & 4 \\ d & \mapsto & 4 \\ e & \mapsto & 4 \\ f & \mapsto & 3 \\ g & \mapsto & 3 \end{array}$.

Définition VII.2. Un *graphe non orienté* $G = (S, A)$ est la donnée :
- d'un ensemble S dont les éléments sont les sommets du graphe,
- d'un ensemble A dont les éléments, les arêtes du graphe, sont des parties à un ou deux éléments de S.

Le ou les sommets d'une arête sont appelés extrémités de l'arête. Les arêtes n'ayant qu'une seule extrémité sont des boucles.

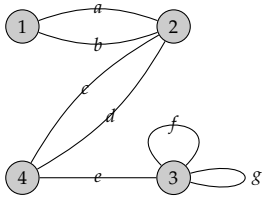
On peut de la même façon un graphe non-orienté *multi-arêtes*. Formellement, $G = (S, A, \alpha)$ est la donnée :
- d'un ensemble S dont les éléments sont des sommets ;
- d'un ensemble A dont les éléments sont les arêtes ;
- d'une fonction α de A dans les parties à un ou deux éléments de S.

Exemple VII.2. Exemple de graphe non-orienté :



$G = (S, A)$ où
— $S = \{1, 2, 3, 4\}$,
— $A = \{\{1, 2\}, \{2, 4\}, \{3, 4\}, \{3\}\}$.

Exemple de graphe non orienté multi-arêtes :



$G = (S, A, \alpha)$ où
— $S = \{1, 2, 3, 4\}$,
— $A = \{a, b, c, d, e, f, g, h\}$,
— $\alpha : \begin{array}{lcl} a & \mapsto & \{1,2\} \\ b & \mapsto & \{1,2\} \\ c & \mapsto & \{2,4\} \\ d & \mapsto & \{2,4\} \\ e & \mapsto & \{3,4\} \\ f & \mapsto & \{3\} \\ g & \mapsto & \{3\} \end{array}$.

Si un arc ou une arête à ses deux extrémités constituées du même sommet, on dit que c'est une *boucle*.

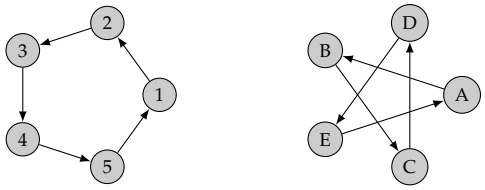
Un graphe est *simple* s'il est non-orienté, s'il a au plus une arête entre deux sommets et s'il n'a pas de boucle.

L'*ordre* d'un graphe est le nombre de sommets $|S|$ et la *taille* d'un graphe est le nombre d'arêtes ou d'arcs.

VII.2.2 Isomorphisme de graphe

Deux graphes orientés $G = (S, A)$ et $G' = (S', A')$ sont *isomorphes* s'il existe une application bijective $\varphi : S \to S'$ telle que pour tout $s, s' \in S$ on $(s, s') \in A \iff (\varphi(s), \varphi(s')) \in A'$. L'application φ est alors un *isomorphisme de graphes orientés*.

Exemple VII.3. Les deux graphes suivants sont isomorphes par l'isomorphisme $\varphi : 1 \mapsto A, 2 \mapsto B, 3 \mapsto C, 4 \mapsto D, 5 \mapsto E$.



De même, deux graphes non-orientés $G = (S, A)$ et $G' = (S', A')$ sont *isomorphes* s'il existe une application bijective $\varphi : S \to S'$ telle que pour tout $s, s' \in S$ on $\{s, s'\} \in A \iff \{\varphi(s), \varphi(s')\} \in A$. L'application φ est alors un *isomorphisme de graphes non-orientés*.

VII.2.3 Degré

Pour un graphe orienté, on appèle *degré entrant* d'un sommet s, noté $d_-(s)$ (resp. *degré sortant* d'un sommet s, noté $d_+(s)$) le nombre d'arcs dont le sommet est prédécesseur (resp. successeur).

Pour un graphe non-orienté, on appelle *degré* d'un sommet s, noté $d(s)$ le nombre d'arêtes dont le sommet est une extrémité.

Théorème VII.1 Lemme de la poignée de main

Soit $G = (S, A)$ un graphe orienté. On alors les égalités suivantes :

$$\sum_{s \in S} d_+(s) = \sum_{s \in S} d_-(s) = |A|.$$

Soit $G = (S, A)$ un graphe non-orienté. On a alors l'égalité suivante :

$$\sum_{s \in S} d(s) = 2|A|.$$

Démonstration : Pour un graphe orienté $G = (S, A)$, chaque arc a un successeur et un prédécesseur d'ou la première égalité.

Pour obtenir la deuxième égalité, il suffit d'orienté le graphe non-orienté et remarquer que pour chaque sommet $d(s) = d_+(s) + d_-(s)$. ■

Une conséquence directe de ce théorème est que dans un graphe, le nombre de sommets dont le degré est impair est toujours pair.

Corollaire VII.2

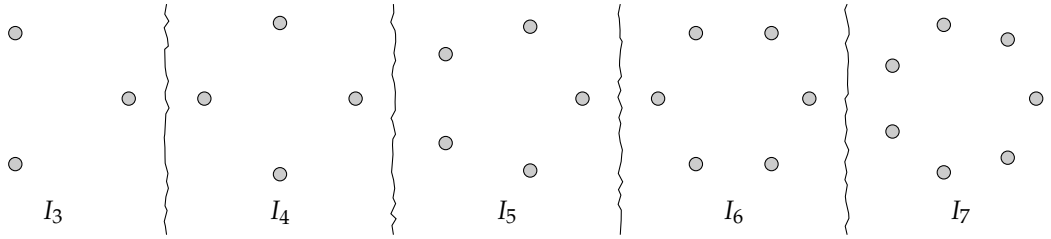
Dans un graphe, le nombre de sommets dont le degré est impair est toujours pair.

VII.3 Quelques classes de graphe importantes

On s'intéresse ici à définir quelques classes de graphes non-orientés dont la plupart sont simple (non multi-arête et sans boucle).

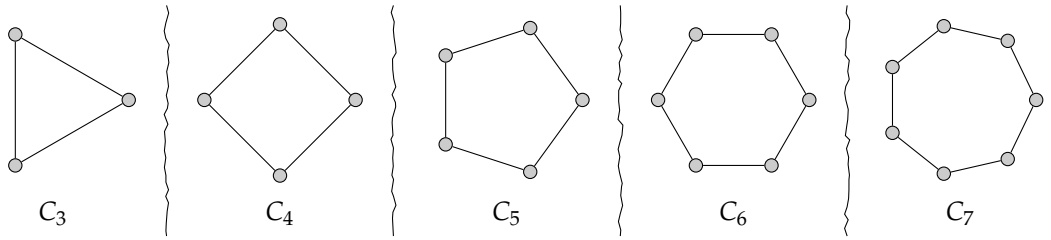
VII.3.1 Graphes isolés

Le *graphe isolé d'ordre n* est un graphe à n sommets sans arête, on le note I_n.



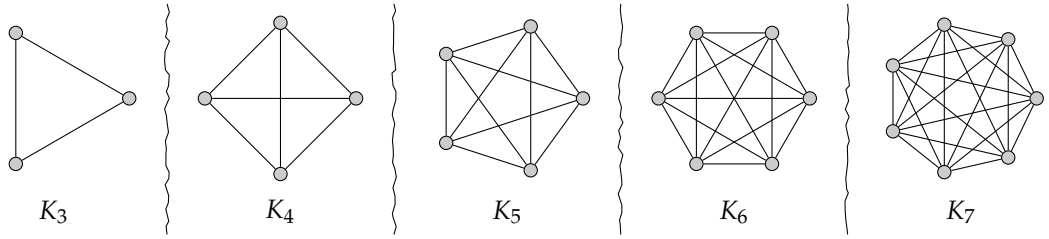
VII.3.2 Graphes cycliques

Le *graphe cyclique d'ordre n* est le graphe à n sommets $S = \{s_1, \ldots, s_n\}$ tels que les arêtes sont $A = \{\{s_i, s_{i+1}\} : i \in [1, n]\} \cup \{\{s_n, s_1\}\}$, on le note C_n.



VII.3.3 Graphes complets

Le *graphe complet d'ordre n* est le graphe simple à n sommets dont tous les sommets sont reliés deux à deux, on le note K_n.

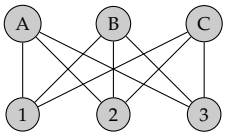


K_3 K_4 K_5 K_6 K_7

VII.3.4 Graphe biparti

Un graphe est *biparti* s'il existe une partition de son ensemble de sommets en deux sous-ensembles X et Y telle que chaque arête ait une extrémité dans X et l'autre dans Y.

On définit le *graphe biparti complet* entre un ensemble de n sommets et un ensemble à m sommets comme le graphe simple tel que chaque sommet du premier ensemble est relié à chaque sommet su deuxième ensemble. On le note $K_{n,m}$.

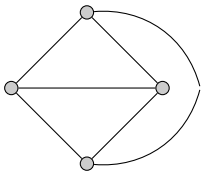


$K_{3,3}$

VII.3.5 Graphes planaires

Un graphe non-orienté (pas forcément simple) est *planaire* s'il admet une représentation sagittale dans un plan sans que les arêtes se croisent.

Exemple VII.4. K_4 est planaire puisque on peut le représenter de la façon suivante :



K_4

Est ce que K_5 et $K_{3,3}$ sont planaires ?

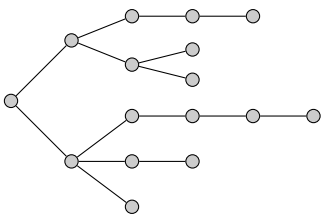
VII.3.6 Arbres

Définition VII.3. Un *arbre* se définit de manière inductive par :

— le graphe formé par un sommet est un arbre ;

— si $G = (S, A)$ est un arbre, alors pour $s \in S$ et x un élément quelconque n'appartenant pas à S, le graphe $G' = (S \cup \{x\}, A \cup \{\{x, s\}\})$ est un arbre.

Un exemple d'arbre :



VII.4 Problèmes de coloriages

VII.4.1 Position du problème

Définition VII.4. Soit $G = (S, A)$ un graphe non orienté simple (sans boucle et pas multi-arêtes). Un *coloriage* de G consiste à assigner une couleur (ou un nombre) à chaque sommet de telle sorte que deux sommets adjacents soient de couleurs différentes. Un graphe G est *k-coloriable* s'il existe un coloriage avec k couleurs.

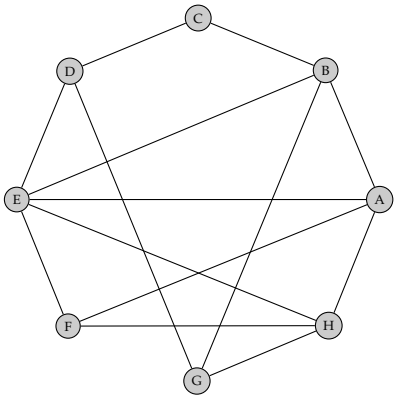
Le *nombre chromatique* du graphe G, noté $\chi(G)$ est le nombre minimal de couleurs nécessaire pour colorier un graphe.

VII.4.2 Exemples d'applications

Problème de compatibilité Dans un groupe de 14 étudiants, on doit former des groupes de telle sorte que les étudiants d'un même groupe ne s'entendent pas trop mal. On connaît les incompatibilités suivantes :

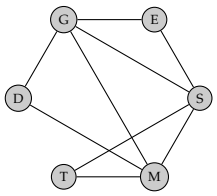
l'étudiant	A	B	C	D	E	F	G	H
ne s'entend pas avec	B,E,F,H	A,C,E,G	B,D	C,E,G	A,D,F,H	A,E,H	B,D,H	A,E,F,G

Le nombre minimal de groupes nécessaire correspond au nombre chromatique du graphe des incompatibilités.



Problème d'emploi du temps Pendant un festival, on veut organiser des tournois de scrable (S), échecs (E), go (G), dames (D), tarot (T) et master-mind (M). Plusieurs personnes se sont inscrites à la fois pour les tournois E, S, G, d'autres personnes pour les tournois G, D, M, et enfin d'autres personnes pour les tournois M, T, S. Il est entendu qu'une participation simultanée à plusieurs tournois est impossible et que les organisateurs veulent satisfaire tout le monde.

Quel est le nombre maximum de tournois qui pourraient se dérouler en même temps ?



Coloriage de carte On cherche à colorier une carte de telle sorte que deux pays frontaliers soient de couleurs différentes. Pour résoudre ce problème, plus historique qu'autre chose, on peut se ramener au coloriage d'un graphe planaire construit de la façon suivante : les sommet correspondent aux pays et il y a une arête entre deux sommets si les pays correspondant sont frontaliers.

VII.4.3 Nombre chromatique de graphes classiques

Il est facile de déterminer le nombre chromatique de certains graphes classiques :
— graphe isolé d'ordre n : $\chi(I_n) = 1$;
— graphe cyclique d'ordre n : $\chi(C_n) = 2$ si n pair et 3 si n impair ;
— graphe complet d'ordre n : $\chi(K_n) = n$;
— G graphe biparti avec au moins une arête : $\chi(G) = 2$ (en fait un graphe est 2-coloriable si et seulement s'il est biparti) ;
— G arbre avec au moins une arête : $\chi(G) = 2$.

VII.4.4 Comment calculer un nombre chromatique ?

Il est intéressant d'avoir des outils pour encadrer le nombre chromatique. On note qu'obtenir un coloriage à k couleurs d'un graphe G permet d'affirmer que $\chi(G) \leq k$. La difficulté réside pour trouver une minoration.

Proposition VII.3

Soit G un graphe et G' un sous graphe, on a $\chi(G') \leq \chi(G)$.

On va introduire deux nouvelles notions.

Définition VII.5. Soit G un graphe non orienté.
Une *clique* est un sous-graphe complet de G.
Une *stable* est un sous-graphe induit de G sans arcs (ou arêtes).

Ces notions donnes des informations sur le nombre chromatique :
— les sommets d'une même clique doivent être coloriés d'une couleur différente, ainsi trouver une clique à k sommets permet d'affirmer que $\chi(G) \geq k$;
— les sommets d'une même stable peuvent être coloriés de la même couleur.

VII.4.5 Résolution algorithmique

Dans cette section on s'intéresse aux algorithmes qui permettent de trouver un coloriage ou le nombre chromatique.

Algorithme glouton

On considère ici un coloriage comme une fonction des sommets dans les entiers. L'algorithme glouton nous donne facilement un coloriage du graphe, le principe consiste à prendre les sommets les uns après les autres et pour chaque sommet s d'affecter la couleur minimale qui n'apparait pas dans les voisins coloriés de s.

Algorithm 2: Algorithme glouton de coloriage d'un graphe

Data: Un graphe $G = (S, A)$
Result: Une coloration $\varphi : S \to \mathbb{N}^*$ de G

for $s \in S$ **do**
  $\varphi(s) \leftarrow$ plus petite couleur non utilisé par les voisins de s;

Terminaison L'algorithme termine une fois que l'on a visité tous les sommets.

Correction A chaque fois que l'on attribue une couleur à un sommet, elle est différentes des couleurs des sommets voisins pour lesquels on a attribué une couleur. Ainsi le coloriage obtenu est valide.

Complexité On passe $|S|$ fois dans la boucle, chaque fois que l'on passe dans la boucle on regarde tous les voisins du sommet considéré, on a au plus $\Delta(G)$ voisin à regarder où $\Delta(G)$ est le degré maximal du graphe. Dans le pire des cas, on une complexité $O(\Delta(G)|S|)$.

A t'on un coloriage optimal avec cet algorithme ? Le résultat dépend généralement de l'ordre dans lequel on choisit les sommets et il est facile de trouver des exemples ou l'ordre donné ne donne pas un coloriage optimal. On peut jouer sur l'ordre des sommets choisis, par exemple les prendre dans l'ordre des degrés décroissants.

Algorithme de Welsh-Powell

Il est possible d'améliorer cet algorithme en coloriant d'abord les sommets qui imposent le plus de contraintes (sommet de plus haut degré) et en utilisant la couleur que l'on vient d'utiliser là ou cela est possible. On appèle ce principe l'algorithme de Welsh-Powell. Pour certaine classe de graphe cet algorithme donne même systématiquement le coloriage optimal.

Algorithm 3: Algorithme de Welsh-Powell pour colorier un graphe

Data: Un graphe $G = (S, A)$
Result: Une coloration $\varphi : S \to \mathbb{N}$ de G

$L \leftarrow$ liste des sommets ordonnés par degré décroissant ;
couleur-courante $\leftarrow 0$;
while $L \neq \emptyset$ **do**
  couleur-courante $\leftarrow$ couleur-courante $+1$;
  Colorier s le premier sommet de L avec couleur-courante;
  Eliminer s de L;
  $V \leftarrow$ voisins de s;
  for $x \in L$ **do**
    if $x \notin V$ **then**
      Colorier x avec la couleur-courante;
      Eliminer x de L;
      Ajouter les voisins de x à V;

Terminaison Il est clair que, puisque le nombre de sommets dans L (et donc non coloriés) diminue d'au moins une unité à chaque fois que l'on exécute la boucle.

Correction Cette algorithme fournit bien un coloriage de G, en effet chaque fois que l'on colorie un sommet, on place dans V les sommets voisins à ce sommet de telle sorte que l'on ne colorie plus de cette couleur les sommets de V. Ainsi deux sommets voisins sont de couleurs différentes.

Complexité De manière grossière, on passe $|S|$ fois dans la boucle **while** puis $|S|$ fois dans la boucle **for**, on a donc une complexité grossière en $O(|S|^2)$. Cependant, on peut être plus précis. Dans la preuve de la proposition VII.4, on voit que l'on passe au maximum $\Delta(G) + 1$ fois dans la boucle **while**. On a donc une complexité en $O(\Delta(G)|S|)$.

Proposition VII.4

Soit $\Delta(G)$ le degré maximal d'un graphe G, on a $\chi(G) \leq \Delta(G) + 1$.

Démonstration : Soit s le dernier sommet colorié par l'algorithme 2. Si s n'a pas été colorié avant, c'est que pour chacune des couleurs précédentes, un sommet adjacent à s a été colorié de cette couleur. Par suite, le nombre de couleurs utilisées avant de colorier s ne peut dépasser $d(s)$. Ainsi, en tenant compte de la couleur de s, on déduit que le nombre total de couleurs utilisées par l'algorithme ne dépasse pas $d(s) + 1$. ■

A t'on un coloriage optimal avec cet algorithme ? Là encore il existe des exemples ou cet algorithme n'est pas optimal même si dans la majorité des cas il donne un coloriage optimal.

Existe t'il un algorithme pour trouver le nombre chromatique d'un graphe ?

On cherche un algorithme qui prend en argument un graphe $G = (S, A)$ et renvoie le nombre chromatique de ce graphe. Pour cela on teste tous les 2-coloriages, il y en a $2^{|S|}$ s'il y a en a un valide, on a $\chi(G) = 2$, sinon on teste tous les 3-coloriages et ainsi de suite. L'algorithme termine car il y a un coloriage à $\Delta(G) + 1$ couleurs et il nous donne un coloriage optimal car on a essayer toutes les possibilités avec moins de couleurs.

Cependant cet algorithme a une complexité en $O((\Delta(G) + 1)^{|S|})$ dans le pire des cas, cette complexité est par exemple atteinte pour le graphe complet. Cette complexité est exponentielle en la taille du graphe et en pratique, pour des graphes un peu grand, il faut attendre des temps extrêmement long pour le voir terminer. On estime que les complexités qui permettent d'avoir un algorithme utilisable sont les complexité en $O(n^d)$ pour une valeur d donnée. Pour le problème du nombre chromatique on ne sait pas s'il existe un algorithme polynomial qui permet de le résoudre.

Toutefois, il existe des classes de graphes pour lesquelles l'algorithme glouton (et donc de complexité polynomiale) donne même systématiquement le coloriage optimal. En TD on verra qu'un algorithme glouton avec un bon ordre sur les sommets donne un coloriage optimal pour les graphes d'intervalles.

Remarque VII.1. En général on s'intéresse aux problèmes de décisions, par exemple :

Problème 1 : Etant donné $a, b, c \in \mathbb{Z}$, est ce que $ax^2 + bx + c = 0$ admet une solution rélle ?

Problème 2 : Etant donné un graphe G est ce que G admet un 3-coloriage ?

On s'intéresse aux complexités qui résolvent ces problèmes, on définit les classes de problèmes suivant :

— Classe $\mathcal{P}$: classe de problèmes que l'on peut résoudre en temps polynomial (par exemple Problème 1) ;

— Classe $\mathcal{NP}$: classe de problèmes tel que si on donne une solution on peut vérifier que c'est bien une solution du problème (par exemple Problème 2) ;

— Classe $\mathcal{E}xp$: classe de problèmes que l'on peut résoudre en temps exponentiel.

On a $\mathcal{P} \subseteq \mathcal{NP} \subseteq \mathcal{E}xp$. On sait que $\mathcal{P} \neq \mathcal{E}xp$ mais on ne sait pas si $\mathcal{P} = \mathcal{NP}$, c'est **le** problème ouvert de l'informatique théorique.

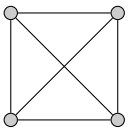
Il existe une autre classe, la classe des problèmes $\mathcal{NP}$-complet, ce sont les problèmes tels que si on les résout en temps polynomial, on résout tous les problèmes $\mathcal{NP}$ en temps polynomial. En particulier le problème de 3-coloriage est $\mathcal{NP}$-complet.

VII.4.6 Cas des graphes planaires

Les graphes planaires sont une classe graphe avec des propriétés intéressantes du point de vu du coloriage.

Définition VII.6 (Graphe planaire)**.** Un graphe $G = (S, A)$ est *planaire* s'il existe une représentation dans le plan où les arêtes ne s'intersectent pas.

Exemple VII.5. Le graphe suivant est planaire si on déplace les sommets



G_3

En fait, le nombre de couleur maximal pour colorié un graphe planaire est 4. Ce théorème est connu comme l'un des premiers ou la preuve nécessite un ordinateur pour explorer l'explosion combinatoire des différents cas de base.

Théorème VII.5

Tout graphe planaire est coloriable avec 4 *couleurs, son nombre chromatique est donc inférieur ou égal à 4.*

VII.5 Problèmes de chemins dans un graphe

VII.5.1 Définitions

Définition VII.7. Soit $G = (S, A)$ un graphe orienté (resp. non-orienté). Un *chemin* (resp. une *chaîne*) dans G est une suite de sommets $C = (s_0, s_1, s_2, \ldots, s_k)$ telle qu'il existe un arc (resp. une arête) entre chaque couple de sommets successifs de C. Ce qui s'écrit :

— si $G = (S, A)$ est orienté alors pour tout $i \in [0, k-1]$ on a $(s_i, s_{i+1}) \in A$,
— si $G = (S, A)$ est non-orienté alors pour tout $i \in [0, k-1]$ on a $\{s_i, s_{i+1}\} \in A$,

On appellera :

Chemin (resp. chaîne) simple **:** un chemin (resp. chaîne) dont tous les arcs (resp. arêtes) sont différents.

Chemin (resp. chaîne) élémentaire **:** un chemin (resp. chaîne) dont tous les sommets sont différents sauf peut être le départ et l'arrivée (pour autoriser les circuits ou cycles).

Circuit dans un graphe orienté **:** un chemin simple finissant à son point de départ.

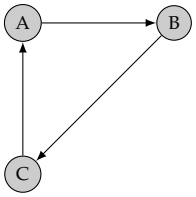
Cycle dans un graphe non-orienté **:** une chaîne simple finissant à son point de départ.

VII.5.2 Connexité

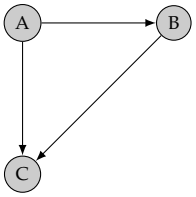
Définition VII.8 (Connexité et forte connexité)**.** Un graphe non-orienté est *connexe* si pour tout couple de sommets s et s', il existe une chaîne reliant s à s'.

Un graphe orienté est connexe si le graphe non orienté associé est connexe. Un graphe orienté est *fortement connexe* si pour tout couple de sommets s et s', il existe une chemin reliant s à s'.

Exemple VII.6 (Graphe connexe et fortement connexe). G_1 est fortement connexe tandis que G_2 est connexe mais non fortement connexe.



G_1



G_1

Définition VII.9 (Composantes connexes et fortement connexes)**.** Une composante connexe (resp. fortement connexe) C d'un graphe $G = (S, A)$ est un sous-ensemble maximal de sommets tels que deux quelconques d'entre eux soient reliés par une chaîne (resp. un chemin). Formellement, si $s \in C$ alors on a :

— pour tout $s' \in C$ il existe une chaîne (resp. un chemin) reliant s à s',
— pour tout $s' \in S \setminus C$, il n'existe pas de chaîne (resp. chemin) reliant s à s'.

Quelques propriétés :

— Les composantes connexes (resp. fortement connexe) d'un graphe $G = (S, A)$ forment une partition de S.
— Un graphe est connexe (resp. fortement connexe) si et seulement s'il a une seule composante connexe (resp. fortement connexe).
— Le sous-graphe induit par une composante connexe (resp. fortement connexe) est connexe (resp. fortement connexe).

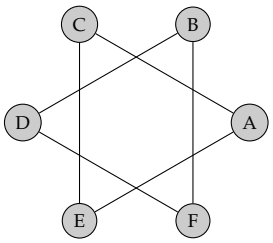
— La composante connexe C qui contient un sommet $s \in S$ est

$$C = \{s' \in S \text{ tel qu'il existe une chaîne reliant } s \text{ à } s'\}$$

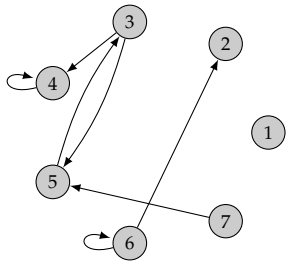
— La composante fortement connexe C qui contient un sommet $s \in S$ est

$$C = \{s' \in S \text{ tel qu'il existe un chemin reliant } s \text{ à } s' \text{ et un chemin reliant } s' \text{ à } s\}$$

Exemple VII.7 (Composantes connexes). Les composantes connexes de G_1 sont $\{A, C, E\}$ et $\{B, D, F\}$ tandis que celles de G_4 sont $\{1\}$, $\{2, 6\}$, $\{3, 5, 7\}$ et $\{4\}$.

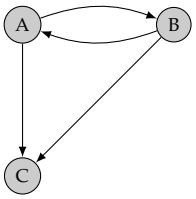


G_1

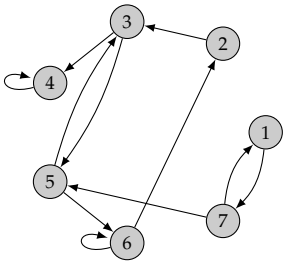


G_2

Exemple VII.8 (Composantes fortement connexes). Les composantes fortement connexes de G_1 sont $\{A, B\}$ et $\{C\}$ tandis que celles de G_2 sont $\{1, 7\}$, $\{2, 3, 5, 6\}$ et $\{4\}$.



G_1

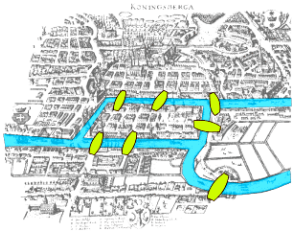


G_2

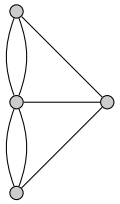
VII.5.3 Chemin Eulérien

Problématique

Au $XVIII^{\text{ème}}$ siècle un casse-tête est populaire chez les habitants de Königsberg : est-il possible de se promener dans la ville en ne passant qu'une seule fois par chacun des sept ponts de Königsberg ? C'est le célèbre mathématicien Euler qui montre le premier que ce problème n'a pas de solution, en utilisant pour la première fois la notion de graphe. Le problème se reformule ainsi en terme de graphes : existe-t-il un cycle qui passe exactement une fois par toutes les arêtes dans le graphe (multi-arête) ci-dessous ?



Ville de Königsberg

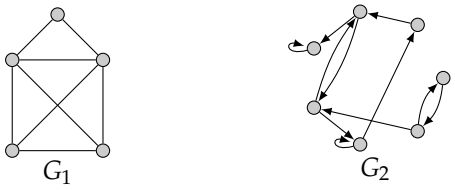


G

Définition VII.10. Soit G un graphe non orienté. Une chaîne (resp. un cycle) *eulérienne* est une chaîne (resp. un cycle) qui passe une et une seule fois par toutes les arêtes de G.

On définit les mêmes notions pour un graphe orienté G : un chemin (resp. un circuit eulérien) est un chemin (resp. un circuit) passant une et une seule fois par tous les arcs de G.

Exemple VII.9. Le graphe G_1 admet un cycle eulérien. Le graphe G_2 admet un chemin eulérien mais pas un circuit.



Caractérisation des chemins eulériens

Avant de prouver la caractérisation des chemin eulériens, on a besoin du résultat suivant.

Proposition VII.6

Un graphe dont tous les sommets sont de degré supérieur ou égal à 2 possède au moins un cycle.

Démonstration : La preuve utilise un algorithme de marquage. Initialement tous les sommets sont non marqués. Un sommet s_1 est marqué arbitrairement. L'algorithme construit alors une séquence $s_1, \ldots, s_k$ de sommets marqués en choisissant arbitrairement pour s_{i+1} un sommet non marqué adjacent à s_i. L'algorithme s'arrête lorsque s_k ne possède plus de voisin non marqué. Puisque ce sommet est de degré au moins 2, il possède un voisin $s_j \neq s_{k-1}$ dans la séquence, $j < k-1$. On en déduit que $(s_k, s_j, s_{j+1}, \ldots, s_{k-1}, s_k)$ est un cycle. ■

Théorème VII.7

Soit $G = (S, A)$ un graphe non orienté connexe. Il admet un cycle eulérien si et seulement si $d(s)$ est pair pour tout $s \in S$.

Si seulement deux sommets ne vérifient pas les conditions précédentes alors G admet une chaîne Eulériene.

Démonstration : Soit $G = (S, A)$ un graphe connexe. Pour qu'il admette un cycle Eulérien il faut qu'en chaque sommet lorsqu'on arrive par une arête on puisse repartir par un autre arête. On obtient donc que $d(s)$ est pair si le graphe est orienté pour chaque sommet $s \in S$.

Réciproquement, on démontre par récurrence sur le nombre d'arcs que pour un graphe connexe G, si chaque sommet $s \in S$ est de degré pair alors G admet un cycle eulérien.

Initialisation : Si $|A| = 0$, on a un graphe connexe sans arêtes, c'est à dire un seul sommet isolé qui admet un cycle eulérien.

Induction : On suppose que le théorème est vrai pour tout graphe ayant un nombre d'arêtes inférieur ou égal à n (hypothèse de récurrence forte). Soit $G = (S, A)$ un graphe connexe tel que $|A| = n+1$ et pour chaque sommet $s \in S$ est de degré pair. Comme le graphe est connexe et que le degré de chaque sommet est pair, on en déduit que G admet un cycle élémentaire $C = (s_1, s_2, \ldots, s_k, s_1)$.

Soit G' le sous-graphe de G auquel on a supprimé les arêtes de C. Le graphe G' n'est pas forcément connexe mais vérifie $d(s)$ pairs pour chacun de ses sommet s. On applique l'hypothèse de récurrence sur chacune de ses composantes qui admettent donc des cycles eulériens. On combine alors ces différents cycles eulériens avec le cycle C, pour former un cycle eulérien sur G de la façon suivante : on parcourt C depuis un sommet initial arbitraire et, à chaque fois que l'on rencontre une des composantes connexes de G' pour la première fois, on insère le cycle eulérien considéré sur cette composante. S'agissant d'un cycle, on est assuré de pouvoir poursuivre le parcours de C après ce détour. Il est facile de vérifier qu'on a ainsi bien construit un cycle eulérien sur G.

Si G admet une chaîne Eulérienne et admet un sommet de degré impair, soit c'est le point de départ de la chaîne, soit il arrive un moment où l'on ne pourra plus repartir ce qui constitue le sommet terminal de la chaîne. Ainsi, si seulement deux sommets sont de degré impair il peuvent servir de point de départ et d'arrivé d'un chemin passant par tous les arêtes du graphe, le graphe peut donc admettre une chaîne Eulérienne. ■

Dans le cas orienté on montre de manière similaire le résultat suivant.

Théorème VII.8

Soit $G = (S, A)$ un graphe orienté fortement connexe. Il admet un circuit eulérien si et seulement si $d_+(s) = d_-(s)$ pour tout $s \in S$.

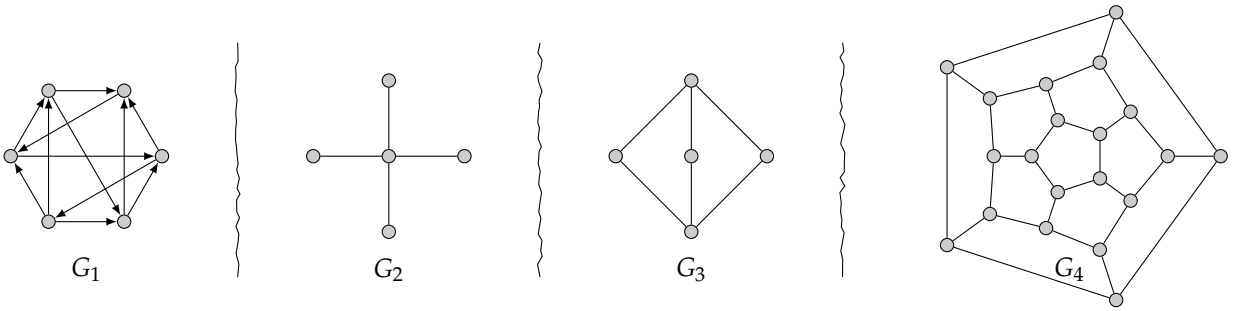
Si seulement deux sommets vérifient $|d_+(s) - d_-(s)| = 1$ alors G admet un chemin Eulérien.

VII.5.4 Chemins hamiltonien

Définition VII.11. Soit G un graphe non orienté. Un *cycle* (respectivement une *chaîne*) *hamiltonien* est un cycle (resp. une chaîne) qui passe une et une seule fois par tous les sommets de G.

On définit les mêmes notions pour un graphe orienté G : un *circuit* ou un *chemin hamiltonien* est un circuit ou un chemin passant une et une seule fois par tous les sommets de G

Exemple VII.10. G_1 admet un circuit hamiltonien, G_2 n'admet ni chaîne ni cycle hamiltoniens, G_3 admet une chaîne hamiltonienne mais pas de cycles hamiltoniens et G_4 admet un cycle hamiltonien.



On ne connaît pas de condition nécessaire et suffisante exploitable dans la pratique pour décider si un graphe est hamiltonien ou non. De manière générale, la recherche de cycle, chaîne, circuit ou chemin Hamiltonien est un problème algorithmiquement difficile. En fait, on peut montrer que c'est un problème NP-complet.