Mathématiques pour l'informatique

Christophe GUYEUX et Jean-François COUCHOT
guyeux[arobase]iut-bm.univ-fcomte[point]fr
couchot[arobase]iut-bm.univ-fcomte[point]fr

3 novembre 2010

Table des matières

Première partie

Théorie des ensembles

Chapitre 1

Introduction à la théorie des ensembles

I Rappels de théorie des ensembles

I.1 Notion première d'ensemble

Ensemble Notion première qui ne se définit pas. C'est une collection d'objets réunis en vertu d'une propriété commune.

On peut définir un ensemble de deux manières :

- *en extension* : on donne la liste exhaustive des éléments qui y figurent,
- *en compréhension* : en donnant la propriété que doivent posséder les éléments de l'ensemble.

Exercice 1.1. *Définir les ensembles suivants en compréhension :*

1. *A = {1,2,4,8,16,32,64}*
2. *B = {1,2,7,14}*
3. *C = {4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20 }*

Réponse : 1) Les puissances de 2 inférieures ou égales à 64. 2) Les diviseurs de 14. 3) Les entiers inférieurs ou égaux à 20 qui ont au moins 3 diviseurs (les nombres non premiers entre 2 et 20).

NOTATION : On note $\mathbb{N}_n$ l'ensemble des entiers inférieurs ou égaux à n.

Exercice 1.2. *Définir les ensembles suivants en extension*

1. $A = \{x \in \mathbb{R} | x(x+5) = 14\}$
2. $B = \{x \in \mathbb{N} | x(2x+3) = 14\}$
3. $C = \{x \in \mathbb{N}_{10}^* | x^4 - 1$ *est divisible par 5* $\}$

Réponse : A = {2, -7}, B = {2}, et C = {1, 2, 3, 4, 6, 7, 8, 9} (factoriser $x^4 - 1$).

I.2 Règles de fonctionnement

Relation d'appartenance. On admet être capable de décider si un objet est ou non élément d'un ensemble. Le fait que l'élément x appartienne à l'ensemble X se note : $x \in X$.

Objets distincts. On admet aussi être capable de distinguer entre eux les éléments d'un ensemble. En particulier, un ensemble ne peut pas contenir deux fois le même objet.

Ensemble vide. Il existe un ensemble ne contenant aucun élément, appelé ensemble vide. Symbole : le cercle barré [1] $\varnothing$.

L'ensemble vide ne correspond pas à rien ; c'est en fait un ensemble qui ne contient rien, mais en tant qu'ensemble il n'est pas rien : un sac vide est vide, mais le sac en lui même existe.

La notation $\{\varnothing\}$ n'a pas le même sens que $\varnothing$. La dernière notation décrit un ensemble qui ne contient rien alors que le premier décrit un ensemble contenant un élément : l'ensemble vide. On peut, afin de mieux comprendre, reprendre l'analogie du sac vide. Un tiroir contenant un sac vide - $\{\varnothing\}$ - n'est pas vide - $\varnothing$ - et contient bien un objet.

D'ailleurs, l'ensemble $\{\varnothing\}$ contient un élément (qui est un ensemble), quand l'ensemble $\varnothing$ n'en contient aucun...

Dernière règle de fonctionnement des ensembles. Un ensemble ne peut pas s'appartenir à lui-même.

Cette dernière règle de fonctionnement peut sembler obscure, pas naturelle.

Dans l'euphorie de la naissance de la théorie des ensembles, les mathématiciens ne voyaient pas d'objection à envisager un ensemble Ω dont les éléments seraient tous les ensembles (en particulier, $\Omega \in \Omega$) : l'ensemble des ensembles, à l'origine de tout !

Leur enthousiasme fut stoppé lorsque Russell leur opposa le paradoxe...

Exercice 1.3 (Paradoxe de Bertrand Russell (1872-1970)). *Soit X l'ensemble de tous les éléments qui ne sont pas éléments d'eux-mêmes.*

A-t-on $X \in X$? A-t-on $X \notin X$?

REMARQUE 1.1. Dit autrement : le barbier qui rase tous les barbiers qui ne se rasent pas eux-mêmes...se rase-t-il lui-même ?

REMARQUE 1.2. On en déduit donc que l'on ne peut pas parler de l'ensemble de tous les ensembles (cet ensemble devrait s'appartenir lui-même). Il ne faut pas négliger l'impact d'une telle révélation.

I.3 Sous-ensembles, ensemble des parties

Sous-ensemble. Les sous-ensembles sont définis par la relation d'inclusion...

DÉFINITION 1.1. *A est un sous-ensemble de B ($A \subset B$) » si et seulement si tout élément de A appartient à B. On dit aussi que A est une partie de B.* ♢

PROPRIÉTÉ 1.1 : L'ensemble vide est inclus dans n'importe quel ensemble.

PREUVE D'après la définition d'un sous-ensemble, cela veut dire que pour tout élément x de $\varnothing$, x appartient à A. Raisonnons a contrario : si l'ensemble vide n'est pas inclus dans A, alors il existe au moins un élément de l'ensemble vide qui n'appartient pas à A. Or, il n'y a aucun élément dans l'ensemble vide, donc plus particulièrement aucun élément de l'ensemble vide qui n'appartienne pas à A. On en conclut donc que tout élément de $\varnothing$ appartient à A, et donc que $\varnothing$ est un sous-ensemble de A. ■

Plus généralement, toute proposition commençant par « pour tout élément de $\varnothing$» est vraie. On a de plus le résultat suivant :

1. La notation $\varnothing$ a été introduite par le mathématicien français André Weil du groupe Bourbaki. Unicode : U+00D8

PROPRIÉTÉ 1.2 : Tout ensemble est inclus dans lui-même.

Ensemble des parties.

DÉFINITION 1.2. *Soit A un ensemble. L'ensemble des parties de A, noté $\mathcal{P}(A)$, est l'ensemble de tous les sous-ensembles de A.* ♢

On sait déjà que $\varnothing$ et A sont deux parties de A...

PROPRIÉTÉ 1.3 : Pour tout ensemble A, on a $\varnothing, A \in \mathcal{P}(A)$.

EXEMPLE 1.4. Si $A = \{1, 2, 3\}$, alors $\mathcal{P}(A) = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$

EXEMPLE 1.5. Si $A = \varnothing, \mathcal{P}(A) = \{\varnothing\}, \mathcal{P}\left(\mathcal{P}(a)\right) = \{\varnothing, \{\varnothing\}\}$. Cela n'est pas qu'un jeu de l'esprit :
- On définit 0 comme étant $\varnothing$,
- 1 correspond alors à $\mathcal{P}(\varnothing)$,
- 2 est alors $\mathcal{P}(\mathcal{P}(\varnothing))$,
- *etc.*

D'autres définitions de l'ensemble des entiers naturels existent.

De manière plus générale, si A possède n éléments, $\mathcal{P}(A)$ en possède 2^n.

Exercice 1.6. *Justifier le fait que le nombre d'éléments de $\mathcal{P}(A)$ est égal à 2^n, où n représente le nombre d'éléments de A.*

Exercice 1.7. *On considère A = {1,2}. Dire quelles assertions sont exactes :*
- $1 \in A$,
- $1 \subset A$,
- $\{1\} \in A$,
- $\{1\} \subset A$,
- $\varnothing \in A$,
- $\varnothing \subset A$.

Exercice 1.8. *Reprendre l'exercice précédent, avec $A = \{\{1\}, \{2\}\}$.*

I.4 Représentation graphique

On peut représenter ensembles et sous-ensembles à l'aide d'un diagramme de Venn (les célèbres « patates »)...

Exercice 1.9 (Diagramme de Venn). *A partir des affirmations*
1. *les poëtes sont des gens heureux,*
2. *tous les docteurs sont riches et*
3. *nul être heureux n'est riche,*

déterminer la validité de chacune des conclusions suivantes
1. *Aucun poëte n'est riche.*
2. *Les docteurs sont des gens heureux.*
3. *Nul ne peut être à la fois docteur et poëte.*

I.5 Exercices

Exercice 1.10. *Est-ce que* $\{a\} \in \{a, b, c\}$ *? Former la liste des parties de* $\{a, b, c\}$.

Exercice 1.11. *Montrer que* $\mathcal{P}(A) \subset \mathcal{P}(B)$ *quand* $A \subset B$.

Exercice 1.12. *Soit* $\mathbb{B} = \{0, 1\}$.

1. *A-t-on* $\mathbb{B} \in \mathbb{B}$ *?*
2. *Quels sont les éléments de* $\mathcal{P}(\mathbb{B})$ *?*
3. *Quels sont les éléments de* $\mathcal{P}(\mathcal{P}(\mathbb{B}))$ *?*

II Opérations sur les ensembles

II.1 Égalite de deux ensembles

DÉFINITION 1.3. *Deux ensembles sont* égaux *si et seulement si ils ont les mêmes éléments.* ♢

$A \subset B$ et $B \subset A \iff A = B$.

Exercice 1.13. *Dans chacun des cas suivants, déterminer si les ensembles sont égaux :*

1. $A = \{x \in \mathbb{R} | x > 0\}$ *et* $B = \{x \in \mathbb{R} | x \geqslant |x|\}$
2. $A = \{x \in \mathbb{R} | x > 0\}$ *et* $B = \{x \in \mathbb{R} | x \leqslant |x|\}$
3. $A = \mathbb{Z}$ *et* $B = \{x \in \mathbb{Z} | x^2 - x$ *pair* $\}$
4. $A = \{x \in \mathbb{N}_{10} \mid x$ *impair, non divisible par 3* $\}$ *et* $B = \{x \in \mathbb{N}_{10} \mid$ *24 divise* $x^2 - 1\}$

Réponse : Pour le 3, $x^2 - x = x(x - 1)$, et réfléchir sur la parité de ce produit.

II.2 Réunion, intersection

Réunion A et B sont deux ensembles, on considère la réunion de A et de B, notée $A \cup B$, l'ensemble des éléments qui sont éléments de A ou de B.

EXEMPLE 1.14. $A = \{1, 2, 3\}, B = \{1, 4, 5\},$ alors $A \cup B = \{1, 2, 3, 4, 5\}$

Exercice 1.15. *Faire la réunion des ensembles* $A = \{x \in \mathbb{R} | 0 \leqslant x \leqslant 3\}$, $B = \{x \in \mathbb{R} | -2 < x \leqslant 1\}$.

PROPRIÉTÉ 1.4 (PROPRIÉTÉS DE LA RÉUNION) : La réunion de deux ensembles possède certaines propriétés :

- idempotence : $A \cup A = A$
- commutativité : $A \cup B = B \cup A$
- associativité : $A \cup (B \cup C) = (A \cup B) \cup C$
- élément neutre : $A \cup \varnothing = A$

Exercice 1.16. *Donner des exemples d'opérateurs idempotents, commutatifs, associatifs, et possédant un élément neutre, par exemple en arithmétique, ou en analyse.*

Intersection L'intersection de deux ensembles A et B est l'ensemble, noté $A \cap B$ des éléments communs à A et à B.

Exercice 1.17. *Dans chacun des cas suivants, faire l'intersection des ensembles A et B.*

1. *A = l'ensemble des rectangles, et B = l'ensemble des losanges.*
2. $A = \{x \in \mathbb{R} | 0 \leqslant x \leqslant 3\}$, $B = \{x \in \mathbb{R} | -2 < x \leqslant 1\}$

PROPRIÉTÉ 1.5 (PROPRIÉTÉS DE L'INTERSECTION) : L'intersection de deux ensembles possède certaines propriétés :
- idempotence : $A \cap A = A$
- commutativité : $A \cap B = B \cap A$
- associativité : $A \cap (B \cap C) = (A \cap B) \cap C$
- élément neutre : si l'on se place dans un ensemble E et que A est une partie de E, alors E est élément neutre pour l'intersection : $A \cap E = A$

Propriétés mutuelles de ces deux opérations Ces deux opérations ont des propriétés symétriques...

PROPRIÉTÉ 1.6 (DISTRIBUTIVITÉS DE $\cup$ ET $\cap$) : On a les distributivités :
- de $\cup$ sur $\cap$: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- de $\cap$ sur $\cup$: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Exercice 1.18. *On se donne trois ensembles A, B, C tels que $A \cap B \cap C = \varnothing$. Sont-ils nécessairement disjoints deux à deux ? Donner des exemples.*

II.3 Complémentation

DÉFINITION 1.4 (COMPLÉMENTATION). *Pour $A \subset E$, on définit le* complémentaire *de A par rapport à E comme l'ensemble des éléments de E qui ne sont pas éléments de A.* ♢

NOTATION : Il existe plusieurs manières de noter le complémentaire de A dans E : $E \setminus A$ (« E moins A »), $\bar{A}$, ou encore $\mathcal{C}_E A$.

REMARQUE 1.3. Il faut donc se placer, pour la définition de la complémentation, dans $\mathcal{P}(E)$ (où E est un ensemble fixé) : *la complémentation se définit par rapport à un ensemble.*

PROPRIÉTÉ 1.7 : La complémentation a plusieurs propriétés remarquables :
- involution : $\bar{\bar{A}} = A$,
- loi de De Morgan : $A \bar{\cup} B = \bar{A} \cap \bar{B}$, et $A \bar{\cap} B = \bar{A} \cup \bar{B}$.

Exercice 1.19. *Connaissez-vous d'autres opérations involutives ?*

Exercice 1.20. *Illustrez, à l'aide d'un diagramme de Venn, les lois de De Morgan.*

Exercice 1.21. *Faire la réunion des ensembles A et B, quand $A = \{x \in \mathbb{N} | x$ impair $\}$, et $B = \{x \in \mathbb{N} | x$ pas divisible par 3 $\}$.*

Réponse : Rechercher à quoi correspond le complémentaire de la réunion de A et B.

II.4 Produit cartésien

Le produit cartésien des ensembles A et B (dans cet ordre) est l'ensemble, que l'on note $A \times B$ (« A croix B ») des couples ordonnés (a, b) où $a \in A$ et $b \in B$.

Dans le couple (a, b),
– (a, b) n'est pas un ensemble et
– (a, b) est distinct de (b, a).

Exercice 1.22. *Représenter graphiquement la réunion des ensembles $A = \{(x, y) \in \mathbb{R}^2 | x + y \leqslant 2\}$, et $B = \{(x, y) \in \mathbb{R}^2 | 2 < 3x - y\}$.*

Exercice 1.23. *Représenter graphiquement l'intersection des ensembles $A = \{(x, y) \in \mathbb{R}^2 | x + y \leqslant 2\}$, et $B = \{(x, y) \in \mathbb{R}^2 | 2 < 3x - y\}$.*

III Exercices

III.1 Ensembles de base

Exercice 1.24. *Rappelez la définition et la notation des ensembles fondamentaux suivants : entiers naturels, entiers relatifs, nombres rationnels, décimaux, réels et complexes.*

Exercice 1.25. *Connaissez-vous d'autres ensembles de nombres ? Quelle en est la définition ?*

Exercice 1.26. *Réalisez un diagramme de Venn des ensembles des deux précédents exercices.*

III.2 La différence symétrique

DÉFINITION 1.5. *Pour deux ensembles A et B, on appelle différence symétrique, note $A\Delta B$, l'ensemble défini par*

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

c'est-à-dire que $A\Delta B$ est constitué des éléments qui appartiennent soit à A, soit à B, mais pas aux deux. ♢

Exercice 1.27. *Soit $A = \{1, 2, 3, 4, 5, 6\}$, $B = \{1, 3, 5, 7, 9\}$, $C = \{4, 5, 6, 7, 8, 9\}$ et $D = \{2, 3, 5, 7, 8\}$. Trouver $A\Delta B$, $C\Delta B$, $A \cap (B\Delta D)$, $B\Delta C$, $A\Delta D$ et $(A \cap B)\Delta(A \cap D)$.*

Exercice 1.28. *Montrez que $A\Delta B = [A \cap (E \setminus B)] \cup [(E \setminus A) \cap B]$*

Exercice 1.29. *Calculer $A\Delta A$, $A\Delta(E \setminus A)$, $A\Delta E$ et $E \setminus (A\triangle B)$.*

Exercice 1.30. *Montrer que, si $A\triangle B = C$, alors $A\triangle C = B$ et $B\triangle C = A$.*

Exercice 1.31. *Montrer que la différence ensembliste est commutative, et possède un élément neutre.*

Exercice 1.32. *Montrer que si $A\Delta B = A\Delta C$ alors $B = C$.*

III.3 Généraux

Exercice 1.33. *Soit E un ensemble.*

Démontrer que, quelles que soient les parties A, B, X, Y de E, l'implication suivante est vraie :

$$(X \cap A = X \cap B) \text{ et } Y \subset X \Rightarrow Y \cap A = Y \cap B.$$

Exercice 1.34. *On se place dans l'ensemble $\mathcal{P}(E)$ des parties d'un ensemble non vide E ; A, B et C sont des parties de E.*

Montrer que $(A \cup C) \subset (A \cup B)$ et $(A \cap C) \subset (A \cap B)$ implique que $C \subset B$.

Exercice 1.35. *Soit E un ensemble non vide et $\mathcal{P}(E)$ l'ensemble de ses parties.*

Soit f une application croissante, pour l'inclusion, de $\mathcal{P}(E)$ dans lui-même (c'est-à-dire : si X et Y sont deux parties de E et si $X \subset Y$, alors $f(X) \subset f(Y)$).

1. *Montrer que, pour tout couple (X, Y) de parties de E, on a : $f(X) \cup f(Y) \subset f(X \cup Y)$*
2. *On dit qu'une partie X de E est régulière si et seulement si $f(X) \subset X$. Montrer qu'il existe au moins une partie régulière dans E et que, si X est régulière, il en est de même de $f(X)$.*
3. *Soit A l'intersection de toutes les parties régulières de E. Montrer que A est régulière et que $f(A) = A$.*

Exercice 1.36. *Soit E un ensemble et A, B, C des parties de E.*

Démontrer la proposition suivante :

$$[A \subset (B \cap C)] \text{ et } [(B \cup C) \subset A] \Rightarrow [A = B = C].$$

Exercice 1.37. *Sous les mêmes hypothèses, montrer que $A \cap (A \cup B) = A \cup (A \cap B) = A$*

Exercice 1.38. *Montrer que $(A \cup B) \cap (B \cup C) \cap (C \cup A) = (A \cap B) \cup (B \cap C) \cup (C \cap A)$.*

Exercice 1.39 (Archives). *Le jour où il ne faut pas, vous découvrez que*

- *vous avez besoin d'un fichier client C et du fichier prospects P qui contenait la liste des clients prospects, c.à.d. des clients actuels ou potentiels visités par les représentants au dernier semestre ;*
- *Le stagiaire les a effacé par mégarde, en répondant au hasard à une question du système qu'il ne comprenait pas.*

Au cours d'une réunion de crise, vous apprenez cependant qu'il reste

- *le fichier F des clients non prospects de ce dernier trimestre ;*
- *le fichier G des prospects du dernier trimestre non encore client ;*
- *le fichier H des clients et/ou propspects mélangés sans distinction.*

En déduire comment reconstruire P et C.

Exercice 1.40. *Soit les affirmations :*

- *J'ai planté tous mes arbres onéreux l'an passé.*
- *Tous mes arbres fruitiers sont dans mon verger.*
- *Aucun des arbres fruitiers n'a été planté l'an passé.*
- *J'ai un orme, qui est un arbre onéreux, mais pas dans mon verger.*

Dire si les affirmations suivantes sont justes ou fausses ou impossibles à répondre.

1. *Aucun de mes arbres fruitiers n'est onéreux.*
2. *Tous mes arbres plantés l'an passé l'ont été dans le verger.*
3. *J'ai planté au moins un arbre l'an passé.*

Exercice 1.41 (Fonction caractéristique des parties d'un ensemble). *On appelle fonction caractéristique de la partie A de l'ensemble E ($E \neq \emptyset$, $A \neq \emptyset$, $A \subset E$) l'application $f_A : E \Rightarrow \{0, 1\}$, définie par*

– $\forall x \in A,\ f_A(x) = 1$,
– $\forall x \in E \setminus A,\ f_A(x) = 0$.

On pose de plus $\forall x \in E, f_\emptyset(x) = 0$ *et* $f_E(x) = 1$.

Étudier les fonctions caractéristiques d'une réunion, d'une intersection de deux parties, ainsi que celle du complémentaire d'une partie.

Exercice 1.42. *On définit, dans* $\{0,1\}$*, trois lois de composition, de manière que,* $\forall x \in E$*, on ait* $f_A(x) \cdot f_B(x) = f_{A\cap B}(x)$*,* $f_A(x) + f_B(x) = f_{A\cup B}(x)$ *et* $\overline{f_A(x)} = f_{E\setminus A}(x)$*.*

Montrer que $(\{0,1\}, +, \cdot, \overline{\ })$ *est une algèbre de Boole binaire.*

Fin du Chapitre

Séance du 16 Janvier 2019 ESTI L1

Relation binaire
Relation d'ordre
Relation d'équivalence

Chapitre 2

Relations binaires entre ensembles

I Relations

On se donne deux ensembles E et F.

DÉFINITION 2.1 (RELATION BINAIRE, GRAPHE). *On dit que :*
- *l'on a défini une* relation binaire $\mathcal{R}$ *entre ces deux ensembles lorsque l'on s'est donné une partie G de l'ensemble produit $E \times F$ ($G \subset E \times F$).*
- *Cette partie est appelée* graphe *de la relation binaire.*
- *Si x dans E et y dans F sont tels que $(x, y) \in G$, on dit que x est* en rela tion avec y *par la relation $\mathcal{R}$ et on note $x\mathcal{R}y$* ♢

Exercice 2.1. *On se place dans l'ensemble $E = \{1, 2, 3, \ldots, 20\}$. Représenter, dans le plan rapporté à deux axes de coordonnées rectangulaires, les graphes des relations binaires sur E dont les définitions suivent :*
- $x\mathcal{R}y \iff x \leqslant y$.
- $x\mathcal{R}y \iff x|y$ *: x divise y.*
- $x\mathcal{R}y \iff x \equiv y[3]$ *: x est congru à y modulo 3.*
- $x\mathcal{R}y \iff y = x^2$.

REMARQUE 2.1. Lorsque $E = F$, on parle de relation binaire définie dans l'ensemble E. Son graphe est une partie de E^2.

REMARQUE 2.2. Il est possible que $x\mathcal{R}y$ sans que $y\mathcal{R}x$.

Exercice 2.2. *A-t-on $y\mathcal{R}x$ quand $x\mathcal{R}y$, dans les relations binaires définies dans l'exercice précédent ?*

Exercice 2.3. *Sur l'ensemble des mots de la langue française, on définit la relation : « le mot M est lié au mot N s'ils coïncident quand on écrit M à l'envers ». Déterminer quelques couples de mots en relation, ainsi que des mots en relation avec eux-mêmes. Comment appelle-t-on de tels mots ?*

Exercice 2.4. *Sur l'ensemble $\mathbb{Z}$ des entiers relatifs, on définit deux relations, notées respectivement Σ et Δ, de la façon suivante :*
- *$x\Sigma y$ quand la somme $x + y$ est paire*
- *$x\Delta y$ quand la différence $x - y$ est paire*

Sont-elles égales ?

II Relations d'ordre

Dans ce paragraphe, on se place dans le cas où $E = F$. Soit $\mathcal{R}$ une relation binaire définie dans un ensemble E, de graphe G.

II.1 Réflexivité, antisymétrie, transitivité

DÉFINITION 2.2 (RÉFLEXIVITÉ). *$\mathcal{R}$ est dite* réflexive *quand tout élément de E est en relation avec lui-même : $\forall x \in E,\ x\mathcal{R}x$.* ♢

DÉFINITION 2.3 (ANTISYMÉTRIE). *$\mathcal{R}$ est dite* antisymétrique *si, lorsque x est en relation avec y, alors y ne peut pas être en relation avec x (sauf si $x = y$) : $\forall (x, y) \in E^2,\ x\mathcal{R}y$ et $y\mathcal{R}x \Rightarrow\ x = y$* ♢

DÉFINITION 2.4 (TRANSITIVITÉ). *$\mathcal{R}$ est dite* transitive *lorsque, si x est en relation avec y, et si y l'est avec z, alors x est en relation avec z : $\forall x \in E,\ \forall y \in E, \forall z \in E,\ x\mathcal{R}y$ et $y\mathcal{R}z \Rightarrow x\mathcal{R}z$.* ♢

Exercice 2.5. *Les relations suivantes sont-elles réflexives, antisymétriques ou transitives ?*

1. *$A = \mathbb{R}$ et $x\mathcal{R}y$ si $|x| = |y|$.*
2. *$A = \mathbb{R}$ et $x\mathcal{R}y$ si $sin^2 x + \cos^2 y = 1$.*
3. *$A = \mathbb{N}$ et $x\mathcal{R}y$ s'il existe p et q entiers tels que $y = px^q$.*
4. *A est l'ensemble des points du plan, et $x\mathcal{R}y$ si la distance de x à y est inférieure à 52,7 km.*

Exercice 2.6. *Sur $\mathbb{N}^*$ on définit la relation $a\mathcal{R}b$ si et seulement si $a^b \leq b^a$.*

1. *Vérifier que cette relation est réflexive et transitive.*
2. *Comparer 2 et 4. La relation est-elle antisymétrique ?*

Exercice 2.7. *Soit $\mathcal{R}$ et $\mathcal{S}$ deux relations dans A.*

1. *Montrer que si $\mathcal{R}$ et $\mathcal{S}$ sont transitives alors $\mathcal{R} \cap \mathcal{S}$ est transitive.*
2. *Si $\mathcal{R}$ est antisymétrique alors $\mathcal{R} \cap \mathcal{S}$ est antisymétrique.*

II.2 Relation d'ordre

DÉFINITION 2.5 (RELATION D'ORDRE). *$\mathcal{R}$ est une* relation d'ordre *lorsqu'elle est réflexive, antisymétrique et transitive.* ♢

EXEMPLE 2.8 (EXEMPLES DE RELATIONS D'ORDRE). Quelques relations d'ordre :
- $(\mathbb{R}, \leqslant)$
- $(\mathcal{P}(E), \subset)$

EXEMPLE 2.9 (RELATION DE DIVISIBILITÉ). On note $a|b$ si et seulement si b est un multiple de a ($\exists k \in \mathbb{N}^*, b = ka$). C'est une relation d'ordre définie dans $\mathbb{N}^*$. En effet, elle est

reflexive : $a = 1a$, donc $a|a$ est vrai,

antisymétrique : si $a|b$ et $b|a$, alors $\exists k, k' \in \mathbb{N}^*, a = kb$ et $b = k'a$. Donc $a = kk'a$. Comme $a \neq 0$, $kk' = 1$. Mais $k, k' \in \mathbb{N}^*$, donc $k = k' = 1$, et $a = b$.

transitive : si $a|b$ et $b|c$, alors alors $\exists k, k' \in \mathbb{N}^*, a = kb$ et $b = k'c$. Donc $a = kk'c$: il existe $k'' \in \mathbb{N}^*$ ($k'' = kk'$) tel que $a = k''c$: $a|c$.

La structure algébrique constituée par l'ensemble E, muni de la relation d'ordre $\mathcal{R}$, (c'est-à-dire : le couple $(E, \mathcal{R})$) est celle d'*ensemble ordonné*.

II.3 Ordre partiel, ordre total

Une relation d'ordre définie dans un ensemble E peut posséder une propriété supplémentaire, celle selon laquelle tous les éléments de E sont comparables entre eux. On formalise comme suit :

DÉFINITION 2.6 (RELATION D'ORDRE TOTALE). *Une relation d'ordre qui possède cette dernière propriété est dite relation d'*ordre total*, et la structure algébrique correspondante est celle d'*ensemble totalement ordonné*.* ♢

REMARQUE 2.3. Cette propriété est aussi équivalente à :

$$\forall x \in E,\ \forall y \in E,\ x\mathcal{R}y \text{ ou } y\mathcal{R}x$$

ou encore : « si x n'est pas en relation avec y, alors y est en relation avec x ».

DÉFINITION 2.7 (RELATION D'ORDRE PARTIEL). *Dans le cas contraire, il existe des éléments qui ne sont pas comparables : on parle alors d'*ordre partiel *.* ♢

EXEMPLE 2.10. $\leqslant$ est une relation d'ordre totale dans $\mathbb{R}$.

Exercice 2.11. *On définit une relation binaire $\mathcal{R}$ sur $\mathbb{R} \times \mathbb{R}^+$ par $(x, y)\mathcal{R}(x', y')$ ssi $x^2 + y^2 < x'^2 + y'^2$ ou ($x^2 + y^2 = x'^2 + y'^2$ et $x \leq x'$).*

Montrer qu'il s'agit d'une relation d'ordre totale

II.4 Éléments maximaux

Soit $(E, \mathcal{R})$ un ensemble ordonné et A une partie de E. Quelques définitions...

DÉFINITION 2.8 (MAJORANT). *On appelle* majorant *de A tout élément M de E tel que, quel que soit $a \in A$, $a\mathcal{R}M$.* ♢

DÉFINITION 2.9 (PARTIE MAJORÉE). *La partie A de E est dite* majorée *s'il existe un majorant de A.*♢

DÉFINITION 2.10 (MINORANT). *On appelle* minorant *de A tout élément m de E tel que, quel que soit $a \in A$, $m\mathcal{R}a$.*

On parle aussi de partie minorée. ♢

DÉFINITION 2.11 (ÉLÉMENT MAXIMUM). *On appelle* élément maximum *de A un élément de A qui est majorant de A.* ♢

Exercice 2.12. *Trouvez des exemples d'élément maximum sur $\mathbb{N}$ et $\mathbb{R}$.*

NOTATION : $\operatorname{Max} A$.

REMARQUE 2.4. Si A est non majorée, il est exclu qu'elle admette un élément maximum. Cet élément maximum n'existe pas toujours, même pour une partie majorée. Ainsi, l'intervalle réel]2,3[est majoré, mais n'a pas d'élément maximum. Cependant, s'il existe, cet élément est unique.

DÉFINITION 2.12 (ÉLÉMENT MINIMUM). *On appelle* élément minimum *de A un élément de A qui est minorant de A.* ♢

NOTATION : $\operatorname{Min} A$.

Exercice 2.13. *Etant donné $B = \{1, 2, 3, 4, 5\}$ ordonné selon la relation $4 < 2$, $5 < 2$, $5 < 3$, $2 < 1$, $3 < 1$. Trouver $\operatorname{Min} A$ et $\operatorname{Max} A$.*

Exercice 2.14 (Relations d'ordre en Algèbre de Boole). *Soit $\mathcal{A}$ une algèbre de Boole.*
On considère la relation binaire, de symbole $<$, définie par

$$a < b \Leftrightarrow a + b = b.$$

1. *Montrer qu'il s'agit d'une relation d'ordre.*
2. *Montrer que $a < b \Leftrightarrow a \cdot b = a$.*
3. *Montrer que, $\forall (a, b, c) \in \mathcal{A}^3,\ b \cdot c < a \cdot b + \overline{a} \cdot c$.*
4. *On définit la relation binaire $\subset$ par : $a \subset b$ si et seulement si $a \cdot \overline{b} = 0$; montrer que c'est une relation d'ordre.*
5. *Comparer $<$ et $\subset$.*
6. *En utilisant l'une ou l'autre des définitions ci-dessus pour la relation d'ordre, trouver $\operatorname{Max} \mathcal{A}$ et $\operatorname{Min} \mathcal{A}$.*

Exercice 2.15 (Diagrammes de transitivité). *On considère...*

1. *$E = \{1\,,\,2\,,\,3\,,\,4\,,\,5\,,\,6\,,\,7\,,\,8\,,\,9\}$ et on définit la relation binaire $\mathcal{R}$ dans E par son graphe $G = \{$ (1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (1,8), (1,9), (2,2), (2,3), (2,4), (2,6), (2,8), (2,9), (3,3), (4,3), (4,4), (4,6), (4,8), (4,9), (5,3), (5,4), (5,5), (5,6), (5,7), (5,8), (5,9), (6,6), (6,8), (6,9), (7,7), (7,8), (7,9), (8,8), (9,9) $\}$ (c'est-à-dire : $1\mathcal{R}1$, etc...). Montrer que cette relation est une relation d'ordre. E est-il totalement ordonné par cette relation ?*
2. *Mêmes questions pour $E' = \{1\,,\,2\,,\,3\,,\,4\,,\,5\,,\,6\}$ et $G' = \{$ (1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,2), (2,4), (2,5), (2,6), (3,3), (3,4), (3,6), (4,4), (4,6), (5,5), (5,6), (6,6)$\}$.*

III Relations d'équivalence

On se place encore dans ce paragraphe dans le cas où $E = F$. Soit $\mathcal{R}$ une relation binaire définie dans un ensemble (non vide) E, de graphe G.

DÉFINITION 2.13 (RELATION SYMÉTRIQUE). *$\mathcal{R}$ est dite* symétrique *si, dès que x est en relation avec y, alors y est en relation avec x*

$$\forall x \in E,\ \forall y \in E, (x, y) \in G \Rightarrow (y, x) \in G$$

REMARQUE 2.5. Ou encore : $\forall x \in E, \forall y \in E,\ x\mathcal{R}y \Rightarrow y\mathcal{R}x$.

Exercice 2.16. *Est-ce qu'une relation sur un ensemble A dont le graphe est constitué uniquement de couples (x, x) est symétrique ? transitive ?*

DÉFINITION 2.14 (RELATION D'ÉQUIVALENCE). *$\mathcal{R}$ est une relation d'équivalence lorsqu'elle est réflexive, symétrique et transitive.* ♢

EXEMPLE 2.17. L'égalité est une relation d'équivalence.

EXEMPLE 2.18 (RELATION DE CONGRUENCE MODULO n DANS $\mathbb{Z}$). Par définition :

$$x \equiv y\ [n] \text{(lire : « } x \text{ est congru à } y \text{ modulo } n \text{ »)} \Leftrightarrow \exists k \in \mathbb{Z},\ x - y = k \cdot n$$

.

- réflexivité : $x \equiv x[n]$: en effet, $x - x = 0 \cdot n$, et $0 \in \mathbb{Z}$.
- symétrie : si $x \equiv y\ [n]$, $\exists k \in \mathbb{Z}$, $x - y = k \cdot n$; alors $y - x = (-k) \cdot n$; or, si $k \in \mathbb{Z}$, $(-k) \in \mathbb{Z}$, donc $y \equiv x\ [n]$.
- transitivité : si $x \equiv y\ [n]$ et $y \equiv z\ [n]$, $\exists k \in \mathbb{Z}$, $x - y = k \cdot n$ et $\exists l \in \mathbb{Z}$, $y - z = l \cdot n$. En aditionnant membre à membre ces deux égalités, on obtient $x - z = (k + l) \cdot n$, or $(k, l) \in \mathbb{Z}^2$, donc $k + l \in \mathbb{Z}$, donc $x \equiv z\ [n]$.

C'est bien une relation d'équivalence.

Exercice 2.19. *Sur $\mathbb{Z}$, on écrit « $x\mathcal{R}y$ quand $x + y$ est pair. » Montrez que $\mathcal{R}$ est une relation d'équivalence.*

Exercice 2.20. *Sur $\mathbb{R}$, on définit la relation « $x\mathcal{R}y$ quand $\cos(2x) = \cos(2y)$ ». Montrez que $\mathcal{R}$ est une relation d'équivalence.*

III.1 Classes d'équivalence

DÉFINITION 2.15 (CLASSE D'ÉQUIVALENCE). *Soit x un élément de E, et $\mathcal{R}$ une relation d'équivalence sur E. On appelle* classe d'équivalence *de cet élément l'ensemble des éléments de E qui sont en relation avec x (on dit encore : « qui sont équivalents à x »).* ♢

NOTATION : On note $\dot{x}$ la classe de l'élément x : $\dot{x} = \{y \in E \mid y\mathcal{R}x\}$.

Exercice 2.21. *Dans $\mathbb{R}$, on considère la relation binaire $\mathcal{R}$ définie par : $x\mathcal{R}y$ ssi $x^2 - y^2 = x - y$.*

1. *Vérifier que $\mathcal{R}$ est une relation d'équivalence.*
2. *Pour tout réel x, déterminer $\dot{x}$.*

Exercice 2.22. *Dans $\mathbb{R}$, on considère la relation binaire $\mathcal{R}$ définie par : $x\mathcal{R}y$ ssi $x.e^y = y.e^x$.*

1. *Vérifier que $\mathcal{R}$ est une relation d'équivalence.*
2. *Pour tout réel x, déterminer le nombre d'éléments de $\dot{x}$.*

PROPRIÉTÉ 2.1 : L'intersection de deux classes d'équivalence distinctes est vide.

REMARQUE 2.6. On dit aussi que les classes sont deux à deux disjointes.

PREUVE 1 : *On considère deux classes, $\dot{x}$ et $\dot{y}$, soit $z \in \dot{x} \cap \dot{y}$; $\forall t \in \dot{x}$, on a $(t, x) \in G$; mais $z \in \dot{x}$, donc $(z, x) \in G$, donc (symétrie) $(x, z) \in G$, donc (transitivité) $(t, z) \in G$; mais $z \in \dot{y}$, donc $(z, y) \in G$, donc (transitivité) $(t, y) \in G$, donc (finalement) $t \in \dot{y}$, et donc $\dot{x} \subset \dot{y}$; raisonnement analogue pour tout $t \in \dot{y}$, qui aboutit à $\dot{y} \subset \dot{x}$, et enfin (par double inclusion) $\dot{x} = \dot{y}$; si deux classes ont un élément commun, elles sont confondues ; donc deux classes distinctes sont disjointes).* †

DÉFINITION 2.16 (PARTITION D'UN ENSEMBLE). *Une* partition *d'un ensemble E est une famille de sous-ensembles de E, 2 à 2 disjoints, et dont la réunion est égale à E.* ♢

PROPRIÉTÉ 2.2 : Les classes d'équivalence réalisent une partition de E.

PREUVE 2 : *Comme les classes sont des parties de E, leur réunion est une partie de E. Réciproquement, tout élément de E appartient à une classe (« tout élément est classé »). Donc E est une partie de la réunion des classes ; et E est égal à la réunion des classes.* †

EXEMPLE 2.23. On reprend la congruence modulo n, par exemple pour $n = 4$.On a :
$\dot{0} = \{\ldots, -12, -8, -4, 0, 4, 8, 12, \ldots\}$
$\dot{1} = \{\ldots, -11, -7, -3, 1, 5, 9, 13, \ldots\}$
$\dot{2} = \{\ldots, -10, -6, -2, 2, 6, 10, 14, \ldots\}$
$\dot{3} = \{\ldots, -9, -5, -1, 3, 7, 11, 15, \ldots\}$

Exercice 2.24. *Soit $\mathcal{R}$ la relation d'équivalence suivante dans l'ensemble $A = \{1, 2, 3, 4, 5, 6\}$:*

$$\mathcal{R} = \{(1,1), (1,5), (2,2), (2,3), (2,6), (3,2), (3,3), (3,6), (4,4), (5,1), (5,5), (6,2), (6,3), (6,6)\}$$

Trouver la partition de A induite par $\mathcal{R}$, c'est-à-dire trouver les classes d'équivalence de $\mathcal{R}$.

Exercice 2.25 (Une relation d'équivalence). *On considère l'ensemble des points du plan rapporté à deux axes de coordonnées rectangulaires et deux points P_1 et P_2 de coordonnées respectives (x_1, y_1) et (x_2, y_2) ; on définit dans cet ensemble la relation binaire $\mathcal{R}$ par :*

$$P_1 \mathcal{R} P_2 \Leftrightarrow x_1 y_1 = x_2 y_2$$

- *S'agit-il d'une relation d'équivalence ? Si oui, étudier les classes d'équivalence.*
- *Mêmes questions pour la relation $\mathcal{R}'$, définie par*

$$P_1 \mathcal{R}' P_2 \Leftrightarrow x_1 y_1 = x_2 y_2 \text{ et } x_1 x_2 \geqslant 0$$

Exercice 2.26. *Définir, par leurs graphes, les relations d'équivalence dans E qui comportent respectivement le moins et le plus possible de points. Que peut-on dire de ces relations ?*

III.2 Ensemble-quotient

DÉFINITION 2.17 (ENSEMBLE-QUOTIENT). *Il s'agit de l'ensemble des classes d'équivalence de tous les éléments de E.* ♢

NOTATION : $E/\mathcal{R}$.

Pour parler aisément d'une classe, on choisit un de ses éléments, et cet élément, surmonté d'un point, sert à représenter la classe en question. Une fois que ce choix est fait, il est définitif, et il n'est plus question d'évoquer les autres éléments de cette classe, il faut se tenir, sous peine d'incohérence, au choix qui a été fait.

EXEMPLE 2.27 (CONGRUENCE MODULO 4). On choisit pour représentants les entiers < 4, donc 0, 1, 2 et 3. L'ensemble-quotient est $\mathbb{Z}/4\mathbb{Z} = \{\dot{0}, \dot{1}, \dot{2}, \dot{3}\}$.

IV Compatibilité entre une opération et une relation binaire

DÉFINITION 2.18. *La relation binaire (dans E) de symbole $\mathcal{R}$ est dite* compatible avec l'opération *(définie dans E) de symbole $\circ$ lorsque, quels que soient les éléments x, x', y et y' de E : si $x\mathcal{R}x'$ et si $y\mathcal{R}y'$, alors $(x \circ y)\mathcal{R}(x' \circ y')$* ♢

Autrement dit, l'opération conserve la relation.

EXEMPLE 2.28. On considère la relation classique d'inégalité dans $\mathbb{R}$: si on a $x \leqslant x'$ et $y \leqslant y'$, on peut écrire $x + x' \leqslant y + y'$.

Ce résultat est bien connu : on a le droit « d'additionner des inégalités membre à membre ». En d'autres termes, l'addition des réels est compatible avec l'inégalité.

Mais, de $-2 \leqslant 1$ et de $-3 \leqslant -1$, on ne peut pas déduire que $6 \leqslant -1\ldots$ On n'a pas le droit de « multiplier des inégalités membre à membre ».

La multiplication des réels, quant a elle, n'est donc pas compatible avec l'inégalité.

Exercice 2.29 (Congruences modulo n). *Montrer que la relation de congruence modulo n dans $\mathbb{Z}$ définie en cours est compatible avec addition et multiplication.*

Établir les tables des opérations que l'on peut alors définir dans les ensembles $\mathbb{Z}/5\mathbb{Z}$, $\mathbb{Z}/6\mathbb{Z}$.

Lorsqu'une relation d'équivalence est compatible avec une opération, on peut définir dans l'ensemble-quotient une opération, dite *induite* de celle qui existe dans l'ensemble d'origine.

Fin du Chapitre

Chapitre 3

Application d'un ensemble dans un autre

I Application et relation fonctionnelle

DÉFINITION 3.1 (APPLICATION). *Une* application *de l'ensemble E dans l'ensemble F est une relation binaire particulière $\mathcal{R}$ entre E et F, dont le graphe G doit posséder les propriétés suivantes :*
- *pour tout élément x de E, il doit exister un élément y de F tel que (x, y) soit élément de G ;*
- *cet élément y doit être unique.* ♢

Voici la formalisation (partielle) de ces propositions :
- $\forall x \in E,\ \exists y \in F,\ (x, y) \in G$
- $\forall x \in E,\ \forall y \in F,\ \forall y' \in F, [(x, y) \in G \text{ et } (x, y') \in G \Longrightarrow y = y']$.

Il existe un intermédiaire entre relation et application ...

DÉFINITION 3.2 (RELATION FONCTIONNELLE). *On parle de* relation fonctionnelle *quand tout élément de l'ensemble de départ possède au plus une image.* ♢

REMARQUE 3.1. Une application est donc une relation fonctionnelle particulière : tout élément de l'ensemble de départ possède exactement une image.

Exercice 3.1. *Parmi les relations suivantes de $\mathbb{R}$ vers $\mathbb{R}$, repérez les relations fonctionnelles, repérez les applications :*
1. $\mathcal{R} = \{(x, y) \in \mathbb{R}^2, |y| = \sqrt{x}\}$
2. $\mathcal{R} = \{(x, y) \in \mathbb{R}^2, xy = 1\}$
3. $\mathcal{R} = \{(x, y) \in \mathbb{R}^2, y - x + 2 = 0\}$

II Image et antécédent d'un élément

On suppose dorénavant que $\mathcal{R}$ est une application. Pour un x donné de E, il lui correspond un et un seul y de F qui est en relation avec lui par $\mathcal{R}$.

DÉFINITION 3.3. *Cet unique y est alors appelé* image *de x par l'application définie par $\mathcal{R}$.* ♢

NOTATION : Si l'on désigne par f cette application, l'expression « y est l'image de x par f » est formalisée par $y = f(x)$. De plus, on formalise la proposition « f est une application de E dans F » par $f : E \to F$. La proposition « y est l'image de x par f » peut aussi être traduite par : $f : x \longmapsto y$.

Exercice 3.2. *Interpréter chacune des situations suivantes au moyen d'une application. Pour cela, on définira deux ensembles A et B ainsi que $f : A \to B$. Préciser dans chaque cas pourquoi il s'agit bien d'une application.*

1. *Le registre d'un hôtel qui possède 55 chambres.*
2. *Le numéro d'INSEE.*
3. *La parité d'un entier naturel.*

Réciproquement, soit f une application...

DÉFINITION 3.4 (ANTÉCÉDENT). *Si y est l'image de x par f, alors x est appelé un* antécédent *de y par f.* ♢

Un quelconque élément y de F ne possède pas forcément d'antécédant par une application $f : E \longrightarrow F$. Et il n'y a pas forcément unicité quand il en possède : un y de F peut avoir plusieurs antécédants par une application f.

Les cas particuliers où tout y de F possède au plus un antécédant, et au moins un antécédant, sont étudiés dans les deux sections suivantes.

III Applications injectives

DÉFINITION 3.5 (INJECTIVITÉ). *L'application $f : E \longrightarrow F$ est dite* injective *quand tout $y \in F$ possède au plus un antécédant par f.* ♢

REMARQUE 3.2. Le terme injection est synonyme d'« application injective ».

L'injectivité d'une application peut se caractériser de la manière suivante.

PROPRIÉTÉ 3.1 (CARACTÉRISATION DES FONCTIONS INJECTIVES) :

« $f : E \longrightarrow F$ est une application injective » $\iff [\forall x, x' \in E, f(x) = f(x') \Rightarrow x = x']$

Exercice 3.3. *Prouver, en utilisant la caractérisation ci-dessus, que les applications suivantes sont injectives :*

1. $f : \mathbb{R} \longrightarrow \mathbb{R}, x \longmapsto 2x + 3$.
2. $f : \mathbb{R}^+ \longrightarrow \mathbb{R}, x \longmapsto 3x^2 + 1$.

Exercice 3.4. *Prouver, en utilisant la caractérisation ci-dessus, que les applications suivantes ne sont pas injectives :*

1. $f : \mathbb{R} \longrightarrow \mathbb{R}, x \longmapsto |x|$.
2. $f : \mathbb{R} \longrightarrow \mathbb{R}, x \longmapsto x^2$.

Exercice 3.5. *Tracez le graphe d'une application qui est injective, et d'une application qui ne l'est pas. Trouver une caractérisation de l'injectivité d'une application $f : E \longrightarrow F$, à partir du nombre d'intersections entre la courbe $\mathcal{C}_f$ et les droites verticales $y = b, b \in F$.*

Exercice 3.6. *Soit $f : E \longrightarrow F$ une application injective. Peut-elle perdre ce caractère injectif si on réduit l'ensemble d'arrivée ? Et si l'on réduit l'ensemble de départ ?*

Que se passe-t-il si l'on change «réduit» en «augmente» dans les précédentes questions ?

Exercice 3.7. *On suppose gof injective. Montrer que f est injective. Est-ce que g est obligatoirement injective ?*

IV Applications surjectives

La définition d'une application f exige seulement que chaque élément x de E admette une image (unique) y dans F, mais pas que tout élément y de F admette un antécédent dans E. S'il en est néanmoins ainsi, l'application est dite *surjective* :

DÉFINITION 3.6. *Une application* surjective $f : E \to F$ *est une application telle que tout* y *de* F *admette un antécédent dans* E. ♢

REMARQUE 3.3. *Surjection* est synonyme d'« application surjective ».

Exercice 3.8. *Tracez le graphe d'une application qui est surjective, et d'une application qui ne l'est pas.*

Exercice 3.9. *Donnez des exemples (sous forme analytique) de fonctions surjectives, et de fonction qui ne le sont pas.*

V Image d'un ensemble par une application

D'une manière générale, on peut considérer l'ensemble des images des éléments de E par une application f de E dans F (ils en ont tous une, et une seule).

DÉFINITION 3.7 (IMAGE D'UN ENSEMBLE PAR UNE APPLICATION). *Cet ensemble, qui est évidemment une partie de* F*, est noté* $f < E >$*, et est appelé* image de E par f *:*

$$f < E >= \{f(x) \in F \mid x \in E\}$$

REMARQUE 3.4. Si tous les éléments de F ont un antécédent dans E (f est surjective), cela signifie que tout élément de F est élément de $f < E >$, donc que $F \subset f < E >$. Comme on a remarqué par ailleurs que $f < E > \subset F$, on a, dans ce cas, $f < E >= F$.

Cette dernière remarque permet la formalisation suivante :

PROPRIÉTÉ 3.2 (CARACTÉRISATION DE LA SURJECTIVITÉ) :

« f est (une application) surjective » $\Leftrightarrow f < E >= F$

EXEMPLE 3.10. Soit l'application « élévation au carré » $f : x \longmapsto x^2$ de $\mathbb{R}$ dans $\mathbb{R}$. Elle est :
- non surjective : $f < \mathbb{R} >= \mathbb{R}^+$,
- non injective : $f(-2) = f(2) = 4$.

Exercice 3.11. *On suppose* gof *surjective. Montrer que* g *est surjective. Est-ce que* f *est obligatoirement surjective ?*

VI Applications bijectives

DÉFINITION 3.8 (APPLICATIONS BIJECTIVES). *Une application qui est à la fois injective et surjective est dite* bijective . ♢

REMARQUE 3.5. Synonyme d'« application bijective » : bijection.

Exercice 3.12. *Dans chaque cas, dire si l'application $f : A \to B$ est injective, surjective ou bijective.*

1. $A = \mathbb{R}, B = \mathbb{R}, f(x) = x + 7$
2. $A = \mathbb{R}, B = \mathbb{R}, f(x) = x^2 + 2x - 3$
3. $A = \{x \in \mathbb{R} | 4 \leqslant x \leqslant 9\}, B = \{x \in \mathbb{R} | 21 \leqslant x \leqslant 96\}, f(x) = x^2 + 2x - 3$
4. $A = \mathbb{R}, B = \mathbb{R}, f(x) = 3x - 2|x|$
5. $A = \mathbb{R}, B = \mathbb{R}, f(x) = e^x + 1$
6. $A = \mathbb{N}, B = \mathbb{N}, f(x) = x(x + 1)$

PROPRIÉTÉ 3.3 : Dans le cas d'une bijection, à chaque élément x de E correspond un et un seul élément y de F (définition d'une application) et, réciproquement, à chaque (surjectivité) élément y de F correspond un et un seul (injectivité) élément x de E.

Cette dernière proposition est précisément l'affirmation de l'existence d'une application g de F dans E, telle que $x = g(y) \iff f(x) = y$.

DÉFINITION 3.9 (APPLICATION INVERSE). *Cette application est appelée* application inverse *de l'application f.* ♢

NOTATION : On la note f^{-1}

Exercice 3.13. *Reprendre l'exercice précédent, en trouvant l'application réciproque des applications bijectives.*

EXEMPLE 3.14. Soit l'application « multiplication par 2 » $f : x \longmapsto 2x$ de $\mathbb{R}$ dans $\mathbb{R}$. Elle est surjective et injective. Elle admet donc une application inverse : $f^{-1} : x \longmapsto \dfrac{x}{2}$.

Exercice 3.15. *Soit $f : \mathbb{Z} \to \mathbb{Z}$ définie par $f(n) = n + (-1)^n$.*

1. *Montrer que n et $f(n)$ sont toujours de parité différente.*
2. *Montrer que f est bijective.*
3. *Calculer $f(f(n))$. En déduire une expression de f^{-1} et résoudre l'équation $347 = n + (-1)^n$.*

Chapitre 4

Relations n-aires

I Définitions

I.1 Relations orientées et non orientées

Exactement comme dans le cas des relations binaires, on considère une partie G de l'ensemble produit cartésien de n ensembles $(E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_n)$, soit $G \subset E_1 \times E_2 \times \cdots \times E_n$.

DÉFINITION 4.1 (RELATION n-AIRE). *Cette partie définit une relation n-aire entre ces ensembles.* $\diamondsuit$

NOTATION : Pour un n-uplet $(x_1\ ,\ x_2\ ,\ \ldots\ ,\ x_n)$ d'éléments de $E_1 \times E_2 \times \cdots \times E_n$, on notera $(x_1\ ,\ x_2\ ,\ \ldots\ ,\ x_n) \in G$ ou $\mathcal{R}(x_1\ ,\ x_2\ ,\ \ldots\ ,\ x_n)$ le fait que ces éléments sont en relation par la relation $\mathcal{R}$ de graphe G.

Comme dans le cas des relations binaires, les n-uplets sont ordonnés et, même si deux des ensembles E_i et E_j sont identiques (pour $i \neq j$), le couple d'éléments (x_i, y_j) est considéré comme différent du couple (y_j, x_i) lorsque $x_i \neq y_j$.

Cependant, dans la plupart des applications pratiques des relations n-aires, et dans toutes celles que nous verrons en tout cas, on « étiquette les colonnes », ce qui permet de s'affranchir de cet ordre, et de considérer ce que l'on appelle des relations n-aires *non orientées*, dont les *domaines* sont les ensembles $(E_1\ ,\ E_2\ ,\ \ldots\ ,\ E_n)$, dans un ordre non spécifié, car ils sont nommés.

I.1.1 Exemple de relation ternaire orientée

Soient
- $E_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$,
- $E_2 = \{1988, 1989, 1990, 1991, 1992, 1993, 1994\}$
- $E_3 = \{\text{Alsace}, \text{Beaujolais}, \text{Côtes du Rhône}\}$.

et soit

$G =$ { (3,1988,Alsace), (4,1991,Alsace), (8,1989,Beaujolais), (4,1989,Côtes du Rhône) }.

G est le graphe d'une relation ternaire orientée qui représente une cave à vins.

On peut la représenter par le tableau :

3	1988	Alsace
4	1991	Alsace
8	1989	Beaujolais
4	1989	Côtes du Rhône

Il est évident que l'ordre des éléments du n-uplet élément de G a une importance fondamentale, surtout lorsque l'intersection des domaines n'est pas vide.

Autrement dit, cette relation doit être considérée comme différente de la relation définie sur $E_3 \times E_2 \times E_1$ par le graphe G' représenté par le tableau

Alsace	1988	3
Alsace	1991	4
Beaujolais	1989	8
Côtes du Rhône	1989	4

I.1.2 Exemple de relation ternaire non orientée

Pour s'affranchir de l'ordre en évitant toute ambiguïté, il faut nommer les colonnes du tableau, c'est-à-dire ajouter un ensemble d'*attributs* (ou clés, ou étiquettes) qui pourraient être ici {Nombre, Année, Région}.

On obtiendrait

Nombre	Année	Région
3	1988	Alsace
4	1991	Alsace
8	1989	Beaujolais
4	1989	Côtes du Rhône

Cette relation ternaire ne sera pas considérée comme différente de la relation représentée par

Région	Année	Nombre
Alsace	1988	3
Alsace	1991	4
Beaujolais	1989	8
Côtes du Rhône	1989	4

En effet, les attributs ne sont pas ordonnés, l'ensemble {Région, Année, Nombre} est égal à l'ensemble {Nombre, Année, Région}.

Dans la suite, le terme de relation n-aire sera réservé aux relations non orientées.

On peut toujours associer à une relation n-aire une relation n-aire orientée, définie sur $D_1 \times D_2 \times \cdots \times D_n$, où les D_i sont les domaines attachés aux attributs de A, énoncés dans un certain ordre.

Bien entendu, si les attributs sont énoncés dans un ordre différent, la relation n-aire orientée associée peut ne pas être la même, mais, pour une même relation n-aire, toutes les relations n-aires orientées associées se déduisent les unes des autres par une permutation sur les domaines.

C'est pourquoi on s'autorisera à utiliser l'abus de notation $\mathcal{R}(x_1\,,\,x_2\,,\,\ldots\,,\,x_n)$, pour exprimer que les x_i sont en relation par la relation n-aire (non orientée) $\mathcal{R}$, en se référant à l'une quelconque des relations n-aires orientées associées (celle qui correspond à l'ordre des domaines D_i lorsque les x_i sont énoncés).

NOTATION : On notera $\mathcal{R}[A]$ une relation n-aire (non orientée) d'attributs A.

I.2 Relations équivalentes, relations égales

DÉFINITION 4.2 (RELATIONS n-AIRES ÉQUIVALENTES). *Deux relations n-aires (non orientées) sont* équivalentes *lorsque leurs domaines sont les mêmes et qu'il existe une permutation de ces domaines telle que les relations orientées associées sont égales (au sens de l'égalité des ensembles, puisqu'une relation n-aire orientée est définie comme un ensemble).* ♢

DÉFINITION 4.3 (RELATIONS n-AIRES ÉGALES). *Deux relations n-aires (non orientées) sont* égales *lorsqu'elles sont équivalentes et que leurs attributs sont les mêmes.* ♢

Groupe	Nom	Age
1	A	18
1	B	17
2	C	18

Une relation $\mathcal{R}$

Age	Nom	Groupe
18	A	1
17	B	1
18	C	2

Une relation égale à $\mathcal{R}$

Note	Matière	Nombre
18	A	1
17	B	1
18	C	2

Une relation équivalente à $\mathcal{R}$

I.3 Interprétation fonctionnelle

Chaque ligne du tableau d'une relation n-aire $\mathcal{R}[A]$ aux attributs A, de domaines $(D_1\ ,\ D_2\ ,\ \ldots\ ,\ D_n)$, peut être interprétée comme une application de A (l'ensemble des attributs) dans $D_1 \cup D_2 \cup \cdots \cup D_n$.

EXEMPLE 4.1. Par exemple, pour la première relation du paragraphe précédent, on peut considérer les fonctions f_1, f_2 et f_3 définies par $f_1(\text{Groupe}) = 1$, $f_1(\text{Nom}) = A$, $f_1(\text{Age}) = 18$, $f_2(\text{Groupe}) = 1$, etc.

I.4 SGBD

DÉFINITION 4.4 (SGBD). *Un Système de Gestion de Base de Données Relationnelles (SGBD) est une application informatique de définition et de travail sur des relations n-aires (non orientées).* ♢

Cette application met en général à la disposition de l'utilisateur un langage (le plus souvent, SQL) qui permet

- de définir les objets et leurs liens, de les modifier et d'enrichir la base de données,
- de retrouver l'information contenue dans la base de données par la formulation de requêtes.

II Projections

II.1 Définitions

Soit $\mathcal{R}[A]$ une relation n-aire d'attributs A, et $a \in A$.

On pose $A = \{a\} \cup B$, et on suppose que le domaine de a est D_1 et que les domaines des attributs de B sont $D_2\ ,\ \ldots\ ,\ D_n$.

DÉFINITION 4.5 (PROJECTION D'UNE RELATION). *La projection de la relation $\mathcal{R}$ suivant a sur B, notée $\mathcal{R}_a$ (on autorise aussi $\mathcal{R}[B]$), est définie par :*

$$\mathcal{R}_a(x_2\ ,\ \ldots\ ,\ x_n) \qquad \Leftrightarrow \qquad \exists x_1 \in D_1,\ \mathcal{R}(x_1\ ,\ x_2\ ,\ \ldots\ ,\ x_n).$$

REMARQUE 4.1. Dans la pratique, on obtient la projection d'une relation :

- en supprimant la colonne de l'attribut selon lequel se fait la projection,
- et en ne conservant qu'une seule occurrence de lignes qui seraient devenues identiques.

II.2 Théorème des projections

Soit $\mathcal{R}[A]$ une relation n-aire d'attributs A, $a \in A$, $b \in A$ ($b \neq a$).

PROPRIÉTÉ 4.1 (THÉORÈME DES PROJECTIONS) :

$$(\mathcal{R}_a)_b = (\mathcal{R}_b)_a \ .$$

PREUVE (Démonstration immédiate). ■

REMARQUE 4.2. Ce dernier résultat nous autorise à considérer la projection d'une relation suivant un sous-ensemble d'attributs (et sur le complémentaire de ce sous-ensemble d'attributs).

NOTATION : On notera cette projection $\mathcal{R}_B$ (ou $\mathcal{R}[A \setminus B]$) (si $B \subset A$, c'est la projection suivant B de $\mathcal{R}$ sur $C = A \setminus B$.

III Opérations sur les relations n-aires

III.1 Somme et produit

Soit $\mathcal{R}$ une relation d'attributs A et $\mathcal{S}$ une relation d'attributs B, pour lesquelles les attributs de même nom ont même domaine.

Les relations somme $\mathcal{R} + \mathcal{S}$ et produit $\mathcal{R} * \mathcal{S}$ ont pour attributs $A \cup B$.

Pour l'énoncé de la définition, comme l'ordre dans lequel on énonce les attributs est sans importance, on suppose que, dans $A \cup B$, les éléments sont énumérés dans l'ordre suivant

– les attributs de A qui ne sont pas dans B, les domaines sont $D_1 , \ldots , D_p$,
– les attributs communs à A et à B, les domaines sont $D_{p+1} , \ldots , D_q$,
– les attributs de B qui ne sont pas dans A, les domaines sont $D_{q+1} , \ldots , D_n$.

(l'un de ces sous-ensembles peut être vide).

DÉFINITION 4.6. *On a alors, par définition*

– $(\mathcal{R} + \mathcal{S})(x_1 , \ldots , x_p , x_{p+1} , \ldots , x_q , \ldots , x_{q+1} , x_n)$ *si et seulement si* $\mathcal{R}(x_1 , \ldots , x_q)$ *ou* $\mathcal{S}(x_{p+1} , \ldots , x_n)$,
– $(\mathcal{R} * \mathcal{S})(x_1 , \ldots , x_p , \ldots , x_q , \ldots , x_n)$ *si et seulement si* $\mathcal{R}(x_1 , \ldots , x_q)$ *et* $\mathcal{S}(x_{p+1} , \ldots , x_n)$. ♢

NOTATION : On note $(\mathcal{R} + \mathcal{S})\,[A \cup B]$ et $(\mathcal{R} * \mathcal{S})\,[A \cup B]$.

EXEMPLE 4.2. Le domaine de l'attribut Groupe est $\{1, 2, 3\}$, celui de Nom est {A,B,C} et celui de Age est $\{19, 20, 21\}$.

Groupe	Nom
1	A
1	B
2	C

Une relation $\mathcal{R}$

Groupe	Age
1	20
1	21
2	21

Une relation $\mathcal{S}$

Groupe	Nom	Age
1	A	20
1	A	21
1	B	20
1	B	21
2	C	21

La relation $\mathcal{R} * \mathcal{S}$

Groupe	Nom	Age
1	A	19
1	A	20
1	A	21
1	B	19
1	B	20
1	B	21
1	C	20
1	C	21
2	A	21
2	B	21
2	C	19
2	C	20
2	C	21

La relation $\mathcal{R} + \mathcal{S}$

III.2 Réunion et intersection

C'est le cas particulier de la somme et du produit de deux relations d'attributs A et B dans le cas où $A = B$.

Donc, dans le cas où $A = B$, $\mathcal{R} \cup \mathcal{S} = \mathcal{R} + \mathcal{S}$ et $\mathcal{R} \cap \mathcal{S} = \mathcal{R} * \mathcal{S}$.

NOTATION : On note donc $(\mathcal{R} \cup \mathcal{S})\,[A]$ et $(\mathcal{R} \cap \mathcal{S})\,[A]$

III.3 Produit cartésien

Il s'agit du cas particulier du produit de deux relations dans le cas où $A \cap B = \emptyset$.

Donc, dans le cas où $A \cap B = \emptyset$, $\mathcal{R} \times \mathcal{S} = \mathcal{R} * \mathcal{S}$.

NOTATION : On note donc $(\mathcal{R} \times \mathcal{S})\,[A \cup B]$.

IV Sélection d'une relation n-aire

Soit $\mathcal{R}[A]$ une relation n-aire d'attributs A et F une formule de logique dans laquelle les variables sont des éléments de A et les constantes des éléments du domaine des attributs.

DÉFINITION 4.7. *La sélection de $\mathcal{R}$ suivant F est une relation ayant les mêmes attributs A, notée $(\mathcal{R} : F)\,[A]$ et telle que $\mathcal{R}(x_1\,,\,x_2\,,\,\ldots\,,\,x_n)$ et $F(x_1\,,\,x_2\,,\,\ldots\,,\,x_n)$.* ♢

Autrement dit, il s'agit des éléments des domaines des attributs qui sont en relation par $\mathcal{R}$ et qui satisfont la formule F donnée.

EXEMPLE 4.3. Sur une relation d'attributs $\{\text{Nom}\,,\,\text{Age}\,,\,\text{Note}\}$ on pourra définir la relation $[(\text{Age} \leqslant 19)$ et $(\text{Note} \geqslant 16)]$ pour sélectionner les étudiants admis à s'inscrire au département d'Informatique.

V Dépendances fonctionnelles et clés

V.1 Dépendances fonctionnelles

Il s'agit, lorsque c'est possible, de remplacer une relation n-aire par une autre, plus simple, et sans perte d'information.

Soit $\mathcal{R}[A]$ une relation d'attributs A telle que A soit de la forme $X \cup Y \cup Z$.
On suppose pour simplifier :
- que les domaines des attributs de X sont $D_1\,,\,D_2\,,\,\ldots\,,\,D_p$,
- que ceux de Y sont $D_{p+1}\,,\,\ldots\,,\,D_q$
- que ceux de Z sont $D_{q+1}\,,\,\ldots\,,\,D_n$.

DÉFINITION 4.8 (DÉPENDANCE FONCTIONNELLE). *On dit que Y dépend fonctionnellement de X lorsque l'on a*

$$\mathcal{R}(x_1\,,\,\ldots\,,\,x_p\,,\,x_{p+1}\,,\,\ldots\,,\,x_q\,,\,x_{q+1}\,,\,\ldots\,,\,x_n)$$

et

$$\mathcal{R}(x_1\,,\,\ldots\,,\,x_p\,,\,x'_{p+1}\,,\,\ldots\,,\,x'_q\,,\,x'_{q+1}\,,\,\ldots\,,\,x'_n)$$

si et seulement si

$$x_{p+1} = x'_{p+1}\,,\,\ldots\,,\,x_q = x'_q$$

. ♢

NOTATION : Dans la suite, et pour une situation du même type, on s'autorisera à utiliser les notations suivantes :

- D_X pour D_1 , D_2 , ... , D_p,
- D_Y pour D_{p+1} , ... , D_q,
- D_Z pour D_{q+1} , ... , D_n,
- x pour $(x_1$, ... , $x_p)$,
- y pour $(x_{p+1}$, ... , $x_q)$
- et z pour $(x_{q+1}$, ... , $x_n)$.

Ainsi, la condition énoncée peut s'écrire plus simplement $\mathcal{R}(x, y, z)$ et $\mathcal{R}(x, y', z')$ si et seulement si $y = y'$ (cette égalité devant être considérée comme une égalité de n-uplets, c'est-à-dire l'égalité composante par composante.

EXEMPLE 4.4. Dans la relation suivante,

Groupe	Nom	Niveau	Age
1	A	1	20
2	B	3	21
1	C	3	20
1	A	3	20
3	B	1	21
1	C	1	20
2	A	1	20
3	B	2	21
1	C	2	20

On distingue les dépendances fonctionnelles

- {Nom} $\Rightarrow$ {Age}
- {Groupe , Niveau} $\Rightarrow$ {Age}

V.2 Théorème des dépendances fonctionnelles

PROPRIÉTÉ 4.2 (THÉORÈME DES DÉPENDANCES FONCTIONNELLES) : Si la relation $\mathcal{R}[A]$ d'attributs $A = X \cup Y \cup Z$ admet une dépendance fonctionnelle $X \Rightarrow Y$, elle est le produit de ses projections sur $X \cup Y$ et $X \cup Z$.

EXEMPLE 4.5. La relation précédente est le produit de ses deux projections $\mathcal{R}[\{\text{Nom , Age}\}]$ et $\mathcal{R}[\{\text{Nom , Groupe , Nive}$

V.3 Clés

DÉFINITION 4.9 (CLÉ). *Pour une relation $\mathcal{R}[A]$ d'attributs A, une* clé *est un sous-ensemble minimal K de A tel qu'il existe une dépendance fonctionnelle $C \Rightarrow A \setminus K$.*

(K est un sous-ensemble minimal au sens qu'il n'y a pas de partie stricte K' de K pour laquelle il existe une dépendance fonctionnelle $K' \Rightarrow A \setminus K'$). ♢

REMARQUE 4.3. Cette « minimalité » n'entraîne en aucune manière l'unicité de la clé pour une relation donnée.

PROPRIÉTÉ 4.3 : Pour toute relation, il est possible d'introduire un attribut dont les valeurs sont toutes différentes, et qui constitue donc une clé pour la nouvelle relation obtenue (par exemple, une numérotation).

Fin du Chapitre

Deuxième partie

Arithmétique

Chapitre 5

Ensembles de nombres entiers

I Nombres entiers naturels ($\mathbb{N}$)

I.1 Définition de $\mathbb{N}$

I.1.1 Définition

DÉFINITION 5.1 (ENSEMBLE DES ENTIERS NATURELS). *On appelle* ensemble des nombres entiers naturels $\mathbb{N}$ *tout ensemble possédant les propriétés suivantes*

1. *Il existe une injection de* $\mathbb{N}$ *dans* $\mathbb{N}$.
   Cette injection, appelée fonction de succession*, sera notée* s *dans la suite.*
   L'image d'un entier n *par la fonction de succession* s*, soit* $s(n)$*, est appelée* successeur *de* n.

2. *Il existe un élément de* $\mathbb{N}$ *qui n'est le successeur d'aucun élément de* $\mathbb{N}$.
   Cet élément est appelé « zéro *» et noté 0 dans la suite.*
3. *Le «* Principe de récurrence *» est satisfait :*
   Soit M *la partie de* $\mathbb{N}$ *constituée par les entiers qui possèdent une certaine propriété* p*. On note «* $p(n)$ *» le fait que l'entier* n *possède la propriété* p.

> PROPRIÉTÉ 5.1 (PRINCIPE DE RÉCURRENCE) : Il s'énonce ainsi : « Si M contient 0 et le successeur de chacun de ses éléments, alors $M = \mathbb{N}$. »

Sous forme formalisée...
Soit $M = \{n \in \mathbb{N} \mid p(n)\}$ *; si* $0 \in M$ *et si* $[n \in M \Rightarrow s(n) \in M]$*, alors* $M = \mathbb{N}$. ♢

REMARQUE 5.1. $M = \mathbb{N}$ signifie évidemment que la propriété est possédée par tous les entiers naturels. C'est, en général, la conclusion attendue d'un « raisonnement par récurrence »

I.1.2 Sur la récurrence

Il existe une version « affaiblie » du principe de récurrence : la récurrence restreinte, qui permet de s'assurer qu'une propriété est vraie à partir d'un certain rang...

> PROPRIÉTÉ 5.2 (RÉCURRENCE RESTREINTE) : Soit $M = \{n \in \mathbb{N} \mid p(n)\}$.
> Si $p \in M$ et si, $[n \in M \Rightarrow s(n) \in M]$, alors M est de la forme $\{p, p+1, p+2, \ldots\}$.

Il existe encore une version « renforcée » : la récurrence généralisée, qui permet de « supposer la propriété vraie jusqu'à l'ordre n »...

PROPRIÉTÉ 5.3 (RÉCURRENCE GÉNÉRALISÉE) : Soit $M = \{n \in \mathbb{N} \mid p(n)\}$.
Si, $\forall p \in M$, $\{0, 1, 2, \ldots, p\} \subset M$ et si $s(p) \in M$, alors $M = \mathbb{N}$.

PREUVE Elle se démontre à partir de la récurrence « normale ». ■

REMARQUE 5.2. La récurrence généralisée permet d'éviter un double raisonnement par récurrence.

Manière correcte de rédiger un raisonnement par récurrence :

1. Soit M l'ensemble des entiers naturels qui vérifient ... (mettre ici l'énoncé de la propriété que l'on cherche à démontrer)
2. Initialisation de la récurrence : vérifier que 0 est élément de M (« la propriété est vraie pour $n = 0$ »)
3. Caractère héréditaire de la propriété : soit n un élément de M (cela a un sens, puisque l'on sait maintenant que M n'est pas vide : il contient au moins 0), vérifions que $s(n)$ est encore élément de M (« la propriété est vraie pour $n + 1$ »)

REMARQUE 5.3. Toute phrase, telle que celles que l'on peut souvent lire, de la forme « supposons la propriété vraie pour n » devrait immédiatement appeler la question : qu'est-ce que c'est que n ?

- Si n est « un entier quelconque » , alors vous supposez la propriété vraie pour un entier quelconque, et il ne vous reste plus grand chose à démontrer....
- Si n est un entier fixé, mettons 47, alors vous allez démontrer la propriété pour 48, et il vous restera pas mal de chemin à faire...

Non, ce que vous supposez, ce n'est pas que la propriété est vraie (pour quoi que ce soit), mais que n est un entier pour lequel la propriété est vérifiée (cet entier étant évidemment quelconque parmi ceux pour lesquels la propriété est vérifiée), ce n'est pas du tout la même chose.

I.1.3 Exercices

Une première récurrence

Exercice 5.1. *1. Calculez 1, 1+3, 1+3+5, et 1+3+5+7.*

2. A quoi $1 + 3 + 5 + 7 + ... + (2n - 1) + (2n + 1)$ *semble-t-il être égal (en fonction de* n*) ?*

3. Démontrer que l'on a effectivement l'égalité

Somme d'entiers élevés à une puissance donnée On montre, dans ce qui suit, une application de la technique de récurrence, pour calculer $S_k(n) = 1^k + 2^k + ... + n^k, \forall k, n \in \mathbb{N}$ (d'autres techniques, plus efficaces, existent).

Exercice 5.2. *On souhaite calculer* $S_1(n) = 1 + 2 + ... + n$*.*

1. *Cherchez un bon candidat* $S_1(n)$ *pour cette formule.*
   - *On pourra chercher un lien logique entre* $S_1(1), S_1(2), S_1(3), S_1(4), ...$
   - *On pourra aussi faire le lien avec les suites arithmétiques.*
   - *Ou encore, retrouver la méthode de Gauss :* $S = 1+2+...+n$*, et* $S = n+(n-1)+...+2+1$*. Si on somme ces deux expressions...*

2. *Prouvez, par récurrence, que la somme est bien égale à ce candidat.*
3. *Quelle est la « forme » de ce candidat (fonction tangente ? polynôme ?)*

Exercice 5.3. *On souhaite calculer* $S_2(n) = 1^2 + 2^2 + ... + n^2$.

1. *Cherchez un bon candidat* $S_2(n)$ *pour cette formule.*
   - *On pourra chercher un lien logique entre* $S_2(1), S_2(2), S_2(3), S_2(4), ...$
   - *Ou encore,*
     - *Regardez la forme de* $S_0(n) = 1^0 + 2^0 + ... + n^0$, *et de* $S_1(n) = 1^1 + 2^1 + ... + n^1$
     - *Interpolez la formule pour* $S_2(n)$. *On pourra imaginer que* $S_2(n)$ *est toujours un polynôme en* n. *Quel serait son degré le plus probable ? Quelle en serait donc la forme ? On aura à déterminer les coefficients intervenant dans ce polynôme. Pour ce faire, il suffit de considérer que cette formule doit convenir pour n=1, 2, etc.*
2. *Démontrez, par récurrence, que l'on a bien égalité entre* $1^2 + 2^2 + ... + n^2$ *et votre candidat.*

Exercice 5.4. *Poursuivre le raisonnement pour* $S_3(n)$. *Cette méthode permet-elle de calculer* $S_k(n), \forall k, n$ *?*

Autres exercices sur la récurrence

Exercice 5.5. *Montrer que* $\forall n \in \mathbb{N}$, *7 divise* $3^{2n+1} + 2^{n+2}$.

Exercice 5.6. *Soit* $(u_n)_{n\in\mathbb{N}}$ *une suite réelle telle que* $\forall n \in \mathbb{N}$,

$$u_{n+2} - 5u_{n+1} + 6u_n = 0$$

Montrez qu'il existe $\alpha, \beta \in \mathbb{N}$ *tels quel* $\forall n \in \mathbb{N}, u_n = \alpha 3^n + \beta 2^n$.

Exercice 5.7. *Montrer que* $\forall m, n \in \mathbb{N}^*, \forall r \in \mathbb{N}, m^{2r+1} + n^{2r+1}$ *est divisible par* $m + n$.

I.2 Opérations et relation d'ordre dans $\mathbb{N}$

On suppose ici connues les opérations et la relation d'ordre classiques qui existent dans $\mathbb{N}$: addition, multiplication, relation d'inégalité au sens large.

Ces éléments peuvent être définis rigoureusement, et toutes les propriétés démontrées par récurrence.

EXEMPLE 5.8. Par exemple, on peut définir la relation $p \leqslant n$ par $\exists q \in \mathbb{N},\ n = p + q$.

PROPRIÉTÉ 5.4 : Les opérations précédentes ont pour propriétés :
- l'addition est commutative, associative, il existe un élément neutre (0),
- la multiplication est commutative, associative et admet aussi un élément neutre (1),
- la multiplication est distributive sur l'addition,
- les entiers sont totalement ordonnés par l'inégalité, et cette relation d'ordre est compatible avec l'addition et avec la multiplication.

I.3 Nombres premiers

I.3.1 Définitions

DÉFINITION 5.2 (MULTIPLE, DIVISEUR). *Si un entier* n *peut s'écrire sous la forme* $n = pq$, *où* p *et* q *sont des entiers, on dit que* n *est un* multiple *de* p *et que* p *est un* diviseur *de* n. ♢

Exercice 5.9. *Soit $m = 2^3 * 5 * 7^2 * 13^3$. Combien le nombre m a-t-il de diviseurs naturels ?*

Réponse : (3+1)*(1+1)*(2+1)*(3+1)=96.

DÉFINITION 5.3 (NOMBRE PREMIER). *Un* nombre premier *est un nombre entier strictement supérieur à 1 qui n'est divisible que par 1 et par lui-même.* ♢

EXEMPLE 5.10. Ainsi, le plus petit nombre premier (et le seul qui soit pair) est 2.

PROPRIÉTÉ 5.5 : Il existe une infinité de nombres premiers.

REMARQUE 5.4. Le problème de la primalité d'un nombre (très grand, évidemment) est difficile.

I.3.2 Décomposition en facteurs premiers

DÉFINITION 5.4 (DÉCOMPOSITION EN FACTEURS PREMIERS). *L'écriture d'un entier n sous la forme $n = a^\alpha b^\beta c^\gamma \ldots$, où a , b , c , $\ldots$ sont les diviseurs premiers distincts de n et où les exposants α , β , γ , $\ldots$ sont tels que, par exemple, n est divisible par a^α mais pas par $a^{\alpha+1}$ s'appelle la* décomposition en facteurs premiers *de n.*

On dit que les exposants α, β, γ, ... sont les ordres de multiplicité des diviseurs a, b, c, ...) ♢

PROPRIÉTÉ 5.6 : La décomposition d'un entier en ses facteurs premiers est unique.

Exercice 5.11. *Écrivez les nombres 3850 et 1911 sous forme de produits de nombres premiers.*

Réponses : $2 * 5^2 * 7 * 11$ et $3 * 7^2 * 13$.

Exercice 5.12 (Nombres de Fermat). *On appelle nombres de Fermat les nombres de la forme $2^{2^p} + 1$.*

1. *Montrer que, pour que $2^n + 1$ soit premier, il faut que n soit une puissance de 2.*
2. *La réciproque n'est pas vraie : donner un exemple de nombre de Fermat qui ne soit pas premier.*
3. *Montrer que, pour $k \geqslant 1$, F_p divise $F_{p+k} - 2$.*
4. *En déduire que F_p et F_{p+k} sont premiers entre eux.*
5. *En déduire qu'il existe une infinité de nombres premiers.*

I.4 Relation de divisibilité

On a vu dans le chapitre sur les relations entre ensembles la relation binaire de divisibilité définie dans $\mathbb{N}^*$.

Cette relation est une relation d'ordre partiel : il existe des paires d'entiers non comparables par cette relation.

EXEMPLE 5.13. 3 ne divise pas 7 et 7 ne divise pas 3.

Ces deux entiers ne sont donc pas comparables du point de vue de la divisibilité.

Cet ordre n'est donc que partiel, mais il existe, pour chaque couple d'entiers distincts, une borne inférieure et une borne supérieure...

DÉFINITION 5.5 (PGCD, PPCM). *Tout ensemble fini de nombres entiers strictement positifs admet une borne sup et une borne inf pour la relation de divisibilité.*

Cette borne inférieure et cette borne supérieure sont respectivement appelées plus grand commun diviseur *et* plus petit commun multiple *de ces deux entiers.* ♢

NOTATION : Ils sont respectivement notés $a \wedge b$ et $a \vee b$.

PREUVE L'existence du PGCD découle de l'existence de la décomposition en facteurs premiers : il suffit de comparer les décompositions des deux nombres pour découvrir leur PGCD.

Le PPCM, lui, vaut $a \vee b = ab/(a \wedge b)$. ■

EXEMPLE 5.14. Comme $48 = 2^4 3$ et que $56 = 2^3 7$, on voit aisément que $48 \wedge 56 = 2^3$.

Exercice 5.15. *Calculez* $ppcm(102, 138)$.

Réponse : 2346.

PROPRIÉTÉ 5.7 : $\mathbb{N}^*$ est un treillis pour la divisibilité.

On peut de plus montrer que :

- ce treillis est distributif, c'est-à-dire que $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ et que $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$,
- il admet un élément minimum (1), mais pas d'élément maximum,
- les nombres premiers sont les éléments minimaux de $(\mathbb{N}^* \setminus \{1\})$.

DÉFINITION 5.6. *Deux nombres entiers strictement positifs* a *et* b *sont dits* premiers entre eux *lorsque* $a \wedge b = 1$. ♢

Exercice 5.16. *Soient* a, b, c, d *des entiers naturels non nuls tels que* $ad = bc$.
Prouvez que si a *et* b *sont premiers entre eux, alors* $b|d$

Réponse : En se plongeant dans le calcul modulo b, on a : ad = 0.
Comme a et b sont premiers entre eux, a est inversible, et donc $d = 0$.
On en déduit que d est un multiple de b.

I.5 Entiers relatifs

L'ensemble habituellement noté $\mathbb{Z}$ des entiers relatifs est obtenu à partir de $\mathbb{N}$ par le procédé de symétrisation pour l'addition.

Sans s'étendre sur le sujet, disons que cela consiste à introduire les entiers strictement négatifs comme opposés des positifs correspondants, par $n + (-n) = 0$.

On sait que les porpriétés des opérations sont conservées ; la seule propriété perdue dans cette extension est la compatibilté entre la relation d'ordre et la multiplication.

En revanche, on gagne évidemment l'existence d'un opposé pour chaque entier.

II Division euclidienne dans $\mathbb{Z}$ et applications

II.1 Définition

On se donne deux entiers relatifs a et b, b non nul.

PROPRIÉTÉ 5.8 : Il existe un et un seul couple d'entiers relatifs q et r qui vérifient la relation suivante : $a = bq + r$, avec $0 \leqslant r < |b|$.

DÉFINITION 5.7 (DIVISION EUCLIDIENNE). *Obtenir les valeurs de q et de r, c'est effectuer la* division euclidienne *de a par b.*

q est appelé quotient*, r est appelé* reste *(dans la division euclidienne).*

Enfin, lorsque r est nul, a est dit divisible *par b, ou b est un* diviseur *de a.* ♢

EXEMPLE 5.17. Tout nombre non nul est au moins divisible par 1 et par lui-même ($a = a \times 1 + 0$).

EXEMPLE 5.18. 0 est divisible par tout nombre entier non nul $(0 = 0 \times b + 0)$.

Exercice 5.19. *Quels sont le quotient et le reste de la division euclidienne de m par n dans le cas où :*

1. $m = -38$ *et* $n = 6$,
2. $m = 165$ *et* $n = -14$.

Réponses : (-7,4) et (-11,11).

Exercice 5.20 (Divisibilité dans $\mathbb{N}$). *On se place dans l'ensemble $\mathbb{N}$.*

1. *Trouver les restes dans la division par 5 du carré d'un entier.*
2. *Trouver les restes dans la division par 8 du carré d'un entier impair.*
3. *Trouver les restes dans la division par 11 de 37^n (pour $n \in \mathbb{N}^*$).*
4. *Montrer que $10^n(9n - 1) + 1$ est divisible par 9.*

II.2 Représentation des nombres entiers

II.2.1 Définition

DÉFINITION 5.8 (PRINCIPE DE LA NUMÉRATION DE POSITION). *Il consiste à choisir une base b de numération, et b symboles qui constitueront les chiffres dans la représentation d'un entier positif en base b.*

Celle-ci s'écrira alors

$$n = n_p b^p + n_{p-1} b^{p-1} + \cdots + n_1 b^1 + n_0$$

NOTATION : Cette écriture est abrégée en $(\overline{n_p n_{p-1} \ldots n_0})_b$.

REMARQUE 5.5. En informatique, on utilise couramment les bases 2, 8 et 16.

II.2.2 Obtention de cette représentation

L'algorithme pour obtenir la représentation en base b d'un entier est :

1. Effectuer la division euclidienne de cet entier par b, division qui donne un premier quotient et un premier reste.
2. Le quotient est à sont tour divisé par b pour donner un second quotient et un second reste, et ainsi de suite jusqu'à obtenir un quotient nul.
3. Les restes successifs (tous strictement inférieurs à b), et en commençant par le dernier, constituent la représentation en base b de l'entier donné.

II.2.3 Algorithme de Hörner

Réciproquement, étant donnée la représentation en base b d'un entier, on obtient sa valeur par application de l'algorithme de Hörner :

$n = n_p b^p + n_{p-1} b^{p-1} + \cdots + n_1 b^1 + n_0$ est calculé par $(.....(((n_p b + n_{p-1})b + n_{p-2})b + \cdots + n_1)b + n_0)$

II.2.4 Exercices

Exercice 5.21 (Numération, changements de base). *1. Chercher les entiers dont le carré a, en représentation décimale, mêmes chiffres des dizaines et des unités.*

2. *On pose $a = 2p - 1$, $b = 2p + 1$, $c = 2p + 3$; trouver l'entier p de manière que $a^2 + b^2 + c^2$ soit de la forme $\overline{xxxx}_{10}$.*
3. *L'entier n s'écrit $\overline{341}_{10}$ et $\overline{2331}_a$. Trouver a.*
4. *Montrer que, dans toute base b supérieure ou égale à 3, l'entier qui s'écrit $\overline{11211}_b$ n'est pas premier.*
5. *soit $n \geqslant 7$. Donner l'écriture de $(n + 1)^4$ en base n.*

Exercice 5.22 (Développement décimal). *On considère le nombre réel x dont le développement décimal s'écrit $x = 0,012\ 345\ 679\ 012\ 345\ 679\ \ldots\ \ldots\ \ldots$ (la séquence $012\ 345\ 679$ est reproduite indéfiniment). Ce développement décimal est périodique, de période 9.*

1. *Montrer que x vérifie une équation de la forme $10^k x = n + x$, où k et n sont des entiers à déterminer. En résolvant cette équation, montrer que x est un nombre rationnel, et le mettre sous la forme $x = \frac{p}{q}$, où p et q sont premiers entre eux.*
2. *Appliquer la même méthode au "nombre" y dont le développement décimal est $y = 0,999\,999\,999\,999\ \ldots$ (périodique de période 1). Quelle conclusion peut-on en tirer ?*
3. *Démontrer que tout nombre réel dont le développement décimal est fini ou périodique à partir d'un certain rang est un nombre rationnel.*
4. *Réciproquement, on se propose de démontrer que le développement décimal de tout nombre rationnel est fini ou périodique à partir d'un certain rang. Pour cela, on considère un rationnel $x = \frac{p}{q}$, avec $q > 0$, $p \in \mathbb{Z}$, p et q premiers entre eux, et on étudiera successivement les cas suivants :*
   - *x est entier (c'est à dire $q = 1$)*
   - *x est rationnel non entier, et q est premier avec 10 (On pourra montrer que, si q est premier avec 10, il existe un entier k, non nul, tel que $10^k \equiv 1\ [q]$).*
   - *x est rationnel non entier, mais q n'est pas premier avec 10.*

II.3 Arithmétique modulo n

On rappelle ici la définition de la relation dite de « congruence modulo n » définie dans $\mathbb{Z}$ étudiée dans le chapitre consacré aux relations entre ensembles.

DÉFINITION 5.9 (CONGRUENCE MODULO n). *Soit n un entier strictement supérieur à 1 et x et y deux éléments de $\mathbb{Z}$.*

On dit que « x est congru *à y* modulo *n » lorsque x et y possèdent le même reste dans la division (euclidienne) par n :*

$$x \equiv y[n] \Leftrightarrow \exists k \in \mathbb{Z}, x - y = k \cdot n$$

PROPRIÉTÉ 5.9 : Il s'agit d'une relation d'équivalence dans $\mathbb{Z}$.

PREUVE En effet :

- $\forall x \in \mathbb{Z}, x - x = 0 = 0 \cdot n$; or $0 \in \mathbb{Z}$, donc $x \equiv x[n]$ (réflexivité).
- Si $x \equiv y[n]$,$\exists k \in \mathbb{Z}$, $x - y = k \cdot n$; alors $y - x = (-k) \cdot n$, et, puisque $k \in \mathbb{Z}$, $(-k) \in \mathbb{Z}$, donc $y \equiv x[n]$ (symétrie).
- Si $x \equiv y[n]$,$\exists k \in \mathbb{Z}$, $x - y = k \cdot n$; si, de plus, $y \equiv z[n]$, $\exists l \in \mathbb{Z}$, $y - z = l \cdot n$; alors (par addition), $x - z = (k + l) \times n$; comme $k \in \mathbb{Z}$ et $l \in \mathbb{Z}$, $(k + l) \in \mathbb{Z}$, donc $x \equiv z[n]$ (transitivité). ■

La classe d'équivalence d'un entier donné comprend donc cet entier et tous ceux qui ont le même reste que lui dans la division euclidienne par n.

EXEMPLE 5.23. Si $n = 3$, il y a trois classes distinctes :

- $\dot{0} = \{\ldots, -6, -3, 0, 3, 6, 9, \ldots\}$,
- $\dot{1} = \{\ldots, -5, -2, 1, 4, 7, 10, \ldots\}$,
- $\dot{2} = \{\ldots, -4, -1, 2, 5, 8, 11, \ldots\}$.

On retrouve ensuite les mêmes éléments : $\dot{3} = \dot{0}$, etc...

D'une manière générale, pour n quelconque, il y a exactement n classes d'équivalence, notées de $\dot{0}$ à $(n - 1)$, c'est-à-dire, il faut le remarquer, un nombre fini.

PROPRIÉTÉ 5.10 : L'ensemble-quotient (ensemble des classes d'équivalence) de la relation de congruence modulo n est un ensemble fini.

NOTATION : Il est noté $\mathbb{Z}/n\mathbb{Z}$.

EXEMPLE 5.24. $\mathbb{Z}/3\mathbb{Z} = \{\dot{0}, \dot{1}, \dot{2}\}$.

PROPRIÉTÉ 5.11 : La relation de « congruence modulo n » est compatible avec l'addition et la multiplication des nombres entiers.

PREUVE En effet, on suppose que :

– $x \equiv x'[n] \Leftrightarrow \exists k \in \mathbb{Z},\ x - x' = k \cdot n$ et que
– $y \equiv y'[n] \Leftrightarrow \exists l \in \mathbb{Z}, y - y' = l \cdot n$.
– Alors, par addition, $(x+y)-(x'+y') = (k+l)\cdot n$; $(k+l) \in \mathbb{Z}$, donc $(x+y) \equiv (x'+y')[n]$: la congruence modulo n est compatible avec l'addition dans $\mathbb{Z}$.

En multipliant la première égalité par y : $xy - x'y = (ky)\cdot n$ et la seconde par x' : $x'y - x'y' = (x'l)\cdot n$.

Alors, par addition, $xy - x'y' = (ky + lx') \cdot n$. $(ky + lx') \in \mathbb{Z}$, donc $x \cdot y \equiv x' \cdot y'[n]$: la congruence modulo n est aussi compatible avec la multiplication dans $\mathbb{Z}$. ■

REMARQUE 5.6. C'est cette propriété qui permet de définir dans l'ensemble quotient $\mathbb{Z}/n\mathbb{Z}$ des opérations, dites *induites* par celles qui existent dans $\mathbb{Z}$...

DÉFINITION 5.10. *Par définition, on pose* $\dot{x} + \dot{y} = \dot{(x + y)}$ *et* $\dot{x} \cdot \dot{y} = \dot{(xy)}$. ♢

EXEMPLE 5.25. C'est ainsi qu'on obtient les tables d'opérations suivantes dans $\mathbb{Z}/4\mathbb{Z}$:

$+$	$\dot{0}$	$\dot{1}$	$\dot{2}$	$\dot{3}$
$\dot{0}$	$\dot{0}$	$\dot{1}$	$\dot{2}$	$\dot{3}$
$\dot{1}$	$\dot{1}$	$\dot{2}$	$\dot{3}$	$\dot{0}$
$\dot{2}$	$\dot{2}$	$\dot{3}$	$\dot{0}$	$\dot{1}$
$\dot{3}$	$\dot{3}$	$\dot{0}$	$\dot{1}$	$\dot{2}$

$\times$	$\dot{0}$	$\dot{1}$	$\dot{2}$	$\dot{3}$
$\dot{0}$	$\dot{0}$	$\dot{0}$	$\dot{0}$	$\dot{0}$
$\dot{1}$	$\dot{0}$	$\dot{1}$	$\dot{2}$	$\dot{3}$
$\dot{2}$	$\dot{0}$	$\dot{2}$	$\dot{0}$	$\dot{2}$
$\dot{3}$	$\dot{0}$	$\dot{3}$	$\dot{2}$	$\dot{1}$

REMARQUE 5.7. On aperçoit la présence de « diviseurs de zéro » ($\dot{2} \times \dot{2} = \dot{0}$), mais aussi l'apparition d'un inverse pour certains éléments ($\dot{3} \times \dot{3} = \dot{1}$).

Exercice 5.26. *Calculez :*

1. $3 * 10^9 mod 97$,
2. $3^{1024} mod 1037$.

Réponses : 5 et 630.

Exercice 5.27 (Systèmes de congruences). *Il s'agit de trouver des entiers x qui satisfont des systèmes de la forme*

$$\begin{cases} x & \equiv & a\ [p] \\ x & \equiv & b\ [q] \end{cases}$$

Un tel système peut ne pas avoir de solution (par exemple, $a = 1,\ p = 2,\ b = 0,\ q = 4$: un nombre impair ne peut être un multiple de 4).

Une condition suffisante d'existence de solutions est que p et q soient premiers entre eux.

C'est le cas que nous traiterons ici ; dans ce cas, il existe deux entiers u et v tels que $pu + qv = 1$ (théorème de Bezout).

Donc $pu \equiv 1\ [q]$ et $qv \equiv 1\ [p]$, et $x = bpu + aqv$ est une solution du système (pourquoi ? ?) ; les autres sont de la forme $x + kpq$, où k est un entier quelconque.

1. *Résoudre le système de congruences*

$$\begin{cases} x & \equiv & 2\ [88] \\ x & \equiv & 1\ [27] \end{cases}$$

2. *Application : Problème du cuisinier : Une bande de 17 pirates s'est emparée d'un butin composé de pièces d'or, toutes d'égale valeur.*
   Ils décident de se les partager également et de donner le reste éventuel au cuisinier. Celui-ci recevrait alors 3 pièces d'or.
   Malheureusement, une querelle éclate, au cours de laquelle 6 pirates sont tués. Le cuisinier recevrait alors 4 pièces d'or.
   Dans un naufrage ultérieur, seuls le butin, 6 pirates et le cuisinier sont sauvés. Le partage laisserait alors 5 pièces à ce dernier.
   Quel est le plus petit nombre de pièces d'or qu'il espère lorsqu'il décide d'empoisonner les derniers pirates ?

Exercice 5.28. *Résolvez modulo 18 les équations suivantes :*

1. $2x + 17 = 15$,
2. $3x + 4 = 12$,
3. $5x + 13 = 16$.

Réponses : {8,17}, { } et {15}.

Exercice 5.29. *Si m est un entier naturel plus grand que 2, quel est l'inverse de $m - 1$ modulo m ?*

Réponse : $m - 1$.

Exercice 5.30. *Un nombre « pseudo-premier de base b » est un entier naturel non premier p tel que $(b^p - b) mod p = 0$.*
Vérifier que 561 est pseudo-premier de base 3 et que 341 est pseudo-premier de base 2.

II.4 Division « entière » informatique et division euclidienne

La plupart des langages de programmation utilisés en informatique disposent d'un type de données pour représenter ce que les informaticiens appellent les entiers signés (les entiers relatifs) et possèdent des opérateurs pour effectuer les calculs classiques sur ces nombres.

En C ou java, par exemple, le symbole / représente le quotient dans la « division entière » et le symbole % représente ce que les informaticiens appellent improprement le modulo (le reste dans leur « division entière »).

Pour des raisons pratiques de réalisation des micro-circuits des processeurs qui réalisent ces opérations, la « division entière » ne donne pas exactement le même résultat que la division euclidienne.

Considérons par exemple les 4 cas possibles de division euclidienne de a par b lorsque $|a| = 29$ et $|b| = 7$ (en n'oubliant pas que le reste d'une division euclidienne ne peut être que positif)

a	b	division euclidienne	q	r	a/b	$a\%b$
29	7	$29 = 4 \times 7 + 1$	4	1	4	1
29	-7	$29 = (-4) \times (-7) + 1$	-4	1	-4	1
-29	7	$-29 = (-5) \times 7 + 6$	-5	6	-4	-1
-29	-7	$-29 = 5 \times (-7) + 6$	5	6	4	-1

Autrement dit, mathématiquement, le quotient est positif lorsque les deux nombres ont le même signe et le reste est toujours positif, et, pour que le reste soit toujours positif, le quotient peut ne pas être le quotient des valeurs absolues.

Informatiquement, le « quotient » est positif lorsque les nombres ont le même signe, le « reste » a le signe du dividende, et la valeur absolue du « quotient » est toujours le quotient des valeurs absolues.

Dans les applications de calcul arithmétique, par exemple un calcul de PGCD, ce n'est pas gênant parce que les restes « informatiques » sont congrus aux restes mathématiques modulo la valeur absolue du diviseur, et qu'il ne s'agit alors que du choix d'un représentant de la classe concernée (addition et multiplication étant compatibles avec la congruence modulo n).

Mais il faut quand même savoir que l'on peut obtenir un « reste » négatif et prendre ses dispositions le cas échéant...

II.5 Arithmétique modulo 2^n dans les ordinateurs

II.5.1 Présentation générale

Les calculs sur les entiers, dans un ordinateur, se font dans $\mathbb{Z}/2^n\mathbb{Z}$, où n est le nombre de bits utilisés dans la représentation de ces nombres.

Dans la plupart des microprocesseurs, les entiers sont représentés sur 32 bits, les calculs se font donc dans $\mathbb{Z}/2^{32}\mathbb{Z}$ (et qu'ils le soient sur 64 bits ne change rien au problème).

Disposer d'entiers signés ou d'entiers non signés est uniquement une question de choix du représentant dans les classes d'équivalence, mais la représentation physique est la même.

Comme il nous est difficile de représenter ici la liste compléte de tous ces entiers, nous allons illustrer ce propos en supposant que les entiers sont représentés sur 4 bits.

II.5.2 Illustration dans le cas de 4 bits.

Pour des mots de 4 bits, il y a alors 16 entiers représentables : (a.s.= arithmétique signée, a.n.s. = arithmétique non signée)

code binaire		a.s.	a.n.s.
0000	interprété par	0	0
0001	interprété par	1	1
0010	interprété par	2	2
0011	interprété par	3	3
0100	interprété par	4	4
0101	interprété par	5	5
0110	interprété par	6	6
0111	interprété par	7	7
1000	interprété par	8	-8
1001	interprété par	9	-7
1010	interprété par	10	-6
1011	interprété par	11	-5
1100	interprété par	12	-4
1101	interprété par	13	-3
1110	interprété par	14	-2
1111	interprété par	15	-1

Pourquoi ce choix ? Pourquoi ne pas avoir, en a.s., représenté les entiers dans l'ordre croissant de 0000 (-8) à 1111 (7) ?

- Tout simplement pour des raisons d'efficacité : 0 doit toujours être représenté par le code « nul » 0000.
- Ensuite, il faut pouvoir comparer efficacement ces codes entre eux, ce qui explique que 0 doit être suivi de 1, arithmétique signée ou pas.

Ces principes ont ainsi conduit à placer les codes interprétés comme entiers négatifs après ceux qui représentent les entiers positifs.

Par ailleurs, on s'aperçoit que, de cette manière, les codes des entiers négatifs commencent tous par 1. On parle improprement de « bit de signe » : s'il s'agissait d'un véritable bit de signe, le code 1001 devrait être celui de -1, or c'est celui de -7. Mais il n'en reste pas moins que tous les entiers négatifs commencent par 1).

Ainsi, il est facile de déduire la comparaison signée de la comparaison non signée : les codes qui commencent par 1 sont « plus petits » que ceux qui commencent par 0, et, s'ils commencent par le même bit, c'est la comparaison non signée qui peut être utilisée.

Mais il y a quand même deux instructions assembleur distinctes pour la comparaison signée et pour la comparaison non signée.

II.5.3 Quelques exemples de calculs.

Pour l'addition et la soustraction, les opérations et les tests de validité des résultats sont les mêmes en arithmétique signée et non signée.

Pour la multiplication, l'instruction assembleur n'est pas la même (le dépassement de capacité doit être ignoré en a.s. dans le dernier exemple).

EXEMPLE 5.31. Premiers résultats, corrects :

Opération binaire	Entiers non signés	Entiers signés
0010	2	2
+ 1001	+ 9	+(-7)
1011	11	(-5)

EXEMPLE 5.32. Un résultat correct en arithmétique non signée, et négatif en arithmétique signée, mais correct modulo 16 (-6 et 10 sont dans la même classe, mais cette classe est représentée par 10 en a.n.s. et par -6 en a.s.) :

Opération binaire	Entiers non signés	Entiers signés
0100	4	4
+ 0110	+ 6	+ 6
1010	10	(-6)

EXEMPLE 5.33. Un dépassement de capacité dans les deux cas, mais le résultat est correct modulo 16 : les classes de 21, de -11 et de 5 sont les mêmes :

Opération binaire	Entiers non signés	Entiers signés
1100	12	(-4)
+ 1001	+ 9	+(-7)
(1)0101	5	5

Le résultat (correct modulo 16) est disponible dans tous les cas, les « dépassement de capacité » et « résultat négatif » sont signalés par le positionnement d'un bit dans un registre spécial.

EXEMPLE 5.34. Un résultat correct en a.n.s., résultat négatif en a.s., mais correct modulo 16 :

Opération binaire	Entiers non signés	Entiers signés
0101	5	5
× 0010	× 2	× 2
1010	10	(-6)

EXEMPLE 5.35. Dépassement de capacité dans les deux cas, résultat négatif en a.s., mais résultat correct modulo 16, compte tenu du choix des représentants dans les deux arithmétiques :

Opération binaire	Entiers non signés	Entiers signés
0101	5	5
× 0110	× 6	× 6
(1)1110	14	(-2)

EXEMPLE 5.36. Dépassement de capacité dans les deux cas, résultat correct en a.s., correct modulo 16 en a.n.s.

Opération binaire	Entiers non signés	Entiers signés
1101	13	(-3)
$\times$ 1110	$\times$ 14	$\times$ (-2)
(1011)0110	6	6

III Algorithmes d'Euclide et applications

III.1 PGCD de deux entiers

On a vu plus haut la justification de l'existence du PGCD de deux nombres strictement positifs par comparaison de leurs décompositions en facteurs premiers.

Par définition, le PGCD de a non nul avec 0 est a (défintion raisonnable, car 0 est divisible par tout entier non nul, donc par a, qui l'est aussi par a) et enfin le PGCD de 0 et de 0 n'est pas défini.

Il est possible de considérer des nombres négatifs (bien que ce soit sans grand intérêt dans les applications pratiques), mais le PGCD est celui des valeurs absolues.

L'algorithme consistant à comparer les décompositions en facteurs premiers n'est pas efficace, la découverte de diviseurs de nombres très grands est un problème difficile dont nous reparlerons plus loin.

III.2 Algorithme d'Euclide

III.2.1 Algorithme

On se limite ici au cas de deux entiers a et b strictement positifs.

Supposons par exemple $a > b$...

1. La division euclidienne de a par b peut s'écrire $a = bq + r$ avec $0 \leqslant r < b$.
2. Soit d un diviseur commun à a et b, qui peuvent alors s'écrire $a = da'$ et $b = db'$.
3. L'égalité $a = bq + r$ devient $da' = db'q + r$ ou encore $r = d(a' - b'q)$, donc d est aussi un diviseur commun à b et r.
4. Réciproquement, soit d un diviseur commun à b et r, qui peuvent alors s'écrire $b = db'$ et $r = dr'$ et l'égalité $a = bq + r$ devient $a = d(b'q + r')$.

   Donc d est un diviseur commun à a et b, et, par inclusion réciproque, les ensembles des diviseurs communs à a et b d'une part et à b et r d'autre part sont identiques.

   En particulier $a \wedge b = b \wedge r$.
5. Si $r = 0$, le $a \wedge b = b$, sinon on peut effectuer la division euclidienne de b par r, qui donne un reste r_1, tel que $r_1 < r$ et $b \wedge r = r \wedge r_1$.
6. Cet algorithme est itéré jusqu'à l'obtention d'un reste nul, ce qui se produit obligatoirement puisqu'il s'agit d'entiers et que la suite des restes ainsi construite est strictement décroissante.
7. Le PGCD est alors l'avant-dernier reste (le dernier non nul).

REMARQUE 5.8. Cet algorithme permet donc d'obtenir le PGCD de deux nombres sans connaître leurs décompositions en facteurs premiers.

III.2.2 Programmation

Voici sa programmation itérative en C :

```
int pgcd ( int a , int b ) {
    int r;
    while ( b != 0 ) {
        r = a % b;
        a = b;
        b = r;
    }
    return a;
}
```

(en toute rigueur, il faudrait vérifier que a et b sont bien positifs ; par ailleurs, cette fonction retourne 0 comme PGCD de 0 et de 0 : à vérifier avant l'appel).

Voici sa programmation récursive :

```
int pgcd ( int a , int b ) {
    if ( b == 0 )
        return a;
    else
        return pgcd ( b , a % b );
}
```

III.3 Théorème de Bézout

On considère deux nombres entiers strictement positifs a et b.

PROPRIÉTÉ 5.12 (THÉORÈME DE BÉZOUT) : Il existe un couple d'entiers u et v tels que $au-bv = d$, où d est le PGCD de a et de b.

PREUVE On peut se ramener au cas où $a \wedge b = 1$.

En effet, si $d > 1$, on peut écrire $a = a'd$ et $b = b'd$ avec $a' \wedge b' = 1$; si le théorème est établi dans le cas du PGCD égal à 1, on peut affirmer l'existence de u et de v tels que $a'u - b'v = 1$; en multipliant les deux membres de cette égalité par d, on obtient $a'du - b'dv = d$, soit $au - bv = d$.

Il suffit donc d'établir le théorème dans le cas où $d = 1$ (a et b premiers entre eux). Plaçons nous dans $(\mathbb{Z}/b\mathbb{Z})^*$ et considérons l'application de cet ensemble dans lui-même définie par $x \longmapsto ax$. Essayons de résoudre $ax = ax'$, soit $a(x - x') = 0$, soit encore $a(x - x') \equiv 0[b]$, ou finalement $a(x - x') = kb$, avec $k \in \mathbb{Z}$.

Comme $a \wedge b = 1$, a ne divise pas b, donc divise k ; on peut écrire $k = k'a$, il reste $x - x' = k'b$, donc $x \equiv x'[b]$, donc $x = x'$; finalement $ax = ax' \Rightarrow x = x'$, donc l'application envisagée est injective ; comme il s'agit d'un ensemble fini, elle est évidemment aussi surjective, donc il existe u tel que $au = 1$, ce qui s'écrit encore $au \equiv 1[b]$, ou encore $au = bv + 1$, finalement $au - bv = 1$. ■

REMARQUE 5.9. Ce couple n'est pas unique.

PREUVE En effet, si (u, v) est un couple de Bézout pour (a, b), donc tel que $au-bv = d$, où $d = a \wedge b$, alors, pout tout k dans $\mathbb{Z}$, $a(u + kb) - b(v + ka) = au - bv + kab - kab = au - bv = d$ aussi. ■

Exercice 5.37. *Montrez que, si m est multiple de deux nombres premiers entre eux a et b, alors m est multiple de ab.*

Réponse : $1 = aa' + bb'$, donc $m = maa' + mbb'$. Or $m = ax = by$, donc $m = ab(ya' + xb')$.

Exercice 5.38. *Montrez que, si on divise deux entiers naturels a et b par leur pgcd, alors les quotients obtenus sont premiers entre eux.*

Réciproquement, montrer que, si les quotients obtenus en divisant a et b par un diviseur commun d sont premiers entre eux, alors $d = pgcd(a, b)$.

Réponse : Soit $d = pgcd(a, b)$, et q_1 et q_2 les quotients de a et b par d. Alors $d = aa' + bb' = dq_1a' + dq_2b'$. Donc $1 = q_1a' + q_2b'$: q_1 et q_2 sont premiers entre eux. La réciproque est du même genre.

III.4 Algorithme d'Euclide généralisé

III.4.1 Idée de base.

Pour deux entiers positifs a et b, on a vu que l'algorithme d'Euclide s'écrit : $a \wedge b = b \wedge r$, où r est le reste dans la division euclidienne de a par b.

En supposant $a > b$, si on pose $a = r_0$ et $b = r_1$, on définit une famille finie $(r_0, r_1, \ldots, r_k, r_{k+1})$ par $r_i = q_{i+1}r_{i+1} + r_{i+2}$ (c'est-à-dire que r_{i+2} est le reste dans la division euclidienne de r_i par r_{i+1}).

Cette famille...
- est strictement décroissante,
- est telle que $r_{k+1} = 0$,
- vérifie $r_0 \wedge r_1 = r_1 \wedge r_2 = \ldots = r_{k-1} \wedge r_k = r_k \wedge r_{k+1} = r_k \wedge 0 = r_k$.

On remarque que r_{k-1} est un multiple de r_k, puisque la division euclidienne de r_{k-1} par r_k s'écrit $r_{k-1} = q_k r_k$.

Soit d le PGCD de a et de b (évidemment, $d = r_k$), on peut écrire $1 \times r_k - 0 \times r_{k-1} = d$ puis $1 \times r_{k-2} - q_{k-1} \times r_{k-1} = d$.

D'une manière générale, si (u, v) est un couple de Bézout pour r_{i+1} et r_{i+2}, soit $u \cdot r_{i+1} + v \cdot r_{i+2} = d$, comme $r_i = q_{i+1} \cdot r_{i+1} + r_{i+2}$, on a $u \cdot r_{i+1} + v \cdot (r_i - q_{i+1} \cdot r_{i+1}) = d$, soit $(u - q_{i+1} \cdot v) \cdot r_{i+1} + v \cdot r_i = d$.

III.4.2 L'algorithme.

Ceci donne l'idée de construire deux familles par les relations :
- $u_0 = 1$, $u_1 = 0$,$u_{i+2} = u_i - q_{i+1} \cdot u_{i+1}$
- $v_0 = 0$, $v_1 = 1$, $v_{i+2} = v_i - q_{i+1} \cdot v_{i+1}$.

C'est ce que l'on appelle algorithme d'Euclide généralisé. On a alors $(u_k, v_k, r_k) = (u, v, d)$, u et v tels que $a \cdot u + b \cdot v = d$.

PREUVE 3 : *Pour cela, il suffit de montrer par récurrence que* $\forall i \in \{0, \ldots, k\}, r_0 \cdot u_i + r_1 \cdot v_i = r_i$.
- *Initialisation de la récurrence : la relation est vraie pour* $i = 0$*, en effet* $r_0 \cdot u_0 + r_1 \cdot v_0 = r_0$*, puisque* $u_0 = 1$ *et* $v_0 = 0$.
- *Caractère héréditaire de la propriété : en supposant que* i *est un entier pour lequel* $r_0 \cdot u_i + r_1 \cdot v_i = r_i$ *et* $r_0 \cdot u_{i+1} + r_1 \cdot v_{i+1} = r_{i+1}$*, calculons* $r_0 \cdot u_{i+2} + r_1 \cdot v_{i+2} = r_0 \cdot (u_i - q_{i+1} \cdot u_{i+1}) + r_1 \cdot (v_i - q_{i+1} \cdot v_{i+1}) = r_0 \cdot u_i + r_1 \cdot v_i - q_{i+1} \cdot (r_0 \cdot u_{i+1} + r_1 \cdot v_{i+1}) = r_i - q_{i+1} \cdot r_{i+1} = r_{i+2}$. †

III.4.3 Exemple.

Illustrons la mise en œuvre de cet algorithme...

EXEMPLE 5.39. Soit à obtenir un couple de Bézout pour (23,17) :

(23,1,0)	(17,0,1)	$\longrightarrow$	$q = 1$
(17,0,1)	(6,1,-1)	$\longrightarrow$	$q = 2$
(6,1,-1)	(5,-2,3)	$\longrightarrow$	$q = 1$
(5,-2,3)	(1,3,-4)	$\longrightarrow$	$q = 5$
(1,3,-4)	(0,-17,23)	$\longrightarrow$	FIN

On a bien $3 \times 23 - 4 \times 17 = 1$.

REMARQUE 5.10. Il est possible d'obtenir -1 (ou $-d$ en général) comme résultat, donc $au - bv = -1$, cela dépend de la parité du nombre d'itérations effectuées dans l'algorithme précédent.

Ce n'est pas un résultat faux, puisqu'alors $bv - au = 1$ et qu'on a quand même un couple de Bézout pour (b, a).

S'il est nécessaire d'obtenir un couple (u, v) tel que $au - bv = 1$ et où a et b figurent dans cet ordre, et que l'algorithme a fourni un couple (u', v') tel que $bv' - au' = 1$, il suffit de prendre $u = b - u'$ et $v = a - v'$ et, dans ces conditions $au - bv = a(b - u') - b(a - v') = ab - au' - ab + bv' = bv' - au' = 1$.

Exercice 5.40. *Exprimer $pgcd(1330, 602)$ comme combinaison à coefficients entiers des nombres 1330 et 602.*

Réponse $14 = 1330 * (-19) + 602 * 42$.

Fin du Chapitre

Chapitre 6

Représentation des nombres réels en machine

I Introduction

Pour des raisons évidentes, il est impossible de représenter exactement en machine un nombre réel dont le développement binaire, et a fortiori décimal, est infini.

EXEMPLE 6.1. Par exemple 1/3, mais aussi 1/10 (dont le développement décimal 0,1 est fini, mais pas le développement binaire) ne sont pas exactement représentables.

Cette limitation interdit la représentation de tout nombre irrationnel, dont le développement est toujours infini et non périodique. On ne peut donc représenter que :
– des nombres rationnels,
– et, parmi ceux-ci, seuls ceux qui admettent un développement binaire fini et « pas trop long », c'est-à-dire, au total, un nombre fini de nombres rationnels.

La représentation généralement adoptée est la représentation dite « en virgule flottante », parce qu'elle permet de traiter de manière à peu près satisfaisante les opérations sur deux opérandes de grandeurs très différentes.

EXEMPLE 6.2. En « virgule fixe », les limitations physiques des machines interdiraient de représenter simultanément, par exemple, 10^{100} et 10^{-100} , alors que la représentation en virgule flottante le permet.

Bien entendu, l'addition de ces deux nombres donnera le premier (exactement) comme résultat, ce qui n'est pas gênant, mais leur multiplication donnera bien 1 comme résultat (aux erreurs de représentation et de calcul près, car aucun de ces deux nombres n'est représentable exactement en machine).

II Les formats IEEE

II.1 La norme IEEE 754

La représentation des nombres réels en machine (en « virgule flottante ») fait l'objet d'une norme (norme IEEE 754).

Cette norme reconnaît trois formats :
– « single » (réel représenté sur 32 bits),
– « double » (64 bits),
– « extended » (80 bits).

Les formats « single » et « double » sont analogues, à la taille des diverses composantes près.

Cette même norme prévoit un certain nombre de spécifications qui concernent les calculs sur les réels représentés (indépendamment du format retenu) : aucune opération sur les réels ne doit provoquer, par elle-même, d'interruption du déroulement normal du programme. Ni une division par 0, ni un dépassement de capacité, ni une tentative de calcul impossible.

C'est au logiciel qui gouverne les calculs de vérifier le résultat et de provoquer, s'il le juge utile (et c'est ce que font en général les compilateurs), une interruption.

Néanmoins, il a fallu prévoir des représentations spéciales pour ces cas particuliers :
– l'une s'appelle « INF » (infini),
– l'autre « NAN » (abréviation pour « not a number »).

EXEMPLE 6.3. D'après ces spécifications, le résultat de $1/0$ (en réels) doit être INF, celui de $0/0$ doit être NAN.

On doit obtenir aussi $\sqrt{-1} = \mathrm{NAN}$, $\ln 0 = -\mathrm{INF}$, $1/\mathrm{INF} = 0$, mais $\mathrm{INF}/\mathrm{INF} = \mathrm{NAN}$, de même que toute opération dont l'un des opérandes est NAN, par exemple $\sin(\mathrm{NAN}) = \mathrm{NAN}$, $1 + \mathrm{NAN} = \mathrm{NAN}$...

REMARQUE 6.1. Si vous voulez observer ces résultats, effectifs, vous serez obligés d'opérer depuis l'assembleur, aucun compilateur ne vous autorisera à tenter d'obtenir de pareilles horreurs !

Dans la représentation d'un nombre réel, on numérotera les bits à partir de 0 et à partir de la droite (bit « le moins significatif ») jusqu'à (respectivement) 31, 63 ou 79 (bit « dominant »).

II.2 Format « single »

s	e (8 bits)	m (23 bits)

On retrouve la valeur du réel x représenté de la manière suivante :

– si $0 < e < 255$, alors $x = (-1)^s \cdot 2^{e-127} \cdot (1, m)$
– si $e = 0$ et $m \neq 0$, alors $x = (-1)^s \cdot 2^{-126} \cdot (0, m)$
– si $e = 0$ et $m = 0$, alors $x = 0$
– si $e = 255$ et $m = 0$, alors $x = (-1)^s \cdot INF$
– si $e = 255$ et $m \neq 0$, alors x est un NAN
  Comme, dans ce cas, m peut prendre n'importe quelle valeur non nulle, il est possible de conserver de cette manière un code qui permet de reconnaître l'origine de l'erreur.

II.3 Format « double »

s	e (11 bits)	m (52 bits)

On retrouve la valeur du réel x représenté de la manière suivante :

– si $0 < e < 2047$, alors $x = (-1)^s \cdot 2^{e-1023} \cdot (1, m)$
– si $e = 0$ et $m \neq 0$, alors $x = (-1)^s \cdot 2^{-1022} \cdot (0, m)$
– si $e = 0$ et $m = 0$, alors $x = 0$
– si $e = 2047$ et $m = 0$, alors $x = (-1)^s \cdot INF$
– si $e = 2047$ et $m \neq 0$, alors x est un NAN

II.4 Format « extended »

s	e (15 bits)	i	m (63 bits)

On retrouve la valeur du réel x représenté de la manière suivante :

- si $0 \leqslant e < 32767$, alors $x = (-1)^s \cdot 2^{e-16383} \cdot (i, m)$
- si $e = 32767$ et $m = 0$, alors $x = (-1)^s \cdot INF$ (quelle que soit la valeur de i)
- si $e = 32767$ et $m \neq 0$, alors x est un NAN (quelle que soit la valeur de i)

II.5 D'une manière générale...

1. s, représenté sur 1 bit, est le signe du nombre (0 pour +, 1 pour -)

2. e est l'« exposant biaisé », *i.e.* l'exposant translaté.
   Cette translation a été introduite de manière à faciliter la comparaison des réels représentés entre eux :
   - Pour deux réels positifs non nuls, le plus grand est évidemment celui qui a le plus grand exposant (s'ils ont le même, on compare alors les « mantisses » m).
   - Or, la représentation ordinaire dans les formats « single » et « double » ne permet pas la représentation de 0 : $[x = (-1)^s \cdot 2^{e-t} \cdot (1, m)]$ ne peut pas être nul, même si m et $e - t$ sont nuls, auquel cas on obtient 1 (ou -1).
   - Par ailleurs, comme 0 est le plus petit réel positif, il est logique de lui attribuer le plus petit exposant (c'est-à-dire -128 ou -1024), et de lui attribuer évidemment une « mantisse » nulle.
   - Mais il est plus simple (pour les tests) que 0 possède un exposant nul, ce qui oblige à rendre tous les autres exposants positifs par la translation indiquée.
   - Pour retrouver le véritable exposant, il faut donc retrancher cette quantité à l'exposant de la représentation.

3. La notation $1, m$ (ou $0, m$ ou i, m) signifie que le nombre entier m doit être considéré comme la partie fractionnaire d'un nombre dont la représentation binaire a pour partie entière 1 (ou 0 ou i).

4. Pour les formats « single » et « double », la formule à appliquer est différente dans le cas où l'exposant e est nul.
   Il s'agit de ce que l'on appelle un « réel dénormalisé », introduit pour le motif suivant :
   - les chiffres significatifs du réel représenté sont contenus dans la mantisse ;
   - celle-ci est de longueur fixe pour un format donné,
   - donc, quel que soit l'ordre de grandeur du réel, sa représentation est obtenue avec la même précision relative, ce qui permet de connaître la précision du résultat d'un calcul.

   Cette mantisse $(1, m)$ représente un nombre compris entre 1 (inclus, si $m = 0$) et 2 (exclu).

   On obtient ce nombre en multipliant ou en divisant le réel à représenter par 2 jusqu'à ce que le résultat soit compris entre 1 et 2. Le nombre d'opérations effectuées donne l'exposant (le vrai, négatif dans le cas de multiplications, positif dans le cas de divisions).

   Pour des nombres réels trop petits, l'exposant peut alors être lui-même trop petit pour être représentable dans la plage qui lui est fixée. On admet alors que, pour la plus petite valeur de l'exposant (« biaisé »), c'est-à-dire 0, la mantisse est à interpréter sous la forme $0, m$, ce qui permet de représenter encore quelques réels trop petits pour être représentés dans la représentation normalisée (les Anglo-Saxons parlent de « progressive underflow »).

   Ces réels « dénormalisés » sont distingués des autres, parce qu'ils sont représentés avec une précision moindre (la mantisse a moins de 52 chiffres binaires significatifs).

Autrement dit, la précision d'un calcul qui utilise un réel dénormalisé n'est plus assurée, mais le risque d'une division par 0 (alors que le « vrai » nombre n'est pas nul) est diminué.

5. La distinction entre réels « normalisés » et « dénormalisés » disparaît dans le format « extended », puisque la partie entière de la mantisse y figure explicitement (le bit i, valeur 0 ou 1).
   L'inconvénient est qu'il existe alors plusieurs représentations possibles pour un même nombre réel.

   EXEMPLE 6.4. 1 peut être représenté par $i = 1$, $m = 0$, $e = 16383$, mais aussi par $i = 0$, $m = 100....0$, $e = 16384$, ou encore $i = 0$, $m = 010...0$, $e = 16385$, etc.

   Mais il est clair que, pour la précision d'un calcul, il vaut mieux utiliser tous les bits disponibles dans la mantisse (pour avoir le maximum de chiffres significatifs).
   C'est-à-dire qu'il faut choisir, parmi toutes les représentations possibles pour un nombre réel, celle pour laquelle $i = 1$: c'est ce que fait la machine (que les opérations soient implantées logiciellement ou effectuées par un coprocesseur arithmétique).
   Autrement dit, on réservera la valeur 0 pour i au cas où l'exposant « biaisé » est nul, comme pour les autres formats, et le problème de la précision se pose de la même manière.

II.6 Format « extended » des microprocesseurs.

Dans un langage tel que C (ou java),
– le format « single » est obtenu avec les valeurs de type « float »,
– le format « double » est disponible dans les valeurs de type « double »,
– le format « extended » dans les valeurs de type « long double ».

Pour être complet, il est nécessaire de préciser que les microprocesseurs modernes possèdent presque tous, intégrée, une unité de calcul spécialisée dans le calcul sur les réels, ayant des registres de taille adaptée à la représentation de ces nombres (donc plus longs que les registres de l'unité de calcul arithmétique et logique principale).

Plus anciennement, ce rôle était confié à une unité externe que l'on appelait « coprocesseur arithmétique » et qui était quelquefois optionnelle.
Si, dans ce cas, l'option n'avait pas été retenue, les opérations sur les réels étaient implémentées logiciellement au prix d'un dramatique allongement des temps de calcul.

Toujours est-il que les formats disponibles dans une unité de calcul sur les flottants dépendent de la taille des registres, et ceux-ci sont parfois limités à 64 bits, ce qui interdit le format « extended » en natif sur la machine (si le langage de programmation utilisé y donne accès, les opérations sont alors implémentées logiciellement).

Lorsque le format « extended » est disponible en natif dans la machine, les registres sont en général de taille 96 bits (et non 80).
Les 16 bits supplémentaires, s'ils sont évidemment utilisés par le processeur pour sa cuisine interne, ne sont jamais significatifs dans les résultats accessibles à l'utilisateur, et sont mis à 0.
Autrement dit, on obtient le réel au format « extended » en supprimant les 16 bits nuls.

III Réels représentables et précision

Tous les réels normalisés représentés en machine comportent le même nombre de chiffres binaires significatifs (dans un format donné).

Comme deux nombres dont les expressions binaires comportent le même nombre de chiffres n'ont pas nécessairement le même nombre de chiffre en représentation décimale, le nombre de chiffres significatifs en base 10 peut varier d'une unité.

EXEMPLE 6.5. 1000 en base 2 est 8 en décimal : 4 chiffres binaires, 1 chiffre décimal, 1100 binaire est 12 décimal : 4 chiffres binaires, 2 chiffres décimaux.

Ainsi,
- en format « single », on a 6 ou 7 chiffres significatifs,
- en format « double », 15 ou 16 chiffres,
- en format « extended », 19 ou 20.

Une telle précision peut sembler totalement superflue : elle est cependant largement insuffisante pour, par exemple, les calculs en astronomie (trajectoires de satellites, etc.), pour lesquels il est nécessaire de faire appel à des précisions nettement supérieures...

Le plus grand nombre réel représentable en format « single » est tel que
- $e = 254$, donc le véritable exposant est $254 - 127 = 127$
- m est constitué de 23 « 1 » , la mantisse a donc pour valeur $1,1\ldots1$, c'est-à-dire $1 + 2^{-1} + 2^{-2} + 2^{-3} + \cdots + 2^{-23}$ (somme d'une progression géométrique de raison 1/2, donc) $= 2 - 2^{-23}$.
- Il vaut donc exactement $2^{127}(2-2^{-23}) = 2^{128}-2^{104}$, c'est à dire approximativement $3,403\cdot10^{38}$.

Le plus petit réel positif normalisé (« single ») est tel que
- $e = 1$, donc le véritable exposant est $1 - 127 = -126$
- $m = 0$, donc la mantisse vaut 1
- Il vaut donc exactement 2^{-126} c'est-à-dire approximativement $1,175 \cdot 10^{-38}$.

Le plus petit réel positif dénormalisé (« single ») est tel que
- $e = 0$, donc le véritable exposant est -126
- $m = 0\ldots01$, donc la mantisse vaut $0,0\ldots01$, soit 2^{-23}
- Il vaut donc exactement 2^{-149} c'est-à-dire approximativement $1,401 \cdot 10^{-45}$.

En format « double », les nombres correspondants sont $1,7 \cdot 10^{308}$, $2,3 \cdot 10^{-308}$, $5 \cdot 10^{-324}$.

En format « extended », les nombres correspondants sont $1,1 \cdot 10^{4932}$, $1,7 \cdot 10^{-4932}$, $1,9 \cdot 10^{-4951}$.

Les réels qui sont représentés en machine sont exacts ; par contre, tous les nombres réels ne sont pas représentables (la représentation est évidemment discrète).

EXEMPLE 6.6. Considérons le réel 1 en format « extended » : « vrai exposant » : 0, $s = 0$, $i = 1$, $m = 0$, donc $e = 16383$, c'est-à-dire (en hexadécimal) :
- 3FFF pour les 16 premiers bits,
- 8000 hexadecimal pour les 16 suivants,
- et tous les derniers sont nuls,

donc : 3FFF 8000 0000 0000 0000.

Le réel représentable en machine, supérieur à 1 et le plus proche de 1, a évidemment un m égal à $0\ldots01$, il s'agit donc de 3FFF 8000 0000 0000 0001.

La différence des mantisses (i, m) de ces deux nombres est $0,0\ldots01$, soit (exactement) 2^{-63}, ou encore (environ) $1,08 \cdot 10^{-19}$.

En d'autres termes...
- Dans l'intervalle $[1, 2[$, les réels représentables varient de 2^{-63} en 2^{-63}.
- Dans l'intervalle $[2, 4[$, les réels représentables varient de 2^{-62} en 2^{-62}.
- etc.
- Dans l'intervalle $[2^{63}, 2^{64}[$, les réels représentables varient de 2^0 en 2^0, donc d'unité en unité : on ne peut plus représenter que des nombres entiers, mais il s'agit d'entiers qui sont plus grands que les entiers de la machine (sur 32 bits seulement).
- Dans l'intervalle suivant, on ne peut plus représenter tous les entiers, on n'en représente plus qu'un sur deux, puis un sur quatre, etc.

REMARQUE 6.2. Les calculs sur les réels en machine sont exacts dans le sens suivant :

Si, par exemple, on additionne, soustrait ou multiplie deux entiers représentables sous forme de réels en machine, et si le résultat est aussi représentable sous forme de réel en machine, alors ce résultat est exact.

Autrement dit, il s'agit encore d'un entier exactement.

Le problème de la division est différent, parce que c'est évidemment l'algorithme de la division des réels qui est appliqué et non celui de la division euclidienne des entiers.

EXEMPLE 6.7. Soit à représenter $8,5$ sous forme de réel double précision (format « double »)

1. Le ramener entre 1 et 2 par divisions par 2 : $8,5 = 8 \times 1,0625$.
2. L'exposant est donc 3, et $e = 3 + 1023 = 1026$, les 12 premiers bits sont donc $0100\,0000\,0010$ (en effet, $1026 = 1024 + 2$, et 1024 est 2^{10}, dont l'écriture binaire est « 1 » suivi de 10 « 0 » . On rajoute 2, soit 10 binaire).
3. $1,0625 = 1 + 0,0625$, et $0,0625 = 2^{-4}$, soit (en binaire) $0,0001$, donc $m = 0001\,00\ldots$

On obtient : 0100 0000 0010 0001 0000 0000 . . ., soit 4021 0000 0000 0000 en hexadécimal.

EXEMPLE 6.8. Soit à représenter $0,1$ sous forme de réel double précision

1. Le ramener entre 1 et 2 par multiplications par 2 : $16 \times 0,1 = 1,6$
2. L'exposant est donc -4, et $e = -4 + 1023 = 1019$, les 12 premiers bits sont (comme $s = 0$) 0011 1111 1011 (3FB hexadécimal)
3. $1,6 = 1 + 0,6$.

   Pour obtenir la représentation binaire de $0,6$, il faut effectuer la division de 6 par 10 en base 2, ou, mieux, celle de 3 par 5 (donc 11 par 101 en base 2).

   On obtient : 0,1001 1001 1001 . . ..

   Ce développement binaire est infini mais périodique, il suffit de le tronquer à 52 chiffres et d'effectuer l'arrondi.

   Les 4 derniers bits sont 1001 et le suivant serait 1, donc l'arrondi est fait à 1010 pour les 4 derniers, soit A hexadécimal (la représentation n'est donc pas exacte, on l'a signalé plus haut).

   Les précédents (par groupe de 4) sont égaux à 1001, soit 9 hexadécimal ; on obtient finalement : 3FB9 9999 9999 999A.

Attention : en « single » , la représentation de 0,1 serait 3DCC CCCD et, en « extended » : 3FFB CCCC CCCC CCCC CCCD (faites-le aussi !).

Autrement dit : le passage d'un format à l'autre n'est pas évident en hexadécimal (il ne suffit pas de « raccourcir » ou de « rallonger » !).

Fin du Chapitre

Chapitre 7

Cryptologie et arithmétique.

I Méthodes de cryptage « à clé publique »

I.1 Principe

Supposons qu'un individu A soit obligé de transmettre à un autre individu B un message M en utilisant un réseau de communication public, par exemple les ondes hertziennes. N'importe quel individu peut se mettre à l'écoute et intercepter le message.

Le problème est donc :
- le message doit être inintelligible pour tout individu autre que A et B.
- B doit pouvoir le comprendre.
- B doit pouvoir s'assurer que le message provient bien de A (et non d'un plaisantin quelconque).

L'idée est de doter tous les participants de la même méthode de cryptage. Les résultats du cryptage d'un même message par divers individus sont cependant différents, car chacun d'entre eux emploie une « clé » qui lui est propre.

EXEMPLE 7.1. Lorsque l'on remplace 'a' par 'c', 'b' par 'd', etc..., la méthode de cryptage est « décalage des lettres de l'alphabet » et la clé est la longueur du décalage, ici 2.

La méthode de cryptage est fondée sur l'existence de fonctions f, dépendant d'un paramètre (la « clé »), inversibles, mais pour lesquelles la détermination de l'inverse est matériellement impossible, en l'état actuel des connaissances humaines.

Soit f_A la fonction de cryptage qui utilise la clé propre à l'individu A.
- La clé de A est publique, ainsi n'importe qui est en mesure d'appliquer la fonction f_A à un message M quelconque.
- Par contre, seul A connaît la fonction inverse f_A^{-1} qui permet de retrouver le message initial.

Au message M , A applique en fait f_A^{-1} (il est le seul à pouvoir le faire).
Puis, à ce message $f_A^{-1}(M)$, il applique la fonction de cryptage de B, soit f_B (il peut le faire, la clé de B est publique), pour obtenir $f_B \circ f_A^{-1}(M)$, incompréhensible car les clés sont évidemment uniques, et donc $f_B \circ f_A^{-1}$ n'est pas l'identité.

C'est ce message « doublement » crypté qui est envoyé. B le reçoit et lui applique aussitôt f_B^{-1}, ce qu'il est le seul à pouvoir faire, pour obtenir $f_A^{-1}(M)$, auquel il applique f_A : si le résultat est compréhensible, B est sûr que le message lui était bien destiné, et qu'il a bien été envoyé par A.

I.2 Utilisation de l'indicatrice d'Euler

Exercice 7.2 (Fonction indicatrice d'Euler). *Soit n un entier strictement positif ; on note $\varphi(n)$ le nombre des entiers inférieurs à n qui sont premiers avec n.*

L'application de $\mathbb{N}^*$ dans $\mathbb{N}^*$ ainsi définie est appelée fonction indicatrice d'Euler.

1. *Montrer que, pour p premier, $\varphi(p) = p - 1$*
2. *Montrer que, pour p premier, $\varphi(p^\alpha) = p^\alpha - p^{\alpha-1}$*
3. *On considère les nombres de la forme $ap + bq$, pour p et q premiers entre eux et l'application de $(\mathbb{Z}/q\mathbb{Z}) \times (\mathbb{Z}/p\mathbb{Z})$ dans $(\mathbb{Z}/pq\mathbb{Z})$ définie par $(a, b) \longmapsto ap+bq\ [pq]$; montrer que cette application est injective et surjective.*
4. *En déduire (en utilisant les nombres de la forme $ap + bq\ [pq]$) que, pour p et q premiers entre eux, $\varphi(pq) = \varphi(p)\varphi(q)$.*
5. *Utiliser le résultat précédent et le théorème de Fermat ci-dessus pour prouver que, pour tout entier a premier avec n, et pour tout entier positif n dépourvu de facteur carré, $a^{\varphi(n)} \equiv 1\ [n]$.*

I.2.1 Résultat de base

Diverses fonctions « à inverse difficile à déterminer » ont été proposées. Les plus satisfaisantes sont celles qui utilisent le résultat suivant :

PROPRIÉTÉ 7.1 : s'il est très facile d'obtenir un très grand nombre entier composé par produit de deux nombres premiers eux-mêmes grands, la décomposition en facteurs premiers d'un nombre composé est très difficile.

I.2.2 Méthode de cryptage

La méthode de cryptage est la suivante :

1. Soit donc $n = pq$ un entier, produit de deux nombres entiers premiers, par exemple tels que $p \equiv q \equiv 2[3]$.
2. Soit M le message, préalablement chiffré (sans précautions particulières, par exemple en remplaçant les lettres par leurs codes ASCII).
3. Si $M \geqslant n$, on décompose M en plusieurs sous-messages, ses « chiffres » en base n, par exemple.
4. Si n est la clé choisie par A, et pour $M < n$, $f_A(M) = C$, avec $C \equiv M^3\ [n]$. Comme n est connu de tous, n'importe qui peut calculer C très rapidement. Par contre, les facteurs premiers p et q de n sont soigneusement tenus secrets par A.
5. Un résultat (élémentaire) d'arithmétique indique que, comme n n'a pas de facteur carré, si M est premier avec n, alors $M^{\varphi(n)} \equiv 1\ [n]$ (dans cette expression, ϕ est la fonction indicatrice d'Euler, c'est-à-dire que $\varphi(n)$ est le nombre de nombres strictement positifs inférieurs à n qui sont premiers avec n).
6. Un autre résultat (élémentaire) d'arithmétique dit que, comme $n = pq$, avec p et q premiers, $\varphi(n) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$.
7. On a donc, en combinant ces deux résultats, $M^{(p-1)(q-1)} \equiv 1\ [n]$, donc $M^{2(p-1)(q-1)} \equiv 1\ [n]$, et finalement $M^{2(p-1)(q-1)+1} \equiv M\ [n]$.
8. Comme on a choisi $p \equiv q \equiv 2\ [3]$, $(p-1)(q-1) \equiv 1\ [3]$, $2(p-1)(q-1) \equiv 2\ [3]$ et $2(p-1)(q-1)+1 \equiv 0\ [3]$. Il s'agit donc d'un multiple de 3, on peut poser $2(p-1)(q-1)+1 = 3k$, et on a $M^{3k} \equiv M\ [n]$.

9. Or $M^{3k} = (M^3)^k$, donc, si le message crypté est $C \equiv M^3\ [n]$, $C^k \equiv M\ [n]$ et la connaissance de $k = \dfrac{2(p-1)(q-1)+1}{3}$ permet de retrouver le message original.

EXEMPLE 7.3. Avec $p = 5$, $q = 11$, $n = pq = 55$.
Le message à envoyer est chiffré $M = 7$.
Alors $7^2 \equiv 49\ [55]$, $7^3 \equiv 13\ [55]$.
Le message crypté est $C = 13$.
Ici $k = \dfrac{2 \times 4 \times 10 + 1}{3} = 27$, donc $M \equiv 13^{27}\ [55]$.
On a $13^{27} = 13^{16+8+2+1}$, or $13^2 \equiv 4\ [55]$, $13^4 \equiv 4 \times 4 \equiv 16\ [55]$, $13^8 \equiv 16 \times 16 \equiv 256 \equiv 36\ [55]$ $13^{16} \equiv 36 \times 36 \equiv 1296 \equiv 21\ [55]$, donc $13^3 \equiv 4 \times 13 \equiv 52\ [55]$, $13^{11} \equiv 52 \times 36 \equiv 37\ [55]$, $13^{27} \equiv 37 \times 21 \equiv 7\ [55]$.

Si, par malchance, M est un multiple de p ou de q, il suffit de modifier légèrement le premier chiffrage, par exemple en introduisant un espace supplémentaire dans les caractères du message d'origine, ce qui ne modifie pas son sens. Ne pas oublier cette précaution indispensable.

II Choix d'un nombre n

Dans l'exemple ci-dessus, le cryptage est immédiatement percé à jour, puisque la décomposition de 55 en ses facteurs premiers 5 et 11 est immédiate. On peut en dire autant de tout entier représentable sur 32 bits.

Il faut aller chercher bien plus loin pour assurer un minimum de sécurité. Pour fixer les idées, les clés utilisées sont à l'heure actuelle le produit de deux nombres qui ont entre 100 et 200 chiffres dans leur représentation décimale.

II.1 Nombres premiers

Pour produire un nombre n utilisable, il faut tout d'abord trouver deux nombres p et q premiers, suffisamment grands.

On choisit deux nombres se terminant par 1, 3, 7 ou 9 dans leur représentation décimale et de longueurs comparables (mais pas trop proches : il existe un algorithme de décomposition qui est capable de décomposer rapidement un nombre qui est le produit de deux nombres de longueurs très proches).

Il faut vérifier qu'ils sont premiers et, pour cela, disposer d'un critère de primalité (voir plus loin).

Lorsque le nombre produit au hasard n'est pas premier, il suffit de lui ajouter 2 , puis encore 2 etc., jusqu'à obtenir un nombre premier, ce qui interviendra très rapidement.

Avec ces deux nombres premiers p et q ainsi obtenus, on obtient la clé n.

II.2 Décomposition en facteurs premiers

Théoriquement, bien sûr, la décomposition d'un nombre composé (non premier) est un problème résolu : il suffit de tenter de le diviser par tous les nombres premiers jusqu'à sa racine carrée.

Pratiquement, cet algorithme est totalement impraticable dès que la longueur du nombre dépasse une vingtaine de chiffres décimaux (durée d'exécution trop élevée).

La durée d'exécution d'un algorithme de décomposition en facteurs premiers dépend, bien sûr, de la longueur du nombre à décomposer. Mais il n'y a pas proportionnalité stricte : cela dépend aussi de l'algorithme utilisé. Pour prendre un exemple limite, 1000 !, qui est un nombre dont la représentation décimale occupe plus de 4000 chiffres, est décomposé en quelques fractions de seconde par le plus rudimentaire des algorithmes.

Le seul moyen, donc, pour savoir si un nombre n obtenu comme ci-dessus est une « bonne » clé, est de tenter de le décomposer par tous les algorithmes connus. S'il résiste vaillamment, on peut l'adopter, sinon, il faut en changer.

La conclusion de cette présentation est qu'il est donc nécessaire de disposer d'un test de primalité et d'algorithmes de décomposition en facteurs premiers, questions que nous allons aborder dans les paragraphes suivants.

Chapitre 8

Tests de primalité

I Théorème de Fermat

PROPRIÉTÉ 8.1 (PETIT THÉORÈME DE FERMAT) : Si n est premier et si $a \neq 0$, $a^n \equiv 1\ [n]$.

Exercice 8.1 (Théorème de Fermat). *Soit n un nombre premier,*

1. *montrer que, pour p entier tel que $0 < p < n$, n divise C_n^p*
2. *montrer que, pour tout $a \in \mathbb{N}$, $(a+1)^n - a^n - 1$ est divisible par n.*
3. *montrer que, pour tout $b \in \mathbb{N}$, si $b^n - b$ est divisible par n, $(b+1)^n - (b+1)$ l'est aussi.*
4. *En déduire le théorème de Fermat : pour n premier et $a \in \mathbb{N}$, $a^n \equiv a\ [n]$.*

Exercice 8.2 (Théorème de Wilson). *Soit p un nombre entier strictement supérieur à 1.*
$(p-1)! + 1$ est divisible par p si et seulement si p est premier.
On demande la démonstration de ce théorème.

Ce théorème ne peut servir de test de primalité, mais seulement de test de non-primalité. C'est-à-dire que si l'on trouve un nombre $a \not\equiv 0\ [n]$ tel que $a^{n-1} \not\equiv 1\ [n]$, on en conclut que n est composé.

Les nombres a tels que $a^{n-1} \equiv 1\ [n]$ alors que n n'est pas premier ne sont pas nombreux. C'est pourquoi si, après l'essai de quelques valeurs de a, on trouve toujours $a^{n-1} \equiv 1\ [n]$, ce nombre n sera envoyé à un véritable test de primalité.

Ce pré-test a l'avantage d'être simple et rapide.

II Test de Miller-Rabin

Soit n un nombre impair, que l'on met sous la forme $n - 1 = 2^t m$, avec m impair.

DÉFINITION 8.1 (NOMBRE PSEUDO-PREMIER FORT). *Ce nombre n est dit* pseudo-premier fort dans la base a *si l'on peut trouver a tel que :*

- *ou bien $a^m \equiv 1\ [n]$,*
- *ou bien on peut trouver u tel que $0 \leqslant u \leqslant t-1$, $a^{2^u m} \equiv -1\ [n]$.* ♢

On montre que :

PROPRIÉTÉ 8.2 : Tout nombre premier est pseudo-premier fort dans n'importe quelle base et qu'un nombre composé est pseudo-premier fort dans au plus $\frac{n}{4}$ bases différentes, et « en général » aucune.

Bien entendu, dès que n est un tant soit peu grand, il est exclu de tester autant de bases.

Il n'en reste pas moins que si, après une dizaine de bases, n est pseudo-premier fort dans chacune de ces bases, il a de « très bonnes chances » d'être premier.

Ce test n'est cependant pas, lui non plus, un véritable test de primalité, mais il est presque aussi rapide que celui de Fermat, et il sert d'aiguillage entre les nombres que l'on enverra à un algorithme de décomposition et ceux que l'on enverra plutôt à un véritable test de primalité.

III Tests de Lucas, Selfridge et Pocklington

Le test de Lucas peut s'exprimer de la manière suivante :

PROPRIÉTÉ 8.3 (TEST DE LUCAS) : Si on peut trouver un entier a pour lequel $a^{n-1} \equiv 1\ [n]$, mais $a^{\frac{n-1}{q}} \not\equiv 1\ [n]$ pour tous les diviseurs premiers q de $n-1$, alors n est premier.

REMARQUE 8.1. Selfridge a montré qu'il n'était pas nécessaire d'utiliser la même valeur de a pour tous ces diviseurs.

Ce test est théoriquement satisfaisant (c'est un test qui peut répondre : « oui, n est premier »), pratiquement il l'est beaucoup moins : il exige la décomposition en facteurs premiers de $n-1$ qui est une opération en général longue et difficile (voir les algorithmes qui suivent).

De plus, il connaît un cas d'échec, dans lequel il ne donne pas de réponse.

Le critère de Pocklington permet d'atténuer cette difficulté :

PROPRIÉTÉ 8.4 (CRITÈRE DE POCKLINGTON) : Si n n'est que « partiellement décomposé », dans le sens où il a été mis sous la forme $n = FR$, où F est totalement décomposé en facteurs premiers, mais R n'est pas premier, alors :
- si le critère de Selfridge appliqué aux diviseurs premiers de F aboutit à un succès,
- et si $F > R$,

alors n est premier.

Fin du Chapitre

Chapitre 9

Décomposition en facteurs premiers

I Divisions successives

L'algorithme est très simple : tenter de diviser le nombre par les nombres premiers successifs, dont on dispose dans un tableau.

Cet algorithme n'est pas efficace, mais il est nécessaire d'en disposer : tous les autres algorithmes, conçus pour trouver de « grands » diviseurs, connaissent des cas d'échec, qui sont d'autant plus fréquents que les diviseurs sont petits.

Avant d'envoyer un nombre à un autre algorithme, il est donc indispensable de l'avoir débarrassé de ses « petits » diviseurs. Pour fixer les idées, il s'agit des nombres premiers représentables sur 16 bits, jusqu'à 65 535 ($= 2^{16} - 1$) (le plus grand est 65 521, et il y en a au total 6 542).

Cette première phase de la décomposition nécessite en général un temps si faible qu'il n'est pas mesurable.

REMARQUE 9.1. On pourrait alors envisager aussi d'aller plus loin, c'est-à-dire, par exemple, de tenter la division par tous les nombres premiers représentables sur 32 bits (c'est-à-dire inférieurs à $2^{32} = 4\,294\,967\,296$: 10 chiffres « seulement » !).
Il faut alors savoir qu'il y en a 203 280 221 et que la manière la plus économique de les stocker nécessite environ 194 Mo...

On n'ose évoquer le temps d'exécution d'un algorithme qui parcourrait un tel tableau... pour ne même pas obtenir de résultat, ce qui est le cas dès que le plus petit diviseur du nombre à décomposer possède plus de 10 chiffres 'ce qui est fort peu pour des nombres qui dépassent les 100 chiffres décimaux).
Il faut bien saisir sur ces exemples l'ampleur du problème !

II Algorithme de Monte-Carlo (1975)

II.1 Présentation

Cet algorithme, dont l'efficacité est tout-à-fait surprenante, utilise un générateur de nombres au hasard (c'est de l'intervention de ce « hasard » que l'algorithme tire son nom).

Soit
- f cette fonction (le générateur),
- A la valeur d'initialisation,
- n le nombre à décomposer,

– p un de ses facteurs premiers.

On considère les suites de nombres entiers définies par

$$x_0 = y_0 = A, x_{m+1} = f(x_m)[n], y_{m+1} = f(y_m)[p]$$

REMARQUE 9.2. On ne connaît pas p, bien sûr, mais on sait que $y_m = x_m[p]$, et cela suffit.

Soit h la plus grande puissance de 2 qui est inférieure ou égale à m (par exemple, pour $m = 50, h = 32$). On peut alors montrer qu'il existe un entier m tel que $y_m = y_{h-1}$, c'est-à-dire $x_m - x_{h-1} \equiv 0[p]$.

C'est donc un multiple de p, qu'on pourra obtenir en calculant le PGCD de ce nombre avec n.

II.2 L'algorithme

L'algorithme peut se décrire dans les termes suivants :

```
Initialisations : n au nombre à factoriser
x à 5, x' à 2, k à 1, h à 1, g à 1.

TANT QUE (n n'est pas premier) ET QUE (g est différent de n)
    FAIRE
        REPETER
            g ⟵ (x − x') ∧ n
            SI ( g est différent de 1) ALORS
                SI ( g est différent de n) ALORS
                    IMPRIMER g
                    IMPRIMER » est un diviseur de »
                    IMPRIMER n
                    n ⟵ n/g
                    x ⟵ x%n
                    x' ⟵ x'%n
                FINSI
            SINON
                k ⟵ k − 1
                SI ( k = 0) ALORS
                    x' ⟵ x
                    h ⟵ 2h
                    k ⟵ h
                FINSI
                x ⟵ (x² + 1)%n
            FINSI
        JUSQU'A (g est différent de 1)
    FAIT
FIN
```

On notera que l'algorithme ainsi décrit, si l'on ne se trouve pas dans le cas d'erreur, « tourne » tant qu'il n'a pas terminé la décomposition du nombre, ce qui peut durer très longtemps... Il faut, bien sûr, en plus, prévoir un arrêt au bout d'un certain temps.

II.3 Discussion

Même si, quelquefois, cet algorithme permet la factorisation de nombres plus grands, il ne peut pas prétendre arriver à décomposer tous les nombres de 20 chiffres ou moins.

Cette méthode est idéale pour les calculettes programmables.

III Algorithme du crible quadratique QS de Pomerance

L'idée, dans cet algorithme comme dans de nombreux autres, et d'obtenir, si possible, des congruences de la forme $x^2 \equiv y^2\ [n]$, x n'étant ni congru à y, ni à $-y$. Dans ce cas, $(x-y) \wedge n$ sera un diviseur non trivial de n.

Ce qui distingue ces méthodes entre elles est la manière d'obtenir ces résidus quadratiques modulo n.

Ici, on prend les valeurs sur les entiers du polynôme $P(x) = (x + E(\sqrt{n}))^2 - n$. Ces valeurs fournissent des congruences de la forme $y^2 \equiv r\ [n]$, où r est le résidu quadratique.

On peut repérer plus facilement ceux qui se factorisent aisément (par la méthode précédente, donc qui sont assez petits) en criblant les valeurs de $P(x)$ pour obtenir la congruence recherchée $x^2 \equiv y^2\ [n]$ de manière efficace.

L'intérêt de cette méthode est qu'elle donne ses meilleurs résultats sur les nombres qui font échouer la suivante, mais elle est très difficile à programmer : le criblage n'est pas évident, et il y a énormément de « petites astuces », qui ne peuvent être examinées ici, pour retrouver les congruences recherchées aussi rapidement que possible.

C'est la méthode la plus utilisée avec celle, plus récente, des courbes elliptiques. On peut espérer décomposer des nombres jusqu'à 70 chiffres à peu près dans des temps raisonnables (on veut dire : quelques jours…).

IV Algorithme $(p-1)$ de Pollard

Soit p un diviseur premier du nombre n à décomposer.

Si a est premier avec p, $a^p \equiv a[p]$ (théorème de Fermat).

Comme a est premier avec p, il est inversible modulo p, donc $a^{p-1} \equiv 1\ [p]$, soit $(a^{p-1} - 1) \equiv 0\ [p]$, ou encore $(a^{p-1} - 1) = kp$.

Donc $(a^{p-1} - 1) \wedge n \neq 1$: ce PGCD est donc un diviseur de n.

Le cas d'échec est celui où $k = 0$, on trouve alors que le PGCD est n, et on n'obtient aucun renseignement sur un éventuel diviseur de n.

Par ailleurs, évidemment, on ne peut pas calculer a^{p-1} quand on ne connaît pas p.

Il suffit en fait d'utiliser un multiple quelconque de $p-1$, soit $h(p-1)$.

On aura aussi $a^{h(p-1)} \equiv 1\ [p]$, il faudra donc utiliser comme exposant un nombre qui comporte le plus possible de facteurs premiers distincts, de manière à ce que les diviseurs de $p-1$ figurent tous dans la liste (c'est-à-dire que cet exposant soit de la forme $h(p-1)$)

Concrètement, on opère étape par étape, en utilisant le PPCM des entiers depuis 1 jusqu'à un maximum fixé.

EXEMPLE 9.1. Soit à décomposer le nombre $n = R_7 = 1\,111\,111 = 239 \times 4\,649$.

1. On doit choisir a, premier avec p, sans connaître p.
   C'est facile, il suffit de choisir un nombre premier : s'il n'est pas égal à p, il est premier avec p. Par exemple : 2 (mais il vaut mieux prendre, en général, 3 ; il y a beaucoup plus de cas d'échec avec 2).

2. Le PPCM des entiers depuis 1 jusqu'à un maximum fixé dans la recherche est égal à $2 \times 3 \times 2 \times 5 \times 7 \times 2 \times 3 \times 11 \times 13 \times 2 \times 17 \times 19 \times 23 \times 5 \times 3 \times 29 \times \ldots$
   (Ces nombres figurent dans une table, déterminée à l'avance, et obtenue par l'algorithme suivant :
   – On parcourt les entiers depuis 2 jusqu'au maximum fixé, tout en disposant de la table des nombres premiers inférieurs ou égaux à ce même maximum.
   – Chaque fois que l'on rencontre un nombre premier, on rajoute ce nombre premier.
   – Chaque fois que l'on rencontre une puissance d'un nombre premier, on rajoute un facteur égal à ce nombre premier, pour compléter le PPCM.
   On obtient donc 2, puis 3, comme nombres premiers, puis, en passant par 4, on rajoute un facteur 2, puis 5, premier, rien à rajouter pour 6, puis 7, premier, un facteur 2 supplémentaire en passant par 8, un facteur 3 à la rencontre de 9, etc...
   Cette table contient 6634 éléments pour le PPCM des entiers depuis 2 jusqu'à 65535, tous les nombres représentables sur 16 bits).

3. On effectue donc les calculs suivants

a	q	a^q [n]	$a^q - 1$	$(a^q - 1) \wedge n$	commentaire
2	2	4	3	1	pas de diviseur
4	3	64	63	1	pas de diviseur
64	2	4 096	4 095	1	pas de diviseur
4 096	5	120 077	120 076	1	pas de diviseur
120 077	7	1 084 896	1 084 895	1	pas de diviseur
1 084 896	2	559 627	559 626	1	pas de diviseur
559 627	3	247 053	247 052	1	pas de diviseur
247 053	11	339 352	339 351	1	pas de diviseur
339 352	13	311 394	311 393	1	pas de diviseur
311 394	2	677 377	677 376	1	pas de diviseur
677 377	17	569 060	569 059	239	diviseur trouvé...

4. Explication : on a calculé en fait $2^{2\times3\times2\times5\times7\times2\times3\times11\times13\times2\times17} = 2^{24\,504\,480}$.
   Or $24\,504\,480 = 102\,960 \times 238$, qui est bien de la forme $h(p-1)$: le calcul de $a^{h(p-1)}$ a permis de trouver p.

La capacité de cet algorithme à trouver un diviseur p d'un nombre n dépend de la taille du plus grand diviseur de $p - 1$, ce qui explique ses résultats très inégaux.

EXEMPLE 9.2. Il trouve instantanément le diviseur $p = 1\,325\,815\,267\,337\,711\,173$ (19 chiffres décimaux) de R_{53}, parce que le plus grand diviseur premier de $p - 1$ est $8\,941$.

Mais il ne peut pas obtenir le diviseur $q = 106\,007\,173\,861\,643$ (15 chiffres décimaux seulement) de R_{61}, parce que le plus grand diviseur premier de $q - 1$ est $868\,911\,261\,161$ (12 chiffres).

Ces considérations, dans l'optique du cryptage RSA, montrent que si l'on choisit deux nombres premiers p et q de la forme $p = 2p' + 1$ et $q = 2q' + 1$ où p' et q' sont eux-mêmes premiers (ce qui est assez facile à fabriquer), il suffira que p' et q' aient en gros au moins 12 chiffres pour que le cryptage soit invulnérable par l'algorithme de Pollard (mais pas par un autre algorithme, peut-être...).

V Algorithme de Lenstra (courbes elliptiques)

V.1 Introduction aux courbes elliptiques

DÉFINITION 9.1 (COURBE ELLIPTIQUE). *Une courbe elliptique sur un corps K a une équation affine de la forme*

$$y^2 = x^3 + ax + b$$

(en supposant que le discriminant $\Delta = 4a^3 + 27b^2$ n'est pas nul, pour qu'elle ne soit pas dégénérée). $\diamondsuit$

REMARQUE 9.3. Elle a un point à l'infini dans la direction de Oy.

On considère deux points $P_1 = (x_1, y_1)$ et $P_2 = (x_2, y_2)$ d'une pareille courbe.

La droite P_1P_2 (la tangente en P_1 à la courbe dans le cas où $P_1 = P_2$) recoupe la cubique en un troisième point de coordonnées $(x_3, -y_3)$...

DÉFINITION 9.2. *Si pose $P_3 = (x_3, y_3)$ et $P_3 = P_1 + P_2$,*
- *Cette addition sur la courbe elliptique est une loi de groupe abélien,*
- *Elle est telle que l'élément neutre est le point à l'infini*
- *... et l'opposé du point $P = (x, y)$ est le point $P' = (x, -y)$.* $\diamondsuit$

Les coordonnées de P_3 sont obtenues comme suit :

$$\text{Si on pose } m = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{si} \quad P_1 \neq P_2 \\ \dfrac{3x_1^2 + a}{2y_1} & \text{si} \quad P_1 = P_2 \end{cases}, \text{ alors } \begin{cases} x_3 &= m^2 - x_1 - x_2 \\ y_3 &= m(x_1 - x_3) - y_1 \end{cases}.$$

V.2 Algorithme de Lenstra

Ici, il s'agit de calculer dans $\mathbb{Z}/n\mathbb{Z}$, qui n'est pas un corps, donc l'opération risque de ne pas être définie.

REMARQUE 9.4. C'est le cas lorsque $\delta = (x_2 - x_1) \wedge n \neq 1$, auquel cas on ne peut poursuivre le calcul, car l'inverse de $(x_2 - x_1)$ n'est pas défini.

Dans ce cas ($\delta \neq 1$), si $\delta \neq n$, on a trouvé un diviseur de n, et on a gagné. Si $\delta = n$, c'est le cas d'échec.

On applique la méthode de Pollard à la courbe elliptique, en calculant, à partir d'un point P quelconque $P' = kP$, en cherchant les coefficients multiplicateurs dans le même tableau (celui des facteurs du PPCM évoqué plus haut).

Si l'on note $E(\mathbb{Z}/p\mathbb{Z})$ le groupe additif de la courbe elliptique utilisée, l'intérêt d'opérer sur une courbe elliptique de cette sorte est que le cardinal de $E(\mathbb{Z}/p\mathbb{Z})$ n'est pas nécessairement $p - 1$ (comme dans la méthode classique $p - 1$ exposée ci-dessus, où on travaille dans $(\mathbb{Z}/p\mathbb{Z})^*$, de cardinal toujours $p - 1$).

C'est un nombre de la forme $p + 1 - t$, où $|t| \leqslant 2\sqrt{p}$, qui varie selon la courbe utilisée.

Ainsi, en travaillant sur plusieurs courbes simultanément, on augmente les chances que le plus grand diviseur de ce cardinal soit petit, ce qui conditionne, comme on l'a remarqué, le succès de la méthode.

Fin du Chapitre

Troisième partie

Logique

Chapitre 10

Algèbre de Boole

I Propriétés générales

DÉFINITION 10.1 (ALGÈBRE DE BOOLE). *On appelle* algèbre de Boole *la structure algébrique* $(\mathcal{A}, +, ., \overline{\ })$ *définie par un ensemble (non vide)* $\mathcal{A}$ *et trois opérations :*

- *la somme booléenne (binaire) : "+",*
- *le produit booléen (binaire) : "." et*
- *la négation booléenne (unaire) : " $\overline{\ }$ " (par exemple* $\overline{a}$*).*

et qui doivent posséder les propriétés données dans les deux premières colonnes du tableau ci-dessous.♢

Propriété	$\mathcal{A}$	$\mathcal{P}(E)$
idempotence	$a + a = a$ $a \cdot a = a$	$A \cup A = A$ $A \cap A = A$
commutativité	$a + b = b + a$ $a \cdot b = b \cdot a$	$A \cup B = B \cup A$ $A \cap B = B \cap A$
associativité	$a + (b + c) = (a + b) + c$ $a \cdot (b \cdot c) = (a \cdot b) \cdot c$	$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$
éléments neutres	$a + 0 = a$ $a \cdot 1 = a$	$A \cup \emptyset = A$ $A \cap E = A$
absorption	$a + 1 = 1$ $a \cdot 0 = 0$	$A \cup E = E$ $A \cap \emptyset = \emptyset$
distributivités	$a \cdot (b + c) = a \cdot b + a \cdot c$ $a + b \cdot c = (a + b) \cdot (a + c)$	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
involution	$\overline{\overline{a}} = a$	$E \setminus (E \setminus A) = A$
complémentation	$\overline{0} = 1$ $\overline{1} = 0$	$E \setminus \emptyset = E$ $E \setminus E = \emptyset$
partition	$a + \overline{a} = 1$ $a \cdot \overline{a} = 0$	$A \cup (E \setminus A) = E$ $A \cap (E \setminus A) = \emptyset$
« Lois de De Morgan »	$\overline{a + b} = \overline{a} \cdot \overline{b}$ $\overline{a \cdot b} = \overline{a} + \overline{b}$	$E \setminus (A \cup B) = (E \setminus A) \cap (E \setminus B)$ $E \setminus (A \cap B) = (E \setminus A) \cup (E \setminus B)$

Propriétés d'une algèbre de Boole

Le tableau précédent met en parallèle les mêmes propriétés, écrites en utilisant
- soit les notations générales d'une algèbre de Boole,
- soit les notations ensemblistes, définies lorsque $\mathcal{A}$ est l'ensemble $\mathcal{P}(E)$ des parties d'un ensemble E : c'est une algèbre de Boole particulière, bien connue, et qui possède des notations spécifiques.

REMARQUE 10.1. Les signes opératoires utilisés sont les mêmes que ceux de l'addition et de la multiplication des réels. Cependant, ces opérations n'ont évidemment pas les mêmes propriétés, et ne portent pas sur les mêmes éléments.

Exercice 10.1 (Somme disjonctive). *On considère une algèbre de Boole quelconque $(E, +, \cdot, \overline{\ })$.*
On définit l'opération « somme disjonctive », notée $\oplus$, par $a \oplus b = \overline{a}b + a\overline{b}$.

1. *Que vaut $a \oplus 0$? $a \oplus 1$?*
2. *Calculez $a \oplus a$ et $a \oplus \overline{a}$.*
3. *Calculez $\overline{a \oplus b}$.*
4. *Montrez que $\oplus$ est associative et commutative.*

Exercice 10.2 (Opérateurs de Sheffer et de Peirce). *Soit $(E, +, \cdot, \overline{\ })$ une algèbre de Boole.*

1. *On définit l'opération de Sheffer [1] par : $a|b = \overline{a} + \overline{b}$.*
   Comment obtenir $\overline{a}$, $a+b$, $a \cdot b$ en n'utilisant que l'opérateur $|$? Faire de même pour $a+\overline{b}$; étudier l'associativité de cette opération.
2. *On définit la flèche de Peirce [2] par : $a \downarrow b = \overline{a} \cdot \overline{b}$.Mêmes questions.*

REMARQUE 10.2. Ces connecteurs sont donc remarquables, puisqu'ils sont universels (tous les autres connecteurs peuvent s'exprimer avec uniquement la barre de Scheffer, ou uniquement avec la flèche de Peirce). Cependant, par manque de concision et de lisibilité, ces connecteurs ne sont pas utilisés en logique.

II Règles de calcul dans une algèbre de Boole

1. Les priorités habituelles sont respectées pour la somme et le produit booléen.
2. Les éléments neutres sont notés 0 et 1, par analogie avec les entiers de même symbole (ne pas oublier que ces calculs ne se déroulent pas dans $\mathbb{R}$...)
3. L'absence d'éléments symétriques pour la somme et pour le produit interdit les simplifications que l'on a l'habitude de pratiquer « sans y réfléchir » :
   - $a + b = a + c$ ne donne pas $b = c$,
   - $ab = ac$ n'entraîne pas $b = c$.

   En particulier, ne jamais perdre de vue que
   - $a + b = 0$ n'est réalisable en algèbre de Boole que si $a = b = 0$
   - $a.b = 1$ n'est réalisable en algèbre de Boole que si $a = b = 1$ ($A \cap B = E \Leftrightarrow A = E$ et $B = E$)
   - $a.b = 0$ peut être réalisé avec $a \neq 0$ et $b \neq 0$ (par exemple, avec $b = \overline{a}$, mais ce n'est pas la seule solution...). On parle de « diviseurs de zéro ». (Ainsi, $A \cap B = \varnothing$ est possible sans avoir obligatoirement $A = \varnothing$ et $B = \varnothing$).
4. Il y a deux distributivités. Celle de la somme (booléenne) sur le produit (booléen) n'est pas habituelle. Par exemple, simplifier $(a + b)(a + c)(a + d)(a + e)(a + f)$
5. Signalons pour finir que, comme ci-dessus, le point pour le produit est souvent omis.

1. D'après le logicien H.M. Sheffer
2. Lorsque les logiciens, dans les années 1930, cherchèrent un symbole pour exprimer le connecteur découvert par C.S. Peirce (1839-1914), "Pierce Arrow" était le nom d'une célèbre marque de voiture !

Dans une expression booléenne, une sous-expression est dite « redondante » lorsqu'on peut la supprimer sans changer la « valeur » de l'expression :

1. Dans une somme booléenne, tout terme absorbe ses multiples.
   Autrement dit : $a + a \cdot b = a$.

   PREUVE En effet, $a+a\cdot b = a\cdot(\overline{b}+b)+a\cdot b = a\cdot\overline{b}+a\cdot b+a\cdot b = a\cdot\overline{b}+a\cdot b$ (par idempotence) $= a \cdot (\overline{b} + b) = a$. ∎

2. Dans un produit booléen, tout facteur absorbe tout autre facteur qui le contient en tant que terme.
   Autrement dit : $a \cdot (a + b) = a$.

   PREUVE En effet, $a \cdot (a + b) = a \cdot a + a \cdot b = a + a \cdot b = a$. ∎

3. Enfin, la troisième règle de redondance s'exprime par :

$$a + \overline{a} \cdot b = a + b$$

   PREUVE $a + \overline{a} \cdot b = (a + \overline{a}) \cdot (a + b) = 1 \cdot (a + b) = a + b$. ∎

   EXEMPLE 10.3. $ab + \overline{a}c + \overline{b}c = ab + (\overline{a} + \overline{b}) \cdot c = ab + \overline{ab} \cdot c = ab + c$

Exercice 10.4. *Montrer que* $a \cdot b + \overline{a} \cdot c + b.c = a \cdot b + \overline{a} \cdot c$

Exercice 10.5 (Somme disjonctive). *Montrez que l'on a* $a = b$ *si et seulement si* $a \oplus b = 0$.

Exercice 10.6 (Calcul booléen élémentaire). *Appliquer au maximum les règles précédentes pour supprimer les redondances dans les calculs suivants.*

1. $(a + b + c) \cdot (a + \overline{b} + c) \cdot (a + \overline{b} + \overline{c})$
2. $a + \overline{a} \cdot b \cdot c + \overline{a} + a \cdot b$
3. $a \cdot b + \overline{a} \cdot b \cdot c + a \cdot \overline{b} \cdot c$
4. $(a + b + c) \cdot (\overline{a} + \overline{b} + \overline{c} + d)$

Exercice 10.7 (Calcul booléen). *Même énoncé qu'à l'exercice précédent.*

1. $(\overline{a} + b)(\overline{c} + \overline{a} \cdot \overline{b} + a \cdot b)$.
2. $(a + \overline{b} + \overline{c}) \cdot (\overline{a} + b) \cdot (\overline{b} + c)$.
3. $(a + c) \cdot (\overline{a} + d) \cdot (\overline{b} + \overline{e}) \cdot (\overline{b} \cdot \overline{c} + b \cdot c) \cdot (\overline{d} + c \cdot e) \cdot (\overline{c} + d)$.
4. $(\overline{a} \cdot \overline{a \cdot (\overline{b} + \overline{c})} + a \cdot (\overline{b} + \overline{c})) \cdot (\overline{b} \cdot \overline{\overline{a} + c} + (\overline{a} + c) \cdot b) \cdot (a \cdot \overline{b} \cdot c + \overline{a \cdot \overline{b}} \cdot \overline{c})$.

III Fonctions booléennes

III.1 Définitions

Soit $\mathcal{A}$ une algèbre de Boole.

DÉFINITION 10.2 (FONCTION BOOLÉENNE). *On appelle* fonction booléenne de n variables *toute application de* $\mathcal{A}^n$ *dans* $\mathcal{A}$ *dont l'expression ne contient que :*
- *les symboles des opérations booléennes,*
- *des symboles de variables, de constantes,*
- *d'éventuelles parenthèses.* ♢

EXEMPLE 10.8. $f(a,b,c) = a \cdot \overline{b} + c$.

REMARQUE 10.3. Si a est une variable booléenne, elle peut intervenir dans l'expression d'une fonction booléenne sous la forme a ou sous la forme $\overline{a}$, qui sont appelées les deux *aspects* de cette variable : affirmé et nié.

DÉFINITION 10.3 (FONCTION BOOLÉENNE NULLE). *On appelle* fonction booléenne nulle *(à n variables) la fonction booléenne qui, à chaque valeur des variables, associe la valeur 0.*

Son expression est $f(x_1, x_2, \ldots, x_n) = 0$. ♢

DÉFINITION 10.4 (FONCTION RÉFÉRENTIEL). *On appelle* fonction référentiel *(à n variables) la fonction booléenne qui, à chaque valeur des variables, associe la valeur 1.*

Son expression est $f(x_1, x_2, \ldots, x_n) = 1$. ♢

III.2 Fonctions booléennes élémentaires

DÉFINITION 10.5 (MINTERME, MAXTERME). *Un* minterme *à n variables est une fonction booléenne à n variables dont l'expression se présente sous la forme du produit d'un aspect et d'un seul de chacune des n variables.*

Définition analogue pour un maxterme *, en remplaçant dans la définition précédente « produit » par « somme ».* ♢

DÉFINITION 10.6 (FONCTIONS BOOLÉENNES ÉLÉMENTAIRES). *Pour un nombre de variables n fixé, les* fonctions booléennes élémentaires *sont les mintermes et les maxtermes (à n variables).* ♢

EXEMPLE 10.9 (MINTERME À TROIS VARIABLES). $a \cdot \overline{b} \cdot c$

EXEMPLE 10.10 (MAXTERME À TROIS VARIABLES). $\overline{a} + b + \overline{c}$.

Exercice 10.11. *Pour 3 variables a, b et c, repérez les mintermes et les maxtermes : $b\overline{c}$, $a + \overline{b} + c$, $a\overline{b}b\overline{c}$, $\overline{a}bc$, $a + \overline{b}c$.*

Exercice 10.12. *Dressez la liste des mintermes et des maxtermes pour deux variables a et b.*

PROPRIÉTÉ 10.1 (NOMBRE DE MINTERMES ET DE MAXTERMES) : Les mintermes et maxtermes, pour un nombre donné n de variables, sont au nombre de 2^n chacun.

NOTATION (REPRÉSENTATION DES MINTERMES ET DES MAXTERMES) : Les mintermes et les maxtermes sur n variables sont respectivement notés $m_i^{(n)}$ et $M_i^{(n)}$. L'indice i varie entre 0 et $2^n - 1$, selon une convention d'ordre de numérotation des mintermes et maxtermes.

La convention est la suivante : à chaque variable, on associe 0 ou 1 selon que cette variable apparaît sous son aspect nié ou sous son aspect affirmé dans l'expression du minterme ou du maxterme. En écrivant ces chiffres les uns à côté des autres, dans le même ordre que les variables correspondantes, on obtient un code binaire du minterme ou du maxterme, qu'on peut considérer comme l'écriture d'un entier positif en base 2.

Pour que cette convention de numérotation ait un sens, il est indispensable de fixer un ordre d'énumération des variables une fois pour toutes, et de s'y tenir.

EXEMPLE 10.13. Si les variables sont a, b, c et d et qu'on décide de les énumérer dans l'ordre alphabétique, il sera, par exemple, strictement interdit d'écrire un produit sous la forme $c \cdot a \cdot d \cdot b$, et ceci, même de manière transitoire au cours d'un calcul : la seule expression admissible est alors $a \cdot b \cdot c \cdot d$.

DÉFINITION 10.7 (INDICE D'UN MINTERME OU D'UN MAXTERME). *L'*indice d'un minterme ou d'un maxterme *est la valeur décimale du code binaire de ce minterme ou de ce maxterme.* ♢

EXEMPLE 10.14. Pour 3 variables a, b et c rangées par ordre alphabétique :

minterme (ou Maxterme)	code binaire associé	indice décimal	représentation
$\overline{a} \cdot b \cdot \overline{c}$	010	2	m_2
$a \cdot \overline{b} \cdot \overline{c}$	100	4	m_4
$a + b + c$	111	7	M_7

Exercice 10.15. *Pour 3 variables a, b et c rangées par ordre alphabétique, trouvez l'indice des mintermes et maxtermes suivants : $\overline{a} + b + \overline{c}$, $\overline{a} + \overline{b} + c$, $a \cdot b \cdot c$ et $\overline{a} \cdot b \cdot c$.*

III.3 Correspondance entre maxtermes et mintermes

PROPRIÉTÉ 10.2 : La négation (booléenne) d'un minterme est un maxterme (et réciproquement).

PREUVE Lois de De Morgan : la négation échange les opérations booléennes binaires... ■

EXEMPLE 10.16. $\overline{\overline{a} \cdot b \cdot \overline{c}} = a + \overline{b} + c$

Si l'indice du minterme (ou du maxterme) dont on prend la négation est i et si l'indice de cette négation est j, on a des expressions du type :

$$\begin{array}{rcl} i & = & 0011\,0100\,1110\,\ldots\ldots\,0110 \\ j & = & 1100\,1011\,0001\,\ldots\ldots\,1001 \\ \hline i+j & = & 1111\,1111\,1111\,\ldots\ldots\,1111 \end{array}$$

L'expression en système binaire de la valeur de $i + j$ est donc, quelles que soient les valeurs de ces deux indices, 111.....1 (n chiffres). La valeur correspondante est $2^n - 1$. Autrement dit,

PROPRIÉTÉ 10.3 : La négation d'un minterme est un maxterme, et réciproquement.

$$\forall i \in \{0, ..., 2^n - 1\}, \overline{m_i^{(n)}} = M_{2^n-1-i}^{(n)} \text{ et } \overline{M_i^{(n)}} = m_{2^n-1-i}^{(n)}.$$

III.4 Principaux résultats concernant mintermes et maxtermes

PROPRIÉTÉ 10.4 : Les mintermes à n variables sont disjoints.

$$\text{Si } i \neq j, \text{ alors } m_i^{(n)} \cdot m_j^{(n)} = 0.$$

Exercice 10.17. *Vérifiez la dernière propriété dans le cas de deux variables.*

PREUVE Si $i \neq j$, les écritures en système binaire des entiers i et j comportent au moins un chiffre différent, en l'occurrence au moins un « 1 » à la place d'un « 0 ».

Il y a donc au moins une variable qui figure sous deux aspects différents.

Or, on sait que $a \cdot \overline{a} = 0$. Donc, lorsque l'on calcule le produit des deux mintermes, celui-ci est nécessairement nul. ■

REMARQUE 10.4. On prend la négation de chacun des membres de l'égalité, et l'on obtient : si $i \neq j$, alors $M_i^{(n)} + M_j^{(n)} = 1$. Ainsi, la somme de deux maxtermes distincts vaut 1.

PROPRIÉTÉ 10.5 : Les mintermes forment une partition de l'unité :

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

PREUVE En effet, il y a un nombre pair de mintermes. Dans cette somme, on ordonne les mintermes par indice croissant, puis on les regroupe deux à deux. Dans chacun de ces groupes, seul diffère l'aspect de la dernière variable. On met les autres en facteur de la somme $\overline{x}_n + x_n$, c'est-à-dire 1 : le facteur qui subsiste est un minterme à $(n-1)$ variables.

Donc $\sum_{i=0}^{2^n-1} m_i^{(n)} = \sum_{i=0}^{2^{n-1}-1} m_i^{(n-1)}$.

Par récurrence, cette somme est égale à $\overline{x}_1 + x_1$, c'est-à-dire finalement 1. ■

Exercice 10.18. *Le vérifier dans le cas de deux variables.*

REMARQUE 10.5. Par négation (booléenne) de cette propriété, on obtient : *le produit de tous les maxtermes à n variables est nul*.

III.5 Formes canoniques d'une fonction booléenne

DÉFINITION 10.8 (MONÔMES). *Un* monôme *est une fonction booléenne produit de variables booléennes éventuellement niées.* ♢

Exercice 10.19. *Parmi les expressions suivantes dire lesquelles sont des monômes et lesquelles ne le sont pas en justifiant : $a + b$, $a + bc$, $a(b + c)$, $a\overline{b}$, b.*

PROPRIÉTÉ 10.6 : Quelle que soit l'expression de la fonction booléenne, il est possible de la mettre sous la forme d'une somme de monômes.

PREUVE En effet, comme elle ne fait intervenir que les trois opérations booléennes, il suffit de lui appliquer les règles du calcul booléen :

1. On développe les négations (en appliquant les règles $\overline{a+b} = \overline{a} \cdot \overline{b}$ et $\overline{a \cdot b} = \overline{a} + \overline{b}$), jusqu'à ce qu'il n'y ait plus de négations que sur les variables ;
2. Puis on développe les produits qui portent sur des sommes, en utilisant la distributivité du produit sur la somme ;

3. On obtient ainsi une expression qui s'écrit sans parenthèses, et qui ne contient que des sommes de produits de variables éventuellement niées. ■

PROPRIÉTÉ 10.7 : Chaque monôme peut ensuite être mis sous la forme d'une somme de mintermes.

PREUVE En effet, si, dans l'expression de ce monôme, toutes les variables interviennent, c'est déjà un minterme.

Dans le cas contraire, il manque (par exemple) la variable a dans son expression : on la fait intervenir sous la forme $(\overline{a}+a)$. On développe, les deux monômes obtenus font intervenir la variable a.

Ou bien, il s'agit de mintermes et le processus est terminé, ou bien il manque encore une variable, qu'on fait intervenir en utilisant le même procédé, et ainsi de suite jusqu'à aboutir aux mintermes. ■

On fait évidemment disparaître du résultat, par idempotence, les occurrences multiples de mintermes, pour pouvoir énoncer le résultat suivant :

PROPRIÉTÉ 10.8 (FORME CANONIQUE DISJONCTIVE) : Toute fonction booléenne à n variables (autre que la fonction nulle) peut se mettre sous la forme d'une somme de mintermes à n variables.

Cette forme, unique, s'appelle *Forme Canonique Disjonctive* (dans la suite, FCD).

REMARQUE 10.6. L'unicité de cette FCD permet la comparaison des fonctions booléennes entre elles.

Par négation booléenne de ce résultat, on obtient :

PROPRIÉTÉ 10.9 (FORME CANONIQUE CONJONCTIVE) : Toute fonction booléenne de n variables (autre que la fonction référentiel) peut se mettre sous la forme d'un produit de maxtermes à n variables.

Cette forme, unique, est la *Forme Canonique Conjonctive* (FCC dans la suite).

III.5.1 Obtention des formes canoniques

La méthode algébrique consiste à :
- tout développer pour mettre l'expression sous la forme d'une somme de monômes,
- dans chaque terme de cette somme, faire apparaître les valeurs qui n'y figurent pas.

EXEMPLE 10.20. On illustre cela :
$f(a,b,c) = a + bc = a(\overline{b}+b)(\overline{c}+c) + (\overline{a}+a)bc$
$= a\overline{b}\overline{c} + a\overline{b}c + ab\overline{c} + abc + \overline{a}bc + abc = m_3 + m_4 + m_5 + m_6 + m_7.$

Pour la FCC, on peut imaginer une méthode analogue.

EXEMPLE 10.21. $f(a,b,c) = a + bc = (a+b)(a+c) = (a+b+\overline{c}c)(a+\overline{b}b+c)$
$= (a+b+\overline{c}) \cdot (a+b+c) \cdot (a+\overline{b}+c) \cdot (a+b+c) = M_5M_6M_7$

REMARQUE 10.7. Si on prend la négation de la FCD, on obtient bien sûr une FCC... mais pas celle de la fonction, celle de sa négation !

Il suffit de prendre la négation de la fonction, de calculer sa FCD puis de prendre la négation du résultat.

Exercice 10.22. *Obtenir la FCC de* $x + \overline{y}z$.

Il existe une autre méthode pour obtenir ces formes canoniques : la méthode des diagrammes.

IV Diagrammes de Karnaugh

La représentation des fonctions booléennes par diagrammes de Karnaugh-Veitch : est fondée sur les propriétés des mintermes (ils réalisent une partition de l'unité),

Ces derniers diagrammes deviennent rapidement inextricables quand le nombre de variables augmente, c'est pourquoi, dans les diagrammes de Karnaugh, on divise systématiquement l'« univers » (le référentiel E) en deux parties égales en superficie pour représenter la partie concernée et son complémentaire.

À chaque introduction de variable supplémentaire, chaque case du précédent diagramme est divisée en 2.

EXEMPLE 10.23. On obtient, par exemple :

	$\overline{a}$	a
$\overline{b}$	$\overline{a}\overline{b}$	$a\overline{b}$
b	$\overline{a}b$	ab

Pour obtenir un diagramme de Karnaugh, on place dans ce diagramme les numéros des mintermes :

b \ a	0	1
0	0	2
1	1	3

EXEMPLE 10.24. Cas de trois variables :

- les deux premières colonnes correspondent à $\overline{a}$, les deux dernières à a,
- la première et la dernière colonne correspondent à $\overline{b}$, les deux centrales à b,
- enfin, la première ligne est associée à $\overline{c}$, la deuxième à c.

...ce qui donne

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

EXEMPLE 10.25. Cas de quatre variables :

cd \ ab	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Dans un tel diagramme, chaque case représente un minterme. Les autres monômes regroupent un nombre de cases qui est une puissance de 2, selon le nombre de variables présentes.

Exercice 10.26. *Faire un diagramme à cinq variables.*

Réponse :

de \ abc	000	001	011	010	110	111	101	100
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

C'est-à-dire :
- les quatre premières colonnes correspondent à $\overline{a}$, les quatre dernières à a,
- les deux premières, et les deux dernières colonnes correspondent à $\overline{b}$, les quatre centrales à b,
- les colonnes 1, 4, 5 et 8 à $\overline{c}$, les autres à c,
- les deux premières lignes à $\overline{d}$, les deux dernières à d,
- enfin, la première et la dernière ligne sont associées à $\overline{e}$, les deux centrales à e.

Les diagrammes peuvent être utilisés en réunion, en intersection ou en complémentation.
Ils permettent :
- d'obtenir la FCD d'une fonction booléenne plus aisément que par le calcul algébrique (utilisé pour découvrir la forme en question),
- une première approche du problème de la simplification des fonctions booléennes (dans des cas simples et pour un petit nombre de variables)...

Utilisation des diagrammes de Karnaugh pour représenter les fonctions booléennes...

En réunion. Soit par exemple $f(a,b,c) = a + \overline{b}c$. Son diagramme est :

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

On lit aisément la FCD de f sur le diagramme : $f(a,b,c) = m_1 + m_4 + m_5 + m_6 + m_7$.

En intersection. Soit $f(a,b,c) = (a + \overline{b})(a + c)$.

On peint en rouge les cases correspondant à $a + \overline{b}$, et on note en italique les nombres correspondant à $a + c$:

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

La représentation de f est contenue dans les cases rouges possédant les nombres en italique. Comme $(a+\overline{b})(a+c) = a+\overline{b}c$, on retrouve la même FCD.

En complémentation. Soit $f(a,b,c) = a+\overline{b}c$, de diagramme :

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Alors la négation de $a+\overline{b}c$ est dans les cases pas rouge : la FCD de $\overline{f}$ est $m_0+m_2+m_3$.

Exercice 10.27 (Fonctions booléennes). *Donner la forme canonique disjonctive de la fonction booléeene dont l'expression est*

$$f(a,b,c,d,e) = \overline{a}\cdot[\overline{b}\cdot\overline{e}\cdot(c+d)+b\cdot(\overline{c}\cdot\overline{d}\cdot\overline{e}+c\cdot\overline{d\cdot e})].$$

Exercice 10.28. *Pour chacune des expressions suivantes...*

$$\begin{aligned} E_1 &= xyz+xy\overline{z}+\overline{x}y\overline{z}+\overline{xy}z \\ E_2 &= xyz+xy\overline{z}+x\overline{y}z+\overline{xy}z \\ E_3 &= xyz+xy\overline{z}+\overline{x}y\overline{z}+\overline{xyz}+\overline{xy}z \end{aligned}$$

donner la forme minimale en exploitant les diagrammes de Karnaugh

Exercice 10.29 (Application de la méthode de Karnaugh). *Trouver une forme minimale de* $E = x\overline{y}+xyz+\overline{xyz}+\overline{x}yz\overline{t}$.

Exercice 10.30 (Composition de la méthode de Karnaugh). *On considère deux fonctions booléennes* u *et* v *des quatres variables* a, b, c, d *définies par* $u = (a+d)(b+c)$ *et* $v = (a+c)(\overline{b}+d)$.

1. *Dessiner les diagrammes de Karnaugh de* u *et de* v.
2. *En déduire le diagramme de Karnaugh de* $w = uv+\overline{uv}$.
3. *Donner une forme minimale pour* w

Exercice 10.31 (BTS-2009). *La société K-Gaz décide de recruter en interne des collaborateurs pour sa filiale en Extrême-Orient. Pour chaque employé, on définit les variables booléennes suivantes :*

- a *= 1 s'il a plus de cinq ans d'ancienneté dans l'entreprise ;*
- b *= 1 s'il possède un B.T.S. informatique de gestion (BTS-IG) ;*
- c *= 1 s'il parle couramment l'anglais.*

La direction des ressources humaines décide que pourront postuler les employés :

- *qui satisfont aux trois conditions,*
- *ou qui ont moins de 5 ans d'ancienneté mais qui maîtrisent l'anglais,*
- *ou qui ne maîtrisent pas l'anglais mais qui possèdent un BTS-IG.*

1. *Écrire une expression booléenne* E *traduisant les critères de la direction.*
2. *Représenter l'expression* E *par un tableau de Karnaugh.*
3. *À l'aide du tableau de Karnaugh, donner une expression simplifiée de* E.

4. *Retrouver ce résultat par le calcul.*
5. *En déduire une version simplifiée des critères de la direction.*

Exercice 10.32 (BTS-2002). *On considère l'expression $E = a.c + b.c + a.b + a.b.c$ dépendant des variables booléennes a, b et c :*

1. *Simplifier l'expression E à l'aide de la lecture d'un tableau de Karnaugh (ou d'une table de vérité).*
2. *Dans un organisme qui aide des personnes au chômage à trouver un emploi, on considère pour ces personnes, trois variables booléennes définies ainsi :*
   – *a = 1 si la personne est âgée de 45 ans ou plus (sinon a = 0) ;*
   – *b = 1 si la personne est au chômage depuis un an ou plus (sinon b = 0) ;*
   – *c = 1 si la personne a déjà suivi une formation l'année précédente (sinonc = 0).*
   Une formation qualifiante sera mise en place pour les personnes vérifiant au moins un des critères suivants :
   – *avoir 45 ans ou plus et être au chômage depuis moins de un an ;*
   – *avoir moins de 45 ans et ne pas avoir suivi de formation l'année précé- dente ;*
   – *être au chômage depuis un an ou plus et ne pas avoir suivi de formation l'année précédente ;*
   – *avoir moins de 45 ans, être au chômage depuis moins de un an et avoir suivi une formation l'année précédente.*
   Les personnes qui ne répondent à aucun de ces quatre critères, pourront par- ticiper à un stage d'insertion en entreprise.
   (a) *Écrire l'expression booléenne F en fonction des variables a, b et c qui traduit le fait que la personne pourra suivre cette formation qualifiante.*
   (b) *En déduire une caractérisation simple des personnes qui participeront à un stage d'insertion en entreprise.*

V Résolution d'équations booléennes

Soit une équation booléenne de la forme la plus générale :

$$f(x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_n)$$

1. Puisque $A = B \iff A \oplus B = 0$, on se ramène immédiatement à une équation du type :

$$F(x_1, x_2, \ldots, x_n) = 0$$

2. On met F sous la forme :

$$F(x_1, x_2, \ldots, x_n) = \overline{x_1} \cdot r + x_1 \cdot s = 0$$

   Cette dernière équation est équivalente, en algèbre de Boole, aux deux équations

$$\begin{cases} (1) & \overline{x_1} \cdot r = 0 \\ (2) & x_1 \cdot s = 0 \end{cases}$$

3. Une équation du type de (2) se résout par introduction d'une variable auxiliaire y_1. En effet,

$$x_1 \cdot s = 0 \iff \exists y_1 \in E, x_1 = y_1 \cdot \overline{s}$$

   .

4. Dans ces conditions, $\overline{x_1} = \overline{y_1} + s$, valeur que l'on porte dans (1), pour obtenir l'équation $\overline{y_1} \cdot r + r \cdot s = 0$, qui est elle-même équivalente aux deux équations

$$\begin{cases}(3) & \overline{y_1}\cdot r = 0\\ (4) & r\cdot s = 0\end{cases}.$$

En utilisant la variable auxiliaire z_1, (3) se résout comme (2) par :

$$\exists z_1 \in E, y_1 = \overline{z_1} + r$$

5. Finalement, l'équation proposée est équivalente aux équations :
   – (5) $x_1 = (\overline{z_1} + r)\cdot \overline{s}$;(qui donne les valeurs de x_1)
   – (4) $r \cdot s = 0$ (qui ne comporte plus que n-1 variables).
   On recommence donc les mêmes opérations pour x_2 dans (4), et ainsi de suite.

Exercice 10.33. *Résoudre l'équation :* $x + y = x + z$.

Exercice 10.34. *Résoudre l'équation :* $x \cdot y + \overline{x} \cdot z = 0$

Exercice 10.35. *Résoudre le système d'équations :* $\begin{cases}x + y = x + z\\ x \cdot y = x \cdot z\end{cases}$

Exercice 10.36 (Fonctions booléennes universelles). *On considère une fonction booléenne de deux variables, mise sous forme canonique disjonctive :* $f(x, y) = \alpha \cdot \overline{x} \cdot \overline{y} + \beta \cdot \overline{x} \cdot y + \gamma \cdot x \cdot \overline{y} + \delta \cdot x \cdot y$, *où* $(\alpha\,,\,\beta\,,\,\gamma\,,\,\delta) \in \{0, 1\}^4$

– *Montrer que cette fonction n'est susceptible d'exprimer la négation que si* $\alpha = 1$ *et* $\delta = 0$.
– *Dans ce cas, montrer qu'il n'y a que deux couples de valeurs possibles pour* β *et* γ, *si l'on veut que f puisse aussi exprimer la somme* $x + y$ *et le produit* $x \cdot y$.

La simplification des fonctions booléennes doit être laissée aux méthodes algébriques dans les cas plus complexes, de manière à pouvoir affirmer avoir trouvé une forme minimale, et éventuellement toutes, si nécessaire. Diverses méthodes visent cet objectif. Nous n'en exposerons ici qu'une seule, la méthode de Quine-Mac Cluskey, dite aussi *méthode des consensus*.

VI Méthode des consensus

La méthode des consensus est une méthode algébrique permettant :
– d'être certain d'obtenir la forme minimale,
– de les obtenir toutes.
Commençons par introduire la notion de consensus...

DÉFINITION 10.9. *Lorsque, dans une somme booléenne, deux monômes admettent dans leur expression une et une seule variable qui se présente sous son aspect affirmé dans l'un et sous son aspect nié dans l'autre, on dit que ces deux monômes* présentent un consensus *(ou sont en consensus).*

Le consensus *de ces deux monômes est alors le produit de toutes les autres variables.* ♢

EXEMPLE 10.37. Les monômes $\overline{a} \cdot b \cdot d \cdot e$ et $a \cdot \overline{c} \cdot d \cdot f$ présentent un consensus, car le premier contient $\overline{a}$ et le second a. Le consensus de ces deux termes est $b \cdot \overline{c} \cdot d \cdot e \cdot f$.

EXEMPLE 10.38. abc et $\overline{b}cd$ présentent un consensus (acd), quand abc et bcd d'une part, et $a\overline{b}c$ et $b\overline{c}d$ d'autre part, n'en présentent pas.

Exercice 10.39. *Trouvez tous les consensus de*

$$f(a, b, c, d) = \overline{a}\overline{b}c + \overline{a}c\overline{d} + a\overline{b}\overline{c}\overline{d} + a\overline{c}d + bcd$$

PROPRIÉTÉ 10.10 (RÉSULTAT FONDAMENTAL) : Rajouter, à une somme booléenne, le consensus de deux termes de la somme (qui en présentent un) ne modifie pas sa valeur.

PREUVE En effet, le consensus de $a \cdot m$ et de $\overline{a} \cdot m'$ est $m \cdot m'$, et on peut constater que $a \cdot m + \overline{a} \cdot m' + m \cdot m' = a \cdot m + \overline{a} \cdot m' + (a + \overline{a}) \cdot m \cdot m' = a \cdot m + \overline{a} \cdot m' + a \cdot m \cdot m' + \overline{a} \cdot m \cdot m' = a \cdot m + \overline{a} \cdot m'$.■

Venons-en à la méthode proprement dite (qui permettra, on le rappelle, de trouver toutes les formes les plus simplifiées d'une expression booléenne donnée). Elle se déroule en trois étapes...

Étape préliminaire. Développer l'expression pour la mettre sous la forme d'une somme de monômes, et en suppprimer les multiples (toute autre tentative de simplification est inutile).

Obtention d'une forme stable par consensus. Répéter les deux phases suivantes, jusqu'à ce que l'expression obtenue soit stable, ne change plus :

1. rajouter tous les consensus des termes qui en présentent un,
2. supprimer les multiples nouvellement introduits.

REMARQUE 10.8. L'introduction des consensus fait parfois apparaître des multiples. La suppression de ces derniers fait parfois apparaître de nouvelles possibilités de consensus...

DÉFINITION 10.10 (EXPRESSION STABLE, MONÔMES PRINCIPAUX). *L'expression obtenue est dite* stable du point de vue des consensus *; elle est unique. Ses termes s'appellent les* monômes principaux. ♢

Exercice 10.40. *Trouvez la forme stable par consensus de*

$$f(a,b,c,d) = \overline{a}\overline{b}c + \overline{a}c\overline{d} + a\overline{b}\overline{c}\overline{d} + a\overline{c}d + bcd$$

La somme des monômes principaux d'une expression booléenne est généralement plus longue que l'expression de départ. Parfois, elle peut s'avérer plus courte, mais même dans ce cas, rien ne prouve qu'il n'existe pas une expression encore plus courte. C'est pourquoi, dans tous les cas, une nouvelle étape est nécessaire.

EXEMPLE 10.41. Dans $a + \overline{a} \cdot b$, les deux termes présentent un consensus, qui est b, et on a alors $a + \overline{a} \cdot b = a + \overline{a} \cdot b + b = a + b$ (comme on le sait, c'est la règle n°3). Ici, il y a simplification.

Mais dans $a \cdot b + \overline{a} \cdot c$, les deux termes présentent un consensus, qui est $b \cdot c$. On a alors $a \cdot b + \overline{a} \cdot c = a \cdot b + \overline{a} \cdot c + b \cdot c$. Ici, il apparaît un terme de plus.

Choix d'un nombre minimal de monômes principaux. C'est la dernière étape de la méthode des consensus, qui fait intervenir la FCD.

Les formes minimales de l'expression algébrique de départ sont des sommes des monômes principaux ci-dessus. Pour savoir quels monômes principaux garder, et quels monômes principaux supprimer, on calcule la FCD de l'expression de départ. Les mintermes de cette FCD sont des multiples des monomes principaux. Les monômes principaux retenus dans les formes minimales sont tels qu'ils possèdent tous les mintermes de la FCD parmi leurs multiples.

Pour bien comprendre cette dernière étape, *i.e.* cette sélection des monômes principaux réellement utiles, on donne plusieurs exemples complets...

EXEMPLE 10.42. Appliquons la méthode des consensus à

$$f(a,b,c) = (a+b)(\overline{a}+\overline{b}+c)$$

1. On développe : $f(a,b,c) = a\overline{b} + ac + \overline{a}b + bc$.
2. Obtention d'une forme stable par consensus.
   - $\overline{a}b, ac$: pas de consensus,
   - $ac, \overline{a}b$: consensus bc,
   - $a\overline{b}, \overline{a}b$: pas de consensus,
   - ac, bc : pas de consensus,
   - $a\overline{b}, bc$: consensus ac,
   - $\overline{a}b, bc$: pas de consensus,

   D'où $f(a,b,c) = a\overline{b} + ac + \overline{a}b + bc + ac + bc$.
   Par idempotence : $f(a,b,c) = a\overline{b} + ac + \overline{a}b + bc$.
   Rajouter des consensus ne change alors rien : c'est notre forme stable.
   Soient p_1, p_2, p_3, p_4 les quatre monômes principaux.
3. Le diagramme de Karnaugh de l'expression de départ est :

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

   D'où la FCD de l'expression de départ : $m_2 + m_3 + m_4 + m_5 + m_7$.
4. Choix d'un nombre minimal de monomes principaux :

monomes principaux \ mintermes	2	3	4	5	7
1			X	X	
2				X	X
3	X	X			
4		X			X
	↑ p_3	↑ p_3 p_4	↑ p_1	↑ p_1 p_2	↑ p_2 p_4

   Tout minterme de la FCD doit être pris au moins une fois. Donc :
   - pour avoir m_2, pas le choix : il faut prendre p_3. Mais, comme on a pris p_3, on a récupéré m_3.
   - pour avoir m_4, il faut prendre p_1. Avec cela, on récolte m_5.
   - enfin, pour avoir m_7, on a le choix entre p_2 et p_4.

   Il y a donc deux formes minimales :
   - p_1, p_2, p_3,
   - p_1, p_3, p_4.

EXEMPLE 10.43. Appliquons la méthode des consensus à

$$S = a \cdot b + \overline{a} \cdot c$$

La somme des monômes principaux de S est $a \cdot b + \overline{a} \cdot c + b \cdot c$.
Posons :

– $P_1 = a \cdot b$,
– $P_2 = \overline{a} \cdot c$
– $P_3 = b \cdot c$.

La FCD de S est $m_1 + m_3 + m_6 + m_7$.

– m_1 est contenu dans P_2. Le choix de P_2 est donc obligatoire, ce que l'on exprime par l'équation booléenne $p_2 = 1$.
– m_3 est contenu dans P_2 et P_3. On a donc le choix entre ces deux monômes, ce que l'on exprime par l'équation booléenne $p_2 + p_3 = 1$ (évidemment, le choix précédent rend cette condition inutile, mais on expose ici la méthode).
– m_6 est contenu dans P_1, soit $p_1 = 1$.
– m_7 est contenu dans P_1 et P_3, soit $p_1 + p_3 = 1$.

Il faut donc développer le produit $p_2(p_2 + p_3)p_1(p_1 + p_3) = p_1p_2$ qui prouve que la forme minimale (unique, dans cet exemple) de la fonction donnée est obtenue avec la somme de P_1 et de P_2 ; il s'agit, bien entendu, de $a \cdot b + \overline{a} \cdot c$.

EXEMPLE 10.44. Appliquons la méthode des consensus à

$$S = \overline{a} \cdot \overline{c} + a \cdot b \cdot d + \overline{a} \cdot b \cdot c + a \cdot \overline{b} \cdot \overline{c} + \overline{b} \cdot c \cdot \overline{d} + \overline{a} \cdot \overline{b} \cdot \overline{c}$$

1. Suppression des multiples : $\overline{a} \cdot \overline{c}$ absorbe $\overline{a} \cdot \overline{b} \cdot \overline{c}$
   Il reste :
   $\overline{a} \cdot \overline{c} + a \cdot b \cdot d + \overline{a} \cdot b \cdot c + a \cdot \overline{b} \cdot \overline{c} + \overline{b} \cdot c \cdot \overline{d}$
2. Premiers consensus :
   $\overline{a}\cdot\overline{c}+a\cdot b\cdot d+\overline{a}\cdot b\cdot c+a\cdot\overline{b}\cdot\overline{c}+\overline{b}\cdot c\cdot\overline{d}+b\cdot\overline{c}\cdot d+\overline{a}\cdot b+\overline{b}\cdot\overline{c}+\overline{a}\cdot\overline{b}\cdot\overline{d}+b\cdot c\cdot d+a\cdot\overline{c}\cdot d+a\cdot c\cdot\overline{d}+\overline{a}\cdot c\cdot\overline{d}+a\cdot\overline{b}\cdot\overline{d}$
3. Suppression des multiples :
   $\overline{a} \cdot b$ absorbe $\overline{a} \cdot b \cdot c$
   $\overline{b} \cdot \overline{c}$ absorbe $a \cdot \overline{b} \cdot \overline{c}$
   Il reste :
   $\overline{a}\cdot\overline{c}+a\cdot b\cdot d+\overline{b}\cdot c\cdot\overline{d}+b\cdot\overline{c}\cdot d+\overline{a}\cdot b+\overline{b}\cdot\overline{c}+\overline{a}\cdot\overline{b}\cdot\overline{d}+b\cdot c\cdot d+a\cdot\overline{c}\cdot d+a\cdot c\cdot\overline{d}+\overline{a}\cdot c\cdot\overline{d}+a\cdot\overline{b}\cdot\overline{d}$
4. Nouveaux consensus (on n'a fait figurer qu'une seule fois chacun d'entre eux) :
   $\overline{a}\cdot\overline{c}+a\cdot b\cdot d+\overline{b}\cdot c\cdot\overline{d}+b\cdot\overline{c}\cdot d+\overline{a}\cdot b+\overline{b}\cdot\overline{c}+\overline{a}\cdot\overline{b}\cdot\overline{d}+b\cdot c\cdot d+a\cdot\overline{c}\cdot d+a\cdot c\cdot\overline{d}+\overline{a}\cdot c\cdot\overline{d}+$
   $a\cdot\overline{b}\cdot\overline{d}+\overline{a}\cdot b\cdot d+\overline{c}\cdot d+\overline{a}\cdot\overline{d}+\overline{b}\cdot\overline{c}\cdot\overline{d}+b\cdot d+a\cdot b\cdot c+\overline{b}\cdot\overline{d}+b\cdot c\cdot\overline{d}+\overline{a}\cdot b\cdot c+a\cdot\overline{b}\cdot\overline{c}+c\cdot\overline{d}$
5. Suppression des multiples :
   $\overline{a} \cdot b$ absorbe $\overline{a} \cdot b \cdot d$ et $\overline{a} \cdot b \cdot c$, $\overline{b} \cdot \overline{c}$ absorbe $\overline{b} \cdot \overline{c} \cdot \overline{d}$ et $a \cdot \overline{b} \cdot \overline{c}$, $\overline{c} \cdot d$ absorbe $b \cdot \overline{c} \cdot d$ et $a \cdot \overline{c} \cdot d$
   $\overline{a} \cdot \overline{d}$ absorbe $\overline{a} \cdot \overline{b} \cdot \overline{d}$ et $\overline{a} \cdot c \cdot \overline{d}$, $b \cdot d$ absorbe $a \cdot b \cdot d$ et $b \cdot c \cdot d$, $\overline{b} \cdot \overline{d}$ absorbe $\overline{b} \cdot c \cdot \overline{d}$ et $a \cdot \overline{b} \cdot \overline{d}$
   $c \cdot \overline{d}$ absorbe $a \cdot c \cdot \overline{d}$ et $b \cdot c \cdot \overline{d}$
   Il reste : $\overline{a} \cdot \overline{c} + \overline{a} \cdot b + \overline{b} \cdot \overline{c} + \overline{c} \cdot d + \overline{a} \cdot \overline{d} + b \cdot d + a \cdot b \cdot c + \overline{b} \cdot \overline{d} + c \cdot \overline{d}$
6. Nouveaux consensus (on n'a fait figurer qu'une seule fois chacun d'entre eux) :
   $\overline{a} \cdot \overline{c} + \overline{a} \cdot b + \overline{b} \cdot \overline{c} + \overline{c} \cdot d + \overline{a} \cdot \overline{d} + b \cdot d + a \cdot b \cdot c + \overline{b} \cdot \overline{d} + c \cdot \overline{d} + b \cdot c + a \cdot b \cdot d + b \cdot c \cdot \overline{d} + a \cdot c \cdot \overline{d}$
7. Suppression des multiples :
   $b \cdot d$ absorbe $a \cdot b \cdot d$, $c \cdot \overline{d}$ absorbe $a \cdot c \cdot \overline{d}$ et $b \cdot c \cdot \overline{d}$, $b \cdot c$ absorbe $a \cdot b \cdot c$
   Il reste : $\overline{a} \cdot \overline{c} + \overline{a} \cdot b + \overline{b} \cdot \overline{c} + \overline{c} \cdot d + \overline{a} \cdot \overline{d} + b \cdot d + \overline{b} \cdot \overline{d} + c \cdot \overline{d} + b \cdot c$
8. Un dernier tour de consensus montre que cette expression est stable par consensus.

A l'aide d'un diagramme de Karnaugh, on détermine les mintermes contenus dans chacun des monômes principaux.

On en déduit la FCD, et, dans le tableau qui suit, on fait apparaître les monômes principaux et les mintermes qu'ils contiennent :

monome \ minterme	m_0	m_1	m_2	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{13}	m_{14}	m_{15}
$P_1 = \overline{a} \cdot \overline{c}$	$\diamond$	$\diamond$		$\diamond$	$\diamond$								
$P_2 = \overline{a} \cdot b$				$\diamond$	$\diamond$	$\diamond$	$\diamond$						
$P_3 = \overline{b} \cdot \overline{c}$	$\diamond$	$\diamond$						$\diamond$	$\diamond$				
$P_4 = \overline{c} \cdot d$		$\diamond$			$\diamond$				$\diamond$		$\diamond$		
$P_5 = \overline{a} \cdot \overline{d}$	$\diamond$		$\diamond$	$\diamond$		$\diamond$							
$P_6 = b \cdot d$					$\diamond$		$\diamond$				$\diamond$		$\diamond$
$P_7 = \overline{b} \cdot \overline{d}$	$\diamond$		$\diamond$					$\diamond$		$\diamond$			
$P_8 = c \cdot \overline{d}$			$\diamond$			$\diamond$				$\diamond$		$\diamond$	
$P_9 = b \cdot c$						$\diamond$	$\diamond$					$\diamond$	$\diamond$

La première colonne, par exemple, s'interprète comme suit : dans toute forme (réduite ou non) prétendant représenter l'expression donnée au départ, il est nécessaire qu'un monôme au moins contienne le minterme m_0, puisque ce dernier figure dans la FCD.

Cette condition peut être réalisée en choisissant le monôme principal P_1, ou P_3, ou P_5, ou P_7. Elle peut être exprimée par l'équation booléenne $p_1 + p_3 + p_5 + p_7 = 1$, etc.

On obtient l'équation booléenne :
$(p_1 + p_3 + p_5 + p_7)(p_1 + p_3 + p_4)(p_5 + p_7 + p_8)(p_1 + p_2 + p_5)(p_1 + p_2 + p_4 + p_6)(p_2 + p_5 + p_8 + p_9)(p_2 + p_6 + p_9)(p_3 + p_7)(p_3 + p_4)(p_7 + p_8)(p_4 + p_6)(p_8 + p_9)(p_6 + p_9) = 1$

On supprime évidemment les conditions qui sont automatiquement réalisées lorsque d'autres le sont (si $p_3 + p_7$ vaut 1, alors $p_1 + p_3 + p_5 + p_7$ aussi), il reste
$(p_1 + p_2 + p_5)(p_3 + p_7)(p_3 + p_4)(p_7 + p_8)(p_4 + p_6)(p_8 + p_9)(p_6 + p_9) = 1$

On développe le produit, mais pas le premier facteur, car il est le seul à contenir les monômes principaux p_1, p_2 et p_5, donc on sait déjà qu'il faudra en prendre un (et un seul, pour une forme minimale...) des trois.

On obtient
$(p_1+p_2+p_5)(p_3+p_4p_7)(p_8+p_7p_9)(p_6+p_4p_9) = (p_1+p_2+p_5)(p_3p_8p_3p_7p_9+p_4p_7p_8+p_4p_7p_9)(p_6+p_4p_9) = (p_1+p_2+p_5)(p_3p_6p_8+p_3p_4p_8p_9+p_3p_6p_7p_9+p_3p_4p_7p_9+p_4p_6p_7p_8+p_4p_7p_8p_9+p_4p_6p_7p_9+p_4p_7p_9) = (p_1 + p_2 + p_5)(p_3p_6p_8 + p_3p_4p_8p_9 + p_3p_6p_7p_9 + p_4p_6p_7p_8 + p_4p_7p_9) = 1$

On constate qu'il est possible de réaliser la condition de la seconde parenthèse en ne choisissant que 3 monômes principaux : P_3, P_6 et P_8, ou encore P_4, P_7 et P_9 (les autres choix possibles en nécessitent 4).

En plus, il faut choisir l'un des trois de la première parenthèse, comme on l'a dit plus haut.

On obtient donc 6 formes minimales :

$$\left\{ \begin{array}{c} \overline{b} \cdot \overline{c} + b \cdot d + c \cdot \overline{d} \\ \text{ou} \\ \overline{c} \cdot d + \overline{b} \cdot \overline{d} + b \cdot c \end{array} \right\} \quad + \quad \left\{ \begin{array}{c} \overline{a} \cdot \overline{c} \\ \text{ou} \\ \overline{a} \cdot b \\ \text{ou} \\ \overline{a} \cdot \overline{d} \end{array} \right\}$$

Exercice 10.45. *On considère* $f(a, b, c, d) = \overline{a}\overline{b}c + \overline{a}c\overline{d} + a\overline{b}\overline{c}\overline{d} + a\overline{c}d + bcd$

- *Trouvez sa FCD.*
- *En déduire ses formes minimales.*

Exercice 10.46. *Utiliser la méthode des consensus pour obtenir toutes les formes minimales des fonctions booléennes suivantes :*

1. *Celles des précédents exemples et exercices.*
2. $\overline{d} \cdot e + \overline{a} \cdot c + b \cdot \overline{c} + a \cdot \overline{b} + a \cdot d \cdot e + \overline{a} \cdot d \cdot \overline{e}$

Fin du Chapitre

Chapitre 11

Calcul propositionnel

I Introduction

Les liens étroits entre logique et informatique ne sont pas récents, avec pour exemple la citation suivante de plus de 40 ans : "It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the last. The development of this relationship demands a concern for both applications and mathematical elegance" [McC64].

Ce chapitre met l'accent sur le calcul des propositions, qui est un des fondements de la logique classique, initié par Friedrich Ludwig Gottlob Frege (1848–1925). Ce chapitre contient des extraits de [CL93, Lip90, LBDG07].

II Les fondements de la logique des propositions

Qu'est-ce donc qu'un raisonnement ? Si l'on sait que tous les écureuils sont des rongeurs, que tous les rongeurs sont des mammifères, que tous les mammifères sont des vertébrés et que tous les vertébrés sont des animaux, on peut en déduire que tous les écureuils sont des animaux.[...].

Ce raisonnement est simple à l'extrême, mais sa structure ne diffère pas fondamentalement de celle d'un raisonnement mathématique. Dans les deux cas, le raisonnement est formé d'une suite de propositions dans laquelle chacune découle logiquement des précédentes, [...]. Dans ce cas, on applique la même règle trois fois. Cette règle permet, si l'on sait déjà que tous les Y sont des X et que tous les Z sont des Y, de déduire que tous les Z sont des X [Dow07].

Cette section formalise la notion de proposition (Sec. II.1), montre comment les propositions peuvent être connectées entre elles (Sec. II.2) et comment elles sont représentées syntaxiquement (Sec. II.3).

II.1 Les propositions

L'homme exprime son raisonnement par un discours, et ce discours utilise une langue (une langue naturelle, français, anglais,...). D'une manière générale, ce discours est articulé en phrases, d'un niveau de complexité variable, et c'est l'étude de ces « énoncés » que se propose de faire la logique.

DÉFINITION 11.1 (PROPOSITION). *Parmi tous les énoncés possibles qui peuvent être formulés dans une langue, on distingue ceux auxquels il est possible d'attribuer une « valeur de vérité » : vrai ou faux. Ces énoncés porteront le nom de* propositions *.* ♢

EXEMPLE 11.1. Ainsi, « Henri IV est mort assassiné en 1610 », « Napoléon Bonaparte a été guillotiné en 1852 » sont des propositions, puisqu'on peut leur attribuer une valeur de vérité (« vrai » pour la première, « faux » pour la seconde).

Le calcul que l'on étudie considère toujours comme acquises les vérités suivantes, élevées au rang d'axiomes.

Principe de non-contradiction : Une proposition ne peut être simultanément vraie et fausse.

Principe du tiers-exclu : Une proposition est vraie ou fausse (il n'y a pas d'autre possibilité).

II.2 Les connecteurs logiques

L'analyse logique d'une phrase (reconnue comme proposition) fait apparaître des sous-phrases qui constituent elles-mêmes des propositions. Ces « membres de phrases » sont reliés entre eux par des « connecteurs logiques », comme expliqué dans la partie suivante…

II.2.1 Analyse logique des propositions

Considérons l'énoncé : « J'ai obtenu une mauvaise note à cet examen parce que je n'ai pas assez travaillé ou parce que le cours est trop difficile ». On suppose qu'il est possible d'attribuer une valeur de vérité à cet énoncé « global », ce qui le classe parmi les propositions.

On peut alors mener l'analyse logique de cette phrase, de manière à en extraire les propositions (au sens grammatical du terme) : « J'ai obtenu une mauvaise note à cet examen », « je n'ai pas assez travaillé », « le cours est trop difficile », qui sont aussi des propositions au sens logique du terme.

Globalement, cet énoncé exprime que « ma mauvaise note » est conséquence de l'une (au moins) des deux causes suivantes :

- « mon manque de travail »,
- « un cours trop difficile », soit :

  (« mon manque de travail » ou « cours trop difficile ») entraîne « ma mauvaise note »

D'une manière générale, le calcul propositionnel ne se préoccupe que des valeurs de vérité, et pas du tout des liens sémantiques qui peuvent exister entre des propositions. Ces dernières sont reliées entre elles syntaxiquement par des connecteurs comme « ou » ou « entraîne ». Les connecteurs logiques sont donc des symboles qui permettent de produire des propositions (« plus complexes ») à partir d'autres propositions (« plus simples »). En calcul propositionnel, ils sont définis axiomatiquement à partir de leurs tables de vérité.

II.2.2 Tables de vérité des connecteurs logiques

Disjonction logique : Connecteur « ou », symbole $\vee$.

À partir de deux propositions P et Q, ce connecteur permet la construction de la nouvelle proposition (P ou Q) [notée $P \vee Q$], dont la valeur de vérité est définie par la table de vérité :

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

On remarque que $P \vee Q$ est fausse si et seulement si les deux propositions P et Q sont fausses.

REMARQUE 11.1. Dans le langage courant, le mot « ou » est souvent employé de deux façons distinctes :

- il est parfois utilisé avec le sens « les deux cas peuvent se produire » (comme ici) et,
- parfois avec le sens « p ou q , mais pas les deux » (e.g. « il ira à Paris ou à Marseille »).

Sauf indication contraire, le « ou » sera toujours employé avec cette première signification.

Conjonction logique : Connecteur « et », symbole $\wedge$.

À partir de deux propositions P et Q, ce connecteur permet la construction de la nouvelle proposition (P et Q) [notée $P \wedge Q$], dont la valeur de vérité est définie par la table de vérité :

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

On remarque que $P \wedge Q$ est vraie si et seulement si les deux propositions P et Q sont vraies.

Exercice 11.2. *Rappelez quels sont les différents comportements des commandes shell suivantes :*

- *commmande1 ; commande2*
- *commande1 && commande2*
- *commande1 || commande2*

Expliquez ces comportements à partir de ce qui précède.

Négation logique : Connecteur « non », symbole $\neg$.

À partir d'une proposition P, ce connecteur permet de construire la nouvelle proposition (non P) [notée $\neg P$], dont la valeur de vérité est définie par la table de vérité

P	$\neg P$
F	V
V	F

Implication logique : Connecteur « si… alors », symbole $\Rightarrow$.

À partir de deux propositions P et Q, ce connecteur permet la construction de la proposition (Si P, alors Q) [notée $P \Rightarrow Q$], dont la valeur de vérité est définie par la table de vérité :

P	Q	$P \Rightarrow Q$
F	F	V
F	V	V
V	F	F
V	V	V

REMARQUE 11.2. Lorsque la proposition P est fausse, la proposition « Si P, alors Q » est vraie, quelle que soit la valeur de vérité de la proposition Q,

Exercice 11.3. *Déterminer la valeur de vérité des propositions suivantes* dans le monde actuel *(c.-à-d. celui dans lequel nous vivons) :*

1. *« si la terre est plate, alors la lune est carrée ; »*
2. *« si le soleil tourne autour de la terre alors la terre est ronde »*
3. *« si la terre est ronde alors le soleil tourne autour de la terre »*
4. *« si vous étudiez la logique alors $E = m.c^2$ »*
5. *« si Napoléon est mort alors il a gagné la bataille de Waterloo »*
6. *« s'il pleut en ce moment alors il pleut en ce moment »*
7. *« si tous les hommes sont passionnés par la logique alors Dieu existe »*

8. *« si le Diable existe alors ceci est un exercice de logique »*

La manière de mener un raisonnement qui utilise éventuellement des propositions qui se présentent sous la forme d'implications logiques est l'objet de la théorie de la déduction qui sera étudiée plus loin.

Équivalence logique : Connecteur « si et seulement si », notation : $\iff$. À partir de deux propositions P et Q, ce connecteur permet la construction de la nouvelle proposition (P si et seulement si Q) [notée $P \iff Q$], dont la valeur de vérité est donnée par la table de vérité

P	Q	$P \iff Q$
F	F	V
F	V	F
V	F	F
V	V	V

REMARQUE 11.3. Même remarque que pour l'implication logique : l'équivalence logique de deux propositions fausses est une proposition vraie.

Exercice (corrigé) 11.4. *En notant M et C les affirmations suivantes :*
- *M = « Jean est fort en Maths »,*
- *C = « Jean est fort en Chimie »,*

représenter les affirmations qui suivent sous forme symbolique, à l'aide des lettres M et C et des connecteurs usuels.

1. *« Jean est fort en Maths mais faible en Chimie »*
2. *« Jean n'est fort ni en Maths ni en Chimie »*
3. *« Jean est fort en Maths ou il est à la fois fort en Chimie et faible en Maths »*
4. *« Jean est fort en Maths s'il est fort en Chimie »*
5. *« Jean est fort en Chimie et en Maths ou il est fort en Chimie et faible en Maths »*

Réponses :
1. $M \wedge (\neg C)$ *; 2.* $(\neg M) \wedge (\neg C)$ *; 3.* $M \vee (C \wedge \neg M)$ *; 4.* $C \Rightarrow M$ *; 5.* $(M \wedge C) \vee (\neg M \wedge Q)$.

Exercice 11.5. *En notant M, C et A les trois affirmations suivantes :*
- *M = « Pierre fait des Maths » ;*
- *C = « Pierre fait de la Chimie » ;*
- *A = « Pierre fait de l'Anglais ».*

Représenter les affirmations qui suivent sous forme symbolique, à l'aide des lettres M, C, A et des connecteurs usuels.

1. *« Pierre fait des Maths et de l'Anglais mais pas de Chimie »*
2. *« Pierre fait des Maths et de la Chimie mais pas à la fois de la Chimie et de l'Anglais »*
3. *« Il est faux que Pierre fasse de l'Anglais sans faire de Maths »*
4. *« Il est faux que Pierre ne fasse pas des Maths et fasse quand même de la chimie »*
5. *« Il est faux que Pierre fasse de l'Anglais ou de la Chimie sans faire des Maths »*
6. *« Pierre ne fait ni Anglais ni Chimie mais il fait des Maths »*

Exercice 11.6. *Énoncer la négation des affirmations suivantes en évitant d'employer l'expression : « il est faux que »*

1. *« S'il pleut ou s'il fait froid je ne sors pas »*

2. *« Le nombre 522 n'est pas divisible par 3 mais il est divisible par 7 »*
3. *« Ce quadrilatère n'est ni un rectangle ni un losange »*
4. *« Si Paul ne va pas travailler ce matin il va perdre son emploi »*
5. *« Tout nombre entier impair peut être divisible par 3 ou par 5 mais jamais par 2 »*
6. *« Tout triangle équilatéral a ses angles égaux à 60° »*

Exercice (corrigé) 11.7. *Quelles sont les valeurs de vérité des propositions suivantes ?*

1. *π vaut 4 et la somme des angles d'un triangle vaut 180°*
2. *π vaut 3,141592... implique que la somme des angles d'un triangle vaut 180°*
3. *π vaut 4 implique que la somme des angles d'un triangle vaut 182°*
4. *Il n'est pas vrai qu'un entier impair ne puisse pas être divisible par 6*
5. *Si 2 est plus grand que 3 alors l'eau bout à 100°C*
6. *Si 6 est plus petit que 7 alors 7 est plus petit que 6*
7. *Si 7 est plus petit que 6 alors 6 est plus petit que 7*
8. *84 est divisible par 7 implique que 121 est divisible par 11*
9. *Si $531^{617}+1$ est divisible par 7 alors $531^{617}+1$ est plus grand que 7*
10. *La décimale d de π qui porte le numéro 10^{400} est 3 implique que si d n'est pas 3 alors d est 3.*

Réponses :
1 F ; 2 V ; 3 V ; 4 F ; 5 V ; 6 F ; 7 V ; 8 V ; 9 V ; 10 V.

II.3 Variables et formules propositionnelles

Comme le calcul propositionnel ne s'occupe que des valeurs de vérité, il est possible, dans une expression logique, de remplacer chaque proposition donnée par un symbole (en général, une lettre de l'alphabet majuscule), ou *variable propositionnelle* et d'étudier ensuite les valeurs de vérité de l'expression en fonction des valeurs de vérité de ces symboles.

PROPRIÉTÉ 11.1 : Les règles (de syntaxe) qui permettent de former des *formules propositionnelles* sont les suivantes :
- toute variable propositionnelle est une formule propositionnelle ;
- si F et G sont des formules propositionnelles, alors $\neg(F)$, $(F)\vee(G)$, $(F)\wedge(G)$, $(F)\Rightarrow(G)$ et $(F)\iff(G)$ sont des formules propositionnelles.

REMARQUE 11.4. Ce ne sont plus des propositions, en ce sens qu'elles n'ont en général pas de valeur de vérité déterminée. Cette dernière est une fonction des valeurs de vérité des variables propositionnelles qui interviennent dans l'expression de la formule propositionnelle considérée.

Exercice 11.8. *A et B sont des variables propositionnelles, susceptibles de représenter n'importe quelle proposition. Formaliser, à l'aide de connecteurs logiques appropriés, les énoncés suivants :*

1. *« A si B »*
2. *« A est condition nécessaire pour B »*
3. *« A sauf si B »*
4. *« A seulement si B »*
5. *« A est condition suffisante pour B »*

6. *« A bien que B »*
7. *« Non seulement A, mais aussi B »*
8. *« A et pourtant B »*
9. *« A à moins que B »*
10. *« Ni A, ni B »*

Exercice 11.9. *Les variables propositionnelles N et T serviront, dans cet exercice, à représenter (respectivement) les propositions « Un étudiant a de bonnes notes » et « Un étudiant travaille ». À l'aide des variables propositionnelles N et T, formaliser les propositions suivantes (si, pour l'une ou l'autre d'entre elles, la traduction vous paraît impossible, dites-le et expliquez pourquoi) :*

1. *C'est seulement si un étudiant travaille qu'il a de bonnes notes.*
2. *Un étudiant n'a de bonnes notes que s'il travaille.*
3. *Pour un étudiant, le travail est une condition nécessaire à l'obtention de bonnes notes.*
4. *Un étudiant a de mauvaises notes, à moins qu'il ne travaille.*
5. *Malgré son travail, un étudiant a de mauvaises notes.*
6. *Un étudiant travaille seulement s'il a de bonnes notes.*
7. *À quoi bon travailler, si c'est pour avoir de mauvaises notes ?*
8. *Un étudiant a de bonnes notes sauf s'il ne travaille pas.*

Exercice 11.10. *Combien de lignes contient la table de vérité d'une formule propositionnelle qui dépend de n variables ?*

Lorsqu'on remplace, dans une formule propositionnelle, les variables propositionnelles par des propositions, l'assemblage obtenu est une proposition. Cependant, une formule propositionnelle n'est pas une proposition : $A \Rightarrow B$ n'est ni vrai ni faux.

PROPRIÉTÉ 11.2 (RÈGLES DE PRIORITÉ DES CONNECTEURS LOGIQUES) : Les conventions de priorité des connecteurs logiques sont les suivantes (par ordre de priorité décroissante) :
- la négation,
- la conjonction et la disjonction (au même niveau),
- l'implication et l'équivalence (au même niveau).

EXEMPLE 11.11. $\neg A \wedge B \Rightarrow C$ doit être interprété par $((\neg A) \wedge B) \Rightarrow C$ et $A \vee B \wedge C$ n'a pas de sens, car les deux connecteurs ont même niveau de priorité.

PROPRIÉTÉ 11.3 (ASSOCIATIVITÉ DES OPÉRATEURS $\vee$ ET $\wedge$) : Les opérateurs $\vee$ et $\wedge$ sont associatifs :
- $(A \vee B) \vee C = A \vee (B \vee C) = A \vee B \vee C$,
- $(A \wedge B) \wedge C = A \wedge (B \wedge C) = A \wedge B \wedge C$.

Mais le parenthésage est obligatoire quand $\vee$ et $\wedge$ se trouvent dans la même proposition, puisqu'il n'y a pas de priorité entre $\vee$ et $\wedge$: $(A \vee B) \wedge C \neq A \vee (B \wedge C)$.

REMARQUE 11.5. L'implication n'est pas associative : $A \Rightarrow (B \Rightarrow C) \neq (A \Rightarrow B) \Rightarrow C$. Donc les parenthèses sont obligatoires. Il en est de même pour $\Longleftrightarrow$, et a fortiori quand ces deux opérateurs sont mélangés dans une même proposition.

Exercice (corrigé) 11.12. *Quelles sont les façons de placer des parenthèses dans $\neg P \vee Q \wedge \neg R$ afin d'obtenir l'expression correcte d'une formule propositionnelle ? Déterminer la table de vérité de chacune des formules obtenues.*

Réponses :

1) $\neg P \vee (Q \wedge \neg R)$ *; 2)* $(\neg P \vee Q) \wedge \neg R$ *; 3)* $(\neg(P \vee Q)) \wedge \neg R$ *; 4)* $\neg(P \vee (Q \wedge \neg R))$ *; 5)* $\neg((P \vee Q) \wedge \neg R)$.

Tables de vérité :

P	Q	R	1	2	3	4	5
V	V	V	F	F	F	F	V
V	V	F	V	V	F	F	F
V	F	V	F	F	F	F	V
V	F	F	F	F	F	F	F
F	V	V	V	F	F	V	V
F	v	F	V	V	F	F	F
F	F	V	V	F	F	V	V
F	F	F	V	V	V	V	V

Exercice (corrigé) 11.13. *Construire les tables de vérité des formules propositionnelles suivantes :*

1. $\neg P \wedge Q$
2. $\neg P \Rightarrow P \vee Q$
3. $\neg(\neg P \wedge \neg Q)$
4. $P \wedge Q \Rightarrow \neg Q$
5. $(P \Rightarrow Q) \vee (Q \Rightarrow P)$
6. $(P \Rightarrow \neg Q) \vee (Q \Rightarrow \neg P)$
7. $(P \vee \neg Q) \wedge (\neg P \vee Q)$
8. $P \Rightarrow (\neg P \Rightarrow P)$

Réponse :

P	Q	1	2	3	4	5	6	7	8
V	V	F	V	V	F	V	F	V	V
V	F	F	V	V	V	V	V	F	V
F	V	V	V	V	V	V	V	F	V
F	F	F	F	F	V	V	V	V	V

Exercice 11.14. *Faire de même avec*

1. $(P \vee Q) \vee (\neg R)$
2. $P \vee (\neg(Q \wedge R))$
3. $(\neg P) \Rightarrow ((\neg Q) \vee R)$
4. $(P \vee R) \Rightarrow (R \vee (\neg P))$
5. $(P \Rightarrow (\neg Q)) \vee (Q \Rightarrow R)$
6. $(P \vee (\neg Q)) \Rightarrow ((\neg P) \vee R)$
7. $(P \Rightarrow (\neg R)) \vee (Q \wedge (\neg R))$
8. $(P \Rightarrow Q) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \Rightarrow R))$

III Sémantique du calcul propositionnel

Dans ce qui suit, on donne un sens aux symboles représentant les connecteurs logiques en fonction de la valeur de vérité des propositions de base (ainsi $\neg$ signifie non).

III.1 Fonctions de vérité

Soit F une formule propositionnelle, dans l'expression de laquelle interviennent les variables propositionnelles P_1, P_2, P_3, ..., P_n. À chacune de ces variables propositionnelles, on associe une variable booléenne (généralement la même lettre de l'alphabet, mais en minuscules), qui représente la valeur de vérité qu'elle peut prendre (faux ou vrai, F ou V, 0 ou 1).

DÉFINITION 11.2 (FONCTION DE VÉRITÉ DE F). *La fonction de vérité de F est la fonction booléenne Φ_F des n variables booléennes concernées, obtenue de la manière suivante :*

1. *Si F est de la forme P, où P est une variable propositionnelle, alors $\Phi_F(p) = p$.*
2. *Si F est de la forme $\neg G$, où G est une formule propositionnelle, alors $\Phi_F = \overline{\Phi_G}$.*
3. *Si F est de la forme $G \vee H$, où G et H sont des formules propositionnelles, alors $\Phi_F = \Phi_G + \Phi_H$.*
4. *Si F est de la forme $G \wedge H$, où G et H sont des formules propositionnelles, alors $\Phi_F = \Phi_G \cdot \Phi_H$.*
5. *Si F est de la forme $G \Rightarrow H$, où G et H sont des formules propositionnelles, alors $\Phi_F = \overline{\Phi_G} + \Phi_H$.*
6. *Si F est de la forme $G \iff H$, où G et H sont des formules propositionnelles, alors $\Phi_F = \overline{\Phi_G} \cdot \overline{\Phi_H} + \Phi_G \cdot \Phi_H$.*

REMARQUE 11.6. Soit $F = P \Rightarrow Q$, $G = \neg Q \Rightarrow \neg P$. On a $\Phi_F(p,q) = \overline{p} + q$ et $\Phi_G(p,q) = \overline{\Phi_{\neg Q}} + \Phi_{\neg P} = \overline{\overline{q}} + \overline{p} = \Phi_F(p,q)$. On remarque que les deux fonctions de vérités $\Phi_F(p,q)$ et $\Phi_G(p,q)$ sont identiques. On en déduit que $P \Rightarrow Q$ et $\neg Q \Rightarrow \neg P$ sont logiquement équivalentes.

$\neg Q \Rightarrow \neg P$ est appelée *implication contraposée* de l'implication $P \Rightarrow Q$.

EXEMPLE 11.15. Soit $F = A \vee \neg B \iff (B \Rightarrow C)$. On a alors :

$$\Phi_F(a,b,c) = \overline{a + \overline{b} \cdot \overline{b + c}} + (a + \overline{b}) \cdot (\overline{b} + c) = \overline{a} \cdot b \cdot b \cdot \overline{c} + \overline{b} + a \cdot c = \overline{b} + \overline{a} \cdot \overline{c} + a \cdot c.$$

REMARQUE 11.7. Il est clair que les « tables de vérité » des connecteurs logiques sont les mêmes que les tables des opérations booléennes sur $\{\textit{faux}, \textit{vrai}\}$

- de la négation booléenne (pour la négation logique),
- de la somme booléenne (pour la disjonction logique),
- du produit booléen (pour la conjonction logique),

Ainsi, la détermination de la valeur de vérité d'une proposition composée se ramène à un simple calcul en algèbre de Boole sur la fonction de vérité de la formule propositionnelle associée.

III.2 Formules propositionnelles particulières

On verra dans cette section deux formules particulières : les tautologies et les antilogies.

III.2.1 Tautologies

DÉFINITION 11.3 (TAUTOLOGIE). *Toute formule propositionnelle dont la fonction de vérité est la fonction référentielle est appelée* tautologie . ♢

Ainsi, une tautologie est une formule propositionnelle dont la fonction de vérité est indépendante des valeurs de vérité associées à ses variables. Autrement dit, quelle que soit la valeur de vérité des propositions par lesquelles on remplacerait les variables propositionnelles, la proposition obtenue serait vraie.

NOTATION : La notation utilisée pour marquer une tautologie F est $\models F$ (se lit : « F est une tautologie »).

EXEMPLE 11.16. Soit $F = A \Rightarrow A$. Comme $\Phi_F(a) = \overline{a} + a = 1$, on a $\models F$.

EXEMPLE 11.17. $F = (A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C))$.

$$\begin{array}{ll} \Phi_F(a,b,c) & = \\ \overline{\Phi_{A\Rightarrow C}}(a,c) + \overline{\Phi_{B\Rightarrow C}}(b,c) + \Phi_{A\vee B\Rightarrow C}(a,b,c) & = \\ \overline{\overline{a}+c} + \overline{\overline{b}+c} + \overline{a+b}+c & = \\ a\overline{c} + b\overline{c} + \overline{a}\overline{b} + c & = \\ a + b + \overline{a}\overline{b} + c & = 1 + c = 1 \end{array}$$

Il ne faudrait pas croire, au vu de ces exemples simples, que les tautologies se ramènent toutes à des trivialités totalement inintéressantes et indignes d'être énoncées. Ainsi, dans une théorie mathématique, tous les théorèmes sont des tautologies ; la reconnaissance de cette propriété n'est cependant pas toujours complètement évidente…

Exercice (corrigé) 11.18. *Les formules propositionnelles suivantes sont-elles des tautologies ?*

1. $(P \wedge Q) \Rightarrow P$
2. $(P \vee Q) \Rightarrow (P \wedge Q)$
3. $(P \wedge Q) \Rightarrow (P \vee Q)$
4. $P \Rightarrow (P \vee Q)$
5. $P \Rightarrow ((\neg P) \Rightarrow P)$
6. $P \Rightarrow (P \Rightarrow Q)$
7. $P \Rightarrow (P \Rightarrow P)$
8. $(P \Rightarrow Q) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \Rightarrow R))$

Réponses : 1., 3., 4., 5., 7. et 8. sont des tautologies.

Exercice 11.19. *Prouver les tautologies suivantes*

1. $\models A \Rightarrow (B \Rightarrow A)$
2. $\models (A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C))$
3. $\models A \Rightarrow (B \Rightarrow A \wedge B)$
4. $\models A \wedge B \Rightarrow A$ $\qquad\qquad$ $\models A \wedge B \Rightarrow B$
5. $\models A \Rightarrow A \vee B$
6. $\models B \Rightarrow A \vee B$
7. $\models (A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C))$
8. $\models (A \Rightarrow B) \Rightarrow ((A \Rightarrow \neg B) \Rightarrow \neg A)$
9. $\models \neg\neg A \Rightarrow A$

III.2.2 Antilogies

DÉFINITION 11.4 (ANTILOGIE). *Toute formule propositionnelle dont la fonction de vérité est la fonction nulle est appelée* antilogie *.* ♢

La proposition obtenue en remplaçant les variables par des propositions ne peut alors jamais être vraie.

EXEMPLE 11.20. Soit $F = A \wedge \neg A$. $\Phi_F(a) = a \cdot \overline{a} = 0$. Donc F est bien une antilogie.

Exercice 11.21. *Calculer les fonctions de vérité des formules propositionnelles suivantes, et dire s'il s'agit éventuellement de tautologies ou d'antilogies :*

1. $(A \Rightarrow B) \wedge (A \vee B) \Rightarrow B$
2. $(A \Rightarrow C) \wedge (B \Rightarrow D) \wedge (A \vee B) \Rightarrow C \vee D$
3. $\neg(A \wedge B) \vee \neg A \vee \neg B \Rightarrow C$
4. $(A \Rightarrow C) \vee (B \Rightarrow D) \Rightarrow (A \vee B \Rightarrow C \vee D)$
5. $(A \Rightarrow C) \wedge (B \Rightarrow D) \Rightarrow (A \wedge B \Rightarrow C \wedge D)$
6. $(A \wedge B) \vee (\neg A \wedge \neg C) \Rightarrow (B \Rightarrow C)$
7. $(\neg A \vee B) \wedge (C \Rightarrow (A \Longleftrightarrow B))$
8. $A \wedge \neg A \Rightarrow (B \vee C \Rightarrow (C \Rightarrow \neg A))$
9. $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$
10. $(A \Rightarrow C) \wedge (B \Rightarrow D) \wedge (\neg C \vee \neg D) \Rightarrow \neg A \vee \neg B$
11. $A \wedge (A \vee B) \Longleftrightarrow A$
12. $(\neg A \vee B \Rightarrow (A \Rightarrow \neg A \vee B)) \Longleftrightarrow (\neg A \vee B \Rightarrow (A \Rightarrow (A \Rightarrow B)))$
13. $(A \Rightarrow B) \wedge (A \vee C) \Rightarrow B \vee C$
14. $(A \Rightarrow B) \wedge (A \vee C) \Rightarrow (A \Rightarrow C)$

III.3 Conséquences logiques

Soit $\mathcal{F} = \{F_1, \ldots, F_n\}$ un ensemble de formules propositionnelles .

DÉFINITION 11.5 (CONSÉQUENCE LOGIQUE). *On dit que la formule propositionnelle A est une* conséquence logique *des formules propositionnelles $F_1, \ldots, F_n$ lorsque, chaque fois que les fonctions de vérité Φ_{F_1}, …, Φ_{F_n} prennent simultanément la valeur « vrai » (ou 1), il en est de même pour la fonction de vérité de la forme A.* ♢

NOTATION : On note ce résultat : $\{F_1, \ldots, F_n\} \models A$ (se lit : A est conséquence logique de $\{F_1, \ldots, F_n\}$).

EXEMPLE 11.22. On reconsidère l'ensemble des deux formules propositionnelles

$$\{P, P \Rightarrow Q\}$$

et on va montrer autrement que Q est conséquence logique de ces deux formules. Autrement dit, on va remontrer que : $\{P, P \Rightarrow Q\} \models Q$.

- $\Phi_P(p) = p$: prend la valeur 1 lorsque p prend la valeur 1.
- $\Phi_{P \Rightarrow Q}(p, q) = \overline{p} + q$: prend la valeur 1 lorsque $p = 0$ (quelle que soit la valeur de q) et lorsque $p = 1$ et $q = 1$.
- $\Phi_P(p)$ et $\Phi_{P \Rightarrow Q}(p, q)$ prennent simultanément la valeur 1 uniquement lorsque $p = 1$ et $q = 1$; dans ce cas, $\Phi_Q(q) = q = 1$ aussi. Donc Q est conséquence logique de $\{P, P \Rightarrow Q\}$.

Exercice 11.23. *Dans chacun des cas suivants, déterminer si le premier ensemble de formules a pour conséquence logique la deuxième formule :*

1	$\{P \wedge Q\}$	P
2	$\{(P \wedge Q) \vee R\}$	$P \ \wedge (Q \vee R)$
3	$\{(P \wedge Q) \Rightarrow R\}$	$(P \Rightarrow R) \wedge (Q \Rightarrow R)$
4	$\{P \Rightarrow (Q \vee R)\}$	$(P \Rightarrow Q) \vee (P \Rightarrow R)$
5	$\{A \Rightarrow (P \vee Q), \neg S \vee A\}$	$(\neg P \vee S) \Rightarrow Q$
5	$\{A \Rightarrow (B \wedge C), \neg C \vee D \vee R, R \Rightarrow \neg B\}$	$(A \wedge D) \Rightarrow \neg R$

Exercice 11.24. *Dans chacun des cas suivants, que peut-on dire d'une formule propositionnelle :*

1. *qui a pour conséquence logique une antilogie,*
2. *qui a pour conséquence logique une tautologie,*
3. *qui est conséquence logique d'une antilogie,*
4. *qui est conséquence logique d'une tautologie.*

Exercice 11.25. *La formule propositionnelle F étant fixée, que peut-on dire d'une formule propositionnelle G qui possède chacune des deux propriétés :*

- *$F \vee G$ est une tautologie,*
- *$F \wedge G$ est une antilogie.*

III.4 Formules équivalentes

DÉFINITION 11.6 (FORMULES ÉQUIVALENTES). *Si la formule propositionnelle G est conséquence logique de la formule propositionnelle F et si F est aussi conséquence logique de G, alors ces deux formules sont dites* équivalentes *(que l'on note $\approx$), soit :*

$$\{F\} \models G \text{ et } \{G\} \models F \text{ si et seulement si } F \approx G.$$

C'est cette notion de formules équivalentes qui autorise le remplacement d'une expression par une autre (équivalente, bien sûr) dans une formule propositionnelle.

REMARQUE 11.8. On est autorisé à remplacer $\neg\neg A$ par A, puisque ces formules sont équivalentes.

Exercice 11.26. *Dans chacun des cas suivants, dire si les deux formules propositionnelles inscrites sur la même ligne sont équivalentes :*

1	$\neg(\neg P)$	P
2	$P \wedge (P \Rightarrow Q)$	$P \wedge Q$
3	$P \Rightarrow Q$	$(\neg P) \vee (P \wedge Q)$
4	$P \Rightarrow Q$	$(\neg P) \Rightarrow (\neg Q)$
5	$P \vee Q$	$\neg((\neg P) \wedge (\neg Q))$
6	$P \wedge Q$	$\neg((\neg P) \vee (\neg Q))$
7	$\neg P$	$(\neg(P \vee Q)) \vee ((\neg P) \wedge Q)$
8	$P \Rightarrow (Q \Rightarrow R)$	$(P \Rightarrow Q) \Rightarrow R$
9	$P \Rightarrow (Q \wedge R)$	$(P \Rightarrow Q) \wedge (P \Rightarrow R)$
10	$P \Rightarrow (Q \vee R)$	$(P \Rightarrow Q) \vee (P \Rightarrow R)$
11	$(P \Rightarrow Q) \wedge (Q \Rightarrow P)$	$(P \wedge Q) \Rightarrow (P \wedge Q)$
12	$(P \wedge Q) \vee (Q \wedge R) \vee (R \wedge P)$	$(P \vee Q) \wedge (Q \vee R) \wedge (P \vee R)$

Exercice (corrigé) 11.27. *Soit F une formule propositionnelle dépendant de trois variables P, Q, R qui possède deux propriétés :*

– *$F(P,Q,R)$ est vraie si P,Q,R sont toutes les trois vraies,*

– *la valeur de vérité de $F(P,Q,R)$ change quand celle d'une seule des trois variables change.*

Construire la table de vérité de F, et déterminer une formule possible pour F.

Réponse : table de vérité

P	Q	R	F
V	V	V	V
V	V	F	F
V	F	V	F
F	V	V	F
V	F	F	V
F	F	V	V
F	V	F	V
F	F	F	F

Formule : $(P \wedge Q \wedge R) \vee (P \wedge \neg Q \wedge \neg R) \vee (\neg P \wedge \neg Q \wedge R) \vee (\neg P \wedge Q \wedge \neg R)$

Exercice 11.28. *Déterminer des formules propositionnelles F, G, H dépendant des variables P, Q, R qui admettent les tables de vérité :*

P	Q	R	F
V	V	V	V
V	V	F	F
V	F	V	V
V	F	F	F
F	V	V	F
F	V	F	V
F	F	V	V
F	F	F	V

P	Q	R	G
V	V	V	F
V	V	F	V
V	F	V	V
V	F	F	F
F	V	V	F
F	V	F	V
F	F	V	V
F	F	F	F

P	Q	R	H
V	V	V	V
V	V	F	V
V	F	V	V
V	F	F	F
F	V	V	F
F	V	F	V
F	F	V	V
F	F	F	V

III.5 Simplification du calcul des fonctions de vérité

III.5.1 Théorème de substitution

PROPRIÉTÉ 11.4 (THÉORÈME DE SUBSTITUTION) : Soit F une formule propositionnelle dans laquelle interviennent les variables propositionnelles P_1 , P_2 , P_3 ,… , P_n. Supposons que l'on remplace ces variables par des formules propositionnelles G_1 , G_2 , G_3 ,… , G_n ; la nouvelle formule propositionnelle obtenue est notée F^*.

Dans ces conditions : si $\models F$, alors $\models F^*$.

PREUVE F étant une tautologie, sa fonction de vérité ne dépend pas des valeurs de vérité des variables booléennes, qui peuvent donc être remplacées par n'importe quelle fonction booléenne. ■

Attention, la réciproque n'est pas vraie…

EXEMPLE 11.29. Soit $F = A \Rightarrow B$ et $F^* = P \wedge \neg P \Rightarrow Q$, obtenue à partir de F en remplaçant A par $P \wedge \neg P$ et B par Q. Comme $\Phi_{F^*}(p,q) = \overline{p \cdot \overline{p}} + q = \overline{0} + q = 1 + q = 1$, alors F^* est une tautologie. Cependant de $\Phi_F(a,b)$, on ne peut pas dire que F est une tautologie.

Exemple d'utilisation de ce résultat :

EXEMPLE 11.30. La formule propositionnelle

$$F^* = ((P \Rightarrow Q \wedge \neg R) \vee (\neg S \Longleftrightarrow T)) \Rightarrow ((P \Rightarrow Q \wedge \neg R) \vee (\neg S \Longleftrightarrow T)),$$

est compliquée puisqu'elle contient 5 variables propositionnelle. il y a donc 32 lignes à calculer pour obtenir les valeurs de la fonction de vérité. Cependant, il suffit de remarquer que F^* est obtenue à partir de $F = A \Rightarrow A$, qui est une tautologie ; donc F^* en est une aussi.

Ce résultat peut évidemment être appliqué aussi à des parties de formules propositionnelles, pour accélérer le calcul de leurs fonctions de vérité : si une partie d'une formule propositionnelle constitue à elle seule une tautologie, la partie correspondante de la fonction de vérité peut être avantageusement remplacée par 1.

III.5.2 Théorème de la validité

PROPRIÉTÉ 11.5 (THÉORÈME DE LA VALIDITÉ) : Soit $\{G_1, G_2, \ldots, G_n\}$ un ensemble de formules propositionnelles et H une formule propositionnelle ; alors :

$$\{G_1, G_2, \ldots, G_{n-1}\} \models G_n \Rightarrow H \text{ si et seulement si } \{G_1, G_2, \ldots, G_n\} \models H$$

PREUVE **Si.** Supposons $\{G_1, G_2, \ldots, G_n\} \models H$, c'est à dire, chaque fois que les formules de $\{G_1, G_2, \ldots, G_n\}$ sont vraies, H l'est aussi). Supposons que les formules de $\{G_1, G_2, \ldots, G_{n-1}\}$ soient vraies :

- Alors, si G_n est vraie, toutes les formules de $\{G_1, G_2, \ldots, G_n\}$ sont vraies, et donc, d'après l'hypothèse, H est vraie. Dans ce cas (voir table de vérité de l'implication logique), $G_n \Rightarrow H$ est vraie.
- Et si G_n n'est pas vraie, alors $G_n \Rightarrow H$ est vraie.

Seulement si. Supposons $\{G_1, G_2, \ldots, G_{n-1}\} \models G_n \Rightarrow H$. En d'autres termes, chaque fois que les formules de $\{G_1, G_2, \ldots, G_{n-1}\}$ sont vraies, $G_n \Rightarrow H$ est vraie. Regardons si H est une conséquence logique de $\{G_1, G_2, \ldots, G_n\}$ en distinguant selon que G_n est vraie ou pas.

- soit lorsque G_n n'est pas vraie, indépendamment de la valeur de vérité de H sur laquelle on ne peut alors rien dire, mais peu importe, puisque, dans ce cas, les formules de $\{G_1, G_2, \ldots, G_n\}$ ne sont pas toutes vraies, puisque G_n n'est pas vraie.
- soit lorsque G_n est vraie, et, dans ce cas, on sait que H est obligatoirement vraie aussi. Ceci se produit chaque fois que toutes les formules de $\{G_1, G_2, \ldots, G_n\}$ sont vraies, et, dans ce cas, H l'est aussi. Donc $\{G_1, G_2, \ldots, G_n\} \models H$. ∎

EXEMPLE 11.31 (EXEMPLE D'APPLICATION). Soit à montrer que :

$$\models (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)).$$

On pourrait bien entendu déterminer la fonction de vérité de cette formule. Mais, d'après le théorème précédent, la démonstration du résultat demandé est équivalente à celle de :

$$\{A \Rightarrow (B \Rightarrow C)\} \models (A \Rightarrow B) \Rightarrow (A \Rightarrow C).$$

Une nouvelle application de ce même théorème nous montre que la démonstration demandée est encore équivalente à celle de :

$$\{A \Rightarrow (B \Rightarrow C), (A \Rightarrow B)\} \models (A \Rightarrow C).$$

Et enfin à celle de :

$$\{A \Rightarrow (B \Rightarrow C), (A \Rightarrow B), A\} \models C.$$

Or les fonctions de vérité de $\{A \Rightarrow (B \Rightarrow C), (A \Rightarrow B), A\}$ sont

$$\begin{cases} \overline{a} + \overline{b} + c \\ \overline{a} + b \\ a \end{cases} \quad \text{qui valent sim. 1 quand} \quad \begin{cases} a = 1 \\ b = 1 \\ c = 1 \end{cases}$$

Ainsi C est vraie et on a terminé la démonstration.

Exercice 11.32. *Trois dirigeants d'une Société (Pierre P., Marc M. et Alain A.) sont prévenus de malversations financières ; au cours de l'enquête, l'agent du fisc enregistre leurs déclarations :*

- *Pierre P. : "Marc est coupable et Alain est innocent".*
- *Marc M. : "Si Pierre est coupable, Alain l'est aussi".*
- *Alain A. : "Je suis innocent, mais l'un au moins des deux autres est coupable".*

1. *Ces trois témoignages sont-ils compatibles ?*
2. *En supposant qu'ils sont tous les trois innocents, lequel a menti ?*
3. *En supposant que chacun dit la vérité, qui est innocent et qui est coupable ?*
4. *En supposant que les innocents disent la vérité et que les coupables mentent, qui est innocent et qui est coupable ?*

Exercice 11.33. *Simplifier le règlement suivant :*

- *Les membres de la Direction Financière doivent être choisis parmi ceux de la Direction Générale.*
- *Nul ne peut être à la fois membre de la Direction Générale et de la Direction Technique s'il n'est membre de la Direction Financière.*
- *Aucun membre de la Direction Technique ne peut être membre de la Direction Financière.*

Exercice 11.34. *Un inspecteur des services de santé visite un hôpital psychiatrique où des phénomènes étranges lui ont été signalés.*

Dans cet hôpital, il n'y a que des malades et des médecins, mais les uns comme les autres peuvent être sains d'esprit ou totalement fous. L'inspecteur doit faire sortir de l'hôpital les personnes qui n'ont rien à y faire, c'est à dire les malades sains d'esprit et les médecins totalement fous (quitte à les réintégrer ultérieurement en tant que malades...). Il part du principe que les personnes saines d'esprit ne disent que des choses vraies, alors que les personnes folles ne disent que des choses fausses.

Dans une salle, il rencontre deux personnes (appelons-les A et B pour préserver leur anonymat). A affirme que B est fou et B affirme que A est médecin.

1. *Après une intense réflexion, l'inspecteur fait sortir l'un des deux de l'hôpital. Lequel (et pourquoi ?)*
2. *Peut-il dire quelque chose au sujet de l'autre ?*

Exercice 11.35. *Le prince de Beaudiscours est dans un cruel embarras. Le voici au pied du manoir où la méchante fée Antinomie maintient prisonnière la douce princesse Vérité. Deux portes y donnent accès. L'une d'elles conduit aux appartements de la princesse, mais l'autre s'ouvre sur l'antre d'un dragon furieux. Le prince sait seulement que l'une de ces deux portes s'ouvre lorsqu'on énonce une proposition vraie, et l'autre si on énonce une proposition fausse.*

Comment peut-il délivrer la princesse ?

Exercice 11.36. *Que dire des raisonnements suivants ?*

1. *Si Jean n'a pas rencontré Pierre l'autre nuit, c'est que Pierre est le meurtrier ou que Jean est un menteur. Si Pierre n'est pas le meurtrier, alors Jean n'a pas rencontré Pierre l'autre nuit et le crime a eu lieu après minuit. Si le crime a eu lieu après minuit, alors Pierre est le meurtrier ou Jean n'est pas un menteur. Donc Pierre est le meurtrier*

2. *Manger de la vache folle est dangereux pour la santé ; manger du poulet aux hormones aussi, d'ailleurs. Quand on ne mange pas de la vache folle, on mange du poulet aux hormones. Notre santé est donc en danger.*
3. *Si je n'étudie pas, j'ai des remords. Mais si je ne vis pas à fond ma jeunesse, j'ai aussi des remords. Or je n'ai pas de remords. C'est donc que j'étudie tout en vivant à fond ma jeunesse.*
4. *Quand Marie est là, c'est qu'elle accompagne Paul ou Jean. Paul ne vient jamais en même temps que son cousin Serge. Si Jean et Serge viennent tous les deux, leur sœur Louise les accompagne. Si Louise se pointe, Raoul ne reste pas. Hier, Raoul et Serge étaient présents jusqu'au bout. Peut-on en conclure que Marie n'était pas présente ?*

III.6 Conclusion

Le calcul sur les fonctions de vérité paraît tout-à-fait satisfaisant et séduisant, lorsqu'il s'agit de calculer des valeurs de vérité ou d'examiner des conséquences logiques. Il est vrai qu'il est simple, nécessite un minimum de réflexion (très important dans le cas des ordinateurs !) et qu'il est très facile à programmer.

Mais, pour une formule propositionnelle qui comporte 10 variables propositionnelles (ce qui n'est pas beaucoup pour les problèmes que l'on cherche à programmer !), la table des valeurs de la fonction de vérité comporte $2^{10} = 1024$ lignes. Celui qui opère à la main a déjà démissionné. L'ordinateur démissionne un peu plus loin, certes, mais il finit aussi par avouer son incapacité :

- Sur les machines modernes, il n'est plus impossible d'envisager d'écrire et d'exécuter une « boucle vide » qui porte sur toutes les valeurs entières représentables sur 32 bits, donc de 0 à $2^{32} - 1$, le temps d'exécution est récemment devenu raisonnable.
- Il ne faut cependant pas exiger que ce temps demeure raisonnable dès qu'il s'agit d'exécuter un algorithme un peu compliqué. Et 32 variables constituent un nombre ridiculement petit pour un système expert, dans lequel les expressions offrent souvent une complexité qui n'a aucune commune mesure avec ce que l'on peut imaginer de plus compliqué…

Les « raccourcis » qui viennent d'être étudiés et qui permettent d'accélérer, voire de supprimer totalement, le calcul d'une fonction de vérité, sont plus utiles lorsque l'on opère « à la main » que pour la programmation d'algorithmes de logique.

Il faut donc garder en réserve la méthode des fonctions de vérité : celle-ci peut être très utile dans certains cas, essentiellement lorsque le problème peut être résolu « à la main », mais il faut aussi trouver une autre méthode pour songer à aborder des problèmes plus complexes.

Cette méthode, qui supprime toute référence aux valeurs de vérité, fait l'objet du chapitre suivant.

Fin du Chapitre

Chapitre 12

Calcul propositionnel : déductions syntaxiques

I Présentation de la théorie de la démonstration

Il s'agit ici d'explorer les mécanismes du raisonnement humain, c'est-à-dire les schémas de pensée qui nous permettent de décider d'agir d'une certaine manière, dans le but d'obtenir un certain résultat.

En théorie de la démonstration, une preuve est un objet mathématique. Elle est classiquement représentée comme une structure de donnée (liste, arbre,...). Elle est construite à l'aide d'axiomes logiques et de règles d'inférence. Plus formellement :

DÉFINITION 12.1 (AXIOME LOGIQUE). *Un* axiome logique *est une tautologie qui sert de « point de départ » aux déductions du système formel.* ♢

DÉFINITION 12.2 (RÈGLE D'INFÉRENCE). *Une* règle d'inférence *est une règle qui, à partir de formule(s)* prémisses*, produit une formule* conclusion*.* ♢

DÉFINITION 12.3 (THÉORÈME LOGIQUE). *Un résultat obtenu par une déduction correcte ou une suite de déductions correctes (c'est-à-dire qui utilisent explicitement les règles d'inférence autorisées) à partir des axiomes logiques et, éventuellement, d'autres résultats du même type déjà établis par ailleurs s'appelle un* théorème logique*.* ♢

On exprime que la formule F est un théorème par la notation : $\vdash F$, qui se lit « F est un théorème ».

DÉFINITION 12.4 (DÉMONSTRATION). *La chaîne de déductions qui conduit à un théorème logique est appelée* démonstration *de ce résultat.* ♢

Il est possible d'utiliser des formules logiques supplémentaires (autres que des axiomes ou des théorèmes) et de mener un raisonnement correct à partir de ces formules (et des axiomes et des théorèmes déjà connus). On parle alors de *démonstration sous hypothèses*. L'affirmation « la formule logique H est démontrée sous les hypothèses $G_1, G_2, \ldots, G_n$ » est notée $\{G_1, G_2, \ldots, G_n\} \vdash H$.

II Axiomes logiques et règles d'inférence du système formel « PR »

Il existe plusieurs systèmes d'axiomes qui permettent de définir la logique propositionnelle. Nous nous en tiendrons à l'ensemble des axiomes suivants, qui n'est ni minimal, ni contradictoire.

Axiomes relatifs à l'implication logique :
- Axiome 1 : $P \Rightarrow (Q \Rightarrow P)$
- Axiome 2 : $(P \Rightarrow Q) \Rightarrow ((P \Rightarrow (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R))$

Axiomes relatifs à la conjonction logique :
- Axiome 3 : $P \Rightarrow (Q \Rightarrow P \wedge Q)$
- Axiome 4 : $P \wedge Q \Rightarrow P$
- Axiome 5 : $P \wedge Q \Rightarrow Q$

Axiomes relatifs à la disjonction logique :
- Axiome 6 : $P \Rightarrow P \vee Q$
- Axiome 7 : $Q \Rightarrow P \vee Q$
- Axiome 8 : $(P \Rightarrow R) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \vee Q \Rightarrow R))$

Axiomes relatifs à la négation logique :
- Axiome 9 : $\neg\neg P \Rightarrow P$
- Axiome 10 : $(P \Rightarrow Q) \Rightarrow ((P \Rightarrow \neg Q) \Rightarrow \neg P)$

Axiomes relatifs à l'équivalence logique :
- Axiome 11 : $(P \Rightarrow Q) \Rightarrow ((Q \Rightarrow P) \Rightarrow (P \Longleftrightarrow Q))$
- Axiome 12 : $(P \Longleftrightarrow Q) \Rightarrow (P \Rightarrow Q)$
- Axiome 13 : $(P \Longleftrightarrow Q) \Rightarrow (Q \Rightarrow P)$

On définit la règle d'inférence du *modus ponens* (le mode « en posant, on pose ») :

$$\{P, P \Rightarrow Q\} \vdash Q$$

DÉFINITION 12.5 (SYSTÈME FORMEL « PR »). *Le système formel composé des 13 axiomes précédents et du modus ponens est nommé «* PR *».* ♢

III Démonstrations avec ou sans hypothèses

Un raisonnement logique peut être rédigé sous forme de démonstration, soit d'un théorème, soit d'une conséquence de certaines hypothèses.

III.1 Démonstration d'un théorème

La démonstration d'un théorème est constituée :

1. d'un en-tête, portant l'indication « Démonstration » ;
2. puis d'un certain nombre de lignes, numérotées (pour pouvoir être référencées dans la suite) ; Chacune comporte deux champs :
   (a) une formule, qui est le « résultat » de la ligne courante ;
   (b) la justification du résultat ;
3. une dernière ligne, non numérotée, qui porte l'en-tête « Conclusion ».

Dans une ligne, on peut avancer :
- un axiome en remplaçant éventuellement une variable par une formule ;
- un théorème considéré comme connu (dont la démonstration a été vue par ailleurs), en remplaçant éventuellement une variable par une formule ;

– un résultat de l'application d'une règle d'inférence sur des formules écrites dans les lignes précédentes.

EXEMPLE 12.1 (THÉORÈME DE RÉFLEXIVITÉ DE L'IMPLICATION). Soit P une formule propositionnelle. On souhaite démontrer le *théorème de réflexivité de l'implication* :

$$\vdash (P \Rightarrow P).$$

Démonstration :

1	$(P \Rightarrow (P \Rightarrow P)) \Rightarrow ((P \Rightarrow ((P \Rightarrow P) \Rightarrow P)) \Rightarrow (P \Rightarrow P))$	Axiome 2 ($P \Rightarrow P/Q, P/R$)
2	$P \Rightarrow (P \Rightarrow P)$	Axiome 1 (P/Q)
3	$(P \Rightarrow ((P \Rightarrow P) \Rightarrow P)) \Rightarrow (P \Rightarrow P)$	m.p. sur 2 et 1
4	$P \Rightarrow ((P \Rightarrow P) \Rightarrow P)$	Axiome 1 ($P \Rightarrow P/Q$)
5	$(P \Rightarrow P)$	m.p. sur 4 et 3

Conclusion : $\vdash (P \Rightarrow P)$

III.2 Démonstration sous hypothèses

Une démonstration sous hypothèses ...

1. commence par une première ligne qui comporte les mots « Démonstration sous les hypothèses » suivie de l'écriture de l'ensemble des hypothèses utilisées ;
2. puis, comme dans une démonstration de théorème, de lignes numérotées ... dans lesquelles peuvent figurer les mêmes éléments, auxquels il faut ajouter les hypothèses, dont on a le droit de se servir comme s'il s'agissait de résultats établis ;
3. une ligne de conclusion qui rappelle les hypothèses.

EXEMPLE 12.2 (MODUS (TOLLENDO) TOLLENS).
L'objectif est d'obtenir $\{P \Rightarrow Q\,,\, \neg Q\} \vdash \neg P$ qui est plus connu sous le nom « modus (tollendo) tollens ».

Soit P et Q des formules propositionnelles quelconques, montrons $\neg P$ sous les hypothèses $P \Rightarrow Q$ et $\neg Q$:

Démonstration sous les hypothèses $\{P \Rightarrow Q\,,\, \neg Q\}$

1	$(P \Rightarrow Q) \Rightarrow ((P \Rightarrow \neg Q) \Rightarrow \neg P)$	Axiome 10
2	$P \Rightarrow Q$	Hypothèse 1
3	$(P \Rightarrow \neg Q) \Rightarrow \neg P$	m.p. sur 2 et 1
4	$\neg Q \Rightarrow (P \Rightarrow \neg Q)$	Axiome 1
5	$\neg Q$	Hypothèse 2
6	$(P \Rightarrow \neg Q)$	m.p. sur 5 et 4
7	$\neg P$	m.p. sur 6 et 3

Conclusion : $\{P \Rightarrow Q\,,\, \neg Q\} \vdash \neg P$.

IV Théorème de la déduction

Les démonstrations sont souvent considérablement simplifiées par l'utilisation du théorème de la déduction donné ci-après.

PROPRIÉTÉ 12.1 (THÉORÈME DE LA DÉDUCTION) : Ce théorème s'énonce par :

$$\{G_1, G_2, \ldots, G_n\} \vdash H \text{ si et seulement si } \{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$$

PREUVE La démonstration s'effectue par récurrence sur la longueur de la déduction.

Seulement si. Hypothèse : $\{G_1, G_2, \ldots, G_n\} \vdash H$. Soit p la longueur de la déduction qui amène à H.

- Si $p = 1$: une « déduction de longueur 1 » n'autorise l'écriture que d'une seule ligne. Cela signifie donc que l'on peut directement écrire H dans celle-ci. Ce n'est possible que si H est un axiome ou une hypothèse.
  - Si H est un axiome :

    Démonstration sous les hypothèses $\{G_1, G_2, \ldots, G_{n-1}\}$:

1	$H \Rightarrow (G_n \Rightarrow H)$	Axiome 1
2	H	Axiome j
3	$G_n \Rightarrow H$	m.p. sur 2 et 1

    Conclusion : $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$

    Dans ce premier cas : $\{G_1, G_2, \ldots, G_n\} \vdash H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$ (Les hypothèses ne sont en fait pas utilisées, donc elles n'interviennent pas).
  - Si H est l'une des hypothèses $\{G_1, G_2, \ldots, G_{n-1}\}$, posons $H = G_i$ $(0 < i < n)$:

    Démonstration sous les hypothèses $\{G_1, G_2, \ldots, G_{n-1}\}$:

1	$G_i \Rightarrow (G_n \Rightarrow G_i)$	Axiome 1
2	G_i	Hypothèse
3	$G_n \Rightarrow G_i$	m.p. sur 2 et 1

    Conclusion : $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$

    Dans ce deuxième cas : $\{G_1, G_2, \ldots, G_n\} \vdash H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$ (Seule l'hypothèse G_i a été utilisée, les autres ne sont en fait pas utilisées, elles n'interviennent pas).
  - Si H est l'hypothèse G_n : Alors on sait que : $\vdash G_n \Rightarrow G_n$ (voir paragraphe précédent).

    Dans ce troisième cas : $\{G_1, G_2, \ldots, G_n\} \vdash H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$.

  Conclusion : la propriété est vraie pour $p = 1$.
- Hypothèse de récurrence : Soit p un entier tel que la propriété soit vraie pour tous les entiers i de 1 à p (récurrence généralisée) ; on suppose que la longueur de la déduction qui mène à H est $(p + 1)$.
  - Si H est un axiome ou l'une des hypothèses, le cas se traite comme ci-dessus.
  - Dans le cas contraire, H ne peut avoir été obtenu que par un « modus ponens » sur des formules P et $P \Rightarrow H$. Ces formules ont elles-mêmes été obtenues par des déductions de longueur inférieure ou égale à p, donc on peut dire que

    $\{G_1, G_2, \ldots, G_n\} \vdash P$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow P$ et que

    $\{G_1, G_2, \ldots, G_n\} \vdash P \Rightarrow H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow (P \Rightarrow H)$.

    Démonstration sous les hypothèses $\{G_1, G_2, \ldots, G_{n-1}\}$:

1	$G_n \Rightarrow P$	Résultat intermédiaire 1
2	$G_n \Rightarrow (P \Rightarrow H)$	Résultat intermédiaire 2
3	$(G_n \Rightarrow P) \Rightarrow ((G_n \Rightarrow (P \Rightarrow H)) \Rightarrow (G_n \Rightarrow H))$	Axiome 2
4	$(G_n \Rightarrow (P \Rightarrow H)) \Rightarrow (G_n \Rightarrow H)$	m.p. sur 1 et 3
5	$G_n \Rightarrow H$	m.p. sur 2 4 4

Conclusion $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$,
et donc : $\{G_1, G_2, \ldots, G_n\} \vdash H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$, lorsque la déduction est de longueur $p+1$.

Si. Réciproquement, supposons $\{G_1, G_2, \ldots, G_{n-1}\} \vdash (G_n \Rightarrow H)$. Alors,
Démonstration sous les hypothèses $\{G_1, G_2, \ldots, G_n\}$

1	$G_n \Rightarrow H$	Résultat obtenu sous les hyp. $\{G_1, G_2, \ldots, G_{n-1}\}$
2	G_n	Hypothèse n
3	H	m.p. sur 2 et 1

Conclusion $\{G_1, G_2, \ldots, G_n\} \vdash H$
Donc : $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \Rightarrow H$ entraîne $\{G_1, G_2, \ldots, G_n\} \vdash H$. ∎

EXEMPLE 12.3. On cherche à montrer le *théorème d'échange des prémisses* :

$$\vdash (P \Rightarrow (Q \Rightarrow R)) \Rightarrow (Q \Rightarrow (P \Rightarrow R))$$

La démonstration de ce théorème équivaut à la démonstration sous hypothèses

$$\{P \Rightarrow (Q \Rightarrow R)\} \vdash (Q \Rightarrow (P \Rightarrow R)),$$

équivalente à la démonstration sous hypothèses

$$\{P \Rightarrow (Q \Rightarrow R), Q\} \vdash (P \Rightarrow R),$$

elle-même équivalente à la démonstration sous hypothèses

$$\{P \Rightarrow (Q \Rightarrow R), Q, P\} \vdash R$$

qui est obtenu comme suit :
Démonstration sous les hypothèses $\{P \Rightarrow (Q \Rightarrow R), Q, P\}$

1	P	Hypothèse
2	$P \Rightarrow (Q \Rightarrow R)$	Hypothèse
3	$Q \Rightarrow R$	m.p. sur 1 et 2
4	Q	Hypothèse
5	R	m.p. sur 4 et 3

Conclusion $\{P \Rightarrow (Q \Rightarrow R), Q, P\} \vdash R$.

REMARQUE 12.1. Cette méthode est beaucoup plus rapide que celle qui consisterait à essayer de démontrer ce théorème à partir des axiomes et de la règle d'inférence.

REMARQUE 12.2. L'utilisation principale du théorème de la déduction consiste à remplacer la démonstration d'implication par des déductions sous hypothèses.

V Quelques théorèmes classiques et quelques règles d'inférence annexes

Au théorème de réflexivité de l'implication ($\vdash\ P \Rightarrow P$) et au théorème d'échange des prémisses ($\vdash (P \Rightarrow (Q \Rightarrow R)) \Rightarrow (Q \Rightarrow (P \Rightarrow R))$) on ajoute ceux qui suivent.

PROPRIÉTÉ 12.2 (THÉORÈME DE TRANSITIVITÉ DE L'IMPLICATION) : Soit P et Q deux formules propositionnelles quelconques.

$$\vdash\ (P \Rightarrow Q) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \Rightarrow R))$$

Exercice 12.4. *Effectuer la démonstration du théorème.*

DÉFINITION 12.6 (CONTRAPOSÉE). *L'implication $\neg Q \Rightarrow \neg P$ est appelée* contraposée *de l'implication $P \Rightarrow Q$.* ♢

PROPRIÉTÉ 12.3 (THÉORÈME DE LA CONTRAPOSÉE) : Soit P et Q deux formules propositionnelles quelconques.

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

Exercice 12.5. *Effectuer la démonstration du théorème.*

PROPRIÉTÉ 12.4 (THÉORÈME DE LA CONTRADICTION) : Soit P et Q deux formules propositionnelles quelconques.

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

Intuitivement, cela signifie que si $\neg P$ et P sont établies, alors on peut en déduire n'importe quoi (Q).

Exercice 12.6. *Effectuer la démonstration.*

On introduit une règle permettant de s'abstraire de l'application de deux modus ponens sur l'axiome 8. En effet, considérons l'axiome 8 :

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

En appliquant deux fois de suite le théorème de la déduction, il est équivalent à la déduction :

$$\{P \Rightarrow R\,,\, Q \Rightarrow R\}\ \vdash\ P \vee Q \Rightarrow R$$

que l'on peut utiliser sous cette forme comme règle d'inférence annexe : elle s'appelle *règle de disjonction des cas*.

PROPRIÉTÉ 12.5 (RÈGLE DE DISJONCTION DES CAS) : On a

$$\{P \Rightarrow R\,,\, Q \Rightarrow R\}\ \vdash\ P \vee Q \Rightarrow R$$

Pour finir, en appliquant deux fois de suite le théorème de la déduction à l'axiome 10 :

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

il est équivalent à la déduction

$$\{P \Rightarrow Q\,,\ P \Rightarrow \neg Q\} \vdash\ \neg P$$

que l'on peut utiliser sous cette forme comme règle d'inférence annexe : elle s'appelle *règle de réduction à l'absurde*.

PROPRIÉTÉ 12.6 (RÈGLE DE RÉDUCTION À L'ABSURDE) : On a

$$\{P \Rightarrow Q\,,\ P \Rightarrow \neg Q\} \vdash\ \neg P$$

Exercice 12.7 (Démonstrations avec ou sans hypothèses). *Démontrer les théorèmes logiques suivants (seuls les axiomes, règles d'inférence, règles d'inférence annexes et théorèmes du cours sont autorisés).*

1. $\vdash (P \Rightarrow (Q \Rightarrow R)) \Longleftrightarrow (P \wedge Q \Rightarrow R)$
2. $\vdash (P \Rightarrow Q) \Longleftrightarrow (\neg Q \Rightarrow \neg P)$
3. $\vdash P \Longleftrightarrow \neg\neg P$
4. $\vdash P \wedge (Q \vee R) \Longleftrightarrow (P \wedge Q) \vee (P \wedge R)$
5. $\vdash P \vee (Q \wedge R) \Longleftrightarrow (P \vee Q) \wedge (P \vee R)$
6. $\{P \vee R, P \Rightarrow Q, R \Longleftrightarrow S\} \vdash Q \vee S$
7. $\{P \wedge \neg S, Q \vee \neg R, S \Rightarrow R\} \vdash (P \Rightarrow Q) \vee (R \Rightarrow S)$

VI Théorèmes de complétude du calcul propositionnel

On a jusqu'à maintenant deux points de vue :

1. La théorie des valeurs de vérité, avec ses
   - tables de vérités,
   - fonctions de vérités,
   - tautologie, conséquence, hypothèse.
2. La théorie de la démonstration, avec ses
   - axiomes,
   - règles d'inférence,
   - démonstrations (ou démonstrations sous hypothèses).

On peut se demander si les résultats obtenus dans chacune des deux théories sont identiques : une formule propositionnelle est-elle démontrable si et seulement si elle est une tautologie ?

Un sens est immédiat, c'est le « seulement si » : toute proposition démontrée résulte d'un axiome ou de l'application d'une règle sur des propositions déjà démontrées. On peut facilement de vérifier que les axiomes fournissent des tautologies et que les règles conservent les tautologies. Toute proposition démontrée est donc une tautologie. On dit que le système déductif PR est *correct*. L'autre sens la démonstration qui consiste à vérifier que toute tautologie admet une démonstration dans PR est un peu plus complexe et admise. Pour ce sens on dit que PR est *complet*.

On retiendra les théorèmes suivants (abusivement nommés théorèmes de complétude).

PROPRIÉTÉ 12.7 (THÉORÈME DE COMPLÉTUDE) : tout théorème est une tautologie et réciproquement, soit :

$$\vdash F \text{ si et seulement si } \models F$$

.

PROPRIÉTÉ 12.8 (THÉORÈME DE COMPLÉTUDE GÉNÉRALISÉ) : On a

$$\{P_1, P_2, \ldots, P_n\} \vdash Q \text{ si et seulement si } \{P_1, P_2, \ldots, P_n\} \models Q$$

Fin du Chapitre

Chapitre 13

Calcul des prédicats

I Introduction

I.1 Introduction aux « prédicats »

En logique des propositions, « Pierre est le père de Marc » ne comporte aucun connecteur logique : elle ne peut se formaliser que par une variable propositionnelle : A. La proposition « Jean est le père de Sylvie » ne peut être formalisée en logique des propositions que par une variable propositionnelle, B. Après la formalisation, on se retrouve avec deux variables propositionnelles A et B, sans lien aucun entre elles, alors qu'il est évident que ces deux propositions évoquent un même lien de parenté entre des individus différents. Il apparaît primordial de créer un langage qui permettrait de décrire des propriétés accordées à des individus.

I.2 Introduction à l'« univers du discours »

La valeur de l'expression « x possède une racine carrée » dépend de x et de l'*univers* dans lequel on fait évoluer x. Cette expression est

- toujours vraie si l'univers du discours est l'ensemble des réels positifs ;
- toujours fausse si l'univers du discours est l'ensemble des réels strictement négatifs ;
- vraie pour certaines valeurs si l'univers du discours est l'ensemble des entiers naturels.

I.3 Introduction à la « quantification »

Considérons les propositions « tous les étudiants sont sérieux » et « certains étudiants sont sérieux ».

La propriété évoquée (« être sérieux ») est accordée, par ces propositions, non pas seulement à un individu bien précis, mais à certains individus, considérés dans leur ensemble, ou à toute une catégorie d'individus. Ce troisième exemple suggère la notion de *quantification* d'une variable (tous les…, certains…).

II Définitions

II.1 Termes

Le calcul des prédicats fait intervenir des variables, qui prennent des valeurs dans un certain ensemble appelé univers du discours qui par la suite sera souvent noté U.

DÉFINITION 13.1 (SYMBOLE FONCTIONNEL). *Soit f une fonction de $U_1 \times U_2 \times \ldots \times U_n$ dans U'. Le symbole fonctionnel f est dit d'*arité n *et on note f_n.* ♢

Lorsque l'expression $f(x_1, \ldots, x_n)$ ne dépend pas de $x_1, \ldots, x_n$, on peut la remplacer par une constante c de l'univers.

DÉFINITION 13.2 (TERME). *Un* terme *est défini de manière récursive par :*
- *une variable,*
- *une constante,*
- *l'expression $f(t_1, t_2, \ldots, t_n)$ si $t_1, t_2, \ldots, t_n$ sont des termes et f un symbole fonctionnel d'arité n* ♢

II.2 Prédicats et atomes

On cherche à formaliser les relations qui peuvent lier des individus de l'univers du discours. Par exemple, « 22 est le double de 11 », « 46 est le double de 45 » sont des propositions qui évoquent la relation « être le double de » entre deux constantes à chaque fois.

DÉFINITION 13.3 (SYMBOLE PRÉDICATIF). *Soit p une fonction de $U_1 \times U_2 \times \ldots \times U_n$ dans $\{$vrai, faux$\}$. Le symbole prédicatif p d'*arité n *(notée p_n) est aussi nommé* prédicat p. ♢

DÉFINITION 13.4 (ATOME). *Un atome est de la forme $p(t_1, ..., t_n)$, où p est un symbole de prédicat d'arité n et $t_1, \ldots, t_n$ sont des termes.* ♢

III Quantificateurs

III.1 Quantificateur universel

Considérons la proposition « Tous les étudiants travaillent les mathématiques » pour l'analyser du point de vue du calcul des prédicats. On peut exprimer ceci à l'aide d'un prédicat binaire qui peut être formalisé par *travaille*(x, y), et qui signifie qu'un individu x travaille une certaine matière y.

La notation utilisée pour cette quantification est « $\forall$ » qui représente un A à l'envers (pour *All*). Ainsi on représente la proposition ci dessus par

$$\forall x \,.\, \textit{travaille}(x, \textit{maths}).$$

DÉFINITION 13.5 (QUANTIFICATEUR UNIVERSEL). $\forall$ *est un symbole de quantificateur, appelé le* quantificateur universel*. On dit alors que $\forall x$ est le quantificateur universel de la variable x.* ♢

REMARQUE 13.1. La présence du symbole de quantification est indispensable pour donner du sens aux formules avec variables.
- $\forall x \,.\, \textit{travaille}(x, \textit{maths})$ est une proposition. La variable x est dite *liée* au quatificateur $\forall$;
- *travaille*$(x,$ *maths*$)$ n'est pas une proposition : on ne peut pas lui attribuer une valeur de vérité. Il est nécessaire de substituer x par un terme distinguant un individu pour que cela le devienne (c.-à-d. *travaille*$(Jean,$ *maths*$)$ ou bien *travaille*(*fils*(*Pierre*), *maths*)). Dans cette expression, la variable x est dite *libre*.

Exercice 13.1. *Pour chacune des formules A suivantes, préciser l'ensemble Var(A) des variables de A, l'ensemble Varliees(A) des occurences liées de Var(A), l'ensemble Varlibres(A) des occurences libres de Var(A).*

1. $A \equiv P(f(x, y)) \vee \forall z \,.\, Q(a, z)$
2. $A \equiv (\forall x \,.\, P(x, y, z)) \vee (\forall z \,.\, Q(z) \Rightarrow R(z))$
3. $A \equiv \forall x \,.\, (\forall y \,.\, P(x, y) \Rightarrow \forall z \,.\, Q(x, y, z))$

III.2 Quantificateur existentiel

Dans la proposition « Tous les étudiants travaillent au moins une matière », intervient le même prédicat binaire *travaille*(x, y) ; on remarque qu'aucun étudiant n'est explicitement nommé, pas plus que la matière qu'il travaille. Si le « Tous » induit une quantification universelle, le « au moins une matière » signifie qu'il existe une matière travaillée par cet étudiant. Il suffit alors d'introduire le quantificateur existentiel, représenté par un E (première lettre de *Exists*) retourné ($\exists$).

Cet exemple se traduit alors par

$$\forall x \,.\, \exists y \,.\ \textit{travaille}(x, y)$$

où la variable x est liée par le quantificateur universel $\forall x$ et la variable y est liée par le quantificateur existentiel $\exists y$.

Exercice 13.2. *Formalisez les affirmations suivantes, en utilisant uniquement les prédicats indiqués, les connecteurs logiques et les quantificateurs.*

1. *Personne n'est parfait (*$p(x)$ *représente que* x *est parfait).*
2. *0 est multiple de chaque nombre entier (*$e(x)$ *et* $m(x, y)$ *signifient respectivement que* x *est un entier et que* x *est un multiple de* y*).*
3. *les absents n'ont pas tous tort (*$a(x)$ *et* $t(x)$ *signifient respectivement que* x *est absent et que* x *a tort).*

Exercice 13.3. *Soit les prédicats curé*(x) *(*x *est un curé), vélo*(y) *(*y *est un vélo) et possède*(x, y) *(*x *possède* y*).*

Traduisez en langage courant les formules quantifiées suivantes :

1. $\forall x$ *.vélo*$(x) \Rightarrow (\exists z \,.\, \textit{curé}(z) \wedge \textit{possède}(z, x))$.
2. $\forall x$ *.curé*$(x) \Rightarrow (\forall y, z \,.\, (\textit{velo}(y) \wedge \textit{velo}(z) \wedge y \neq z) \Rightarrow (\neg \textit{possède}(x, y) \vee \neg \textit{possède}(x, z)))$
3. $\exists x \,.\, \textit{curé}(x) \wedge (\forall y \,.\, \textit{vélo}(y) \Rightarrow \neg \textit{possède}(x, y))$.

III.3 Alternance de quantificateurs

Attention : *il est interdit d'intervertir des quantificateurs de symboles différents*.

EXEMPLE 13.4.

- $\forall x \,.\, \exists y \,.\, \textit{travaille}(x, y)$ signifie que tout étudiant travaille une matière (au moins). Autrement dit, la matière travaillée dépend de cet étudiant, elle n'est éventuellement pas la même pour tous les étudiants.
- La formule $\exists y \,.\, \forall x \,.\ \textit{travaille}(x, y)$ signifie qu'il y a une matière que tous les étudiants travaillent. La différence fondamentale avec le cas précédent est que, dans cette dernière affirmation, on affirme que tous les étudiants travaillent la même matière.

Exercice 13.5. *Dans cet exercice, nous considérons qu'aimer est une relation binaire (non nécessairement symétrique). On note aime*(x, y) *si la personne* x *aime la personne* y*.*

1. *Représentez, à l'aide de propositions logiques, les phrases suivantes :*
   (a) *Quelqu'un aime Valentine.*
   (b) *Personne n'aime Quentin.*
   (c) *Toute personne aime quelqu'un.*
   (d) *Quelqu'un est aimé de tous.*
   (e) *Toute personne s'aime.*
   (f) *Il n'y a personne qui ne s'aime pas lui-même.*

2. *Les deux propositions suivantes sont-elles équivalentes ?*

$$(\exists x\,.\,\mathit{aime}(x, \mathit{caroline})) \wedge (\exists x\,.\,\mathit{aime}(\mathit{caroline}, x))$$

et

$$\exists x\,.\,\mathit{aime}(x, \mathit{caroline}) \wedge \mathit{aime}(\mathit{caroline}, x)$$

Exercice 13.6. *Formaliser les propositions suivantes dans le calcul des prédicats.*

1. *« Il existe une planète plus petite que toutes les autres ; »*
2. *« Il y a une planète plus grande que tout objet du système solaire ; »*
3. *« Toute planète a une planète plus proche du soleil qu'elle ; »*
4. *« Certaines planètes sont plus petites que Neptune, d'autres pas ; »*
5. *« Il n'y a pas de planète qui soit plus grande qu'une autre tout en étant plus proche du soleil qu'elle ; »*
6. *« Tout ce qui tourne autours de la Terre est plus petit que toutes les planètes ; »*
7. *« Certaines planètes sont plus grosses que Neptune, mais aucune n'est plus éloignée du soleil qu'elle ; »*
8. *« Certaines planètes sont plus grosses que Neptune tout en étant plus éloignées du soleil qu'elle. »*

III.4 Portée d'un quantificateur

DÉFINITION 13.6 (PORTÉE D'UN QUANTIFICATEUR). *La* portée d'un quantificateur *dans une formule du calcul des prédicats est la partie de cette formule couverte par ce quantificateur.* ♢

Par convention, dans l'écriture d'une formule, un quantificateur est prioritaire sur tout connecteur logique. Sa portée est donc généralement clairement délimitée par une paire de parenthèses, avant le quantificateur jusqu'après la formule elle-même.

Exercice 13.7. *Dans cet exercice f_n signifie que le symbole f (propositionnel ou prédicatif) est d'arité n. On considère un langage $\mathcal{L} = \{f_1, g_1, h_2, R_1, S_2, T_2, =_2\}$ et les expressions suivantes :*

- $\varphi_1 : \exists x\,.\,((\forall y\,.\,(\exists z\,.\,R(x)))) \vee (\exists y\,.\,(\neg(\forall z\,.\,(S(h(x,z),x)))))$
- $\varphi_2 : (\forall x\,.\,(T(f(x),y)) \Rightarrow (\neg(\exists x\,.\,(f(x,y)))))$
- $\varphi_3 : (\forall z\,.\,T(x,y) \Rightarrow (\exists y\,.\,((\forall x'\,.\,\neg(f(x) = y))) \vee T(y,z)))$
- $\varphi_4 : (\forall x\,.\,(\exists y\,.\,((g(y) = x) \vee (\neg T(y,y))))) \Rightarrow (\exists y\,.\,(\forall x\,.\,(T(y,g(x)))))$

1. *Parmi ces expressions, lesquelles sont des formules de $\mathcal{L}$?*
2. *Pour celles qui sont des formules, supprimer les parenthèses inutiles.*
3. *Déterminer les occurrences liées des variables dans les formules.*

Exercice 13.8. *Classer les huit propositions suivantes dans les deux premières colonnes libres du tableau ci-dessous de manière à ce que les relations annoncées soient vérifiées.*

Compléter le tableau avec les formules correspondantes.

1. *Dans cette assemblée, tout le monde parle l'anglais et le français.*
2. *Dans cette assemblée, tout le monde parle l'anglais ou le français.*
3. *Dans cette assemblée, tout le monde parle l'anglais et tout le monde parle le français.*
4. *Dans cette assemblée, tout le monde parle l'anglais ou tout le monde parle le français.*
5. *Un accompagnateur au moins parlera le russe et un accompagnateur au moins parlera le chinois.*
6. *Un accompagnateur au moins parlera le russe ou un accompagnateur au moins parlera le chinois.*

7. *Un accompagnateur au moins parlera le russe et le chinois.*
8. *Un accompagnateur au moins parlera le russe ou le chinois.*

$\dots$	$\Rightarrow$ / $\Leftarrow$	$\dots$	$\forall x \,.\, P(x) \dots Q(x)$	$\approx$	$\forall x \,.\, P(x) \dots \forall y \,.\, Q(y)$
$\dots$	$\Rightarrow$ / $\not\Leftarrow$	$\dots$	$\dots$	$\not\approx$	$\dots$
$\dots$	$\Rightarrow$ / $\Leftarrow$	$\dots$	$\dots$	$\approx$	$\dots$
$\dots$	$\Rightarrow$ / $\not\Leftarrow$	$\dots$	$\dots$	$\not\approx$	$\dots$

III.5 Formules du calcul des prédicats

DÉFINITION 13.7 (FORMULE). *La définition d'une* formule *est :*
- *un atome est une formule,*
- *si P est une formule, si Q est un symbole de quantificateur et si x est un symbole de variable, alors $Qx \,.\, P$ est une formule,*
- *si P est une formule, alors $\neg(P)$ est une formule,*
- *si P et Q sont des formules, alors $(P \vee Q)$, $(P \wedge Q)$, $(P \Rightarrow Q)$, $(P \iff Q)$ sont des formules,*
- *il n'existe pas d'autres manières de construire une formule qu'en appliquant les règles précédentes un nombre fini de fois.* ♢

Exercice 13.9. *On considère un ensemble d'objets de différentes couleurs. Chacun de ces objet porte un numéro et possède une forme géométrique particulière (cube, sphère prisme). En utilisant les prédicats pair(x), cube(x), vert$(x)\dots$, formalisez les phrase suivantes en logique des prédicats*

1. *Si un objet est sphérique, alors cet objet n'est pas vert ou porte un numéro impair.*
2. *S'il existe un objet vert sphérique, alors il existe un objet vert portant un numéro impair.*
3. *Si tout objet vert porte un numéro impair, alors aucun objet sphérique n'est vert.*

IV Sémantique

IV.1 Valeurs de vérité

Le calcul des prédicats utilise, comme le calcul propositionnel, les connecteurs logiques, et produit des propositions. Il est possible d'étendre la notion de « valeur de vérité » au calcul des prédicats. Mais l'étude de la valeur de vérité d'une formule du calcul des prédicats est beaucoup plus compliquée.

1. Une expression typique (une forme propositionnelle) du calcul propositionnel est $P \Rightarrow Q$. Les « atomes » sont ici des variables propositionnelles qui peuvent être remplacées par n'importe quelle proposition. Quelle que soit cette proposition, sa valeur de vérité ne peut être que « *vrai* » ou « *faux* ». Cela permet de lui associer une simple variable booléenne, sa valeur de vérité, pour obtenir simplement la fonction de vérité $\overline{p} + q$.
2. Une expression analogue (une formule) du calcul des prédicats pourrait être $\forall x \,.\, p(x, y) \Rightarrow q(x, z)$. Les atomes sont ici des prédicats binaires qui, eux aussi, ne peuvent prendre que les valeurs « *vrai* » ou « *faux* », mais pas indépendamment des individus considérés.

Il n'est donc pas possible de remplacer un atome par une simple variable booléenne. Seule une fonction booléenne (de variables non booléennes) peut convenir, pour faire intervenir les valeurs des variables qui sont les arguments du prédicat.

– si $f_p(x, y)$ est la fonction booléenne associée au prédicat $p(x, y)$
– si $f_q(x, z)$ est celle qui est associée à $q(x, z)$,

alors la fonction de vérité de la formule est : $\overline{f_p(x, y)} + f_q(x, z)$.

Attention, x, y et z ne sont pas ici des variables booléennes, mais elles prennent leurs valeurs dans l'univers du discours. Il n'est donc pas question de « calculer avec x, y et z comme en algèbre de Boole ».

Le seul moyen, en général, pour étudier une telle fonction de vérité est de construire le tableau de ses valeurs, en donnant à x, y et z successivement toutes les valeurs possibles dans l'univers du discours (si celui-ci est infini, on imagine aisément les problèmes qui vont se poser...).

Les définitions de tautologie et de conséquence logique s'adaptent comme suit.

DÉFINITION 13.8 (TAUTOLOGIE, CONSÉQUENCE LOGIQUE, ÉQUIVALENCE). *Si P et Q sont des formules du calcul des prédicats*

– $\models P$ *[P est une tautologie] si et seulement si, pour tous les univers du discours possibles, pour tous les prédicats qui interviennent dans P, et pour toutes les valeurs des variables dans chacun des univers, la valeur de vérité de P est « vrai ».*
– $\{P\} \models Q$ *[Q est conséquence logique de P] si et seulement si, dans les mêmes conditions que ci-dessus, chaque fois que P est vraie, Q l'est aussi.*
– $P \approx Q$ *[les formules P et Q sont équivalentes] si et seulement si $\{P\} \models Q$ et $\{Q\} \models P$.*

Il reste à donner la définition de la fonction de vérité pour les nouveaux symboles introduits (les quantificateurs)...

– la valeur de vérité de $\forall x \,.\, p(x)$ est obtenue en faisant la liste des valeurs de vérité de $p(x)$ pour toutes les valeurs possibles de x dans l'univers du discours : si, pour toute valeur de x, la valeur de vérité de $p(x)$ est *vrai*, alors la valeur de vérité de $\forall x \,.\, p(x)$ est *vrai*. S'il y a une seule valeur de x pour laquelle la valeur de vérité de $p(x)$ est *faux*, alors la valeur de vérité de $\forall x \,.\, p(x)$ est *faux*.
– la valeur de vérité de $\exists x \,.\, p(x)$ est obtenue en faisant la liste des valeurs de vérité de $p(x)$ jusqu'à ce qu'on trouve *vrai*. Si on trouve *vrai*, la valeur de vérité de $\exists x \,.\, p(x)$ est *vrai*. Si, pour tout élément x de l'univers du discours, la valeur de vérité de $p(x)$ est *faux*, alors celle de $\exists x \,.\, p(x)$ est *faux*.

Bien entendu, dans certains cas, il n'est pas nécessaire d'établir effectivement la table de vérité d'une formule du calcul des prédicats pour prouver qu'il s'agit d'une tautologie. Par exemple, il est bien clair que, pour un atome $p(x, y, z)$, on a : $\models (\forall x, y, z \,.\, p(x, y, z)) \Rightarrow (\forall x, y, z \,.\, p(x, y, z))$.

Donnons enfin deux exemples pour lesquels la construction d'une table de vérité n'est pas nécessaire :

Exercice 13.10. *Montrer que* $\models (\forall x \,.\, p(x, x)) \Rightarrow (\forall x \,.\, (\exists y \,.\, p(x, y)))$.

IV.2 Simplification de formules quantifiées

De la définition de la valeur de vérité d'une formule quantifiée, on peut déduire :

$$\begin{aligned} \neg(\exists x \,.\, p(x)) &\approx (\forall x \,.\, \neg p(x)) \\ \neg(\forall x \,.\, p(x)) &\approx (\exists x \,.\, \neg p(x)) \end{aligned}$$

Exercice 13.11. *Écrire la négation des formules suivantes*

1. $\forall x \,.\, p(x) \Rightarrow q(x)$
2. $\exists x \,.\, p(x) \wedge q(x)$
3. $\forall x \,.\, p(x) \Leftrightarrow q(x)$

4. $\exists x \,.\, (\forall y \,.\, q(x,y) \Rightarrow (p(x,y) \vee r(x,y)))$

Exercice 13.12. *Écrire la négation des formules suivantes*

1. $\forall x \,.\, (\exists y \,.\, p(x,y) \wedge q(x,y))$
2. $\forall x \,.\, (\exists y \,.\, p(x,y)) \Rightarrow q(x)$
3. $\forall x \,.\, (\exists y \,.\, p(x,y)) \Rightarrow (\forall z \,.\, q(z))$
4. $\forall x \,.\, r(x) \Rightarrow (\exists y \,.\, p(x,y))$

Voici d'autres résultats d'équivalences entre formules permettant de réduire la portée de quantificateurs.

PROPRIÉTÉ 13.1 (RÉDUCTION DE PORTÉE DE QUANTIFICATEURS) : Soit p et q des prédicats unaires. Alors on a les deux équivalences suivantes :

$$(\forall x \,.\, p(x) \wedge q(x)) \quad \approx \quad (\forall x \,.\, p(x)) \wedge (\forall y \,.\, q(y)) \tag{13.1}$$
$$(\exists x \,.\, p(x) \vee q(x)) \quad \approx \quad (\exists x \,.\, p(x)) \vee (\exists y \,.\, q(y)) \tag{13.2}$$

PREUVE

Preuve de (13.1). Si, pour toute valeur de x, $p(x)$ et $q(x)$ sont simultanément vrais, alors, pour toute valeur de x, $p(x)$ est vrai, et, pour toute valeur de y, $q(y)$ est vrai. Réciproquement, si, pour toute valeur de x, $p(x)$ est vrai, et, si, pour toute valeur de y, $q(y)$ est aussi vrai, il est bien évident que, pour toute valeur de x, $p(x)$ et $q(x)$ sont simultanément vrais.

Preuve de (13.2). S'il existe une valeur de x pour laquelle l'une au moins des deux propriétés $p(x)$ ou $q(x)$ est vraie, il est bien clair qu'il existe une valeur de x pour laquelle $p(x)$ est vraie ou qu'il en existe une pour laquelle $q(y)$ est vraie ; la réciproque est aussi évidente. ∎

Exercice 13.13. *Trouver*

1. *un exemple où* $\forall x \,.\, p(x) \vee q(x)$ *est vraie sans que* $(\forall x \,.\, p(x)) \vee (\forall x \,.\, q(x))$ *ne le soit ;*
2. *un exemple où* $(\exists x \,.\, p(x)) \wedge (\exists x \,.\, q(x))$ *est vraie sans que* $\exists x \,.\, p(x) \wedge q(x)$ *ne le soit.*

Exercice 13.14. *Pour chacune des formules suivantes,*

- *dire si c'est une négation, une conjonction, une disjonction, une implication, une quantification (universelle ou existentielle) ;*
- *donner la portée des quatificateurs ;*
- *donner les occurences libres des variables.*

1. $\exists x \,.\, A(x,y) \wedge B(x)$
2. $\exists x \,.\, (\exists y \,.\, A(x,y) \Rightarrow B(x))$
3. $\neg(\exists x \,.\, \exists y \,.\, A(x,y) \Rightarrow B(x))$
4. $\forall x \,.\, \neg(\exists y \,.\, A(x,y))$
5. $\exists x \,.\, A(x,y) \wedge B(x)$
6. $\exists x \,.\, A(x,x) \wedge \exists y \,.\, B(y)$

IV.3 Substitutions

Si t est un terme et φ est une formule pouvant contenir la variable x, alors $\varphi(t/x)$ est le résultat du remplacement de toutes les occurrences libres de x par t dans φ.

Le résultat du remplacement $\varphi(t/x)$ est une formule qui est une conséquence logique de la formule originale φ si aucune des variables libre de t ne devient liée suite à ce remplacement. Pour éviter que de telles variables libres deviennent liées il suffit de changer le nom des variables liées de φ en des noms frais (qui n'apparaissent pas dans les variables libres de t). L'oubli de cette condition est une cause fréquente d'erreurs de raisonnement.

A titre d'exemple, on considère la formule φ définie par $\forall y \,.\, y \leq x$ sur l'univers $\mathcal{U}$.

- Si t est un terme sans la variable libre y, alors $\varphi(t/x)$ signifie juste que t est l'élément maximal.
- A l'opposé, si t est y, la formule $\varphi(y/x)$ est $\forall y \,.\, y \leq y$ qui ne dit plus que y est maximal.

Exercice 13.15. *Soit le langage $=\{a, f_2, R_2, S_1\}$, les termes t_i et les formules φ_j suivantes.*

- $t_1 = f(x, y)$
- $t_2 = f(a, y)$
- $t_3 = a$
- $t_4 = f(x, f(x, x))$
- $\varphi_1 : R(x, y) \Rightarrow \forall y \,.\, S(y)$
- $\varphi_2 : (\forall y \,.\, R(y, a)) \Rightarrow (\exists y \,.\, R(x, y))$
- $\varphi_3 : (\forall x \,.\, \exists z \,.\, R(x, z)) \wedge (\exists x \,.\, \forall y \,.\, R(y, x))$

Déterminer si t_i est substituable à x dans φ_j et calculer, le cas échéant, $\varphi_j(t_i/x)$ pour $1 \leq i \leq 4$ et $1 \leq j \leq 3$. Dans le cas où t_i n'est pas substituable, renommer les variables qui le composent ou les variables de la formule afin qu'il le soit.

Exercice 13.16. *Quelle est la valeur de vérité des formules*

- $\varphi_1 = \exists x, y, z \,.\, x \neq y \wedge y \neq z \wedge z \neq x$
- $\varphi_1 = \forall x, y, z, t \,.\, x \neq y \vee x \neq z \vee x \neq t \vee y \neq z \vee y \neq t \vee z \neq t$

pour les univers suivants

- $\mathcal{U}_1 = \{0\}$ *;*
- $\mathcal{U}_2 = \{0, 1\}$ *;*
- $\mathcal{U}_2 = \{0, 1, 3\}$.

Exercice 13.17. *Dans le langage $\mathcal{L} = \{R_2, =_2\}$, on considère les formules suivantes :*

$$\begin{aligned}
\Gamma_1 &= \forall x \,.\, \exists y \,.\, R(x, y) \\
\Gamma_2 &= \forall x \,.\, (\forall y \,.\, (\exists z \,.\, R(x, z) \wedge R(z, y)) \Rightarrow R(x, y)) \\
\Gamma_3 &= \forall x, y \,.\, (R(x, y) \wedge R(y, x)) \Leftrightarrow y = x \\
\Gamma_4 &= \forall x, y \,.\, (\exists z \,.\, \neg R(z, x) \wedge \neg R(z, y)) \vee x = y
\end{aligned}$$

Déterminer la valeur de vérité de ces formules dans les interprétations suivantes :

- *$\mathcal{U}_1 = \mathbb{N}$ et $R(x, y)$ est vraie si et seulement si $x \leq y$. Le prédicat $=$ a l'interprétation standard.*
- *$\mathcal{U}_2 = \mathbb{N} \setminus \{0\}$ et $R(x, y)$ est vraie si et seulement si $x \neq y$ et x divise y. Le prédicat $=$ a l'interprétation standard.*

Fin du Chapitre

Chapitre 14

Méthode de résolution

Que ce soit dans le cas dans le cas propositionnel (comme montré à la Sec. I) ou dans le cas du calcul des prédicats (cf Sec. III) la méthode de *résolution de Robinson* est un algorithme simple de démonstration automatique.

Sa principale différence avec les systèmes déductifs des chapitres précédents se situe dans le sens de la preuve : tandis que ces deux derniers exploitent le modus ponens et effectuent ainsi une preuve en arrière la résolution va engendrer de nouveaux lemmes jusqu'à saturation et fonctionne donc en avant.

On commencera par voir son application tout d'abord à la logique propositionnelle pour s'intéresser ensuite à son intégration dans le calcul des prédicats.

I Cas propositionnel

I.1 Clauses propositionnelles

DÉFINITION 14.1 (FORME CLAUSALE PROPOSITIONNELLE). *Une* clause propositionnelle *est une disjonction de variables propositionnelles éventuellement niées. Une formule propositionnelle est en* forme clausale *(CNF) si elle est exprimée comme une conjonction de clauses. On la représente classiquement comme un ensemble de clauses.* ♢

Une clause propositionnelle est donc de la forme $A \vee B \vee \neg C \vee \ldots$. Pour parvenir à cette forme on utilise successivement les règles de réécriture :

1. réduire les connecteurs : on ne conserve que $\wedge$, $\vee$ et $\neg$:
$$\begin{aligned} A \Rightarrow B &\rightsquigarrow \neg A \vee B \\ A \Longleftrightarrow B &\rightsquigarrow (\neg A \wedge \neg B) \vee (A \wedge B) \end{aligned}$$
2. distribuer récursivement chaque OU sur un ET :
$$(A \wedge B) \vee C \rightsquigarrow (A \vee C) \wedge (B \vee C)$$
3. réécrire les conjonctions de clauses comme un ensemble de clauses :
$$(B \vee D) \wedge (A \vee C \vee \neg D) \rightsquigarrow \begin{array}{rcl} \{C_1 & = & B \vee D, \\ C_2 & = & A \vee C \vee \neg D\}. \end{array}$$

Enfin, la clause vide (sans littéraux) est représentée par $\square$. Elle est insatisfaisable dans toute interprétation.

Exercice 14.1. *Mettre chacune des formules propositionnelles suivantes sous la forme CNF.*

1. $P \Rightarrow Q$
2. $(P \vee Q) \Rightarrow R$
3. $(P \wedge Q) \vee (R \wedge S)$
4. $(P \Rightarrow Q) \wedge (Q \Rightarrow R) \wedge (\neg R)$

I.2 Résolvantes d'une paire de clauses

Deux clauses C_1 et C_2 forment une *paire résoluble* s'il existe un et un seul littéral t tel que t appartient à C_1 et $\neg t$ appartient à C_2. Dans ce cas, on appelle *résolvante* de C_1 et C_2 la clause notée *res*(C_1, C_2) obtenue en prenant la réunion des littéraux de C_1 et C_2 moins cette paire opposée.

Exercice (corrigé) 14.2. *Soit quatre clausess C_1, C_2, C_3,C_4 définies à l'aide de variables propositionnelles P, Q, R et S :*

$$\begin{aligned} C_1 &= P \vee \neg Q \vee \neg R \\ C_2 &= \neg P \vee S \\ C_3 &= \neg Q \\ C_4 &= P \vee \neg S \end{aligned}$$

Quelles paires de clauses sont résolubles et dans ce cas, quelle est la résolvante ?

Réponse : La seule paire résoluble est (C_1, C_2) et sa résolvante est res$(C_1, C_2) = \neg Q \vee \neg R \vee S$.

Exercice 14.3. *Pour chacun des ensembles de clauses trouvées à l'exercice 14.1, dire quelles paires sont résolubles et donner la résolvante le cas échéant.*

I.3 Résolution d'un ensemble de clauses

A partir d'un ensemble de clauses, on appelle *résolution* l'ensemble obtenu en appliquant la seule règle qui, à partir de toute paire résoluble (C_1, C_2) engendre la résolvante *res*(C_1, C_2). On peut alors appliquer cette méthode à la démonstration d'une formule d'après le théorème suivant :

PROPRIÉTÉ 14.1 (DÉDUCTION SYNTAXIQUE PAR RÉSOLUTION) : Soit Γ un ensemble de clauses et C une clause, alors
- Γ est contradictoire si et seulement si la clause vide $\square$ appartient à la résolution de Γ.
- $\Gamma \vdash C$ si et seulement si $\Gamma \cup \mathit{CNF}(\neg C)$ est contradictoire.

Exercice (corrigé) 14.4. *Montrer que*

$$\{ \begin{array}{ll} C_1 = & a \vee b \vee c, \\ C_2 = & \neg a \vee b \vee c, \\ C_3 = & \neg b \vee c \end{array} \} \vdash c.$$

Réponse : on nomme C_4 la clause $\neg c$; on cherche à démontrer que la résolution de $\{C_1, C_2, C_3, C_4\}$ contient la clause vide.
- *C_1 et C_2 sont résolubles, soit $C_5 = \mathit{Res}(C_1, C_2) = b \vee c$.*
- *C_3 et C_5 sont résolubles, soit $C_6 = \mathit{Res}(C_3, C_5) = c$.*
- *C_4 et C_6 sont résolubles, soit $C_7 = \mathit{Res}(C_3, C_5) = \square$.*

On a donc bien le résultat souhaité.

Exercice 14.5.

1. *Montrer que l'ensemble des clauses engendrées par la formule suivante est contradictoire*

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

2. *Démontrer la déduction suivante :*

$$\{A \vee B \vee \neg D, \neg A \vee C \vee \neg D, \neg B, D\} \vdash C$$

Exercice 14.6 (Les Pures et les Pires [Smu98]). *L'ile de Puro Pira est peuplée de* Purs *qui disent toujours la vérité et de de* Pires *qui ne disent que des mensonges.*

1. *Débarqué sur l'île, l'antropologue Abercrombie rencontre trois indigènes Arthur, Bernard et Charles. Abercrombie demande d'abord à Arthur : « Est-ce que Bernard et Charles sont tous les deux des Purs ? », et Arthur lui répondit :« Oui ». Alors Abercrombie lui demande encore : « Est-ce que Bernard est un Pur ? » ; la réponse d'Arthur fur « Non ». A l'aide de la méthode de résolution de Robinson, déduire dans quel groupe appartient chacun des trois indigènes.*
2. *Abercrombie rencontre ensuite deux autres individus Armand et Bertrand. Armand déclare d'une part « l'un de nous deux au moins est un pire » et d'autre part « l'un de nous deux au plus est un pire ». A l'aide de la méthode de résolution de Robinson, déduire dans quel groupe appartient Armand et Bertrand.*

II Formes normales en logique des prédicats

Pour pouvoir appliquer la méthode de résolution dans la logique des prédicats il est nécessaire d'exprimer ces formules sous la forme clausale qui est une version étendue de la forme clausale propositionnelle.

Cette section présente la mise en forme prénexe, Sec. II.1, la mise en forme de Skolem, Sec. II.2 et la mise en forme clausale, Sec. II.3.

II.1 Forme prénexe

DÉFINITION 14.2 (FORME PRÉNEXE). *Une formule G est en forme prénexe si tous ses quantificateurs ($\forall$ et $\exists$) apparaissent en tête, à gauche, dans cette formule. Elle est ainsi de la forme $Q_1 x_1 \,.\, Q_2 x_2 \,.\, \ldots . Q_n x_n \,.\, B$, où B est une formule sans quantificateurs.* ♢

La méthode à suivre est la suivante :

– Réduire les connecteurs : on ne conserve que $\wedge$, $\vee$, $\neg$:

$$\begin{aligned} A \Rightarrow B &\rightsquigarrow \neg A \vee B \\ A \Longleftrightarrow B &\rightsquigarrow (\neg A \wedge \neg B) \vee (A \wedge B) \end{aligned}$$

– Renommer les variables liées, de manière à ce que toute variable liée ne le soit qu'une seule fois, et qu'aucune variable liée ne présente d'occurrence libre, d'après les égalités suivantes :

$$\begin{aligned} \forall x \,.\, A(x) &\rightsquigarrow \forall y \,.\, A(y) \\ \exists x \,.\, A(x) &\rightsquigarrow \exists y \,.\, A(y) \end{aligned}$$

– Faire « remonter » les quantificateurs en tête, par les réécritures suivantes :

$$\begin{aligned} \neg\neg A &\rightsquigarrow A \\ \neg(\forall x \,.\, A(x)) &\rightsquigarrow \exists x \,.\, \neg A(x) \\ \neg(\exists x \,.\, A(x)) &\rightsquigarrow \forall x \,.\, \neg A(x) \end{aligned}$$

et, si x n'est pas variable libre de C

$$\begin{aligned} C \vee \forall x \,.\, A(x) &\rightsquigarrow \forall x \,.\, (C \vee A(x)) \\ C \vee \exists x \,.\, A(x) &\rightsquigarrow \exists x \,.\, (C \vee A(x)) \\ C \wedge \forall x \,.\, A(x) &\rightsquigarrow \forall x \,.\, (C \wedge A(x)) \\ C \wedge \exists x \,.\, A(x) &\rightsquigarrow \exists x \,.\, (C \wedge A(x)) \end{aligned}$$

Exercice 14.7. *Mettre sous la forme prénexe les formules suivantes*

1. $(\exists x\,.\,p(x)) \Rightarrow (\forall x\,.\,p(x))$
2. $(\neg\exists x\,.\,p(x) \vee \forall x\,q(x)) \wedge (r \Rightarrow \forall x\,.\,s(x))$
3. $\neg((\neg existsx\,.\,p(x) \vee \forall x\,.\,q(x)) \wedge (r \Rightarrow \forall x\,.\,s(x))$
4. $\check{\text{n}}(((\forall x\,.\,p(x)) \wedge (\forall y\,.\,p(y) \Rightarrow q(y))) \Rightarrow (\forall z\,.\,q(z)))$
5.

Exercice 14.8. *On considère les propositions suivantes :*

P_1 *Tout crime a un auteur*
P_2 *Seuls les gens malhonnêtes commettent des crimes*
P_3 *On n'arrête que les gens malhonnêtes*
P_4 *Les gens malhonnêtes arrêtés ne commettent pas de crimes*
P_5 *Des crimes se produisent*

On voudrait en déduire :

Q *Il y a des gens malhonnêtes en liberté*

Sans anticiper sur les méthodes de résolution, on va ajouter aux propositions P_1 à P_5 la proposition $P_6 \approx \neg Q$

Pour cela, on considère les prédicats

– *ar(x) : la personne x est arrêtée ;*
– *mal(x) : la personne x est malhonnête ;*
– *cou(x, y) : la personne x commet l'action y ;*
– *cr(y) : l'action y est un crime.*

Traduisez chacune des propositions ci dessus en logique des prédicats et ecrivez chaque formule sous la forme prenexe.

II.2 Forme de Skolem

DÉFINITION 14.3 (FORME DE SKOLEM). *Une formule est sous forme normale de Skolem si sa forme prénexe contient uniquement des quantificateurs universels.* ♢

La démarche qui consiste a supprimer les quantifications existentielles de la forme prénexe est appelée *skolémisation* et est définie comme suit :

– Soit $\exists x_j$ un quantificateur existentiel qui figure après les n quantificateurs universels $\forall x_{j_1} \forall x_{j_2} \ldots \forall x_{j_n}$. La quantification $\exists x_j$ est supprimée et la variable x_j est remplacé par le terme $f(x_{j_1}, x_{j_2} \ldots, x_{j_n})$ où f est un symbole fonctionnel n-aire qui n'apparaît pas ailleurs.
– Soit $\exists x_j$ un quantificateur existentiel qui n'est précédé par aucun quantificateur universel : on peut le supprimer et remplacer x_j par un symbole de constante frais c_j

Cette transformation est autorisée par le théorème suivant, que l'on admettra.

PROPRIÉTÉ 14.2 (THÉORÈME DE SKOLEM) : Soit $\mathcal{A}$ un ensemble fini de formules et $\mathcal{A}_S$ l'ensemble des formes de Skolem de ces formules. Alors, $\mathcal{A}$ peut être interprétée à « *vrai* » si et seulement si $\mathcal{A}_S$ peut aussi être interprétée à « *vrai* ».

Exercice 14.9. *Reprendre chaque formes prénexe de l'exercice 14.7 et l'ecrire sous la forme de Skolem.*

Exercice 14.10 (Suite exemple 14.8). *Donnez la forme de Skolem de chacune des formules de l'exercice 14.8.*

II.3 Forme clausale

Dans la forme de Skolem d'une formule ne subsistent donc que des quantificateurs universels. Ceux-ci sont purement et simplement supprimés. Cette suppression est autorisée par la méthode de déduction utilisée En effet, pour prouver que la formule T est conséquence de l'ensemble de formules $\mathcal{A}$, on cherche à prouver que $\mathcal{A} \cup \{\neg T\}$ n'admet pas de modèle. Pour cela, il suffit d'exhiber une contradiction dans un « cas particulier » , et il est bien clair que, si, dans l'ensemble $\mathcal{A}$ de formules, on supprime les quantificateurs universels, et qu'on prouve que ces formules, avec $\neg T$, n'admettent pas de modèle, alors les formules « complètes » , avec quantificateurs, et $\neg T$, n'en auront pas non plus.

On étend enfin la notion de clause propositionnelle en considérant ici des littéraux au lieu de variables propositionnelles.

DÉFINITION 14.4 (FORME CLAUSALE). *Une* clause *est une disjonction de littéraux (atome ou négation d'un atome). Une formule est en* forme clausale *(CNF) si elle est exprimée comme une conjonction de clauses. On la représente classiquement comme un ensemble de clauses.* ♢

Exercice 14.11 (Suite de l'exo 14.10). *Donner la forme clausale des formes de Skolem de l'exo 14.10.*

III Résolution en logique des prédicats

La méthode de résolution étendue au calcul des prédicats est plus complexe que sa version en calcul propositionnel puisqu'elle doit prendre en compte l'existence de variables. Elle sert de base au langage Prolog qui est étudié en TP.

III.1 Résolvante d'une paire de clauses

Comme précédemment, on ne considère que des formules mises sous la forme clausale.

Deux clauses C_1 et C_2 forment une paire résoluble si et seulement si elles contiennent au moins une paire de littéraux $P(t_1, \ldots, t_n)$ et $\neg P(t'_1, \ldots, t'_n)$ telle qu'il existe une substitution σ avec $\sigma(t_1) = \sigma(t'_1)$, $\ldots$, $\sigma(t_n) = \sigma(t'_n)$.

pour tout i, $1 \leq i \leq n$, $t_i\ sigma = t'_i\sigma$.

Dans un tel cas, à partir de la liste de paires de termes $\{(t_1, t'_1), \ldots, (t_n, t'_n)\}$, toute substitution telle que $\sigma(t_i) = \sigma(t'_i)$ est appelé *unificateur*. Il s'agit de trouver l'unificateur *le plus général* : c'est celui tel que toute autre substitution se déduit de cet unificateur par composition à partir d'une autre substitution.

En pratique, on cherchera la substitution la moins particulière possible. Dans ce cas, on appelle résolvante de C_1 et C_2 la clause notée *Res*(C_1, C_2) obtenue en prenant la réunion de C_1 et de C_2, en effectuant σ et en supprimant la paire opposée $P(\sigma(t_1), \ldots, \sigma(t_n))$ et $\neg P(\sigma(t'_1), \ldots, \sigma(t'_n))$.

EXEMPLE 14.12. Soit deux clauses $C_1 = P(x) \vee Q(g(x))$ et $C_2 = \neg P(f(y))$. La paire (C_1, C_2) est résoluble en prenant comme substitution $(f(y)/x, y/y)$. Sa résolvante est *Res*$(C_1, C_2) = Q(g(f(y)))$.

Dans la pratique, on ne mentionne que les substitution qui modifient les variables. On omettra ainsi celles du type (y/y).

Exercice 14.13. *Peut-on unifier les deux formules atomiques suivantes ?*

$$P(f(X, g(Z)), X, f(Y, g(b))) \text{ et } P(f(U, g(f(a, b))), X, U)$$

III.2 Résolution d'un ensemble de clauses

La démarche est exactement la même qu'à la section I.3, modulo le fait qu'on considère ici la règle de résolution avec unificateur.

EXEMPLE 14.14. Démontrons que l'ensemble de clauses

$$\left\{ \begin{array}{l} C_1 = P(a,X) \vee Q(Y,X), \\ C_2 = \neg Q(Z,T) \vee R(a,Z), \\ C_3 = \neg P(a,b), \\ C_4 = \neg R(U,V) \end{array} \right\}$$

est contradictoire. Pour cela on résout les clauses et on calcule leur résolvante :

- C_1 et C_3 sont résolubles avec comme substitution (b/X). Soit $C_5 = Q(W,b)$;
- C_2 et C_4 sont résolubles avec comme substitution $(a/U, Z/V)$. Soit $C_6 = \neg Q(Z',T')$;
- C_5 et C_6 sont résolubles avec comme substitution $(W/Z', b/T')$. Soit $C_7 = \square$, ce qui termine la démonstration.

III.3 Mise en œuvre de la résolution

La mise en œuvre de la résolution est moins immédiate que dans le cas propositionnel : en raison de la présence de variables quantifiées :

- la mise en forme clausale est plus complexe car elle nécessite deux étapes supplémentaires (mise en forme prénexe et skolémisation) ;
- la résolution est aussi plus complexe car elle nécessite d'exhiber l'unificateur le plus général.

Exercice 14.15. *Montrer que les figures suivantes sont valides ou contradictoires à l'aide de la méthode de résolution de Robinson.*

$$\begin{array}{rcl} \varphi_1 & = & (\exists X \,.\, (\forall Y \,.\, S(X) \Leftrightarrow S(Y))) \wedge (\forall Z \,.\, S(Z)) \\ \varphi_2 & = & \forall X \,.\, \exists Y, Z \,.\, (R(X,X) \Rightarrow R(Z,Y)) \wedge (\neg R(X,X) \vee R(Y,Z)) \\ \varphi_3 & = & \forall X, Y \,.\, \exists Z \,.\, (R(X,X) \wedge R(X,Y)) \Rightarrow R(X,Z) \\ \varphi_4 & = & \forall X, Y \,.\, \exists Z \,.\, R(X,Y) \wedge (R(Y,Z) \Rightarrow \neg R(Z,Z)) \end{array}$$

Exercice 14.16. *Démontrons la déduction suivante à l'aide de la méthode de résolution de Robinson :*

$$\left\{ \begin{array}{l} \forall X \,.\, S(X,X) \Rightarrow U(X,X) \\ \forall X, Z \,.\, \neg(T(Z,F(X)) \wedge U(Z,F(X))) \\ \forall Y, Z \,.\, R(Y,Z) \vee S(Y,Z) \end{array} \right\} \vdash \exists X, Y \,.\, T(X,Y) \Rightarrow R(X,Y)$$

Exercice 14.17. *Sachant que :*

- *l'inspecteur fouillait tous ceux qui pénétraient dans le bâtiment sauf ceux accompagnés par des membres du personnel de l'entreprise ;*
- *certains des hommes de la bande de F. pénétraient dans le bâtiment sans être accompagnés de personne étrangère à la bande de F. ;*
- *l'inspecteur n'a fouillé aucun des hommes de la bande de F..*

Établir que certains membres de la bande de F. étaient membres du personnel de l'entreprise.

Exercice 14.18.

Sachant que :

- *un oncle est soit le frère d'une mère, sont le frère d'un père ;*
- *Fred est l'oncle de Jack ;*
- *Bob est le père de Jack ;*
- *Carol est la mère de Jack ;*
- *Bob n'a pas de frère ;*

Déduire par la méthode de résolution le nom du frère de Carol.

Exercice 14.19.

Sachant que :

- *Jack possède un chien ;*
- *Chaque possesseur de chien aime les animaux ;*
- *Un amoureux des animaux ne peux pas tuer un animal*
- *Soit Jack ou Curiosity ont tué le chat (dont le nom est Tuna)*

Montrer que c'est Curiosity qui a tué le chat

Fin du Chapitre

Quatrième partie

Langages, grammaires et automates

Chapitre 15

Compilation, langages et grammaires

I Introduction à la compilation

I.1 Le problème posé est...

Donner à un ordinateur un fichier contenant du texte, le lui faire lire et comprendre de manière à lui faire exécuter un certain nombre de tâches associées à ce fichier

$\Rightarrow$ On fait une compilation.

I.2 Les diverses phases d'une compilation

Détaillons succintement les différentes phases d'une compilation...

I.2.1 L'analyse lexicale

On analyse le flux d'entrée de manière à le découper en unités lexicales, ou *lexèmes*.

EXEMPLE 15.1. Dans *if (temps == beau) {* etc., les unités lexicales sont « if », « (», « temps », « == », « beau », «) ».

I.2.2 L'analyse syntaxique

Les contraintes à respecter pour que le texte soit compréhensible sont-elles respectées ? En d'autres termes, le flux de lexèmes est-il conforme à la syntaxe du langage utilisé (par comparaison à la grammaire du langage, *c.f.* ci-dessous) ?

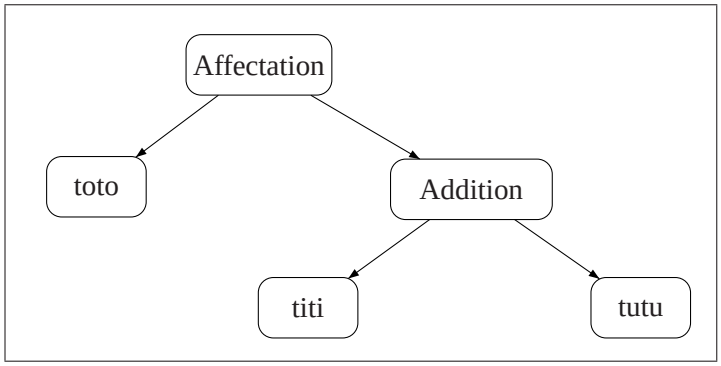
I.2.3 L'analyse sémantique

Reconnaître la signification d'un texte syntaxiquement correct : essayer de comprendre ce que cela signifie (le sens).

Cela implique notemment la transformation de la source en une forme utilisable, qui fasse apparaître le sens du texte.

EXEMPLE 15.2 (D'ANALYSE SÉMANTIQUE). *toto = titi + tutu ;* est une instruction d'affectation à la variable « toto » d'une valeur exprimée par une expression algébrique, constituée de la somme des variables « titi » et « tutu ».

REMARQUE 15.1. Certains compilateurs utilisent des structures arborescentes :



I.2.4 Compilation proprement dite

Utiliser effectivement le résultat de l'analyse sémantique pour obtenir le résultat escompté, ce qui est demandé : production de code machine, traduction d'un texte dans une autre langue, etc.

REMARQUE 15.2. En général, ces différentes phases sont menées en parallèle.

II Les grammaires

II.1 Définition de la notion de grammaire

DÉFINITION 15.1 (GRAMMAIRE). *Une* grammaire *est un ensemble de règles de syntaxe qui décrivent quels sont les constructions correctes qui sont possibles dans le langage utilisé, à l'aide de l'alphabet utilisé (alphabet ou vocabulaire).* ♢

Il existe de nombreux types de grammaires, et encore bien plus de formalismes exprimés pour représenter cette grammaire.

Nous utiliseront pour commencer un seul symbolisme pour remprésenter les grammaires : la formalisation BNF (Backus-Naur Form).

II.2 Le formalisme BNF

Dans la syntaxe BNF, une grammaire est constituée d'un ensemble de règles.

Chaque règle est constituée :
– d'un premier membre,
– suivi du symbole de réécriture (: :=),
– suivi d'un second membre, qui peut être vide.
On utilise (et on distingue) des symboles terminaux et des symboles non-terminaux (ST et SNT).

II.3 Les symboles terminaux

DÉFINITION 15.2 (SYMBOLE TERMINAL). *Un* symbole terminal *est un symbole qui peut effectivement intervenir dans le texte analysé.* ♢

EXEMPLE 15.3. Dans *if (temps == beau), if* est un symbole terminal (on trouve le mot dans le programme).

NOTATION : Les symboles terminaux sont entourés par : « ».

II.4 Les symboles non terminaux

DÉFINITION 15.3 (SYMBOLE NON TERMINAL). *Un* symbole non terminal *(SNT) est un symbole introduit (par commodité, ou plutôt par nécessité) par le rédacteur de la grammaire pour décrire les parties du fichier d'entrée qui représentent un tout logique et permettant de simplifier l'écriture de la grammaire.* ♢

NOTATION : Les symboles non-terminaux sont entourés par des chevrons : $< >$.

Le premier membre d'une règle de grammaire est un SNT (la règle en constitue la définition), le second membre est une famille ordonnée (éventuellement vide) de symboles, terminaux ou non.

Ainsi, chaque règle de la grammaire consiste en la définition d'un symbole non-terminal. Cette dernière est terminée quand tous les SNT ont reçu une définition. Une règle s'écrit finalement sous la forme :

<SNT> : := suite (éventuellement vide) de ST et SNT

EXEMPLE 15.4. Voici un bout de grammaire (pour la définition d'une fonction) :

$$\begin{array}{lll} < fct > & ::= & < type >< nom >'' \ (''< parametres >'')'' < bloc > \\ < type > & ::= & ''int'' \\ & ::= & ''char'' \end{array}$$

DÉFINITION 15.4 (AXIOME DE LA GRAMMAIRE). *Parmi tous les SNT, l'un d'entre eux doit désigner l'ensemble du texte à analyser, on l'appelle* axiome de la grammaire. ♢

EXEMPLE 15.5. $< programme\ en\ C >::=< entete >< suite\ de\ fct >$

La grammaire est terminée quand tous les SNT ont reçu au moins une définition.

II.5 Exercices

Exercice 15.6. *Les mots du langage $\mathcal{L}$ sont constitués d'un nombre, éventuellement nul, de a, suivi d'un b, suivi d'au moins un c.*

Donner une grammaire BNF de ce langage.

Exercice 15.7. *Les mots du langage $\mathcal{L}$ commencent par un caractère a, suivi d'un nombre pair (éventuellement aucun) de caractères b, puis de deux caractères c.*

Donner une grammaire BNF de ce langage.

Exercice 15.8. *Les mots du langage $\mathcal{L}$ sont les noms de variables en C : ils commencent obligatoirement parune lettre (majuscule ou minuscule), et se poursuivent par un nombre quelconque de chiffres, lettres ou underscore.*

Donner une grammaire BNF de ce langage.

Exercice 15.9. *Ecrire la grammaire, en syntaxe BNF, des formes propositionnelles. On rappelle que les opérateurs sont, par ordre de priorité croissante :*

- *Implication et équivalence (au même niveau, le moins prioritaire). Qu'ils ne sont pas associatifs (on ne peut ni les répéter, ni les faire coexister, au même niveau (c'est-à-dire, sans parenthèse), dans une expression (par exemple, $a \longrightarrow b \longleftarrow c$, ou $a \longrightarrow b \longrightarrow c$ sont incorrects).*

– *Disjonction et conjonction (au même niveau, prioritaires sur implication et équivalence). Ils sont associatifs, mais on ne peut pas les mélanger : on peut écrire $a \vee b \vee c$, sans parenthèse, mais pas $a \vee b \wedge c$.*
– *Négation, prioritaire sur tous les autres. Elle peut être répétée. Il s'agit d'un opérateur unaire préfixé.*

Les noms de variable commencent par un caractère alphabétique, suivi éventuellement d'un nombre quelconque de caractères alphanumériques ou de soulignements. Il n'y a pas de constantes dans les expressions. Les opérateurs sont réalisés au clavier par les trois caractères consécutifs $< - >$ pour l'équivalence, les deux caractères consécutifs $- >$ pour l'implication, les lettres consécutives ou pour la disjonction, et pour la conjonction, et non pour la négation. L'expression peut évidemment comporter des parenthèses, et ne peut être vide.

Exercice 15.10. *Le langage considéré est le prototypage des fonctions en C.*
Donner une grammaire BNF de ce langage.

Exercice 15.11. *On accepte les deux types de phrases suivantes :*
– *« Marie est la mère du frère de Sonia. »*
– *« Qui est le père de l'oncle de la mère du petit fils de Paul ? »*
...et tous leurs dérivés. Écrire la grammaire correspondante.

III Un exemple complet

« Les expressions correctes sont constituées d'un nombre quelconque, mais non nul, de 0, suivi d'un nombre quelconque, mais non nul, de 1. »

III.1 Principes généraux

On conseille de suivre cette démarche :

1. Commencer par écrire la grammaire du langage (des expressions correctes).
2. Ecrire l'analyseur syntaxique pur.
3. Passer à l'analyseur syntaxique avec messages d'erreur.
4. Puis à l'analyseur syntaxique avec interprétation sémantique.

Exercice 15.12. *Suivez ce cheminement :*

1. *Ecrivez cette grammaire.*
2. *Programmez, en C, l'analyseur syntaxique pur.*
3. *Ecrire le programme principal associé (le main).*
4. *Le modifier en analyseur syntaxique avec messages d'erreur, et adaptez le programme principal en conséquence.*
5. *Améliorez le programme pour qu'il devienne un analyseur syntaxique avec interprétation sémantique : ici, comptez le nombre de 0 et de 1, en cas de réussite.*

III.2 La grammaire du langage

< expression >	: :=	< groupe0 > < groupe1 >
< groupe0 >	: :=	“0" < suite0 >
< suite0 >	: :=	< groupe0 >
	: :=	
< groupe1 >	: :=	“1" < suite1 >
< suite1 >	: :=	< groupe1 >
	: :=	

Il faut :

– Subdiviser au maximum les expressions en sous-expressions cohérentes, en n'hésitant pas à multiplier les niveaux.
– Retarder au maximum les alternatives (en multipliant les niveaux) pour ne les faire intervenir que lorsqu'on ne peut plus faire autrement.

III.3 Analyseur syntaxique pur

Voici le code de l'analyseur pur : il répond par « bon » ou « mauvais ».

```
#include <stdio.h>

char s[512];
char **ss;

int expression(){
    if (groupe0()==1)
        return groupe1();
    return 0;
}

int groupe0(){
    if (*ss == '0'){
        s++;
        return suite0();
    }
    return 0;
}

int suite0(){
    if (groupe0() == 0)
        return 1;
    return 1;
}
```

... même chose pour groupe1() et suite1().

Une fonction prévue pour analyser une sous-expression ne connaît pas ce qui précède et ne s'occupe pas de ce qui suit.

Passons au programme principal :

```
int main(){
  printf("Une expression a analyser ? \n");
  scanf("%s",s);
  ss = s;
  if (expression()==1)
      if (*ss == '\0')
          printf("Bon \n");
      else
          printf("Mauvais\n");
  else
      printf("Mauvais\n");
}
```

REMARQUE 15.3. Toujours commencer par l'analyseur syntaxique pur.

III.4 Analyseur syntaxique avec messages d'erreur

```
int groupe0(){
    if (*ss == '0'){
```

```
        ss++;
        return suite0();
    }
    printf("L'expression doit commencer par 0\n")
    return 0;
}

int suite0(){
    if (*ss == '0'){
        ss++;
        return suite0();
    }
    return 1;
}
```

Le programme principal devient alors :

```
int main(){
  printf("Une expression a analyser ? \n");
  scanf("%s",s);
  ss = s;
  if (expression()==1)
      if (*ss == '\0')
          printf("Bon \n");
      else if (*ss == '0')
              printf("Pas de 0 apres le(s) un(s).\n");
          else
              printf("Caractere interdit : %c\n",*ss);
}
```

III.5 Analyseur syntaxique avec interprétation sémantique

```
int groupe0(){
    if (*ss == '0'){
        ss++;
        return 1+suite0();
    }
    printf("L'expression doit commencer par 0\n")
    return 0;
}

int suite0(){
    if (*ss == '0'){
        ss++;
        return 1+suite0();
    }
    return 0;
}
```

Pareil pour groupe1 et suite1. Le programme principal devient alors :

```
int main(){
  printf("Une expression a analyser ? \n");
  scanf("%s",s);
  ss = s;
  Expression = expression()

  if (*ss == '\0')
       printf("Bon \n");
  else if (*ss == '0')
           printf("Nombre de 0 : %d, nomre de 1 : %d",expression.zero, expression.un);
```

```
        else
            printf("Caractere interdit : %c\n",*ss);
}
```

où la structure *Expression* et la fonction *expression()* sont ainsi définis :

```
struct Expression{
    int zero;
    int un;
}

struct Expression expression(){
    struct Expression a;
    a.zero = groupe0();
    if (a.zero !=0)
        a.un = groupe1();
    return a;
}
```

Cet exemple, et d'autres, seront (re)vus en TP.

Fin du Chapitre

Chapitre 16

Introduction aux expressions rationnelles

I Présentation

Dans la définition de la syntaxe d'un langage de programmation, par exemple, on rencontre souvent des définitions telles que celle d'un identificateur :

« Un identificateur est un identificateur de symbole ou un identificateur de variable ; un identificateur de symbole commence par deux caractères alphabétiques, suivi par un nombre quelconque, éventuellement nul, de chiffres, suivis, etc. »

Il s'agit ici d'introduire des abréviations pour ce type d'expressions ; ces abréviations conduisent à la notion d'expression rationnelle.

EXEMPLE 16.1. L'expression rationnelle $a(a|b)^*$ signifie : un caractère a, suivi d'un nombre quelconque, éventuellement nul, de caractères choisis dans l'ensemble $\{a, b\}$.

Un langage est évidemment associé à une expression rationnelle. On notera $\mathcal{L}(r)$ le langage associé à l'expression rationnelle r.

Exercice 16.2. *Quel est le langage décrit par l'expression rationnelle $\alpha = (ab^*)^*$?*
L'expression $\beta = a(a|b)^*$ est-elle équivalente à α ?
Trouvez une expression équivalente à α dans laquelle il n'y a qu'une $*$.

Réponse : $\varepsilon \in L_\alpha \setminus L_\beta$; expression équivalente : $(\varepsilon|a(a|b)^*)$.

Définissons dorénavant plus rigoureusement cela, rentrons dans les détails...

II Règles de définition

Voici les règles qui permettent de définir les expressions rationnelles sur un alphabet Σ :

1. ε est une expression rationnelle qui dénote { « » } (l'ensemble constitué de la chaîne vide).
2. Si r et s dénotent les langages $\mathcal{L}(r)$ et $\mathcal{L}(s)$, alors
   - $(r)|(s)$ désigne le langage $\mathcal{L}(r) \cup \mathcal{L}(s)$ (*i.e.* le langage obtenu par réunion des deux langages $\mathcal{L}(r)$ et $\mathcal{L}(s)$).
   - $(r)(s)$ désigne le langage $\mathcal{L}(r)\mathcal{L}(s)$ (*i.e.* le langage obtenu en concaténant, de toutes les manières possibles, un mot du langage $\mathcal{L}(r)$ et un mot du langage $\mathcal{L}(s)$).
   - $(r)^*$ désigne le langage $(\mathcal{L}(r))^*$ (*i.e.* le langage constitué de la chaîne vide, des mots de $\mathcal{L}(r)$ et des mots obtenus en concaténant un nombre quelconque (au moins deux) de mots de $\mathcal{L}(r)$).
   - (r) désigne le langage $\mathcal{L}(r)$.

Exercice 16.3. *Décrire, sur l'alphabet { Acquérir, Sortir, Rentrer, Vendre, Archiver }, la vie d'un document dans une bibliothèque.*

Réponse : Acquérir (Sortir Rentrer)* (Sortir | Vendre | Archiver).

Exercice 16.4. *Écrire, en syntaxe BNF, la grammaire algébrique de toutes les expressions rationnelles sur* Σ.

Réponse :

$<$ expression $>$	: :=	$<$ primitive $>$
	: :=	$<$ parenthèse $>$
	: :=	$<$ concaténation $>$
	: :=	$<$ étoile $>$
$<$ primitive $>$	: :=	ε
	: :=	x (avec $x \in \Sigma$)
$<$ parenthèse $>$	: :=	'(' $<$ expression $>$ | $<$ expression $>$ $<$ suite $>$ ')'
$<$ suite $>$	: :=	$<$ expression $>$ $<$ suite $>$
	: :=	
$<$ concaténation $>$	: :=	$<$ expression $>$ $<$ expression $>$
$<$ étoile $>$	: :=	$<$ primitive $>$ '*'
	: :=	$<$ parenthèse $>$ '*'
	: :=	$<$ concaténation $>$ '*'

On peut réduire le nombre de paires de parenthèses écrites, en adoptant les règles de priorité suivantes :

- La répétition est prioritaire sur tout autre opérateur.
  Autrement dit ab^* est $a|b^*$ doivent être interprétés (respectivement) par $a(b)^*$ et $a|(b^*)$.
- La concaténation est prioritaire sur l'alternative.
  $rs|tu$ doit donc être interprété comme $(rs)|(tu)$.

De plus, on admettra l'écriture a^n pour le mot $aaa \ldots aa$ (n fois).

Exercice 16.5. *Soit l'alphabet* $\Gamma = \{+, \times, a, b, c\}$. *Repérer les expressions rationnelles sur* Γ *parmi les suites de symboles suivantes :*

1. $(a|+)^* + b \times c*$,
2. $+^*|^*\times^*$,
3. $((a+)^*|)b^*c$,
4. $((a^*b)^* \times |ca+^*)$.

Réponse : Seules la première et la dernière suite de symboles sont des expressions rationnelles.

III Propriétés des opérateurs

PROPRIÉTÉ 16.1 : Les propriétés de ces opérateurs sont les suivants :

1. Associativité : $r|(s|t) = (r|s)|t$ et $r(st) = (rs)t$.
2. Commutativité de l'alternative : $r|s = s|r$.
3. Distributivité de l'alternative sur la concaténation : $r(s|t) = rs|rt$ et $(r|s)t = rt|st$.
4. ϵ est élément neutre pour la concaténation.
5. La répétition est idempotente : $r^{**} = r^*$.

Exercice 16.6. *Quel est le langage sur $\{a, b\}$ décrit par l'expression rationnelle : $b^*a(b^*ab^*a)^*b^*$? En trouver une « meilleure » expression.*

Réponse : L'ensemble des mots sur $\{a, b\}$ contenant un nombre impair de a. $b^*a(b|ab^*a)^*$.

IV De nouvelles abréviations

D'autres abréviations peuvent être introduites :
- L'opérateur de fermeture positive : a^+ désigne « au moins une instance de » (*i.e.* $a^+ = aa^*$
- L'opérateur ? qui signifie « zéro ou une occurence de » (*i.e.* $a? = a|\epsilon$).
- Classes de caractères : la notation $[abc]$ est une autre notation pour $a|b|c$, sans intéret, sauf dans le cas de $[a - z] = a|b| \ldots |z$.

EXEMPLE 16.7. On peut représenter « une lettre, suivie d'un nombre quelconque, éventuellement nul, de lettres ou de chiffres » par $[a - zA - Z][a - zA - Z0 - 9]^*$.

V Universalité des expressions rationnelles

Les expressions rationnelles ne sont pas universelles, et ne permettent pas de décrire tous les langages.

EXEMPLE 16.8. Il est impossible de décrire, à l'aide d'expression rationnelles, le langage défini par

$$\{wcw | w \text{ est une chaîne de a et de b}\}$$

En effet, « une chaîne de a ou de b » peut s'exprimer par l'expression rationnelle $(a|b)^*$.

Mais, si l'on écrit ensuite $(a|b)^*c(a|b)^*$, la syntaxe des expressions rationnelles ne permet pas de préciser que les chaînes qui précèdent et suivent le c sont identiques.

Fin du Chapitre

Chapitre 17

Automates Finis

I Automates finis

I.1 Introduction

On va dégager dans ce paragraphe la notion de *machine* comme modèle conceptuel pour la description de dispositifs informatiques aussi variés qu'un ordinateur entier, un logiciel ou un compilateur.

DÉFINITION 17.1 (MACHINE). *Une* machine *est un dispositif doté d'un certain nombre d'états, susceptible d'évoluer d'un état à un autre en fonction de divers paramètres, comme le temps (la machine est alors dotée d'une machine interne).*

Elle est de plus apte à communiquer avec l'extérieur : elle peut accepter des données en provenance de l'extérieur (des entrées) ou communiquer des résultats à l'extérieur (des sorties). ♢

Exercice 17.1. *Donnez des exemples de machines.*

REMARQUE 17.1. À chaque instant, la condition interne de la machine, y compris la mémoire, constitue son état.

I.2 Mécanismes

C'est le type le plus simple de machine :

DÉFINITION 17.2 (MÉCANISME). *Un* mécanisme *est totalement imperméable au monde extérieur, il n'accepte aucune entrée ni aucune sortie.*

C'est une machine à nombre fini d'états, dont le comportement est gouverné uniquement par le temps, mesuré par une horloge interne. ♢

Exercice 17.2. *Donner un exemple de mécanisme.*

Un mécanisme peut être entièrement décrit par un couple (E, t), où E est un ensemble fini d'états et $t : E \to E$ est une fonction de transition des états.

PROPRIÉTÉ 17.1 (EXISTENCE D'UN CYCLE) : Un mécanisme entre nécessairement dans une boucle infinie (on dit : *un cycle*).

En effet, le nombre d'états est fini.

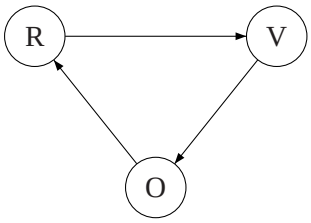
DÉFINITION 17.3 (ÉTAT-REPOS). *S'il existe un état $e \in E$ tel que $t(e) = e$, cet état est appelé* état-repos *.* ♢

REMARQUE 17.2. Un mécanisme qui entre dans un tel état n'en sort évidemment plus.

EXEMPLE 17.3. Un feu de circulation routière peut être décrit par un mécanisme à trois états : V, O et R, donc $E = \{V, O, R\}$.

La fonction de transition des états est telle que $t(V) = O, t(O) = R$ et $t(R) = V$.

On peut représenter ce mécanisme par le graphe de transition des états



ou par la matrice booléenne T représentant t :

$$T = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Exercice 17.4. *L'exemple précédent possède-t-il un cycle ? un état-repos ? Sinon, le modifier pour.*

II Automates finis à comportement déterminé

II.1 Définition

DÉFINITION 17.4 (AUTOMATE FINI À COMPORTEMENT DÉTERMINÉ). *On appelle* automate fini à comportement déterminé *(AFD) tout triplet (E, I, t), où*

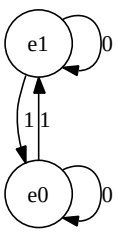
- *E est un ensemble fini (l'ensemble des* états*),*
- *I est le* vocabulare *de l'automate : c'est l'ensemble fini des symboles admis en entrée,*
- *$t : E \times I \to E$ est la* fonction de transition d'états *: si l'automate se trouve dans l'état $e \in E$, il réagit à l'entrée $i \in I$ en passant à l'état $t(e, i)$.*

Pour $i \in I$, on définit la fonction $t_i : E \to E$ par $t_i(e) = t(i, e)$. ♢

EXEMPLE 17.5. Soit $E = \{e_0, e_1\}$, $I = \{0, 1\}$ et t telle que

1. l'entrée 0 laisse inchangé chacun des états,
2. l'entrée 1 échange les états.

Un tel dispositif, en électronique, est appelé un *T-flip-flop* , il est abondamment utilisé dans les ordinateurs...



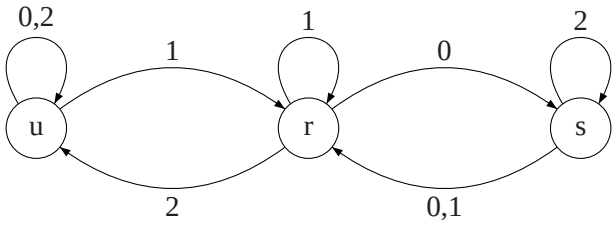
La table qui donne les valeurs de la fonction t est appelée *table de transition d'états* de l'automate considéré :

t	0	1
e_0	e_0	e_1
e_1	e_1	e_0

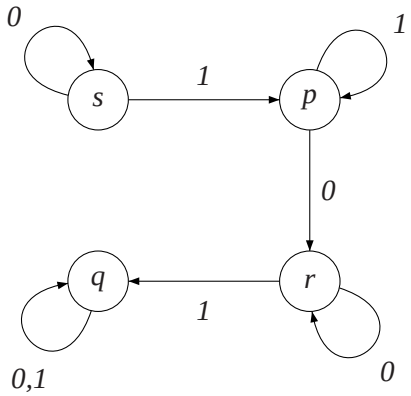
Exercice 17.6. *Représenter le graphe de l'automate fini M dont la table de transition des états est*

t	0	1	2
r	s	r	u
s	r	r	s
u	u	r	u

Réponse :



Exercice 17.7. *Écrire la table de transition d'états de l'automate dont le graphe est représenté dans la figure suivante :*



Réponse :

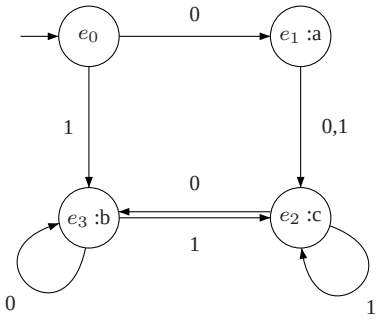
	0	1
p	r	p
q	q	q
r	r	q
s	s	p

II.2 Automates finis avec sorties (machines de Moore et de Mealy)

DÉFINITION 17.5 (MACHINE DE MOORE). *Une* machine de Moore *est un sextuplet* $M = (E, I, t, e_0, V, g)$ *tel que* (E, I, t) *est un AFD, et*

- $e_0 \in E$ *est un état appelé* état initial*, dans lequel se trouve la machine au départ de chaque exécution.*
- V *est un ensemble fini, dit* ensemble des sorties*,*
- $g : t < E, I > \rightarrow V$*, où* $t < E, I >$ *est l'image de* t*, est la* fonction de sortie *telle que, chaque fois que la machine entre dans l'état* e*, elle produise la sortie* $g(e) \in V$*.* ♢

EXEMPLE 17.8. Ici, $E = \{e_0, e_1, e_2, e_3\}$, $I = \{0, 1\}$, $V = \{a, b, c\}$, t est donnée, soit par le graphe de l'automate :



soit par la table de transition d'états

t	0	1
e_0	e_1	e_3
e_1	e_2	e_2
e_2	e_3	e_2
e_3	e_3	e_2

g se lit dans le graphe : $g(e_1) = a, g(e_2) = c, g(e_3) = b$.

REMARQUE 17.3. Une telle machine est aussi appelée *traducteur* (elle « traduit » l'entrée 0001001 en une sortie $acbcbbc$).

DÉFINITION 17.6 (MACHINE DE MEALY). *On obtient une* machine de Mealy *lorsque la sortie est déterminée, non pas par l'état atteint, mais par la transition d'états.*

C'est donc un sextuplet (E, I, t, e_0, V, h) *où la fonction de sortie* h *est une application de* $E \times I$ *vers* V*.* ♢

REMARQUE 17.4. Il est clair que, pour une machine de Moore (E, I, t, e_0, V, g), on peut définir une machine de Mealy équivalente (c'est-à-dire qui produit la même sortie sur toute séquence d'entrée), en posant $h(e, i) = g(t(e, i))$, soit $h = got$.

Réciproquement, en introduisant au besoin des états supplémentaires, on montre qu'on peut remplacer toute machine de Mealy par une machine de Moore équivalente.

Nous ne nous occuperons donc dans la suite que de machines de Moore.

II.3 Automates de Moore

DÉFINITION 17.7 (AUTOMATE DE MOORE). *Une machine de Moore telle que l'ensemble V des sorties est réduit à la paire booléenne $\{FAUX, VRAI\}$ ou $\{0, 1\}$,*

– *tout état qui donne lieu à FAUX est appelé* état de rejet*,*

– *tout état qui donne lieu à la sortie VRAI est appelé* état d'acceptation *.*

est appelée automate de Moore *ou* machine d'acceptation. ♢

REMARQUE 17.5. Inutile d'exhiber ici la fonction de sortie, il suffit de se donner l'ensemble A des états d'acceptation, sous-ensemble de E.

Un automate de Moore est donc défini par le quintuplet (E, I, t, e_0, A).

Sur le graphe représentant un automate de Moore, on représentera un état d'acceptation en l'entourant d'un double cercle.

Les autres états (simplement cerclés) sont les états de rejet.

III Langage associé à un automates de Moore

III.1 Définition du langage

Soit M un automate de Moore.

L'ensemble des entrées I peut être considéré comme l'alphabet d'un système formel.

L'ensemble des « mots » construits avec cet alphabet (suite d'éléments de l'alphabet) qui conduisent la machine à un état d'acceptation peut être considéré comme l'ensemble des formules bien formées de ce système formel.

DÉFINITION 17.8 (LANGAGE). *Ce système formel constitue le* langage associé à l'automate M. ♢

NOTATION : $\mathcal{L}(M)$

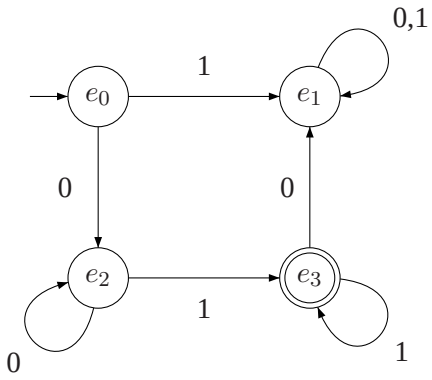
Réciproquement, étant donné un langage $\mathcal{L}$, on peut éventuellement construire un automate de Moore M tel que le langage associé à M soit $\mathcal{L} : \mathcal{L} = \mathcal{L}(M)$.

REMARQUE 17.6. Cela n'est pas possible pour tous les langages. Quand c'est possible, cet automate analyse les mots du langage.

III.2 Exemple et exercices

EXEMPLE 17.9. Construction de l'automate qui reconnaît le langage défini par l'expression suivante...

Un mot du langage est constitué d'un nombre quelconque, mais non nul, de 0, suivi d'un nombre quelconque, mais non nul, de 1.



Exercice 17.10. *Décrire le langage $\mathcal{L}(M)$ de l'automate de Moore M dont la table de transition des états est :*

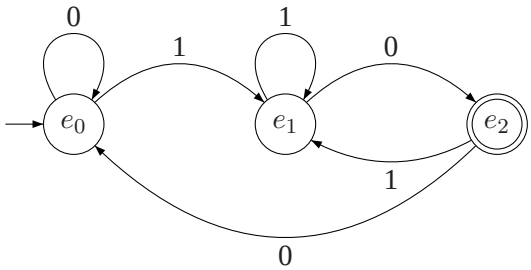
t	0	1
e_0	e_1	e_2
e_1	e_1	e_2
e_2	e_2	e_1

L'état initial est e_0 et le seul état d'acceptation est e_2.

Réponse : Les mots corrects contiennent un nombre impair de 1.

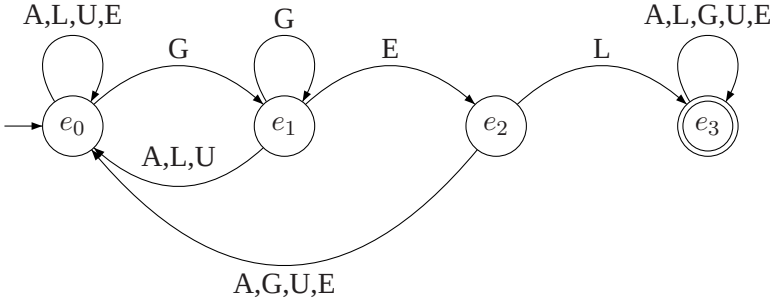
Exercice 17.11. *Sur l'alphabet $I = \{0, 1\}$, construire l'automate de Moore dont le langage est l'ensemble de tous les mots sur I se terminant par 10.*

Réponse :



Exercice 17.12. *Construire un automate de Moore dont l'alphabet est constitué des lettres du mot « ALGUE » qui reconnaît les mots contenant la sous-chaîne « GEL » (et seulement celle-ci).*

Réponse :



IV Automates finis à comportement non déterminé

IV.1 Définitions et exemples

Les automates considérés jusqu'à présent ont un comportement complètement « déterminé » : pour chaque configuration état-entrée $(e, i) \in E \times I$, une et une seule transition d'état est fixée.

Cela résulte du fait qu'ils sont régis par une *fonction* de transition d'états t (de $E \times I$ dans E).

On peut imaginer des automates moins « rigides », pour lesquels, dans certaines configurations, plusieurs transitions d'états sont possibles ou, au contraire, aucune n'est prévue.

Pour un tel automate, qualifié de non-déterministe, t n'est plus une fonction, mais une relation binaire quelconque. Ainsi...

DÉFINITION 17.9 (AUTOMATE FINI NON DÉTERMINISTE). *Un* automate fini non déterministe *à états d'acceptation est défini par* (E, I, t, S, A) *où :*

- E *est un ensemble (fini) d'états,*
- I *est l'ensemble des entrées,*
- t *est la relation de transition des états,*
- S*, partie de* E*, est l'ensemble des états initiaux,*
- A*, partie de* E*, est l'ensemble des états d'acceptation.* ♢

REMARQUE 17.7. Il se peut donc qu'une entrée puisse conduire un automate vers plusieurs états possibles ou qu'elle laisse l'automate indifférent.

EXEMPLE 17.13. Dans cet exemple, lorsque l'automate se trouve dans l'état e_0, l'entrée a peut le faire passer dans l'état e_1 ou dans l'état e_3 et, lorsqu'il se trouve dans l'état e_1, rien n'est prévu pour l'entrée b.

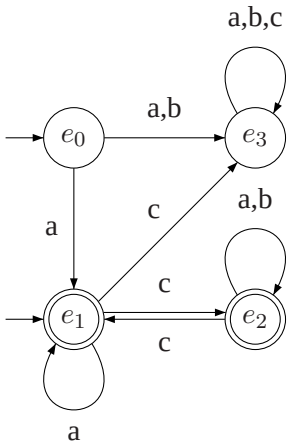


Table de transitions

t	a	b	c
$\{e_0\}$	$\{e_1, e_3\}$	$\{e_3\}$	$\varnothing$
$\{e_1\}$	$\{e_1\}$	$\varnothing$	$\{e_2, e_3\}$
$\{e_2\}$	$\{e_2\}$	$\{e_2\}$	$\{e_1\}$
$\{e_3\}$	$\{e_3\}$	$\{e_3\}$	$\{e_3\}$

Il est évidemment possible de concevoir des AFND produisant des sorties, et, en particulier, des états d'acceptation et de rejet. On admettra par ailleurs qu'il puisse y avoir, dans ces cas, plusieurs états initiaux possibles.

Soit alors $M = (E, I, t, S, A)$ un AFND.

DÉFINITION 17.10 (ENTRÉE RECONNUE). *On dit qu'une suite w d'entrées est* reconnue *par l'automate si cette suite* peut *conduire l'automate à un état d'acceptation.*

EXEMPLE 17.14. Dans l'exemple précédent,
- Comme e_1 est à la fois un état initial et d'acceptation, le mot vide fait partie du langage reconnu par l'automate.
- Le mot $aaacc$ est reconnu par l'automate.
- Les mots refusés sont ceux qui n'ont aucun chemin vers un état d'acceptation, comme bbb.

On définit aussi de cette manière le langage $\mathcal{L}(M)$ associé à un AFND. Il est constitué de l'ensemble des mots qui, depuis l'un des états initiaux, peut conduire à l'un des états d'acceptation.

Exercice 17.15. *On considère l'automate fini M non déterministe dont la relation de transition des états est donnée par la table*

t	0	1
e_0	e_0, e_1	e_1
e_1	$-$	e_2
e_2	e_2	e_1
e_3	$-$	e_0, e_1, e_2

Si e_0 et e_1 sont les états initiaux et si e_2 est le seul état d'acceptation, les mots 001111 et 01001 sont-ils reconnus par M ?

Réponse : 001111 est reconnu, mais pas 01001.

IV.2 Utilité

Les AFND sont beaucoup plus simple à construire que les AFD.

Ainsi, les algorithmes de construction automatique d'automates produisent des AFND, et les algorithmes de simplification d'automate utilisent des AFND.

Mais, étant non déterministes, ils ne sont pas programmables. Heureusement, on sait les déterminiser (*i.e.* construire un automate de Moore qui reconnaît le même langage)...

V Détermination d'un AFND

L'algorithme exposé dans ce paragraphe est appelé *méthode de construction par sous-ensemble*. Il s'agit d'une méthode qui permet d'obtenir un automate de Moore qui reconnaît le même langage qu'un AFND.

V.1 Méthode de construction par sous-ensemble

Soit donc $M = (E, I, t, S, A)$ un AFND à états d'acceptation. Soit Y une partie quelconque de E et $x \in I$ une entrée quelconque.

NOTATION : On note Y_x l'ensemble des états de M accessibles à partir de l'un quelconque des états de Y sur l'entrée x.

EXEMPLE 17.16. Dans l'exemple précédent, et pour $Y = \{e_1, e_3\}$:
- $Y_a = \{e_1\} \cup \{e_3\} = \{e_1, e_3\}$,
- $Y_b = \{e_3\} \cup \varnothing = \{e_3\}$,

– $Y_c = \{e_2, e_3\} \cup \{e_3\} = \{e_2, e_3\}$,

On obtient un automate de Moore $\mathcal{M} = (\mathcal{E}, \mathcal{I}, \mathcal{T}, E_0, \mathcal{A})$ de la manière suivante :

1. L'ensemble $\mathcal{E}$ des états de $\mathcal{M}$ est le sous-ensemble de $\mathcal{P}(E)$ défini par :
   – $S \in \mathcal{E}$,
   – $\forall x \in \mathcal{I}, \forall Y \in \mathcal{E}, Y_x \in \mathcal{E}$.
2. L'état initial de $\mathcal{M}$ est $E_0 = S$.
3. L'ensemble $\mathcal{A}$ des états d'acceptation de $\mathcal{M}$ est défini par $\mathcal{A} = \{Y \in \mathcal{E} \| Y \cap A \neq \emptyset\}$.
4. La fonction de transition d'états est définie par $\mathcal{T} : \mathcal{E} \times \mathcal{I} \to \mathcal{E}, (Y, x) \longmapsto \mathcal{T}(Y, x) = Y_x$.

V.2 En pratique

En pratique, on part de l'état initial de $\mathcal{M}$, c'est-à-dire de S.

Pour chacune des entrées, on forme l'ensemble S_x des états de M que la relation de transition t permet d'atteindre à partir de tous les états de S, et on pose $\mathcal{T}(S, x) = S_x$.

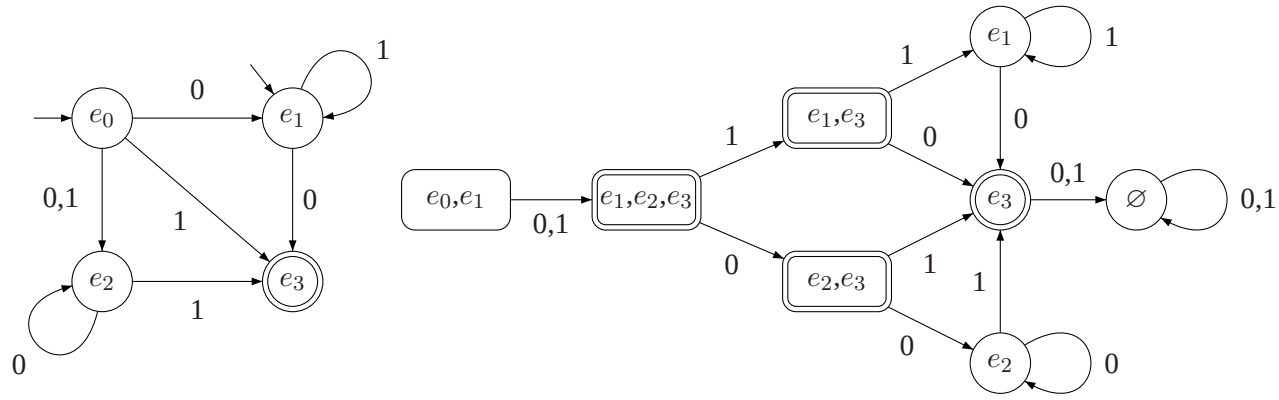
On recommence l'opération pour chacun des états S_x ainsi obtenus (pour les diverses entrées x), etc.

REMARQUE 17.8. Le processus a une fin, parce que E est fini, donc aussi $\mathcal{P}(E)$: si l'automate de départ a n états, l'automate déterminisé en aura au plus 2^n..

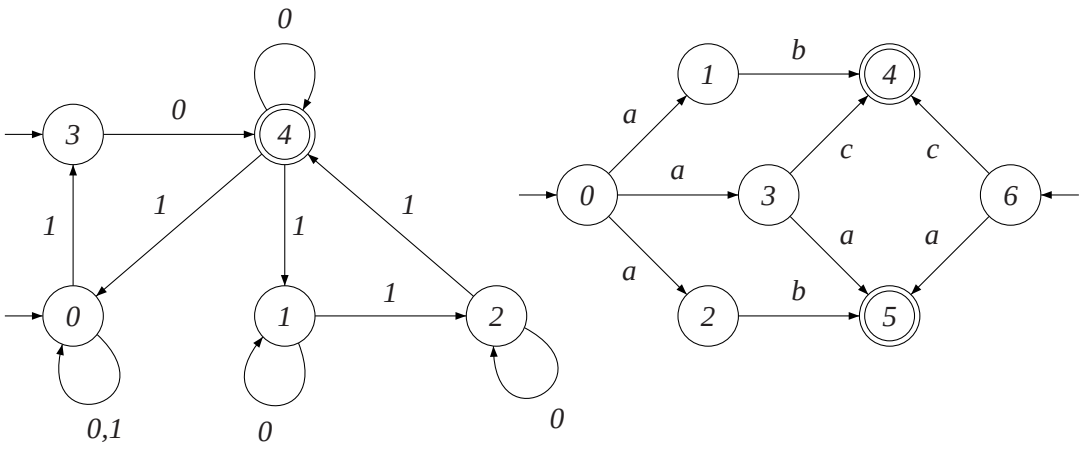
REMARQUE 17.9. Il se peut qu'aucun état ne soit accessible depuis l'un quelconque des états d'un état Y_x de $\mathcal{M}$, sur une entrée y.

On prend alors pour état d'arrivée de $\mathcal{M}$ l'ensemble vide ; celui-ci constitue un état particulier de $\mathcal{M}$, dont on ne peut sortir sur aucune entrée (*c.f.* exemple ci-dessous).

EXEMPLE 17.17. Un exemple...



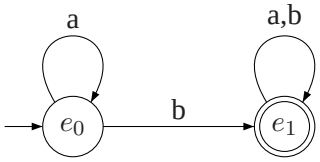
Exercice 17.18. *Construire des automates de Moore équivalents aux AFND ci-dessous :*



Exercice 17.19. *Construire des automates de Moore reconnaissant les langages définis par les expressions régulières :*

1. $(a|b)^*b(a|b)^*$
2. $((a|b)^2)^*|((a|b)^3)^*$
3. $(a^2|b^2)^*|(a^3|b^3)^*$
4. $ba^*|ab|(a|bb)ab^*$.

Réponses : Pour $(a|b)^*b(a|b)^*$



VI Exercices

Exercice 17.20. *Soit $\Sigma = \{a, b\}$.*

1. *Fabriquer un automate qui accepte les mots de longueur pair.*
2. *Fabriquer un automate qui accepte les mots de longueur impair.*
3. *Fabriquer un automate qui accepte les mots dont la longueur est congrue à 1 modulo 4.*

VI.1 Propriétés d'un automate à n états

Exercice 17.21. *Soit un automate fini déterministe A qui a n états et qui n'a pas d'état inaccessible.*

Montrer qu'il existe nécessairement un mot de longueur inférieure où égale à $n-1$ qui est accepté par A.

VI.2 Les palindromes

Exercice 17.22 (Palindrome). *Soit Σ un alphabet dont le nombre de caractères est supérieur ou égal à deux.*

On appelle retournement *l'application $\rho : \Sigma^* \to \Sigma^*$ telle que $\rho(\epsilon) = \epsilon$ et qui associe au mot σ de longueur non nulle le mot τ, nommé* retourné *de σ défini par $\tau(k) = \sigma(n-k+1)$*

1. *Déterminer $\rho(\sigma)$ quand $\sigma = aabcdea$. D'une façon générale, comment le retournement opère-t-il sur la chaîne de caractères qui représente un mot ?*
2. *Exprimer $\rho(\sigma\tau)$ en fonction de $\rho(\sigma)$ et $\rho(\tau)$. Que vaut $\rho(\rho(\sigma))$?*
3. *On dit qu'un mot σ est un* palindrome *si $\rho(\sigma) = \sigma$. Montrer que tout mot de la forme $\rho(\sigma)\sigma$ est un palindrome. Est-ce là tous les palindromes ?*
4. *Si le nombre d'éléments de Σ est n, combien y a-t-il de palindromes de longueur p dans Σ^* ?*

Exercice 17.23 (Suite palindrome). *Soit $\Sigma = \{a, b\}$. Construire un AFD qui accepte les palindromes de longueur 3.*

Exercice 17.24. *Soit $\Sigma = \{a, b\}$. On note L le langage constitué des mots dans lesquels la lettre a, quand elle apparaît, est toujours suivie d'au moins deux lettres b.*

1. *Quels sont les mots de L de longueur inférieure ou égale 6 ?*
2. *Construire un AFD qui accepte L.*
3. *Donner une expression régulière qui décrit L.*

Fin du Chapitre

Chapitre 18

Optimisation d'automates finis

Des automates différents peuvent être associés au même langage.

L'optimisation des programmes d'analyse syntaxique (dont certains sont des réalisations concrètes d'automates finis) rend nécessaire la construction d'un automate minimal (en nombre d'états) qui reconnaissent un langage donné.

On se limitera dans ce chapitre à la simplification d'un automate de Moore (puisque la méthode de construction par sous-ensemble permet de se ramener d'un AFND à un AFD).

I Congruences d'automates

Soit (E, I, t) un AFD et $\mathcal{R}$ une relation d'équivalence sur E.

I.1 Quelques rappels

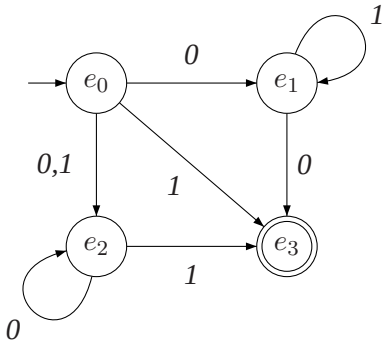
On rappelle que $\mathcal{R}$ est une relation binaire sur l'ensemble E des états, qui a en plus les propriétés suivantes...

- réfléxivité. Pour tout état $e \in E,\ e\mathcal{R}e$,

- symétrie. Pour tout couple d'états $(e_1, e_2) \in E^2$: si $e_1\mathcal{R}e_2$, alors $e_2\mathcal{R}e_1$,

- transitivité. Pour tout triplet d'états $(e_1, e_2, e_3) \in E^3$: si $e_1\mathcal{R}e_2$ et $e_2\mathcal{R}e_3$, alors $e_1\mathcal{R}e_3$.

Exercice 18.1. *On considère l'automate :*



et la relation binaire $e_i \mathcal{R} e_j$ si et seulement si i et j ont la même parité.

Montrez que $\mathcal{R}$ est bien une relation d'équivalence sur E.

On rappelle encore que $t : E \times I \to E$ est la fonction de transition. Par la suite, par souci de concision, on notera $t_x(e)$ pour $t(e, x)$.

I.2 Définition

DÉFINITION 18.1 (CONGRUENCE D'AUTOMATES). *On dit que $\mathcal{R}$ est une* congruence d'automates *si et seulement si*

$$\forall (r, s) \in E^2, \forall x \in I, (r \mathcal{R} s) \Longrightarrow (t_x(r) \mathcal{R} t_x(s))$$

C'est-à-dire si toute paire d'états équivalents modulo $\mathcal{R}$ est transformée par toute entrée en une paire d'états équivalents modulo $\mathcal{R}$. ♢

Exercice 18.2. *La relation d'équivalence $\mathcal{R}$ de l'exercice précédent est-elle une congruence d'automates ?*

I.3 Ensemble quotient

Soit $\mathcal{R}$ une congruence d'automates sur l'AFD (E, I, t).

NOTATION : On note $\tilde{E}$, l'ensemble quotient $E/\mathcal{R}$.

Exercice 18.3. *Représenter le graphe de l'automate M dont la table de transition des états est*

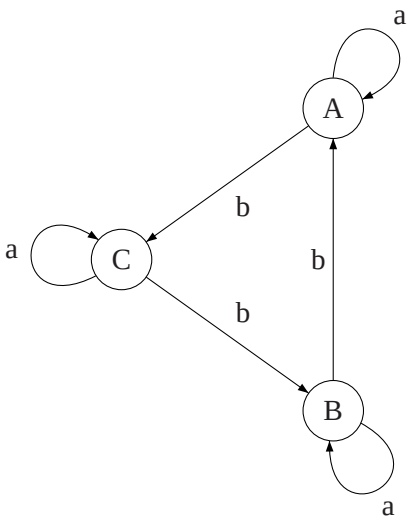
t	a	b
e_0	e_0	e_4
e_1	e_1	e_0
e_2	e_2	e_4
e_3	e_5	e_2
e_4	e_4	e_3
e_5	e_3	e_2

Soit $\mathcal{R}$ la relation d'équivalence pour laquelle $E/\mathcal{R} = \{\{e_0, e_2\}, \{e_1, e_3, e_5\}, \{e_4\}\}$.

1. *Donner la table de transition d'états de l'automate-quotient.*
2. *Représenter son graphe*

Réponses :

E	a	b
$A = \{e_0, e_2\}$	$\{e_0, e_2\}$	$\{e_4\}$
$B = \{e_1, e_3, e_5\}$	$\{e_1, e_3, e_5\}$	$\{e_0, e_2\}$
$C = \{e_4\}$	$\{e_4\}$	$\{e_1, e_3, e_5\}$



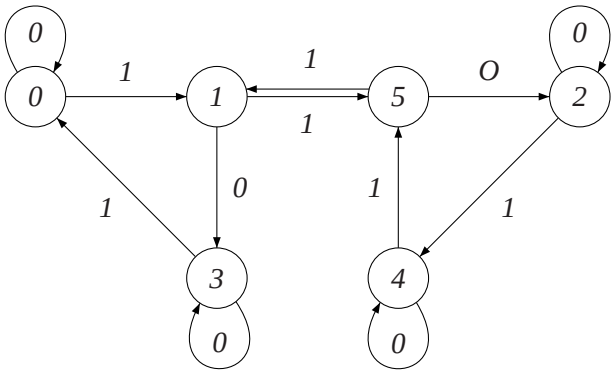
Exercice 18.4. *Soit M l'automate fini dont la table de transition des états est*

t	0	1
1	1	4
2	3	5
3	2	5
4	4	1
5	3	4

Soit $\mathcal{R}$ la relation d'équivalence sur $E = \{1, 2, 3, 4, 5\}$ telle que $1\mathcal{R}4$ et $3\mathcal{R}2$.

1. *Que vaut $E/\mathcal{R}$?*
2. *Montrer que $\mathcal{R}$ est une congruence d'automates.*
3. *Donner la table de transition des états de l'automate-quotient.*
4. *Représenter son graphe.*

Exercice 18.5. *Soit M l'automate fini dont le graphe est représenté par la figure*



Le tableau ci-dessous figure une relation binaire $\mathcal{R}$ dans l'ensemble des états $E = \{0, 1, 2, 3, 4, 5\}$:

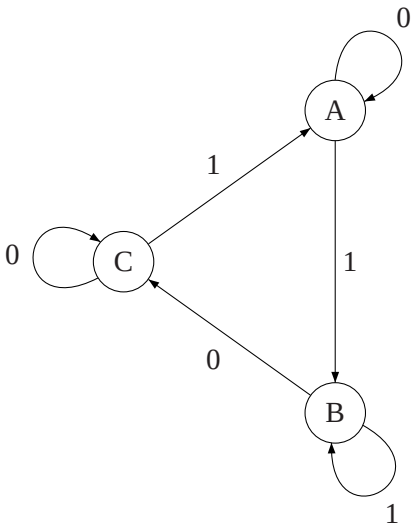
	0	1	2	3	4	5
0	1	0	0	0	1	0
1	0	1	0	0	0	1
2	0	0	1	1	0	0
3	0	0	1	1	0	0
4	1	0	0	0	1	0
5	0	1	0	0	0	1

Un chiffre 1 à l'intersection de la ligne i et de la colonne j signifie que $i\mathcal{R}j$, et un chiffre 0 que ces deux éléments ne sont pas en relation.

1. *Montrer que $\mathcal{R}$ est une relation d'équivalence sur E.*
2. *Montrer que $\mathcal{R}$ est une congruence d'automates.*
3. *Représenter le graphe de l'automate-quotient $M/\mathcal{R}$.*

Réponses :

E	0	1
$A = \{0, 4\}$	$\{0, 4\}$	$\{1, 5\}$
$B = \{1, 5\}$	$\{3, 2\}$	$\{1, 5\}$
$C = \{3, 2\}$	$\{3, 2\}$	$\{0, 4\}$



REMARQUE 18.1. On peut définir une application $\tilde{t}_x : \tilde{E} \to \tilde{E}$ par $\tilde{t}_x(\dot{e}) = [t_x(\dot{e})]$, puis une application $\tilde{t} : \tilde{E} \times I \to \tilde{E}$ par $(\dot{e}, i) \longmapsto \tilde{t}_x(\dot{e})$.

II Équivalence de Nérode

II.1 L'équivalence

Soit $M = (E, I, t, e_0, A)$ un automate de Moore, deux états q et s de E, et $w \in I^*$ un mot d'entrée.

DÉFINITION 18.2 (W-COMPATIBLES). *q est s sont* w-compatible *si et seulement si*

$$t_w(q) \in A \Longleftrightarrow t_w(s) \in A$$

Cette définition permet de définir une relation $\sim$ dans E par

$$(q \sim s) \Longleftrightarrow (\forall w \in I^*, q \text{ et } s \text{ sont } w\text{-compatibles})$$

DÉFINITION 18.3 (ÉQUIVALENCE DE NÉRODE). *Cette relation est manifestement une relation d'équivalence, elle est appelée* équivalence de Nérode *associée à M.* ♢

On démontre facilement que cette équivalence est une congruence d'automates. Tout automate de Moore peut donc être remplacé par son quotient par l'équivalence de Nérode.

Dans ce quotient, le nombre d'états est évidemment plus petit, l'automate obtenu est donc plus « simple ».

II.2 L'algorithme

II.2.1 La théorie

Pour obtenir l'équivalence de Nérode associée à un automate, on dispose de l'algorithme suivant...

Soit $M = \{E, I, t, e_0, A\}$ un automate de Moore.
On définit, pour tout $k \in \mathbb{N}$, une relation $\mathcal{R}_k$ sur E en posant

$$[q\mathcal{R}_k s] \Longleftrightarrow [(\forall w \in I^*), (l(w) \leqslant k \Longrightarrow q \text{ et } s \text{ sont } w\text{-compatibles})]$$

Donc q et s sont en relation par $\mathcal{R}_k$ lorsqu'ils sont w-compatibles, pour tout mot w de longueur inférieure ou égale à k.

Cette relation est clairement une relation d'équivalence, et $\mathcal{R}_{k+1}$ est plus fine que $\mathcal{R}_k$. L'équivalence de Nérode est l'intersection de ces relations $\mathcal{R}_k$, pour toutes les valeurs de k entier.

II.2.2 La pratique

Cet algorithme n'est pas utilisable dans la pratique. On fait plutôt...

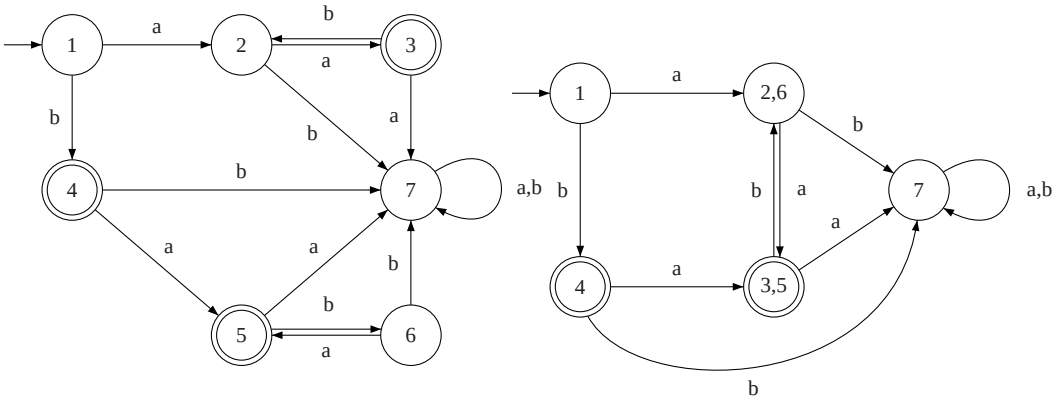
1. Prendre comme partition de départ $P_0 = \{A, E \setminus A\}$.
2. Si $P_k = \{E_1, \ldots, E_n\}$ est la partition correspondant à la relation $\mathcal{R}_k$, morceler éventuellement chaque classe E_i en sous-classes $E_{i1}, E_{i2}, \ldots, E_{ip}$ de manière que deux états q et s appartiennent à la même sous-classe si, pour toute entrée x, les états $t_x(q)$ et $t_x(s)$ appartiennent à la même classe E_j (pouvant dépendre de x).

   L'ensemble des sous-classes obtenues constitue la partition correspondant à la relation $\mathcal{R}_{k+1}$
3. Répéter l'étape précédente jusqu'à ce que $P_{k+1} = P_k$. La relation $\mathcal{R}_k$ est alors l'équivalence de Nérode associée à M.

REMARQUE 18.2. L'étape (3) est nécessairement atteinte, puisque les relations $\mathcal{R}_k$ sont de plus en plus fines.

Au pire, $P = \{\{q\} \| q \in E\}$, la relation est l'égalité : ceci signifie que l'automate n'est pas simplifiable.

EXEMPLE 18.6. Un automate de Moore et l'automate simplifié.



Exercice 18.7. *On donne un AFD par la table de transition des états suivante :*

t	a	b
0	1	2
1	5	3
2	5	1
3	4	5
4	5	4
5	5	5

État initial : 0
États d'acceptation : 4
Cet automate reconnaît le langage défini par l'expression régulière $(a|bb)bab^*$.
Appliquer la méthode de l'équivalence de Nérode pour trouver l'automate minimal reconnaissant le langage.

Exercice 18.8. *Faire de même avec l'automate de Moore dont la table de transition est :*

t	a	b
a	a	c
b	g	d
c	f	e
d	a	d
e	a	d
f	g	f
g	g	c

III Méthode du dual

III.1 Dual d'un automate

Soit $M = (E, I, t, S, A)$ un automate quelconque (AFD ou AFND).

DÉFINITION 18.4. *L'automate dual de* M *est l'automate* $M^{-1} = (E, I, t', A, S)$*, où* t' *est la relation sur* E *obtenue en renversant toutes les flèches sur le graphe de* M*, c'est-à-dire si* $R' \subset (E \times I) \times E$ *est le graphe de la relation* t'*, alors que le graphe de* t *est* R*, on a*

$$((e, i), e') \in R' \iff ((e', i), e) \in R$$

Il est clair que M reconnaît un mot $w \in I^*$ si et seulement si M^{-1} reconnaît le mot w^{-1} (si $w = a_1 a_2 \dots a_n$, alors $w^{-1} = a_n a_{n-1} \dots a_1$).

Le dual d'un automate à comportement déterminé n'est pas nécessairement à comportement déterminé.

III.2 Méthode du dual

Soit M un automate de Moore :

1. Construire l'automate dual M^{-1} de M.
2. Appliquer la construction par sous-ensembles à M^{-1} pour le transformer en automate M' de Moore.
3. Construire le dual M'^{-1} de M'.
4. Appliquer la construction par sous-ensemble à M'^{-1} pour obtenir l'automate de Moore M''.

L'automate M'' est l'automate minimal tel que $\mathcal{L}(M'') = \mathcal{L}(M)$.

Exercice 18.9. *On donne un AFD par la table de transition des états suivante :*

t	a	b
0	1	2
1	3	4
2	3	4
3	5	6
4	5	6
5	7	8
6	7	8
7	9	10
8	9	10
9	11	12
10	11	12
11	1	2
12	1	2

État initial : 0

États d'acceptation : 0 3 4 5 6 7 8 11 12

Cet automate reconnaît le langage défini par l'expression régulière $((a|b)^2)^*|((a|b)^3)^*$.

Appliquer la méthode du dual pour trouver l'automate minimal reconnaissant le langage.

Exercice 18.10. *On donne un automate non déterministe, à transitions instantannées, par la table de*

transition suivante :

t	ε	a	b
0	1, 11		
1	2, 7		
2			3
3	4, 6		
4		5	
5	4, 6		
6	10		
7		8	
8			9
9	10		
10	22		
11	12, 14		
12		13	
13			
14	17		15
15			16
16	17		
17		18	
18	19, 21		
19			20
20	19, 21		
21	22		
22			

État initial : 0
État d'acceptation : 22
Cet automate reconnaît le langage défini par l'expression régulière $ba^*|ab|(a|bb)ab^*$.
Le déterminiser, puis lui appliquer la méthode de votre choix pour obtenir l'automate minimal reconnaissant le langage.

Exercice 18.11. *On donne la table de transition suivante pour un automate fini :*

t	a	b
0	1	2
1	3	4
2	5	6
3	1	2
4	7	8
5	7	8
6	1	2
7	9	10
8	10	11
9	7	8
10	10	10
11	7	8

L'état initial est 0, et les états d'acceptation sont 4, 5, 9 et 11.

Appliquer à cet automate l'algorithme de votre choix pour obtenir l'automate minimal reconnaissant le même langage (équivalence de Nérode ou dual).

(L'automate minimal possède 7 états).

IV Synthèse

IV.1 Outils

IV.1.1 Construction par sous-ensemble

Domaine d'application : Cette méthode s'applique aux automates finis non déterministes. Il est aussi possible de l'utiliser sur un AFD, mais c'est sans intérêt.

Résultat : Automate reconnaissant le même langage.

But : Obtenir un AFD reconnaissant le même langage.

Autre utilisation : Peut permettre d'obtenir l'automate minimal reconnaissant le même langage (méthode du dual).

IV.1.2 Équivalence de Nérode

Domaine d'application : Cette méthode ne s'applique qu'aux automates finis déterministes (AFD).

Résultat : Le quotient de l'automate considéré par l'équivalence de Nérode ; un automate ne reconnaissant généralement pas le même langage.

But : Simplifier l'automate considéré, si c'est possible, grâce à la méthode des quotients.

IV.2 Méthodes d'optimisation

IV.2.1 Méthode des quotients

Domaine d'application : Cette méthode ne s'applique qu'aux automates finis déterministes (AFD).

Moyens : Équivalence de Nérode.

Résultat : On obtient l'automate minimal reconnaissant le même langage.

IV.2.2 Méthode du dual

Domaine d'application : Cette méthode ne s'applique qu'aux automates finis déterministes (AFD).

Moyens : Construction par sous-ensembles.

Résultat : L'automate de Moore minimal reconnaissant le même langage.

Efficacité : Élégant, mais pas efficace.

Fin du Chapitre

Chapitre 19

Construction d'automates finis à partir d'expressions rationnelles

L'algorithme exposé ici est appelé *algorithme de Thompson*. Il permet de construire un AFND à partir d'une expression rationnelle.

I Automates à transitions instantanées

DÉFINITION 19.1 (TRANSITION INSTANTANÉE). *Une* transition instantanée *est une évolution possible de l'automate d'un état vers un autre sans qu'aucune entrée ne soit produite.* ♢

Les automates à transitions instantanées interviennent dans l'algorithme de Thompson de construction automatique d'un automate reconnaissant le langage associé à une expression rationnelle...

II Données et résultat

Données une expression rationnelle r sur un alphabet Σ.

Résultat Un AFND M tel que $\mathcal{L}(M) = \mathcal{L}(r)$, qui ne comporte qu'un seul état initial et un seul état d'acceptation...

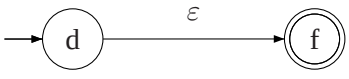
III Algorithme

1. Décomposer l'expression en ses sous-expressions.

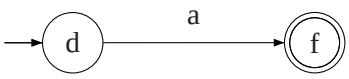
2. En utilisant les règles (a) et (b) ci-dessous, construire un AFND pour les symboles terminaux de la grammaire ou la chaîne vide (si un même symbole a apparaît plusieurs fois dans l'expression rationnelle, un AFND séparé est construit pour chacune de ses occurences).

3. Combiner ensuite récursivement les AFND de base en utilisant la règle (c) jusqu'à obtenir l'AFND pour l'expression rationnelle toute entière.

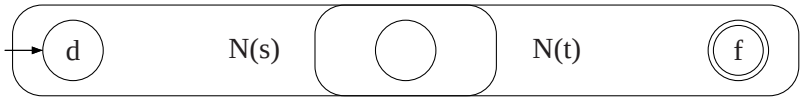
   (a) Pour $<>$, construire l'AFND :



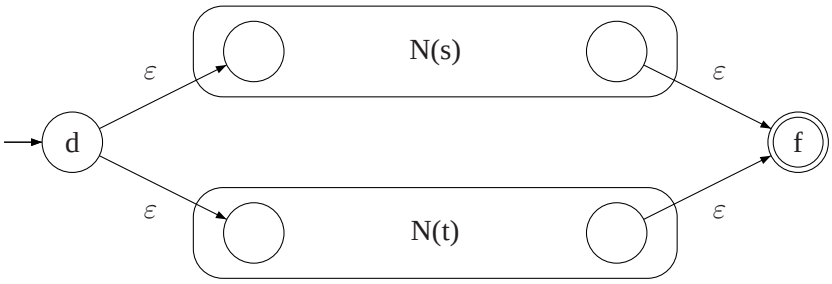
(b) Pour $a \in \Sigma$, construire l'AFND :



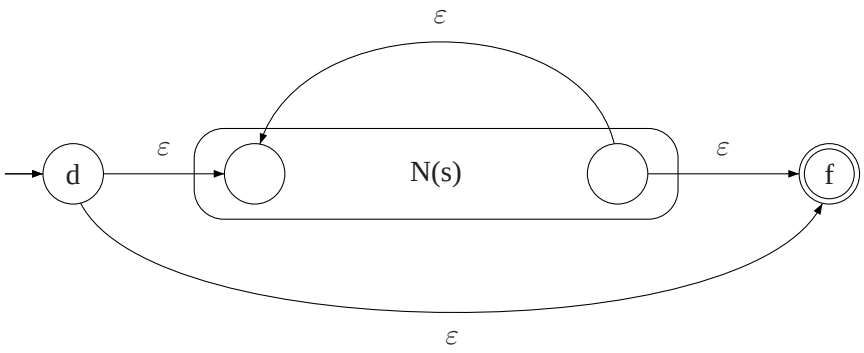
(c) Si $N(s)$ et $N(t)$ sont les AFND pour les expressions rationnelles s et t,
- Pour st, on construit l'AFND :



L'état initial de $N(t)$, qui est état d'acceptation pour $N(s)$, perd ce double caractère dans la nouvelle construction.
- Pour $s|t$, on construit l'AFND



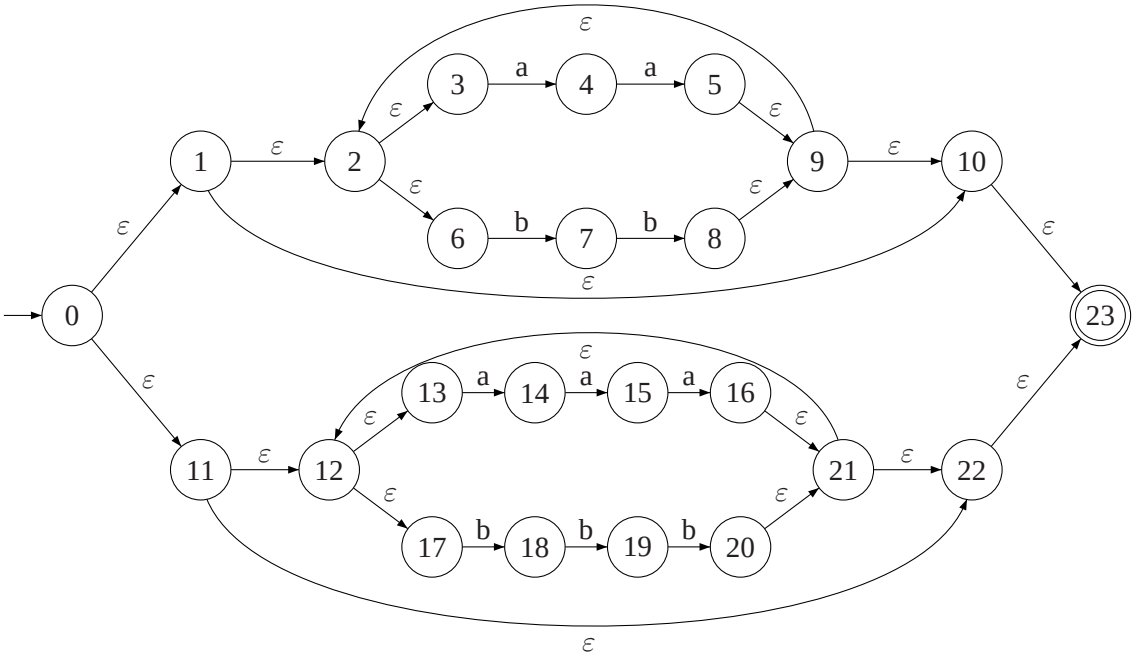
Les états initiaux et les états d'acceptation des AFND de $N(s)$ et de $N(t)$ perdent leur caractère dans le nouvel AFND.
- Pour l'expression rationnelle s^*, on construit l'AFND composé $N(s^*)$:



Les états initiaux et les états d'acceptation de $N(s)$ perdent leurs qualités.
- Pour une expression parenthésée (s), utiliser $N(s)$ lui-même.

IV Exemple

On applique l'algorithme sur l'exemple : $(a^2|b^2)^*|(a^3|b^3)^*$.



Exercice 19.1. *On donne l'expression rationnelle* $(a|b)^*|(a^2|b^2)^*$.

Utiliser l'algorithme de Thompson pour obtenir un automate non déterministe, à transitions instantanées, reconnaissant le langage.

On rappelle que, par définition, sont accessibles sur une entrée donnée x depuis un état donné e : les états accessibles en effectuant successivement

- un nombre arbitraire (éventuellement nul) de transitions instantannées,
- une transition d'entrée x et
- un nombre arbitraire (éventuellement nul) de transitions instantannées.

EXEMPLE 19.2. Pour...

- L'état 3, avec entrée a : on passe à l'état 4.
- Si l'entrée a se produit au départ, les états accessibles sont $\{4, 14\}$.
- Et, à partir de l'état 4 et de l'entrée a : $\{5, 9, 2, 10, 23, 3, 6\}$.

D'où l'algorithme suivant simplifiant l'automate de l'algorithme de Thompson...

V Finalisation

L'automate construit par algorithme de Thompson n'est pas utilisable tel quel.

Il faut en supprimer les transitions instantanées, ce qui se fait par un algorithme voisin de la construction par sous-ensembles. Il faudra, par ailleurs, ensuite, le déterminiser et le minimiser.

Soit M l'automate de Thompson obtenu par l'algorithme précédent, E l'ensemble de ses états, $e \in E$ son (unique) état initial et $a \in E$ son (unique) état d'acceptation.

On remplace cet automate par un automate $\mathcal{M}$ qui reconnaît le même langage, dont l'ensemble des états est $\mathcal{E}$, l'état initial est S, et l'ensemble des états d'acceptation est $A \subset \mathcal{E}$ et qui est obtenu de la manière suivante :

– S est composé de e et de tous les états de M qui sont accessibles depuis e par un nombre quelconque de transitions instantanées (éventuellement aucune). Dans l'exemple précédent,

$$S = \{0, 1, 11, 2, 10, 12, 22, 3, 6, 13, 17, 23\}$$

.
– Soit $Y \in \mathcal{E}$ une partie de l'ensemble des états, et x une entrée. L'image Y_x de Y est constituée des états accessibles depuis un état quelconque de Y par (exactement) une entrée x, suivie d'un nombre quelconque de transitions instantanées.
– $A = \{Y \in \mathcal{E} | Y \cap \{a\} \neq \emptyset\}$, à savoir les parties Y ci-dessus définies de E qui contiennent l'ancien état d'acceptation.

Exercice 19.3. *Finalisez l'automate de l'exercice précédent. Le déterminiser, puis obtenir l'automate minimal reconnaissant le même langage.*

Exercice 19.4 (Reprise d'un exercice précédent, version Thompson). *Construire des automates de Moore reconnaissant les langages définis par les expressions rationnelles :*

1. $(a|b)^*b(a|b)^*$
2. $((a|b)^2)^*|((a|b)^3)^*$
3. $ba^*|ab|(a|bb)ab^*$.

Exercice 19.5. *Donner les automates finis minimaux (table de transition, diagrame) reconnaissant les langages associés aux expressions rationnelles suivantes :*

– $(a|b)^*(aaa|bb)$
– $(a|bb)^*abb^*$

Fin du Chapitre

Chapitre 20

Automates à pile

On l'a dit, les expressions rationnelles ne permettent pas de représenter tous les langages.

EXEMPLE 20.1. Le langage défini par $\{a^n b^n | n \in \mathbb{N}^*\}$ n'est pas représentable par une expression rationnelle.

Exercice 20.2. *Déterminez d'autres langages non reconnus par expression rationnelle.*

On souhaite maintenant étudier « une plus grande classe » de langages, et voir ce qu'il manque à nos automates pour pouvoir les associer à ces langages.

Dans l'exemple précédent, il faudrait « noter quelque part » le nombre de a rencontrés, pour s'assurer qu'il y aura bien autant de b. On peut imaginer y arriver avec une pile jointe à un automate non déterministe.

REMARQUE 20.1. Très précisément, les automates à pile vont jouer pour les langages dits non contextuels (voir chapitre suivant) le rôle des automates finis pour les langages rationnels (: représentables par expressions rationnelles).

I Automates à pile, déteministes ou pas.

I.1 Automate à pile non déterministe

I.1.1 Définition

DÉFINITION 20.1 (AUTOMATE À PILE). *Un* automate à pile *est donné par*

1. *Un alphabet d'entrée Σ (ensemble fini non vide),*
2. *Un ensemble d'états E (fini non vide),*
3. *Un état initial $e_0 \in E$,*
4. *Éventuellement, une partie $A \subset E$ des états d'acceptation (pour un automate à pile dit à* états d'acceptation*),*
5. *Un alphabet de pile P (fini, non vide),*
6. *Un symbole de pile initial p_0,*
7. *Éventuellement, un ensemble $Q \subset P$ de symboles de sommet de pile,*
8. *Enfin, une relation $t : E \times (\Sigma \cup \{\varepsilon\}) \times P \to E \times P^*$.* ♢

REMARQUE 20.2. Le symbole de pile initial n'est pas toujours noté p_0.

I.1.2 Transition

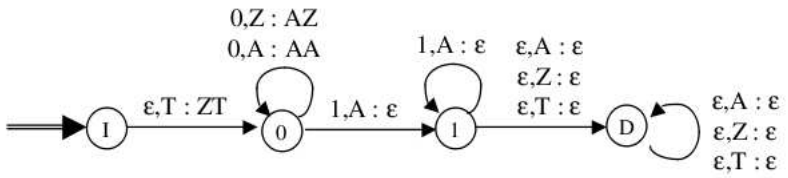
DÉFINITION 20.2 (TRANSITION). *Lorsque* $((e, x, p), (e', q))$ *appartient au graphe de la relation* t*, on parle de la transition* $(e, x, p) \longmapsto (e', q)$*.* ♢

Elle indique que, lorsque l'automate se trouve :
- dans l'état e,
- alors que le symbole de sommet de pile est p,
- sur l'entrée x,

alors il évolue
- vers l'état e',
- le symbole p est dépilé,
- et le mot de pile q (éventuellement plusieurs symboles de pile) est empilé.

Comme t est une relation, il est possible, dans le cas d'un automate à pile non déterministe, qu'il y ait plusieurs transitions possibles dans la même situation (même état, même entrée, même symbole de sommet de pile).

EXEMPLE 20.3 (AUTOMATE À PILE NON DÉTERMINISTE). Ici, l'alphabet d'entrée est $\{0, 1\}$, le fond de pile est T et alphabet de pile $\{T, Z, A\}$:



Cet automate reconnaît, par pile vide, l'ensemble

$$\{0^n 1^m, n \geqslant m > 0\}$$

REMARQUE 20.3. Si un automate à états finis reconnaît un mot lorsqu'il s'arrête dans un état d'acceptation, il n'en est pas de même pour les automates à pile : on verra par la suite que ceux-ci ont plusieurs critères pour décider si un mot est reconnu ou non.

Le critère de reconnaissance *par pile vide* fait partie de ceux-ci : lorsque l'automate s'arrête avec une pile vide, le mot est accepté.

On remarque que les mots 01, 001, 0011 sont acceptés par cet automate. Il n'en est pas de même pour 011.

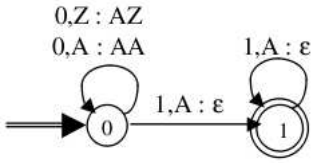
I.2 Automate à pile déterministe

I.2.1 Définition

DÉFINITION 20.3 (AUTOMATE À PILE DÉTERMINISTE). *Dans le cas d'un* automate à pile déterministe*,* t *est une fonction sur son domaine de définition, et* $(e', q) = t(e, x, p)$*.* ♢

REMARQUE 20.4. En particulier si t est définie pour le triplet (e, x, p), $t(e, x, p)$ est unique.

EXEMPLE 20.4 (AUTOMATE À PILE DÉTERMINISTE). Ici, l'alphabet d'entrée est $\{0, 1\}$, le fond de pile est Z et alphabet de pile $\{Z, A\}$:



REMARQUE 20.5. Cet automate à pile déterministe reconnaît, par état final, le même langage que l'exemple précédent.

I.2.2 Transitions

Il est fondamental de comprendre qu'une transition d'un automate à pile, quelle qu'elle soit, exige toujours de dépiler un symbole de pile.

REMARQUE 20.6. Autrement dit, si la pile vient à se vider, l'automate se bloque et ne peut plus évoluer, même si le mot d'entrée n'a pas été entièrement lu.

Ceci explique le « symbole de pile initial » , la plupart du temps sans intérêt autre que celui de permettre le début du calcul dans l'automate.

REMARQUE 20.7. On admet des « transitions vides » , du type $(e, \varepsilon, p) \longmapsto \ldots)$, qui permettent de ne pas avancer sur le mot d'entrée, par exemple pour vider la pile. Il faut les utiliser avec précautions.

II Calcul dans un automate à pile

II.1 Encore quelques définitions...

DÉFINITION 20.4 (CONFIGURATION). *On appelle* configuration d'un automate à pile *un triplet* (e, w, q) *où*

- e *est l'état dans lequel se trouve l'automate à l'instant considéré,*
- w *est le mot à lire,*
- q *est le mot de pile (en tête, le symbole de sommet de pile, en queue, le symbole de fond de pile).*$\diamondsuit$

DÉFINITION 20.5 (DÉRIVATION VALIDE). *Si*

- q *est de la forme* pq' *où* p *est le symbole de sommet de pile,*
- w *est de la forme* xw'*, où* x *est un symbole d'entrée,*
- *il existe une transition* $(e, x, p) \longmapsto (e', q'')$*,*

alors, après application de cette transition, la nouvelle configuration de l'automate est $(e', w', q''q')$ *et la correspondance* $(e, w, q) \vdash (e', w', q''q')$ *est appelée une* dérivation valide dans l'automate. $\diamondsuit$

DÉFINITION 20.6 (CALCUL VALIDE). *Un* calcul valide *dans l'automate est une famille de dérivations* $(e_1, w_1, q_1) \vdash (e_2, w_2, q_2) \vdash \ldots \vdash (e_n, w_n, q_n)$.

On dit que ce calcul valide mène de la configuration (e_1, w_1, q_1) *à la configuration* (e_n, w_n, q_n) $\diamondsuit$

NOTATION : On peut noter cela : $(e_1, w_1, q_1) \overset{*}{\vdash} (e_n, w_n, q_n)$.

DÉFINITION 20.7 (MOT RECONNU). *On dit qu'un mot w est* reconnu *par un automate à pile (état initial e_0, symbole de pile initial p_0) lorsqu'il existe un calcul valide*

$$(e_0, w, q_0) \overset{*}{\vdash} (e, \varepsilon, q)$$

tel que, au choix

- *e est un état d'acceptation : le mot w est dit* reconnu par l'état d'acceptation,
- *q est de la forme $q_s q'$ où $q_s \in Q$: le mot w est dit* reconnu par symbole de sommet de pile,
- *$q = \varepsilon$ (symbole de pile vide) : le mot w est dit* reconnu par pile vide, ♢

REMARQUE 20.8. On peut envisager des reconnaissances par combinaison de deux de ces conditions, voire les trois simultanément.

On démontre que...

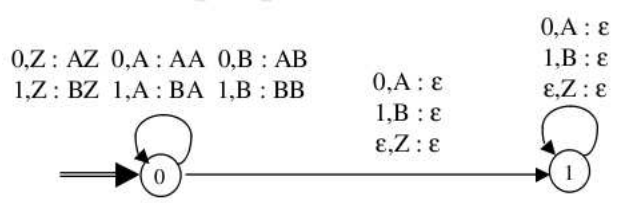
> PROPRIÉTÉ 20.1 : Tous ces types de reconnaissance peuvent se ramener à la seule reconnaissance par pile vide (éventuellement avec un automate beaucoup plus compliqué).

C'est pourquoi nous n'envisagerons plus que cette dernière dans la suite. Enfin,

DÉFINITION 20.8 (LANGAGE RECONNU). *Le* langage reconnu *par automate est l'ensemble des mots reconnus par cet automate (par le même mode de reconnaissance).* ♢

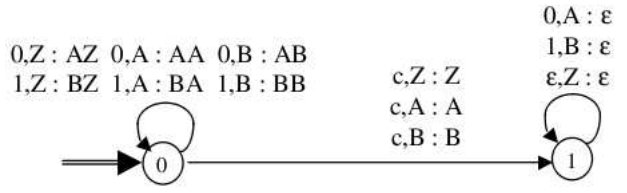
II.2 Premiers exemples

EXEMPLE 20.5. Automate à pile (non déterministe) reconnaissant, par pile vide, l'ensemble des mots de la forme ww^t, concaténation de w (constitué de 0 et de 1) et de son image miroir :



(Alphabet d'entrée $\{0, 1\}$, fond de pile Z, alphabet de pile $\{Z, A, B\}$.)

EXEMPLE 20.6. Automate à pile reconnaissant, par pile vide, l'ensemble des mots de la forme wcw^t, concaténation de w (constitué de 0 et de 1) et de son image miroir séparés par le caractère c :



(Alphabet d'entrée $\{0, 1, c\}$, fond de pile Z, alphabet de pile $\{Z, A, B\}$.)

II.3 Exemple plus complet : le langage $\{0^n 1^n | n \in \mathbb{N}^*\}$

Le principe est le suivant :

1. Tant qu'on lit des 0, on les empile, sans changer d'état,
2. Au premier 1 rencontré, on change d'état (pour ne plus accepter de 0),
3. On dépile alors un à un les symboles de pile (sans jamais rien empiler),
4. Si le mot se vide en même temps que la pile, il comportait autant de 0 que de 1.

Voici les transformations :
- $(e_0, 0, p_0) \to (e_0, 0)$
- $(e_0, 0, 0) \to (e_0, 00)$
- $(e_0, 1, 0) \to (e_1, \varepsilon)$
- $(e_1, 1, 0) \to (e_1, \varepsilon)$

Exercice 20.7. *Représentez cet automate.*

III Construction d'un automate à pile

III.1 Introduction à la méthode

On peut évidemment utiliser la méthode « directe », comme dans l'exemple précédent.

Pour les langages plus complexes, il peut être nécessaire d'avoir recours à un algorithme. Nous l'aborderons par l'exemple de la grammaire écrite pour les expressions algébriques élémentaires :

$$\begin{array}{lll} < expression > & ::= & < terme > \\ & ::= & < terme > \ '+' \ < expression > \\ < terme > & ::= & < facteur > \\ & ::= & < facteur > \ '*' \ < terme > \\ < facteur > & ::= & '(' < expression > \ ')' \\ & ::= & < variable > \end{array}$$

en omettant la définition du SNT « variable », inutile ici.

III.2 Utilisation d'un symbolisme

On introduit un nouveau symbolisme (développé dans la suite) pour cette même grammaire ; il se comprend aisément :

$$\begin{array}{lll} E & -> & T \\ E & -> & T + E \\ T & -> & F \\ T & -> & F * T \\ F & -> & (E) \\ F & -> & a \\ F & -> & b \\ & ... & \\ F & -> & z \end{array}$$

...en admettant comme symboles de variables les caractères alphabétiques minuscules.

III.3 Algorithme de construction

Le principe est de construire un automate à pile non déterministe qui admet des transitions vides :

1. Au départ, sur une transition vide, on empile le SNT de l'axiome de la grammaire (ici, E) : *c.f.* transition (1).
2. associer à chaque règle non terminale une ϵ-transition qui empile les symboles de la partie droite (*c.f.* transitions (2) à (6))
3. associer à chaque règle terminale $A \to x$ une transition $(e_0, x, A) \mapsto (e_0, \epsilon)$ (*c.f.*. transitions (7) et (8))
4. associer à chaque symbole terminal une transition qui reconnaît ce symbole et dépile ce caractère (*c.f.* transitions (9) à (12))

On obtient les transitions suivantes :

$$
\begin{array}{lllll}
(1) & (e_0, \varepsilon, p_0) & \longmapsto & (e_0, E) \\
(2) & (e_0, \varepsilon, E) & \longmapsto & (e_0, T) \\
(3) & (e_0, \varepsilon, E) & \longmapsto & (e_0, T + E) \\
(4) & (e_0, \varepsilon, T) & \longmapsto & (e_0, F) \\
(5) & (e_0, \varepsilon, T) & \longmapsto & (e_0, F * T) \\
(6) & (e_0, \varepsilon, F) & \longmapsto & (e_0, (E)) \\
(7) & (e_0, a, F) & \longmapsto & (e_0, \varepsilon) \\
\vdots & \vdots & \vdots & \vdots \\
(8) & (e_0, z, F) & \longmapsto & (e_0, \varepsilon) \\
(9) & (e_0, +, +) & \longmapsto & (e_0, \varepsilon) \\
(10) & (e_0, *, *) & \longmapsto & (e_0, \varepsilon) \\
(11) & (e_0, (, () & \longmapsto & (e_0, \varepsilon) \\
(12) & (e_0,),)) & \longmapsto & (e_0, \varepsilon)
\end{array}
$$

III.4 Exercices

Exercice 20.8. *Soit l'automate à pile défini par $\Sigma = \{a, b\}$, $E = \{q_0, q_1, q_2\}$, $P = \{p, A\}$ et les transitions suivantes*

$$
\begin{array}{lll}
(q_0, a, p_0) & \mapsto & (q_1, A) \\
(q_1, b, A) & \mapsto & (q_2, \epsilon) \\
(q_1, a, p) & \mapsto & (q_1, Ap) \\
(q_2, b, A) & \mapsto & (q_2, \epsilon)
\end{array}
$$

1. *Donner les enchaînements des transitions permettant d'accepter $aabb$ en précisant s'il y a des points de non déterminisme dans la dérivation.*
2. *Donner l'enchanement des transitions conduisant à l'échec de l'acceptation de la chaîne $aaba$.*
3. *Décrire l'état de l'automate après avoir lu n symboles a en entrée ($n \in \mathbb{N}$). Quelle est alors la seule façon de vider la pile ? En déduire le langage reconnu par cet automate à pile avec arrêt sur pile vide.*

Exercice 20.9. *Pour $u \in \Sigma^*$, on note $|u|_a$ le nombre de a dans u et $|u|_b$ le nombre de b dans u.*
Donner un automate à pile qui reconnaît $L = \{u \in \Sigma^*, |u|_a = |u|_b\}$.

Exercice 20.10. *Soit $\Sigma = \{0, 1\}$. Donner un automate à pile qui accepte un mot $u \in \Sigma^*$ ssi aucun préfixe de u ne contient plus de 1 que de 0. Préciser si l'automate est déterministe.*

Exercice 20.11. *Soit $\Sigma = \{1, 2\}$. Donner un automate à pile qui reconnaît le langage suivant :*

$$\{1^n 2^n | n \geqslant 0\} \cup \{1^n 2^{2n} | n \geqslant 0\}$$

Préciser si l'automate est déterministe.

Exercice 20.12. *Soit $\Sigma = \{a, b, c\}$. Donner un automate à pile qui reconnaît le langage suivant :*

$$\{a^i b^j c^k | i = j \text{ ou } j = k\}$$

Préciser si l'automate est déterministe.

Exercice 20.13. *Soit G la grammaire suivante :*

$$S \to aAB \quad A \to aAB|a \quad B \to bBA|aC \quad C \to BaA.$$

1. *Donner des exemples de mots reconnus.*
2. *Donner un automate à pile qui reconnaît le langage généré par la grammaire G.*

Exercice 20.14. *Soit G la grammaire suivante :*

$$S \to aAB \quad A \to aAB|a \quad B \to bBA|aC|b \quad C \to BaA.$$

Donner un automate à pile qui reconnaît le langage généré par la grammaire G.

Fin du Chapitre

Chapitre 21

Description d'un langage par une grammaire

Dans ce paragraphe, nous précisons les éléments sur les grammaires et sur les langages qui ont déjà été vus jusqu'à présent.

I Langages

NOTATION : Soit Σ un ensemble de symboles, on note par Σ^* l'ensemble des mots sur Σ, c'est-à-dire l'ensemble des assemblages de symboles de Σ.

PROPRIÉTÉ 21.1 : Pour l'opération de concaténation des assemblages de symboles, Σ^* constitue un monoïde (opération associative, admettant un élément neutre, la chaîne vide, notée ε) appelé *monoïde libre sur* Σ.

DÉFINITION 21.1 (LANGAGE SUR Σ). *On appelle* langage *sur* Σ *toute partie de* Σ^*.

Tout le problème consiste à se donner les moyens de définir un langage. L'un d'entre eux est de se donner un système générateur de ce langage, qu'on appelle grammaire.

Nous nous limiterons ici aux grammaires de Chomsky.

II Grammaires

II.1 Définitions

DÉFINITION 21.2 (GRAMMAIRE DE CHOMSKY). *Une* grammaire de Chomsky *est un quadruplet* $G = (\Sigma, N, P, S)$*, où*

- Σ *est un ensemble fini, appelé* alphabet du langage*, ou ensemble des symboles terminaux du langage,*
- N *est un autre ensemble fini, disjoint de* Σ*, et appelé ensemble des symboles non-terminaux, qui constituent un méta-langage dans lequel sera décrit le langage,*

– *P est une partie finie de $((\Sigma \cup N)^* \setminus \Sigma^*) \times (\Sigma \cup N)^*$: c'est l'ensemble des règles de la grammaire,*
– *S est un élément de N (un symbole non-terminal),* symbole initial *ou* axiome *de la grammaire (*$< expression >::= ...$*).* ♢

Les éléments de P sont aussi appelés *productions* .

Ce sont des couples de suites de symboles, la première de ces deux suites comportant au moins un symbole non-terminal. Elles sont de la forme (α, β), où $\alpha \in (\Sigma \cup N)^*$, mais $\alpha \notin \Sigma^*$ et $\beta \in (\Sigma \cup N)^*$.

NOTATION : Une telle production est le plus souvent notée $\alpha \to \beta$, qui se lit « α se réécrit en β ».

REMARQUE 21.1. La flèche n'est pas ici le symbole de l'implication logique, mais celui de réécriture (dans la symbolisation BNF, ou Bakus-Naur form, le symbole de réécriture est « : := »).

II.2 Types de grammaires de Chomsky

On distingue divers types de grammaires de Chomsky :

II.2.1 Les grammaires non restreintes, ou de type 0

Aucune restriction n'est apportée aux productions.

Les langages sont dits récursivement énumérables. Ils sont reconnus par des machines de Turing non déterministes à plusieurs bandes.

II.2.2 Les grammaires contextuelles, ou de type 1

Les langages correspondants sont les langages contextuels.

Ceux-ci constituent un sous-ensemble des langages récursifs (c'est-à-dire récursivement énumérables ainsi que leur complémentaire).

Ils sont reconnus par des machines de Turing déterministes.

II.2.3 Les grammaires algébriques, ou de type 2

Toute production est de la forme $A \to \alpha$, où $A \in N$ et $\alpha \in (\Sigma \cup N)^*$.

Il y a équivalence entre les langages reconnaissables par des automates à pile et les langages algébriques (engendrés par une grammaire algébrique).

II.2.4 Les grammaires régulières, ou de type 3

Chaque production est de l'une des formes $A \to xB$ ou $A \to x$, avec $(A, B) \in N^2$ et $x \in \Sigma^*$.

Il y a équivalence entre les langages reconnaissables par des automates finis et les langages réguliers (certains disent « rationnels » ; engendrés par une grammaire régulière).

II.2.5 Langage associé à une grammaire

Réciproquement, le langage peut être considéré comme langage associé à la grammaire G, $\mathcal{L}(G)$.

III Un exemple de grammaire contextuelle

Nous n'avons jusqu'à présent considéré que des grammaires de type au moins 2, puisque toutes nos règles de grammaire (écrites jusqu'à présent en symbolisme BNF) ont toujours consisté en la définition d'un symbole non-terminal.

EXEMPLE 21.1 (GRAMMAIRE CONTEXTUELLE). Voici un exemple de grammaire contextuelle : celle qui permet de définir le langage $\{a^n b^n c^n | n \in \mathbb{N}^*\}$.

1. $S \to aSBC$
2. $S \to aBC$
3. $CB \to BC$
4. $aB \to ab$
5. $bB \to bb$
6. $bC \to bc$
7. $cC \to cc$

Ces grammaires sont appelées « contextuelles » parce qu'il est impossible de donner une définition « indépendante » de chacun des SNT, comme dans les grammaires algébriques.

EXEMPLE 21.2. On ne peut, par exemple dans la grammaire ci-dessus, pas donner de définition du SNT B indépendamment des symboles qui l'entourent, donc la définition de B est sensible au contexte et la grammaire dans laquelle elle figure est dite contextuelle.

Pour se convaincre de la validité de cet exemple de grammaire, voici l'analyse de la chaîne correcte $aaabbbccc$ en utilisant les règles (ce que l'on appelle une dérivation de chaîne relativement à la grammaire).

– Il faut dériver S
– S se dérive en $aSBC$ (règle 1)
– S se dérive en $aSBC$ (règle 1), donc $aSBC$ en $aaSBCBC$
– CB se dérive en BC (règle 3), donc $aaSBCBC$ en $aaSBBCC$
– S se dérive en aBC (règle 2), donc $aaSBBCC$ en $aaaBCBBCC$
– CB se dérive en BC (règle 3), donc $aaaBCBBCC$ en $aaaBBCBC$
– CB se dérive en BC (règle 3), donc $aaaBBCBCC$ en $aaaBBBCCC$
– aB se dérive en ab (règle 4), donc $aaaBBBCCC$ en $aaabBBCCC$
– bB se dérive en bb (règle 5), donc $aaabBBCCC$ en $aaabbBCCC$
– bB se dérive en bb (règle 5), donc $aaabbBCCC$ en $aaabbbCCC$
– bC se dérive en bc (règle 6), donc $aaabbbCCC$ en $aaabbbcCC$
– cC se dérive en cc (règle 7), donc $aaabbbcCC$ en $aaabbbccC$
– cC se dérive en cc (règle 7), donc $aaabbbccC$ en $aaabbbccc$

La dérivation de $aaabbbccc$ à partir de S est couronnée de succès, l'expression est correct (on n'a pas fait figurer les tentatives d'application de règles qui aboutissent à des échecs, en raison du non-déterminisme de la grammaire).

Fin du Chapitre

Chapitre 22

Exercices sur les grammaires, langages et automates

Exercice 22.1 (Construction par sous-ensembles). *Représenter graphiquement l'AFND dont la table de la relation de transition des états est donnée par :*

t	a	b
0	$0, 1$	3
1	$-$	2
2	2	1
3	$-$	$0, 1, 2$

Les états initiaux sont 0 et 1, le seul état d'acceptation est 2. Le déterminiser en lui appliquant la « construction par sous-ensembles » ; donner le graphe du résultat obtenu.

Exercice 22.2 (Automate-Quotient). *On donne la table de transition des états d'un AFD :*

t	a	b	c
r	r	v	r
s	s	p	s
u	r	v	s
v	r	v	s

Soit $\mathcal{R}$ la plus petite relation d'équivalence sur E (ensemble des états) telle que $u\mathcal{R}v$, $p\mathcal{R}v$ et $s\mathcal{R}$.

1. *Vérifier qu'il s'agit d'une congruence d'automates et dessiner le graphe de l'automate-quotient.*
2. *Sachant que, dans l'automate d'origine, l'état initial est p et que le seul état d'acceptation est u, décrire le langage reconnu par l'automate.*

Exercice 22.3 (Construction d'automates de Moore). *On demande de déssiner un automate de moore reconnaissant le langage :*

1. *décrit par l'expression rationnelle $(a|bb)^*bab^*$,*
2. *défini par l'expression rationnelle $(a|b|c)^*(abc|cba)$,*
3. *défini sur l'alphabet $\{a, b\}$ des mots non vides ne comportant pas plus de 2 lettres b consécutives.*

Fin du Chapitre

Cinquième partie

Théorie des graphes

Chapitre 23

Graphes non orientés

La notion de graphe généralise amplement la notion de relation sur un ensemble ; elle s'intéresse à la façon dont sont liés les objets. Avec les plans de métro, les cartes routières, les schémas de circuits électriques, les formules des molécules, les organigrammes, les arbres généalogiques, on utilise chaque jour des graphes...

I Définitions et premiers exemples

I.1 Définitions

DÉFINITION 23.1 (GRAPHE NON ORIENTÉ, SOMMET, ARÊTE). *Un* graphe non orienté $G = (S, A)$ *est défini par l'ensemble fini* $S = \{s_1, s_2, ..., s_n\}$ *dont les éléments sont appelés* sommets*, et par l'ensemble fini* $A = \{a_1, a_2, ..., a_m\}$ *dont les éléments sont appelés* arêtes.

Une arête a *de l'ensemble* A *est définie par une paire non-ordonnée de sommets, appelés les* extrémités *de* a*. Si les extrémités coïncident, on parle de* boucle.

Si l'arête a *relie les sommets* s_i *et* s_j*, on dira que ces sommets sont* adjacents*, ou incidents avec* a*, ou encore que l'arête* a *est incidente avec les sommets* s_i *et* s_j*.*

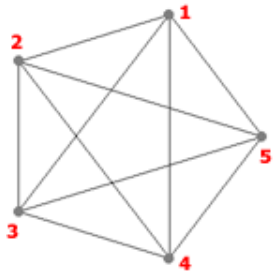
On notera qu'un graphe a au moins un sommet ; on notera par la suite ordre *d'un graphe son nombre de sommets.* ♢

REMARQUE 23.1. Dans le présent chapitre, et ses proches successeurs, graphe signifie graphe non orienté (même quand cela n'est pas spécifié). Il existe aussi des graphes orientés ; ils seront étudiés plus loin.

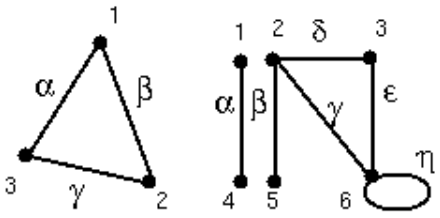
REMARQUE 23.2. La définition précédente n'interdit pas la possibilité que deux mêmes sommets soient reliés par deux arêtes différentes.

I.2 Représentation graphique et notion de graphes pondérés

Les graphes non orientés admettent une représentation graphique permettant leur visualisation :



Signalons aussi dès à présent la possibilité de pondérer les arêtes d'un graphe non orienté (la définition de graphe est alors à adapter) :



I.3 Degré, chaîne

I.3.1 Degré d'un sommet, d'un graphe

DÉFINITION 23.2 (DEGRÉ D'UN SOMMET). *On appelle* degré *d'un sommet s, noté $d(s)$, le nombre d'arêtes dont le sommet s est une extrémité (les boucles comptent pour deux).* ♢

PROPRIÉTÉ 23.1 (LEMME DES POIGNÉES DE MAINS) : La somme des degrés des sommets d'un graphe est égale à deux fois le nombre d'arêtes.

DÉFINITION 23.3 (DEGRÉ D'UN GRAPHE). *Le degré d'un graphe est le degré maximum de tous ses sommets.* ♢

Exercice 23.1. *Calculez les degrés des sommets, et le degré des graphes ci-dessus.*

DÉFINITION 23.4 (GRAPHE RÉGULIER). *Un graphe dont tous les sommets ont le même degré est dit* régulier*. Si le degré commun est k, alors on dit que le graphe est k-régulier.* ♢

Exercice 23.2. *Les graphes précédent sont-ils réguliers ?*

Exercice 23.3. *Représentez un graphe 3-régulier.*

On reviendra sur cette notion dans la section exercices de la fin du paragraphe.

I.3.2 Chaîne

DÉFINITION 23.5 (CHAÎNE). *Une chaîne dans G, est une suite de la forme*

$$(s_0, a_1, s_1, a_2, ..., s_{k-1}, a_k, s_k)$$

- *ayant pour éléments alternativement des sommets (s_i) et des arêtes (a_i),*
- *commençant et se terminant par un sommet,*

– *et telle que les extrémités de a_i soient s_{i-1} et s_i, $i = 1, ..., k$.* ♢

s_0 est appelé le *départ* de la chaîne et s_k l'*arrivée*.

REMARQUE 23.3. On a choisi ici de réserver le terme de *chemin* aux graphes orientés.

DÉFINITION 23.6 (SOMMET ACCESSIBLE). *Dans un graphe (orienté ou non), on dit que le sommet s' est* accessible *à partir du sommet s s'il existe une chaîne menant de s à s'.* ♢

REMARQUE 23.4. On dit aussi qu'on peut *atteindre* s' à partir de s.

DÉFINITION 23.7 (CHAÎNE ÉLÉMENTAIRE). *Une chaîne dans laquelle tous les sommets sont différents s'appelle une* chaîne élémentaire *.* ♢

REMARQUE 23.5. On parle aussi de chaîne *simple*.

REMARQUE 23.6. Une chaîne simple a forcément toutes ses arêtes différentes, et ne contient évidemment pas de boucle.

PROPRIÉTÉ 23.2 (EXISTENCE DE CHAÎNES ÉLÉMENTAIRES) : Étant donné une chaîne qui joint s et s' (différents), on peut toujours lui enlever arêtes et sommets pour obtenir une chaîne *élémentaire* joignant s à s'.

Exercice 23.4. *Réfléchir à la preuve de cette existence.*

I.4 circuit-cycle

DÉFINITION 23.8 (CIRCUIT). *Une chaîne de longueur n dont le départ et l'arrivée coïncident s'appelle un* circuit *de longueur n.* ♢

EXEMPLE 23.5. Une boucle est un circuit de longueur 1.

DÉFINITION 23.9 (CYCLE). *Un circuit dont tous les sommets et toutes les arêtes sont différentes, s'appelle un* cycle. ♢

Exercice 23.6. *Représentez un graphe qui admet :*

1. *un circuit,*
2. *un cycle.*

DÉFINITION 23.10 (GRAPHE SIMPLE). *Un graphe est dit* simple*, s'il ne contient pas de boucles et s'il n'y a pas plus d'une arête reliant deux mêmes sommets.* ♢

Exercice 23.7. *Représentez un graphe simple (resp. qui n'est pas simple).*

I.5 Exercices

Exercice 23.8. *On s'intéresse aux graphes 3-réguliers simples.*

1. *Essayez de construire de tels graphes ayant 4 sommets, 5 sommets, 6 sommets, et 7 sommets.*
2. *Qu'en déduisez-vous ?*

Réponse : D'après le lemme des poignées de mains, la somme des degrés des sommets est égale au double du nombre d'arêtes. Si chaque sommet est de degré 3, la somme des degrés des sommets est :
- paire, si le nombre de sommets est pair,
- impaire, sinon.

Comme cette somme doit être égale à un nombre pair (le double du nombre d'arêtes), seuls les graphes 3-réguliers ayant 4, ou 6 sommets, sont possibles.

Exercice 23.9. *Montrez qu'un graphe simple a un nombre pair de sommets de degré impair.*

Réponse : D'après le lemme des poignées de mains, la somme S des degrés des sommets est égale au double du nombre d'arêtes, donc cette somme est paire. D'autres part, S est égale à la somme :
- des degrés pairs,
- des degrés impairs

La somme des degrés pairs est paire. Étudions la somme S' des degrés impairs : notons i_0 le nombre de sommets de degrés impairs. Cette somme S' est égale à $\sum_{k=1}^{i_0}(2k_i+1)$, puisque chaque degré est ici impairs. Donc $S' = 2\left(\sum_{k=1}^{i_0} k_i\right) + i_0$, soit S' est égale à un nombre pair plus i_0. Quand on met tout bout à bout, on obtient finalement l'équation en parité : pair + pair + i_0 = pair, soit i_0 est pair !

Exercice 23.10. *Est-il possible de relier 15 ordinateurs de sorte que chaque appareil soit relié avec exactement trois autres ?*

Réponse : Non, application directe de l'exercice précédent.

Exercice 23.11. *Un groupe de 15 fans d'un chanteur célèbre, possède les deux particularités suivantes :*
- *Chaque personne connaît au moins 7 autres*
- *Toute information détenue par une personne est répercutée dans la minute qui suit à ses connaissances (et uniquement à elles)*

Quel est le temps maximal entre le moment où une des 15 fans apprend une chose nouvelle sur leur idole, et celui où le groupe entier est au courant ?

Réponse : L'éméteur de l'information est un sommet relié à au moins 7 autres. Notons I l'ensemble de ces sommets. Il reste au plus 7 sommets (15-(7+1)). Notons J cet ensemble. Chacun des sommets de J est nécessairement relié à un des sommets de I, sinon il ne serait relié qu'à 6 sommets. L'information met donc au plus 2 mins.

II Quelques types particuliers de graphes

II.1 Graphes planaires

DÉFINITION 23.11 (GRAPHE PLANAIRE). *Si on arrive à dessiner le graphe sans qu'aucune arête n'en coupe une autre (les arêtes ne sont pas forcément rectilignes), on dit que le graphe est* planaire. ♢

Exercice 23.12. *Représentez un graphe planaire.*

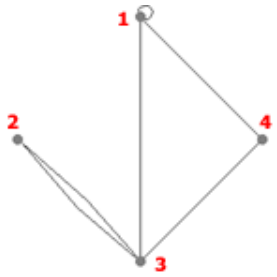
Exercice 23.13. *Représentez un graphe non planaire.*

REMARQUE 23.7. Les graphes planaires seront plus systématiquement étudiés au chapitre suivant.

II.2 Multigraphes

En général, dans ce cours, les graphes étudiés sont simples. On a cependant vu qu'il pouvait, pour un graphe quelconque, exister des boucles, voire des arêtes multiples : on parle, dans ce cas, de *multigraphe*.

EXEMPLE 23.14. Un exemple de multigraphe :



S = {1, 2, 3, 4}
A = {(1,1), (1,3), (1,4), (2,3), (2,3), (3,4)}

II.3 Graphes connexes

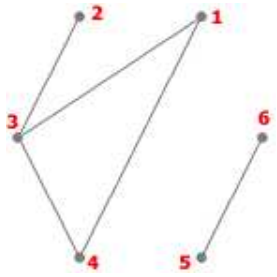
DÉFINITION 23.12 (GRAPHE CONNEXE). *Un graphe est* connexe *s'il est possible, à partir de n'importe quel sommet, d'atteindre n'importe quel autre sommet du graphe (si, pour tout couple de sommets (s, s'), il existe une chaîne reliant s à s').* ♢

REMARQUE 23.8. C'est en particulier le cas lorsqu'à partir d'un sommet on peut atteindre tous les autres sommets.

Exercice 23.15. *Représenter un graphe (non orienté) connexe, et un graphe non connexe.*

DÉFINITION 23.13 (COMPOSANTES CONNEXES). *Un graphe non connexe se décompose en* composantes connexes. ♢

EXEMPLE 23.16. Exemple d'un graphe n'étant pas connexe :



S = {1, 2, 3, 4, 5, 6}
A = {(1,3), (1,4), (2,3), (3,4), (5,6)}

Ici, les composantes connexes sont {1, 2, 3, 4} et {5, 6}.

II.4 Graphes complets

II.4.1 Définition

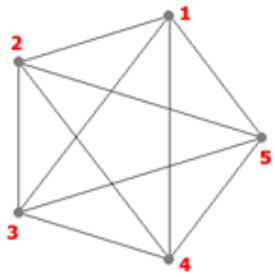
DÉFINITION 23.14 (GRAPHE COMPLET). *Un graphe est* complet *si chaque sommet du graphe est relié directement à tous les autres sommets.* ♢

II.4.2 Exemples et exercices

DÉFINITION 23.15 (K_n). *On note K_n tout graphe non orienté simple d'ordre n, tel que toute paire de sommets est reliée par une unique arête.* ♢

PROPRIÉTÉ 23.3 : $\forall n, K_n$ est complet.

EXEMPLE 23.17 (GRAPHE COMPLET K_5). Exemple d'un graphe complet :



S = {1, 2, 3, 4, 5}
A = {(1,2), (1,3), (1,4), (1,5), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5)}

Exercice 23.18. *Combien d'arêtes possède le graphe K_n ?*

Réponse : Chacun des n sommets possède $n-1$ arêtes, et chaque arête est ainsi comptée deux fois... $\frac{n(n-1)}{2}$.

Exercice 23.19. *Un tournoi d'échecs oppose 6 personnes. Chaque joueur doit affronter tous les autres.*

1. *Construisez un graphe représentant toutes les parties possibles.*
2. *Quel type de graphe obtenez-vous ?*
3. *Si l'on ne joue qu'un match par jour, combien de jours faudra-t-il pour terminer le tournoi ?*
4. *Aidez-vous du graphe pour répondre aux problèmes suivants :*
   - *Si chaque joueur ne joue qu'un match par jour, combien de jours faudra-t-il pour terminer le tournoi ?*
   - *Proposer un calendrier des matches.*

II.5 Graphes biparti

II.5.1 Définition et premières propriétés

DÉFINITION 23.16 (GRAPHES BIPARTI). *Un graphe est* biparti *si ses sommets peuvent être divisés en deux ensembles X et Y, de sorte que toutes les arêtes du graphe relient un sommet dans X à un sommet dans Y.* ♢

On peut se rendre compte que les graphes biparti sont les graphes que l'on peut colorier avec au plus deux couleurs, de sorte que deux sommets adjacents ne possèdent jamais la même couleur. En d'autres termes, les graphes bipartis sont les graphes dont le nombre chromatique est inférieur ou égal à 2 (ce terme sera défini plus proprement dans la suite du cours).

Exercice 23.20. *Responsables d'organiser des speed datings, on souhaite placer les différents individus inscrits à une soirée donnée, dans différentes salles, de telles sorte que nul ne se connaît dans une salle donnée.*

1. *Donner un exemple où cela n'est pas possible.*
2. *Comment modéliser ce problème à l'aide d'un graphe ?*
3. *Peut-on n'utiliser que deux salles, s'il est possible de placer 3 individus autour d'une table de telle sorte que chaque individu connaît ses trois voisins ? Que se passe-t-il si on remplace 3 par 5, par 7 ?*

Le résultat suivant se déduit assez aisément...

PROPRIÉTÉ 23.4 : Un graphe est biparti si et seulement il ne contient pas de cycle impair.

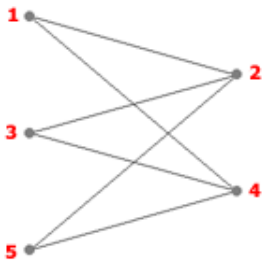
Exercice 23.21. *Relier les notions de graphe biparti et de relation binaire.*

Exercice 23.22. *Relier les graphes N-parti aux relations N-aires, aux bases de données.*

II.5.2 Graphe biparti complet

DÉFINITION 23.17. *Graphes biparti complet Un graphe biparti est dit* biparti complet *(ou encore est appelé une biclique) si chaque sommet de U est relié à chaque sommet de V.* ♢

EXEMPLE 23.23 (GRAPHE BIPARTI COMPLET). Exemple d'un graphe biparti complet :



S = {1, 2, 3, 4, 5}
A = {(1,2), (1,4), (2,3), (2,5), (3,4), (4,5)}
Avec les notations de la définition, on a U={1, 3, 5} et V={2, 4}, ou vice versa.

Un tel graphe se note $K_{3,2}$. Plus généralement,

NOTATION : On note $K_{m,n}$ un graphe biparti complet liant m sommets à n sommets.

PROPRIÉTÉ 23.5 : Ces graphes $K_{m,n}$ possèdent mn arêtes.

II.6 Exercices

Exercice 23.24. *Sur un échiquier 3x3, les deux cavaliers noirs sont placés sur les cases a1 et c1, les deux cavaliers blancs occupant les cases a3 et c3. Aidez-vous d'un graphe pour déterminer les mouvements qui permettront aux cavaliers blancs de prendre les places des cavaliers noirs, et vice versa.*

Réponse : Faire un graphe biparti :
- a_1, a_3, c_1, c_3 d'un côté,
- a_2, b_1, b_2, b_3, c_2 de l'autre.

avec des arêtes quand le passage d'une case à l'autre est possible pour un cavalier (par exemple, entre a_1 et b_3). On oriente alors les arêtes, suivant les parcours à réaliser par les cavaliers. De a_1, on peut envoyer le cavalier en b_3 ou c_2. Mais si on l'envoie en c_2, il se retrouve le coup d'après en a_3.

Exercice 23.25. *Quel est le nombre maximal d'arêtes dans un graphe non orienté d'ordre n qui ne possède pas d'arêtes parallèles ? Et si l'on suppose qu'il ne possède pas de boucle ?*

Réponse : Le cas le pire correspond au graphe complet K_n, et on a déjà calculé son nombre d'arêtes : $(n-1) + ... + 1 = \frac{n(n-1)}{2}$. Rajouter des boucles revient, dans le cas le pire, à rajouter une arête sur chaqu'un des n sommets. On ajoute donc n à ce qui précède, pour trouver : $\frac{n(n+1)}{2}$.

III Représentation des graphes

Nous aimons bien communiquer par des dessins, mais les machines n'en sont pas encore là : il faut donc trouver d'autres façons de représenter les graphes.

La solution consiste à utiliser des matrices, et il y a différentes méthodes. Dans ce qui suit, on considère des graphes non pondérés, mais ces représentations s'adaptent sans problème aux graphes pondérés.

III.1 Matrice d'incidence

III.1.1 Présentation

La *matrice d'incidence* d'un graphe non orienté est une matrice J à coefficients entiers dont les lignes sont repérées par les sommets d'un graphe et les colonnes par ses arêtes :

DÉFINITION 23.18 (MATRICE D'INCIDENCE). *Par définition, $J_{s,\varepsilon}$ vaut :*
- *1 quand s est une extrémité de l'arête ε si celle-ci n'est pas une boucle,*
- *2 quand s est une extrémité de la boucle ε,*
- *0 si s n'est pas une extrémité de ε.* ♢

On peut reconstituer un graphe non orienté à partir de sa matrice d'incidence, car elle donne le nombre de sommets, le nombre d'arêtes et elle dit comment chaque arête est liée à chaque sommet.

Exercice 23.26. *Représentez le graphe non orienté dont la matrice d'incidence est :*

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Exercice 23.27. *Représentez les matrices d'incidences des graphes de ce chapitre.*

Exercice 23.28. *Réfléchir aux avantages et inconvénients d'un tel mode de représentation des graphes.*

III.1.2 Résultat

PROPRIÉTÉ 23.6 : Si $s_1, \ldots, s_n$ sont les sommets d'un graphe non orienté, alors :

$$\begin{pmatrix} d(s_1) \\ d(s_2) \\ \vdots \\ d(s_n) \end{pmatrix} = J \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Exercice 23.29. *Trouvez pourquoi.*

III.2 Matrice d'adjacence

III.2.1 Présentation

On peut représenter un graphe non orienté par une matrice d'adjacence.

DÉFINITION 23.19 (MATRICE D'ADJACENCE). *Dans une* matrice d'adjacences*, les lignes et les colonnes représentent les sommets du graphe.*

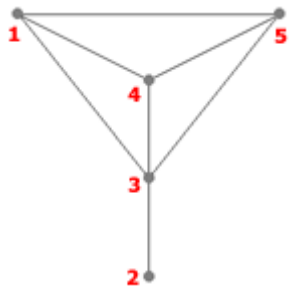
- *Un 1 à la position (i,j) signifie que le sommet i est adjacent au sommet j.*
- *Sinon, on place un 0.*

En cas de boucle (pour le sommet i, par exemple), on place un 1 sur la diagonale (en position (i, i)). ♢

REMARQUE 23.9. On aurait pu convenir de placer un 2 en cas de boucle. L'avantage serait de continuer à obtenir le degré des sommets en faisant les sommes par lignes. Par contre, on perdrait la possibilités, évoquée ci-dessous, de déterminer les chemins de longueur k.

III.2.2 Exemple

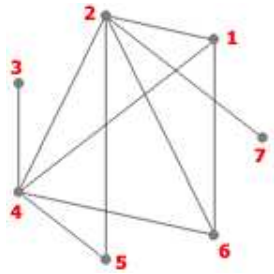
Considérons le graphe G :



Voici la matrice d'adjacences du graphe G :

$$M = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Exercice 23.30. *Décrivez le graphe G ci-dessous par une matrice d'adjacences.*



Exercice 23.31. *Représentez les matrices d'adjacences des graphes de ce chapitre.*

Exercice 23.32. *Réfléchir à la possibilité d'extension des matrices d'adjacences aux graphes :*
- *orientés,*
- *pondérés.*

III.2.3 Propriétés de la matrice d'adjacence

Cette matrice a plusieurs caractéristiques :

PROPRIÉTÉ 23.7 :
1. Elle est carrée : il y a autant de lignes que de colonnes.
2. Un 1 sur la diagonale indique une boucle. Si le graphe n'a pas de boucle, alors la diagonale de sa matrice d'adjacence est nulle.
3. La matrice d'adjacence d'un graphe non orienté est symétrique : $m_{ij} = m_{ji}$.

Exercice 23.33. *Calculez M^2 et M^3 pour la matrice d'adjacence M ci-dessus. Comparer ces matrices aux chaînes de longueur 2 et 3 reliant deux sommets quelconques.*

PROPRIÉTÉ 23.8 : Soit A la matrice d'adjacence d'un graphe G. Le coefficient (s, t) de A^k est le nombre de chaînes de longueur k qui mènent du sommet s au sommet t.

Exercice 23.34. *Démontrez ce résultat, par récurrence.*

Exercice 23.35. *On pose :*

$$J = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

1. *Dessinez le graphe non orienté ayant J pour matrice d'incidence.*
2. *Déterminez sa matrice d'adjacence B.*
3. *Vérifiez les formules précédentes :*
   - *le lien entre matrice d'incidence et degré des sommets,*
   - *le lien entre B^k et les chaînes de longueur k.*

Exercice 23.36. *Quels sont les avantages et les inconvénients de cette méthode de représentation des graphes ? Comparez-la aux matrices d'incidences.*

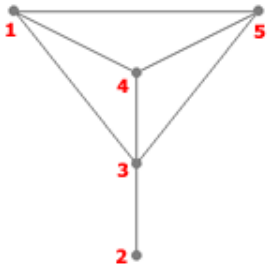
En particulier, pour un graphe à m sommets et n arêtes, quelle représentation est la plus gourmande en espace mémoire ? Cela dépend du nombres d'arêtes ?

III.3 Listes d'adjacence

III.3.1 Présentation

DÉFINITION 23.20 (LISTE D'ADJACENCE). *On peut encore représenter un graphe en donnant pour chacun de ses sommets la liste des sommets auxquels il est adjacent. On parle alors de* liste d'adjacence. ♢

EXEMPLE 23.37. On considère le graphe G suivant :



Voici les listes d'adjacences de G :

1 : 3, 4, 5
2 : 3
3 : 1, 2, 4, 5
4 : 1, 3, 5
5 : 1, 3, 4

Exercice 23.38. *Représentez les listes d'adjacences des graphes de ce chapitre.*

Exercice 23.39. *Discuter des avantages et des inconvénients de cette méthode de représentation. On la comparera aux matrices d'adjacence et d'incidence.*

Fin du Chapitre

Chapitre 24

Problèmes de graphes

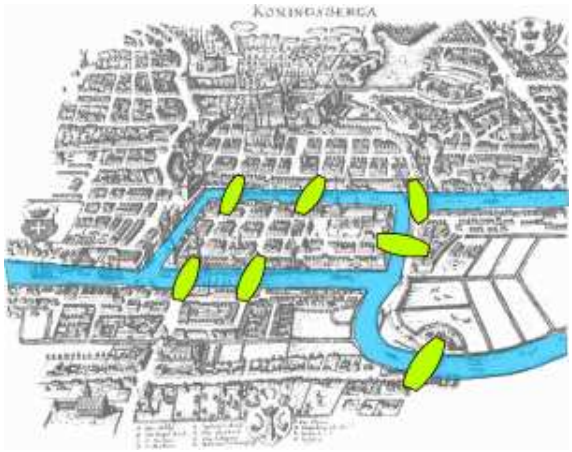
Dans ce chapitre, différents problèmes classiques autour des graphes seront présentés. On verra successivement les problèmes suivants, et on en proposera des solutions :

- Existence d'un circuit eulérien.
- Caractérisation des graphes extraits d'un graphe donné.
- Caractérisation des graphes planaires.
- Dénombrement des régions induites par un graphe planaire.
- Existence d'un circuit hamiltonien.

I Circuits eulériens

I.1 Introduction : les ponts de Königsberg

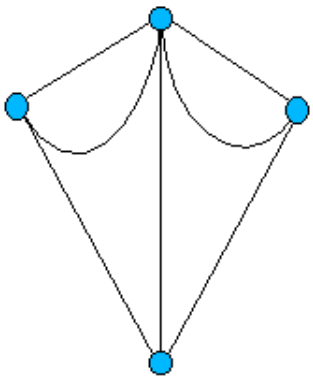
La question à l'origine de la théorie des graphes est due à Euler [1], en 1736 : dans cette partie de la ville de Königsberg :



peut-on, lors d'une promenade, revenir à notre point de départ en empruntant une, et une seule fois, chaque pont ?

Pour y répondre, Euler a introduit le graphe suivant (les arcs symbolisent les ponts ; les sommets, les quatre zones terrestres) :

1. Leonhard Euler, mathématicien suisse (1707-1783). A consacré près de 900 mémoires aux mathématiques, à l'optique, à la science navale, à la musique, l'atronomie, la théorie des assurance... Un monstre, appelé l'aigle des mathématiques. Le plus grand mathématicien du dix-huitième siècle, l'un des plus grands de tous les temps.



Le problème de départ se ramène alors à la question suivante : peut-on trouver un circuit permettant d'emprunter une, et une seule fois chaque arête, en retournant à son point de départ ?

La réponse, dans ce cas particulier, est non : comme
– le point de départ s est aussi le point d'arrivée,
– on ne peut pas emprunter deux fois une même arête,
on en conclut forcément que le degré de s est pair. Or, dans le graphe ci-dessus, tous les sommets ont un degré impair.

I.2 Définitions

Ce problème, introduit par Euler, conduit aux définitions suivantes...

DÉFINITION 24.1 (CHAÎNE EULÉRIEN). *On appelle* chaîne eulérienne *une chaîne contenant une et une seule fois toutes les arêtes du graphe.* ♢

DÉFINITION 24.2 (CIRCUIT EULÉRIEN). *On appelle* circuit eulérien *un circuit contenant une et une seule fois toutes les arêtes du graphe.* ♢

REMARQUE 24.1. On n'a plus affaire à une chaîne, mais à un circuit : le point de départ et celui d'arrivée coïncident.

DÉFINITION 24.3 (GRAPHE EULÉRIEN). *Un* graphe eulérien *est un graphe possédant un circuit eulérien.* ♢

REMARQUE 24.2. On peut passer plusieurs fois par un sommet donné, pourvu que les arêtes empruntées soient toutes différentes.

Exercice 24.1. *Donnez des exemples de graphes possèdant des circuits eulériens, et d'autres exemples de graphes n'en possédant pas.*

I.3 Résultat d'Euler

Du problème initial d'Euler, on peut en déduire le résultat suivant (rappelons qu'il s'agit ici de graphes non orientés)...

PROPRIÉTÉ 24.1 : Il y a équivalence entre :
- posséder une chaîne eulérienne,
- être connexe, avec O ou 2 sommets de degré impair.

De plus, s'il n'y a pas de sommet de degré impair, alors le graphe est Eulérien.

PREUVE 4 : *En parcourant un chemin ou un circuit, pour chaque sommet visité, on utilise une arête pour arriver à ce sommet et une arête pour en repartir, ces deux arêtes ne devant plus être utilisées par la suite.*

Le nombre d'arêtes utilisables en ce sommet diminue donc de deux. Si un sommet est d'ordre impair, une de ses arêtes aboutissant à ce sommet doit donc être soit sur la première arête d'un chemin, soit sur la dernière.

Un chemin n'ayant que deux extrémités, le nombre de sommets d'ordre impair ne peut excéder deux. †

Exercice 24.2. *Soit G un graphe connexe non eulérien. Est-il toujours possible de rendre G eulérien en lui rajoutant un sommet et quelques arêtes ?*

Réponse : Oui. Quand on ajoute un nouveau sommet, son nombre d'arêtes doit être pair, pour ne pas poser de nouveaux problèmes. On peut donc ainsi corriger le problème d'imparité pour un nombre pair de sommets : en reliant chaque couple de deux sommets de degré impairs, à un sommet que l'on crée uniquement pour ce couple. Or, tout graphe simple possède un nombre pair de sommets de degré impair...

I.4 Exercice : les dominos

Exercice 24.3. *On considère des dominos dont les faces sont numérotées 1, 2, 3, 4 ou 5.*

1. *En excluant les dominos doubles, de combien de dominos dispose-t-on ?*
2. *Montrez que l'on peut arranger ces dominos de façon à former une boucle fermée (en utilisant la règle habituelle de contact entre les dominos).*
3. *Pourquoi n'est-il pas nécessaire de considérer les dominos doubles ?*
4. *Si l'on prend maintenant des dominos dont les faces sont numérotées de 1 à n, est-il possible de les arranger de façon à former une boucle fermée ?*

Réponses :

1. On dispose de 4+3+2+1 = 10 dominos.
2. Considérons K_5, le pentagramme complet. A l'arête $(1, 2)$, par exemple, est associée le domino $(1, 2)$. Arriver à constituer une boucle fermée de dominos revient donc à trouver un circuit eulérien dans K_5. C'est possible : K_5 est connexe, avec tous ses sommets de degré pairs.
3. Considérer les dominos doubles revient à placer une boucle à chaque sommet de K_5. On ne change pas sa connexité, ni la parité de ses sommets.
4. On obtient, dans ce cas, K_n, dont chaque sommet a pour degré $n - 1$. Pour arriver à consitituer une boucle fermée, il faut donc (et il suffit) que n soit impair.

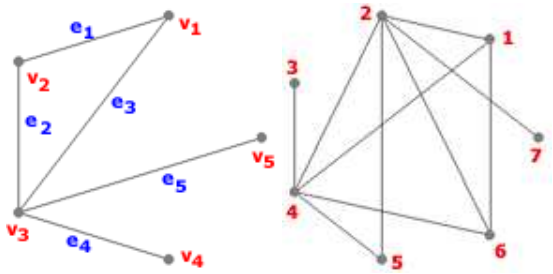
II Graphes partiels et sous-graphes

II.1 Introduction

Dans cette section, on va chercher à étudier quels sont les différents types de graphes que l'on peut obtenir à partir d'un graphe G, en lui enlevant soit des arêtes, soit des sommets.

Prenons l'exemple d'un ordinateur, avec imprimante, souris, etc. Chaque appareil est un sommet du graphe de l'installation informatique, et chaque câble est une arête. On peut alors, au choix, enlever des câbles électriques, ou des appareils (avec leurs câbles), pour obtenir ainsi une nouvelle installation informatique, c'est-à-dire construire un nouveau graphe à partir d'un graphe donné.

On considère, dans tout ce paragraphe, les graphes G_1 et G_2 suivants



Ils nous serviront à illustrer les définitions à venir.

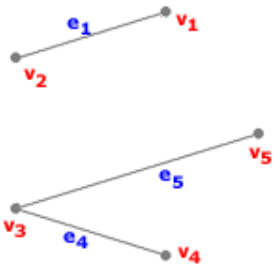
II.2 Graphe partiel et sous-graphe

II.2.1 Graphe partiel

DÉFINITION 24.4 (GRAPHE PARTIEL). *Soit G = (S, A) un graphe. Le graphe G' = (S, A') est un* graphe partiel *de G, si A' est inclus dans A.* ♢

REMARQUE 24.3. Autrement dit, on obtient G' en enlevant une ou plusieurs arêtes au graphe G (sans toucher à ses sommets).

EXEMPLE 24.4 (GRAPHE PARTIEL DE G_1). Voici un exemple de graphe partiel de G_1



Ici,

- S'=S,
- A'=$\{e_3, e_4, e_5\}$.

Exercice 24.5. *Trouver un graphe partiel de G_2.*

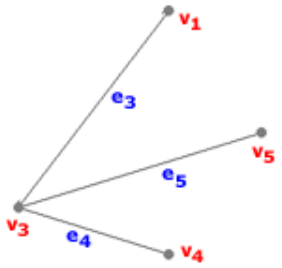
II.2.2 Sous-graphe

DÉFINITION 24.5 (SOUS-GRAPHE). *On dit que le graphe* (S', A') *est un* sous-graphe *du graphe* (S, A) *si*

1. $S' \subset S$,
2. $A' \subset A$,
3. $A' = \{(x, y) \mid (x, y) \in A \wedge x \in S' \wedge y \in S'\}$ ♢

REMARQUE 24.4. Un sous-graphe d'un graphe donné est donc obtenu en enlevant certains sommets, et toutes les arêtes incidentes à ces sommets.

EXEMPLE 24.6 (SOUS-GRAPHE DE G_1). Voici un exemple de sous-graphe de G_1 :



Ici,
- V'=$\{v_1, v_3, v_4, v_5\}$,
- E'=$\{e_3, e_4, e_5\}$.

Exercice 24.7. *Trouver un sous-graphe de* G_2.

Exercice 24.8. *Combien un graphe* G *d'ordre* n *possède-t-il de sous-graphes ?*

Réponse : $2^n - 1$, si l'on ne compte pas G.

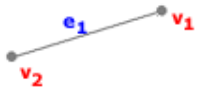
II.3 Sous-graphes particuliers

II.3.1 Sous-graphe partiel

DÉFINITION 24.6 (SOUS-GRAPHE PARTIEL). *Un graphe partiel d'un sous-graphe est un* sous-graphe partiel *de G.* ♢

REMARQUE 24.5. Cette fois-ci, on enlève des sommets (et leurs arêtes incidentes), puis des arêtes.

EXEMPLE 24.9 (SOUS-GRAPHE PARTIEL DE G_1). Voici un sous-graphe partiel de G_1 :



Ici,
- V'=$\{v_1, v_2, v_3, v_4\}$,
- E'=$\{e_1, e_4\}$.

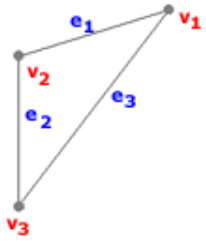
Exercice 24.10. *Trouver un sous-graphe partiel de* G_2.

II.3.2 Clique

DÉFINITION 24.7 (CLIQUE). *On appelle* clique *un sous-graphe complet de G.* ♢

REMARQUE 24.6. On a donc réussi, en enlevant des sommets, à faire en sorte que chaque sommet est adjacent à tous les autres sommets.

EXEMPLE 24.11 (UNE CLIQUE DE G_1). Exemple de clique de G_1 :



Ici,
- V'=$\{v_1, v_2, v_3\}$,
- E'=$\{e_1, e_2, e_3\}$.

Exercice 24.12. *Trouver une clique de G_2.*

II.3.3 Stable

DÉFINITION 24.8 (STABLE). *On appelle* stable *un sous-graphe de G sans arêtes.* ♢

REMARQUE 24.7. On enlève donc des sommets, et leurs arêtes adjacentes. On obtient un stable que si, en ayant enlevé un certain nombre de sommets à G, le sous-graphe résultant ne possède plus d'arête.

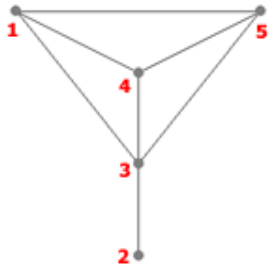
EXEMPLE 24.13 (UN STABLE DE G_1). Voici un stable de G_1 :

Ici,
- V'=$\{v_1, v_4, v_5\}$,
- E'=$\{\}$.

Exercice 24.14. *Trouver un stable de G_2.*

II.4 Exercices

Exercice 24.15. *On considère le graphe G suivant :*



En extraire : un graphe partiel, un sous-graphe, un sous-graphe partiel, une clique et un stable.

Exercice 24.16. *Montrez que dans un groupe de six personnes, il y en a nécessairement trois qui se connaissent mutuellement ou trois qui ne se connaissent pas (on suppose que si A connaît B, B connaît également A).*

Montrez que cela n'est plus nécessairement vrai dans un groupe de cinq personnes.

Réponse : Cela revient à dire que, dans un graphe G à six sommets, il est toujours possible d'obtenir un sous-graphe (*i.e.*, en enlevant trois sommets), qui est :

- soit une clique (les trois sommets sont adjacents deux à deux),
- soit un stable (aucun arc).

Cela n'est plus valable pour les graphes à cinq sommets.

III Graphe planaire

III.1 Définition

On rappelle la définition suivante...

DÉFINITION 24.9 (GRAPHE PLANAIRE). *Un graphe est dit* planaire *s'il admet une représentation graphique dans le plan telle que deux arêtes quelconques ne se coupent pas.* ♢

REMARQUE 24.8. Rappelons que les arêtes ne sont pas forcément rectilignes.

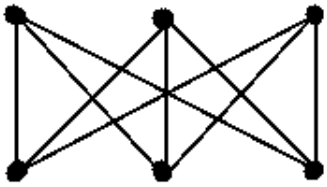
Outre l'intérêt théorique (théorème des quatre couleurs, etc.), ou une représentation plus sympathique, chercher si un graphe possède une représentation planaire peut s'avérer pratiquement important. Par exemple, il était plus facile de concevoir les tout premiers circuits imprimés à transistors quand le graphe du circuit était planaire : on évitait alors de devoir recourir au procédé bicouche ou à des "straps" fragiles pour s'échapper du plan du circuit imprimé.

Exercice 24.17. *Un graphe peut-il être planaire s'il possède un sous-graphe qui ne l'est pas ?*

Réponse : Non : une fois dessiné dans un plan, un graphe planaire a forcément tous ses sous-graphes qui le sont aussi.

III.2 Exemples

EXEMPLE 24.18 ($K_{3,3}$). Il est non planaire



EXEMPLE 24.19 (K_n). On rappelle que l'on note K_n tout graphe non orienté simple d'ordre n, tel que toute paire de sommets est reliée par une unique arête. Alors K_5 n'est pas planaire :

Exercice 24.20. *Dessiner K_1, K_2, K_3 et K_4. Sont-ils planaires ? Et qu'en est-il de $K_n, n \geqslant 5$?*

Réponse : K_1, K_2, K_3 et K_4 sont planaires. K_5, on l'a vu, ne l'est pas. Comme K_5 est un sous-graphe de $K_n, n \geqslant 5$, on en déduit qu'à partir de $n = 5$, tous les K_n ne sont plus planaires.

III.3 Caractérisation des graphes planaires

Pendant de nombreuses années, les mathématiciens ont tenté de caractériser les graphes planaires. Ce problème a été résolu en 1930 par le mathématicien polonais K. Kuratowski.

III.3.1 Expansion d'un graphe

DÉFINITION 24.10 (EXPANSION D'UN GRAPHE). *L'expansion d'un graphe est le résultat de l'ajout d'un ou plusieurs sommets sur une ou plusieurs arêtes (par exemple, transformation de l'arête $\bullet - \bullet$ en $\bullet - \bullet - \bullet$).* ♢

Exercice 24.21. *Représentez deux graphes, le second étant une expansion du premier.*

III.3.2 Théorème de Kuratowski

La réponse au problème de caractérisation des graphes planaires est...

PROPRIÉTÉ 24.2 (KURATOWSKI) : Un graphe fini est planaire si et seulement s'il ne contient pas de sous-graphe qui est une expansion de K_5 ou $K_{3,3}$.

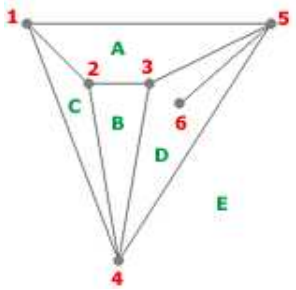
IV Dénombrement des régions d'un graphe planaire

Le prochain problème classique que l'on présentera s'intéresse au nombre de régions qu'un graphe planaire découpe. La célèbre formule d'Euler permettra de relier ce nombre aux nombres d'arêtes et de sommets du graphe considéré.

IV.1 Cartes, régions

DÉFINITION 24.11 (CARTE, RÉGIONS). *Une* carte *est une représentation particulière d'un graphe planaire. On dit qu'une carte est* connexe *si son graphe l'est. Une carte divise le plan en plusieurs* régions. ♢

EXEMPLE 24.22. Par exemple, la carte ci-dessous, avec six sommets et neuf arêtes, divise le plan en cinq régions (A, B, C, D, E).



On remarque que quatre régions sont limitées alors que la cinquième (E), extérieure au diagramme, ne l'est pas.

IV.2 Degré d'une région

DÉFINITION 24.12 (DEGRÉ D'UNE RÉGION). *Le* degré *d'une région r, noté d(r), est la longueur du cycle ou de la chaîne fermée qui limite r.* ♢

EXEMPLE 24.23. Dans le graphe ci-dessus, d(A)=4, d(B)=3, d(C)=3, d(D)=5, d(E)=3.

REMARQUE 24.9. On remarque que toute arête limite deux régions, ou est contenue dans une région et est alors comptée deux fois dans la chaîne fermée. Nous avons donc...

IV.3 Lemme des régions

PROPRIÉTÉ 24.3 (LEMME DES RÉGIONS) : La somme des degrés des régions d'une carte connexe est égale à deux fois le nombre d'arêtes.

Exercice 24.24. *Démontrez ce résulat.*

Réponse : Il y a deux types d'arêtes, celles séparant deux régions données, et celles incluses à l'intérieur d'une région...

IV.4 Formule d'Euler

Euler a établi une formule qui relie le nombre de sommets S, le nombre d'arêtes A et le nombre de régions R d'une carte connexe :

PROPRIÉTÉ 24.4 (EULER) :

$$S - A + R = 2$$

IV.5 Exercices

Exercice 24.25. *Utilisez le résultat d'Euler pour retrouver le fait que $K_{3,3}$ n'est pas planaire.*

DÉFINITION 24.13 (POLYÈDRES RÉGULIERS, SOLIDES DE PLATON). *Un polyèdre est dit régulier s'il est constitué de faces toutes identiques et régulières, et que tous ses sommets sont identiques. Ils sont au nombre de neuf, dont cinq seulement sont convexes. Ces derniers étaient connus de Platon :*

- *le tétraèdre régulier (4 faces en triangle équilatéral),*
- *le cube,*
- *l'octaèdre régulier (8 faces en triangle équilatéral),*
- *le dodécaèdre régulier (12 faces en pentagone),*
- *l'icosaèdre régulier (20 faces en triangle équilatéral).*

Pour cette raison, ces cinq polyèdres réguliers convexes sont appelés Solides de Platon. ♢

REMARQUE 24.10. On appelle parfois polyèdres réguliers uniquement les 5 solides de Platon.

Exercice 24.26. *Les solides platoniciens peuvent être considérés comme des graphes, qui de plus sont planaires. Vérifier la formule d'Euler sur ces solides Platoniciens :*

- *le nombre de sommets,*
- *moins le nombre d'arêtes,*
- *plus le nombre de faces*

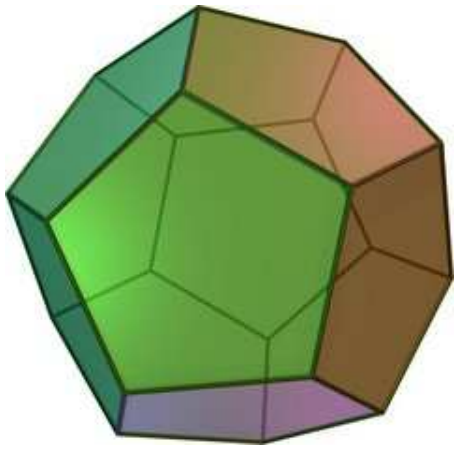
...est toujours égal à 2.

V Circuit hamiltonien

V.1 Les dodécaèdres de Hamilton

Le dodécaèdre est à l'origine d'un autre problème célèbre, dû à Hamilton [2] : comment passer une, et une seule fois, par chacun des sommets du dodécaèdre, de telle manière que le dernier sommet visité est aussi le premier.

2. William Rowan Hamilton, mathématicien et physicien irlandais (1805-1865). Inventeur des quaternions.



V.2 Définition

DÉFINITION 24.14 (CIRCUIT ET GRAPHES HAMILTONIENS). *Un circuit hamiltonien est un circuit qui passe par tous les sommets du graphe, une et une seule fois. Un graphe possédant un tel circuit est qualifié d'hamiltonien.* ♢

REMARQUE 24.11. C'est un circuit : le sommet de départ doit aussi être le sommet d'arrivée.

EXEMPLE 24.27. Les graphes complets K_n sont hamiltoniens, pour tout n.

PREUVE 5 : *On peut inscrire K_n dans un polygone régulier à n sommets : il suffit alors de parcourir ce polygone, sommet par sommet, dans le sens trigonométrique par exemple, jusqu'à retrouver son point de départ.* †

REMARQUE 24.12. Il peut y avoir plusieurs circuits hamiltoniens dans un graphe hamiltonien.

Le problème de la caractérisation des graphes Hamiltoniens n'est pas aussi simple que celui des graphes Eulériens. On peut cependant énoncer quelques conditions, respectivement nécessaires, puis suffisantes, pour qu'un graphe le soit...

V.3 Conditions nécessaires

PROPRIÉTÉ 24.5 : Un graphe hamiltonien d'ordre $n \geqslant 3$ ne possède pas de sommet de degré 1.

PREUVE 6 : *Si un sommet s est de degré 1, il est connecté à un (unique) autre sommet s', qui sera forcément parcouru deux fois (pour aller en s, et pour en sortir).* †

REMARQUE 24.13. K_2 est hamiltonien, et a tous ses sommets d'ordre 1...

PROPRIÉTÉ 24.6 : Si un graphe hamiltonien possède un sommet de degré 2, alors les deux arêtes incidentes à ce sommet doivent faire partie du cycle hamiltonien.

PREUVE 7 : *Supposons que s est de degré 2, connecté à s_1 par l'arête a_1, et à s_2 par a_2. On est arrivé à s à partir d'une arête, mettons a_1 ; si on réempruntait a_1 pour sortir de s, alors on repasserait par a_1, ce qui est impossible.* †

V.4 Conditions suffisantes

PROPRIÉTÉ 24.7 (THÉORÈME DE ORE) : Soit G = (V, E) un graphe simple d'ordre $n \geqslant 3$. Si pour toute paire {x,y} de sommets non adjacents, on a $d(x) + d(y) \geqslant n$, alors G est hamiltonien.

PROPRIÉTÉ 24.8 (COROLLAIRE DE DIRAC) : Soit G = (V, E) un graphe simple d'ordre $n \geqslant 3$. Si pour tout sommet x de G, on a $d(x) \geqslant n/2$, alors G est hamiltonien.

PREUVE 8 : *En effet, un tel graphe vérifie les conditions du théorème précédent. Si x et y ne sont pas adjacents, on a bien :*

$$d(x) + d(y) \geqslant n/2 + n/2 = n$$

Exercice 24.28. *Dessinez un graphe d'ordre au moins 5 qui est...*

1. *hamiltonien et eulérien,*
2. *hamiltonien et non eulérien,*
3. *non hamiltonien et eulérien,*
4. *non hamiltonien et non eulérien.*

V.5 Le problème du voyageur de commerce

Trouver un cycle hamiltonien pour un graphe donné est un problème difficile sur le plan algorithmique – de type NP-complet –, relié au problème dit *du voyageur de commerce* : un voyageur de commerce doit visiter un ensemble de villes, en minimisant son temps de parcours.

Ce problème se formalise aisément en théorie des graphes :
- Les villes sont les sommets d'un graphe pondéré.
- Les arêtes correspondent aux routes reliant ces villes.
- La pondération correspond soit à la distance entre deux ville (en kilomètres), soit à la durée nécessaire pour passer d'une ville à l'autre.
- Ce graphe peut être orienté, si l'on souhaite intégrer des routes à sens unique.

On cherche alors à visiter tous les sommets du graphe (*i.e.* toutes les villes) en minimisant le « poids » du parcours (la distance parcourue, ou le temps total).

Le problème du voyageur de commerce correspond donc à la recherche d'un cycle hamiltonien de poids minimal, dans un graphe pondéré complet. Ce problème est NP-complet, donc impossible à résoudre exactement quand le nombre de villes est grand.

Fin du Chapitre

Chapitre 25

Arbres et arborescence

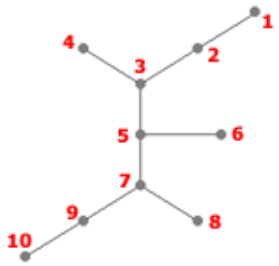
I Présentation générale

I.1 Définitions

DÉFINITION 25.1 (ARBRE, FEUILLES). *On nomme* arbre *un graphe non orienté connexe et acyclique (sa forme évoque en effet la ramification des branches d'un arbre). On distingue deux types de sommets dans un arbre*

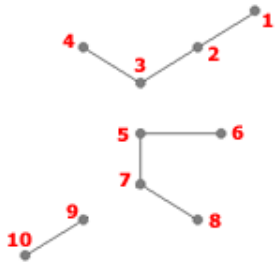
- *les* feuilles *dont le degré est 1 ;*
- *les* nœuds internes *dont le degré est supérieur à 1.* ♢

EXEMPLE 25.1. Un exemple d'arbre :



DÉFINITION 25.2 (FORÊT). *Un graphe sans cycles mais non connexe est appelé une* forêt. ♢

EXEMPLE 25.2. Un exemple de forêt :



I.2 Caractérisation des arbres

PROPRIÉTÉ 25.1 : Les affirmations suivantes sont équivalentes pour tout graphe $G = (V, E)$ à n sommets.

1. G est un arbre,
2. G est sans cycles et connexe,
3. G est sans cycles et comporte $n - 1$ arêtes,
4. G est connexe et comporte $n - 1$ arêtes,
5. Chaque paire (u, v) de sommets distincts est reliée par une seule chaîne simple (et le graphe est sans boucles).

I.3 Nombre minimal de feuilles

PROPRIÉTÉ 25.2 : Tout arbre fini avec au moins deux sommets comporte au moins deux feuilles.

I.4 Exercices

Exercice 25.3. *Démontrez la propriété précédente.*

Indication : Récurrence.

Exercice 25.4. *Combien d'arbres différents existe-t-il avec 3, 4, 5 sommets ?*

Indication : Réfléchir sur le nombre de feuilles.

Exercice 25.5. *Un arbre T a trois sommets de degré 3, quatre sommets de degré 2. Les autres sommets sont tous de degré 1 (des feuilles). Combien y a-t-il de sommets de degré 1 ?*

Réponse : Soit n le nombre total de sommet ; il y a donc $n - 1$ arêtes, et $n - 3 - 4 = n - 7$ sommets de degré 1. Comptons le nombre d'arêtes... Chaque sommet partage son arête avec un autre sommet, donc :

- chaque sommet de degré trois (il y en a 3) a 1,5 arête « à lui »,
- chaque sommet de degré deux (il y en a 4) a 1 arête « à lui »,
- chaque sommet de degré un (il y en a $n - 7$) a 0,5 arête « à lui »

Ce qui fait un total d'arêtes de $3 * 1,5 + 4 * 1 + (n - 7) * 0,5$. Or, comme on a affaire à un arbre à n sommets, on a $n - 1$ arêtes, donc

$$3 * 1,5 + 4 * 1 + (n - 7) * 0,5 = n - 1$$

d'où $n = 12$, et il y a 5 sommets de degré 1.

Exercice 25.6. *Soit un graphe G. Supposons qu'il y ait deux chaînes élémentaires distinctes P_1 et P_2 d'un sommet s à un autre sommet s' de G. G est-il un arbre ? Justifier par une preuve.*

II Codage de Prüfer

II.1 Présentation

Le codage de Prüfer est une manière très compacte de décrire un arbre.

II.2 Codage

II.2.1 La méthode

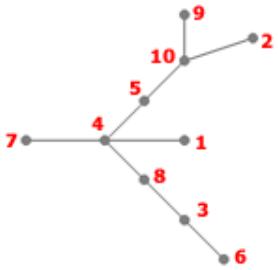
Soit l'arbre $T = (V, E)$ et supposons $V = 1, 2, ..., n$. L'algorithme ci-dessous fournira le code de T, c'est-à-dire une suite S de $n - 2$ termes employant (éventuellement plusieurs fois) des nombres choisis parmi $\{1, ..., n\}$.

PROPRIÉTÉ 25.3 (PAS GÉNÉRAL DE L'ALGORITHME DE CODAGE) : Initialement la suite S est vide. Ce pas général est à répéter tant qu'il reste plus de deux sommets dans l'arbre courant T :

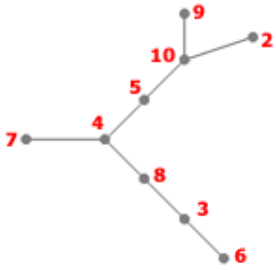
- identifier la feuille v de l'arbre courant ayant le numéro minimum ;
- ajouter à la suite S le seul sommet s adjacent à v dans l'arbre T courant ;
- enlever de l'arbre T courant la feuille v et l'arête incidente à v.

II.2.2 Exemple de codage

Étape 0 : arbre à coder



1 : 4
2 : 10
3 : 6, 8
4 : 1, 5, 7, 8
5 : 4, 10
6 : 3
7 : 4
8 : 3, 4
9 : 10
10 : 2, 5, 9

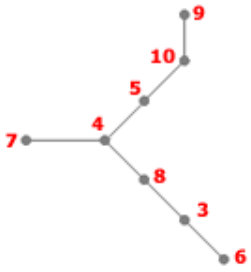


Étape 1 :

2 : 10
3 : 6, 8
4 : 5, 7, 8
5 : 4, 10
6 : 3

7 : 4
8 : 3, 4
9 : 10
10 : 2, 5, 9

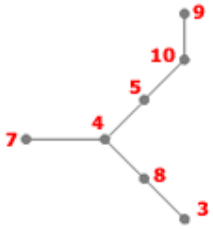
S={4}



Étape 2 :

3 : 6, 8
4 : 5, 7, 8
5 : 4, 10
6 : 3
7 : 4
8 : 3, 4
9 : 10
10 : 5, 9

S={4,10}



Étape 3 :

3 : 8
4 : 5, 7, 8
5 : 4, 10
7 : 4
8 : 3, 4
9 : 10
10 : 5, 9

S={4,10,3}

Étape 4 :

4 : 5, 7, 8

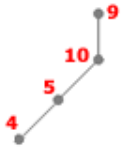
5 : 4, 10
7 : 4
8 : 4
9 : 10
10 : 5, 9

S = {4,10,3,8}

Étape 5 :

4 : 5, 8
5 : 4, 10
8 : 4
9 : 10
10 : 5, 9

S = {4,10,3,8,4}



Étape 6 :

4 : 5
5 : 4, 10
9 : 10
10 : 5, 9

S = {4,10,3,8,4,4}

Étape 7 :

5 : 10
9 : 10
10 : 5, 9

S = {4,10,3,8,4,4,5}

Étape 8 :

9 : 10
10 : 9

S = {4,10,3,8,4,4,5,10}

Étape 9 : S = {4,10,3,8,4,4,5,10} est le codage de Prüfer de l'arbre initial.

II.3 Décodage

II.3.1 La méthode

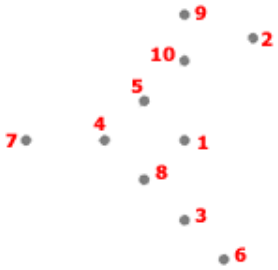
Donnée : suite S de $n-2$ nombres, chacun provenant de {1, ..., n}.
Posons $I = \{1, ..., n\}$.

Pas général de l'algorithme de décodage : à répéter tant qu'il reste des éléments dans S et plus de deux éléments dans I :
- identifier le plus petit élément i de I n'apparaissant pas dans la suite S ;
- relier par une arête de T le sommet i avec le sommet s correspondant au premier élément de la suite S ;
- enlever i de I et s de S.

Finalisation : Les deux éléments qui restent dans I à la fin de l'algorithme constituent les extrémités de la dernière arête à ajouter à T.

II.3.2 Exemple de décodage

Étape 0 : arbre à décoder

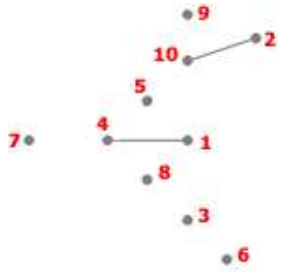


I={1,2,3,4,5,6,7,8,9,10}
S={4,10,3,8,4,4,5,10}

Étape 1 :

1 : 4
4 : 1

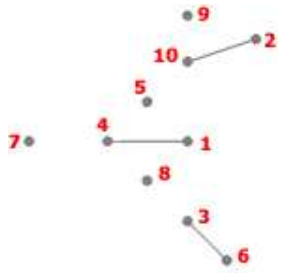
I={2,3,4,5,6,7,8,9,10}
S={10,3,8,4,4,5,10}



Étape 2 :

1 : 4
2 : 10
4 : 1
10 : 2

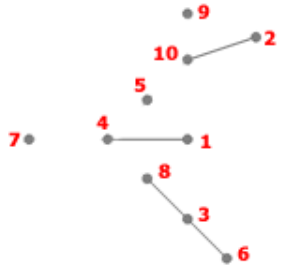
I={3,4,5,6,7,8,9,10}
S={3,8,4,4,5,10}



Étape 3 :

1 : 4
2 : 10
3 : 6
4 : 1
6 : 3
10 : 2

I={3,4,5,7,8,9,10}
S={8,4,4,5,10}

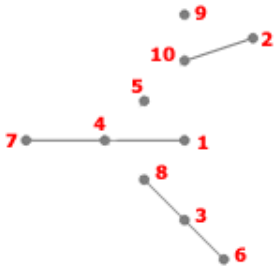


Étape 4 :

1 : 4
2 : 10
3 : 6, 8
4 : 1
6 : 3
8 : 3
10 : 2

I={4,5,7,8,9,10}

S={4,4,5,10}



Étape 5 :

1 : 4

2 : 10

3 : 6, 8

4 : 1, 7

6 : 3

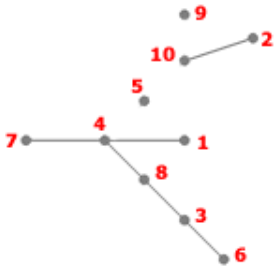
7 : 4

8 : 3

10 : 2

I={4,5,8,9,10}

S={4,5,10}



Étape 6 :

1 : 4

2 : 10

3 : 6, 8

4 : 1, 7, 8

6 : 3

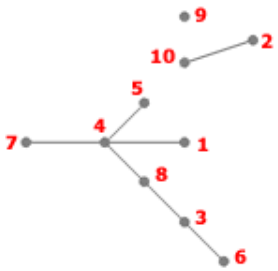
7 : 4

8 : 3, 4

10 : 2

I={4,5,9,10}

S={5,10}

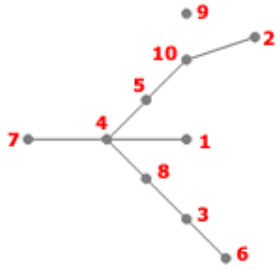


Étape 7 :

1 : 4

2 : 10
3 : 6, 8
4 : 1, 5, 7, 8
5 : 4
6 : 3
7 : 4
8 : 3, 4
10 : 2

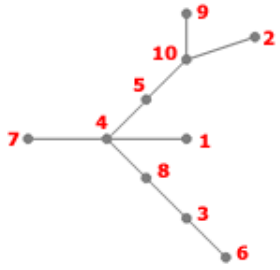
I={5,9,10}
S={10}



Étape 8 :

1 : 4
2 : 10
3 : 6, 8
4 : 1, 5, 7, 8
5 : 4, 10
6 : 3
7 : 4
8 : 3, 4
10 : 2, 5

I={9,10}
S={}

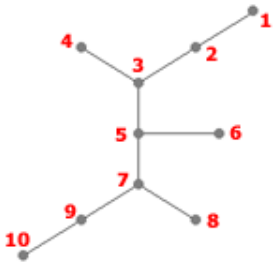


Étape 9 :

1 : 4
2 : 10
3 : 6, 8
4 : 1, 5, 7, 8
5 : 4, 10
6 : 3
7 : 4
8 : 3, 4
9 : 10
10 : 2, 5, 9

I={}
S={}

Exercice 25.7. *Décrivez l'arbre ci-dessous à l'aide du codage de Prüfer.*



Exercice 25.8. *Dessinez l'arbre correspondant à la suite S={1,1,1,1,1,1,1,1}.*

Exercice 25.9. *Dessinez l'arbre correspondant à la suite S={1,2,3,4,5,6,7,8}.*

II.4 Théorème de Cayley

PROPRIÉTÉ 25.4 (CAYLEY, 1857) : Le nombre d'arbres que l'on peut construire sur n ($n > 1$) sommets numérotés est égal à n^{n-2}.

PREUVE 9 : *La preuve est immédiate en utilisant le codage de Prüfer.*

En effet, on vérifie aisément que les deux algorithmes constituent les deux sens d'une bijection entre les arbres sur n sommets numérotés et les mots de $n-2$ lettres sur l'alphabet à n lettres.

Ce constat permet de conclure la preuve du théorème de Cayley. En effet, il existe n^{n-2} mots de longueur $n-2$ sur l'alphabet à n lettres, donc d'arbres sur n sommets numérotés. †

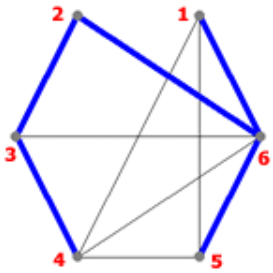
III Arbres couvrants

III.1 Définition

DÉFINITION 25.3 (ARBRE COUVRANT). *Un* arbre couvrant *(ou* arbre maximal*) d'un graphe G, est un graphe partiel de G qui est aussi un arbre.* ♢

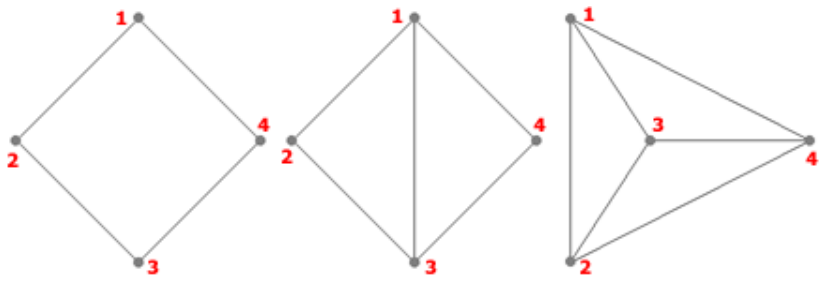
REMARQUE 25.1. On rappelle qu'un graphe partiel de G est obtenu en enlevant des arêtes (mais pas de sommets) à G.

EXEMPLE 25.10. Un des arbres couvrants (en bleu) d'un graphe donné.



III.1.1 Exercices

Exercice 25.11. *Dessiner des arbres couvrants pour chacun des graphes suivants.*



Exercice 25.12. *Combien d'arbres couvrants différents les graphes ci-dessus possèdent-ils ?*

III.2 Arbre maximal de poids minimum

III.2.1 Présentation du problème

On considérera le problème suivant :

« Soit le graphe $G = (V, E)$ avec un poids associé à chacune de ses arêtes. Trouver, dans G, un arbre maximal $A = (V, F)$ de poids total minimum. »

Ce problème se pose, par exemple, lorsqu'on désire relier n villes par un réseau routier de coût minimum. Les sommets du graphe représentent les villes, les arêtes, les tronçons qu'il est possible de construire et les poids des arêtes correspondent aux coûts de construction du tronçon correspondant.

L'algorithme de Kruskal décrit ci-dessous permet de résoudre ce problème.

III.2.2 L'algorithme de Kruskal

Voici un algorithme célèbre permettant de résoudre ce problème :

PROPRIÉTÉ 25.5 (L'ALGORITHME DE KRUSKAL) : Pour obtenir un arbre ou une forêt couvrant(e) de poids minimum à partir d'un graphe pondéré $G = (S, A)$, on procède comme suit :

1. On part du graphe $G' = (S, \varnothing)$ (G sans arête).
2. Tant que le nombre d'arêtes de G' est inférieur à $min(|G| - 1, |A|)$, faire
   – Prendre la plus petite arête (de poids minimal) restante dans G.
   – L'ajouter à G' si cela ne crée pas de cycle.
   – La supprimer de G.

Exercice 25.13. *Le tableau suivant donne les coûts de construction de routes (exprimées en unités adéquates) entre six villes d'Irlande.*

	Athlone	*Dublin*	*Galway*	*Limerick*	*Sligo*	*Wexford*
Athlone	-	*78*	*56*	*73*	*71*	*114*
Dublin	-	-	*132*	*121*	*135*	*96*
Galway	-	-	-	*64*	*85*	*154*
Limerick	-	-	-	-	*144*	*116*
Sligo	-	-	-	-	-	*185*

Trouver une manière de relier ces six villes, en minimisant le coût total de construction.

Exercice 25.14. *Faire de même avec le tableau suivant, qui comptabilise cette fois-ci les kilomètres entre chaque ville (valeurs imaginaires) :*

	Athlone	*Dublin*	*Galway*	*Limerick*	*Sligo*	*Wexford*
Athlone	-	*121*	*95*	*43*	*71*	*74*
Dublin	-	-	*72*	*98*	*115*	*97*
Galway	-	-	-	*64*	*77*	*134*
Limerick	-	-	-	-	*144*	*126*
Sligo	-	-	-	-	-	*85*

IV Arborescence

IV.1 Définitions et exemples

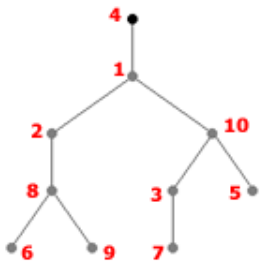
IV.1.1 Définition d'une arborescence

DÉFINITION 25.4 (ARBORESCENCE). *On appelle* arborescence *un arbre avec un sommet distingué, que l'on appelle la racine.* ♢

NOTATION : On représente généralement une arborescence avec la racine en haut du dessin et les feuilles en bas.

IV.1.2 Exemples

EXEMPLE 25.15. Sur l'arborescence ci-dessous, la racine est le sommet 4. Les sommets 5, 6, 7 et 9 sont les feuilles.



EXEMPLE 25.16. Les systèmes de fichiers ($ext3$ sous linux, $ntfs$ sous windows, etc.) sont agencés en arborescence.

IV.1.3 Rang des sommets

On peut, dans une arborescence, assigner un rang aux sommets :

DÉFINITION 25.5 (RANG D'UN SOMMET). *Le* rang *d'un sommet d'une arborescence est la distance de ce sommet à la racine.* ♢

EXEMPLE 25.17. Ainsi, dans l'exemple précédent, le sommet 4 (la racine) a rang 0, le sommet 1 a rang 1, les sommets 2 et 10 ont rang 2, les sommets 3, 5 et 8 ont rang 3 et les sommets 6, 7 et 9 ont rang 4.

DÉFINITION 25.6 (HAUTEUR D'UNE ARBORESCENCE). *On dira que la* hauteur *de l'arborescence est le rang maximum* ♢

EXEMPLE 25.18. L'exemple ci-dessus a une hauteur de 4.

IV.2 Arborescences ordonnées, parcours en largeur et profondeur

IV.2.1 Arborescences ordonnées

DÉFINITION 25.7 (ARBORESCENCE ORDONNÉE). *Une* arborescence ordonnée *est une arborescence dont les arêtes partant de chaque sommet sont ordonnées.* ♢

Une fois qu'une arborescence a été ordonnée, il existe alors une représentation graphique canonique, la seule qui respecte cet ordre, de telle sorte que, pour chaque arête,

- elle possède à sa gauche des arêtes plus petites (pour l'ordre considéré), et à sa droite de plus grandes arêtes,
- au-dessus d'elle, les arêtes sont plus petites pour l'ordre, et plus grandes en-dessous d'elles.

En d'autres termes, on va dans le sens des arêtes croissantes quand on parcourt cette représentation canonique :

- de la gauche vers la droite,
- ou du haut vers le bas.

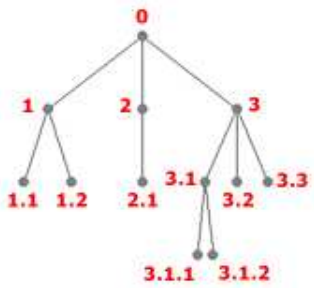
Il est donc possible de parler, sans ambiguïté, de droite et gauche, de haut et bas, pour une arborescence ordonnée.

REMARQUE 25.2. Une manière détournée d'ordonner une arborescence consiste à en donner une représentation graphique, et à considérer cette dernière comme canonique.

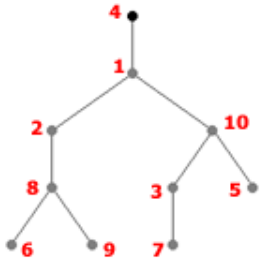
On peut étiqueter sans ambiguïté les sommets d'une arborescence ordonnée, comme suit :

- on attribue 0 à la racine r,
- puis 1, 2, 3, ... aux sommets qui sont adjacents à r, en respectant l'ordre des arêtes issues de la racines.
- Les étiquettes suivantes sont constituées de l'étiquette du sommet "père", suivie d'un chiffre obtenu comme précédemment.
- Ainsi, les sommets "fils" attachés au sommet 2 seront numérotés 2.1, 2.2, 2.3,...

EXEMPLE 25.19. La figure ci-dessous illustre le procédé.



Exercice 25.20. *Étiqueter les sommets de l'arborescence ordonnée dont la représentation canonique est la suivante :*



DÉFINITION 25.8. *Cet ordre des sommets (0, 1, 1.1, 1.2, 2, 2.1, 3, 3.1, 3.1.1, 3.1.2, 3.2, 3.3) est appelé* ordre lexicographique*, puisqu'il est semblable au classement des mots dans un dictionnaire.* ♢

IV.2.2 Parcours en largeur, en profondeur

Définition des parcours

DÉFINITION 25.9 (PARCOURS EN LARGEUR, EN PROFONDEUR). *Dans le cas du parcours d'une arborescence en suivant l'ordre lexicographique, on parle de* parcours en profondeur *de l'arborescence, par opposition au* parcours en largeur *qui serait l'ordre : 0, 1, 2, 3, 1.1, 1.2, 2.1, 3.1, 3.2, 3.3, 3.1.1, 3.1.2.* ♢

Le parcours en largeur consiste à visiter les sommets d'une arborescence ordonnée ligne par ligne, de la gauche vers la droite, les lignes étant parcourues du haut vers le bas.

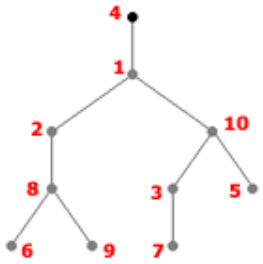
Le parcours en profondeur, par contre, correspond à parcourir de haut en bas la branche la plus à gauche de l'arbre, puis la branche immédiatement à droite, et ainsi de suite jusqu'à la branche la plus à droite.

REMARQUE 25.3. Chaque sommet n'apparaît qu'une fois dans ce parcours : le but d'un parcours étant justement de visiter une et une seule fois chaque sommet.

Un cas concret pourrait être la réalisation récursive d'un traitement pour chaque répertoire d'une arborescence : on souhaite visiter chaque répertoire une et une seule fois.

Exercices.

Exercice 25.21. *Donner la liste ordonnée des sommets visités lors d'un parcours en largeur, et d'un parcours en profondeur, dans l'arborescence suivante :*

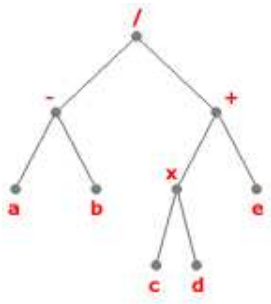


Exercice 25.22. *Réfléchir à un algorithme de parcours en largeur d'une arborescence ordonnée.*

IV.2.3 Lien avec les expressions algébriques

Arborescence d'une expression algébrique. N'importe quelle expression algébrique comprenant des expressions binaires, telle que l'addition, la soustraction, la multiplication et la division, peut être représentée par une arborescence ordonnée.

EXEMPLE 25.23. Par exemple, l'aborescence ci-dessous représente l'expression arithmétique $(a-b)/((c\times d)+e)$:



On observe que les variables de l'expression (a, b, c, d et e) sont les feuilles de l'arborescence, et que les opérations sont les autres sommets.

REMARQUE 25.4. L'arbre doit être ordonné car $a - b$ et $b - a$, qui ne sont pas égaux, conduisent au même arbre, mais pas au même arbre ordonné.

Exercice 25.24. *Construire les arborescences associées aux expressions algébriques :* $(a + b) * (c - (d - e))$ *et* $((a/b) * c) - (d - e)$.

Notation polonaise. Le mathématicien polonais *Lukasiewicz* a remarqué qu'en plaçant les symboles des opérations binaires avant les arguments, c'est-à-dire $+ab$ au lieu de $a+b$ ou $/cd$ au lieu de c/d, nous n'avions plus besoin de parenthèses...

DÉFINITION 25.10 (NOTATION POLONAISE). *On appelle cette nouvelle notation la* notation polonaise *dans sa forme préfixée ou directe.* ♢

Cette notation polonaise revient à effectuer un parcours en profondeur de l'arborescence associée à l'expression algébrique considérée.

EXEMPLE 25.25. L'expression $(a - b)/((c \times d) + e)$ devient ainsi $/ - ab + \times cde$.

Par analogie, en notation polonaise postfixée ou inverse, on place les symboles après les arguments. Certaines calculatrices - notamment des HP - utilisent la notation polonaise inverse.

IV.3 Exercices

Exercice 25.26. *Ecrire les expressions algébriques* $(a + b) \times (c - (d - e))$ *et* $((a/b) \times c) - (d - e)$ *en notation polonaise préfixée, et postfixée.*

Exercice 25.27. *Étant donnée l'expression algébrique* $(2x + y) \times (5a - b)^3$,

1. *Dessinez l'arborescence ordonnée correspondante (on utilisera le symbole* $\wedge$ *pour l'exponentiation).*
2. *Trouvez la* portée *de l'exponentiation (la portée d'un sommet* s *dans une arborescence est le sous-arbre généré par* s *et les sommets qui le suivent, avec* s *pour racine).*
3. *Écrire l'expression algébrique en notation polonaise directe, et inversée.*

IV.4 Codage de Huffman

IV.4.1 Principe

Le *codage de Huffman* [1] (1952) est un algorithme de compression sans perte, utilisant la notion d'arbres, qui permet de réduire la taille de données numériques.

Le principe, élémentaire, consiste à choisir de ne plus coder chaque symbole avec un nombre fixe de bits, mais à coder les symboles les plus fréquents avec un plus petit nombre de bits (quitte à devoir utiliser beaucoup de bits pour les symboles les plus rares).

EXEMPLE 25.28. Par exemple, dans un texte donné, le w apparait 20 fois moins souvent que le e. Plutôt que de coder chaque lettre sur un octet, on va donc :

- Coder le e par 1 (un seul bit, au lieu de 8),
- Coder le w par 010100000100 (12 bits, au lieu de 8).

Si le w apparait 10 fois (donc le e 200 fois), le gain sera de $7 \times 200 - 4 \times 10 = 1360$ bits, rien qu'en considérant ces deux lettres.

Cet algorithme offre des taux de compression démontrés les meilleurs possibles pour un codage par symbole, et il est assez compliqué de mieux compresser.

1. David Albert Huffman

IV.4.2 Propriété de préfixe

Le codage ne peut pas se faire n'importe comment : il faut pouvoir décoder, sans ambiguïté.

EXEMPLE 25.29. Si l'on décide de :
- Coder le e par 0,
- Coder le w par 010100000100,
- Coder le x par 10100000100,

Alors le texte 010100000100 pourra se traduire à la fois par ex, et w. On ne peut pas accepter une telle indétermination.

Le codage de Huffman a donc une propriété de préfixe : une séquence binaire ne peut jamais être à la fois représentative d'un élément codé et constituer le début du code d'un autre élément.

EXEMPLE 25.30. Si un symbole est représenté par la combinaison binaire 100, alors la combinaison 10001 ne peut être le code d'aucune autre information.

Cette caractéristique du codage de Huffman permet une codification à l'aide d'une structure d'arbre binaire...

IV.4.3 Méthode

On part d'une forêt, constituée d'arbres à une racine et une feuille, avec autant d'arbres que de symboles à coder. La feuille contient le symbole, la racine contient la fréquence d'apparition de ce symbole.

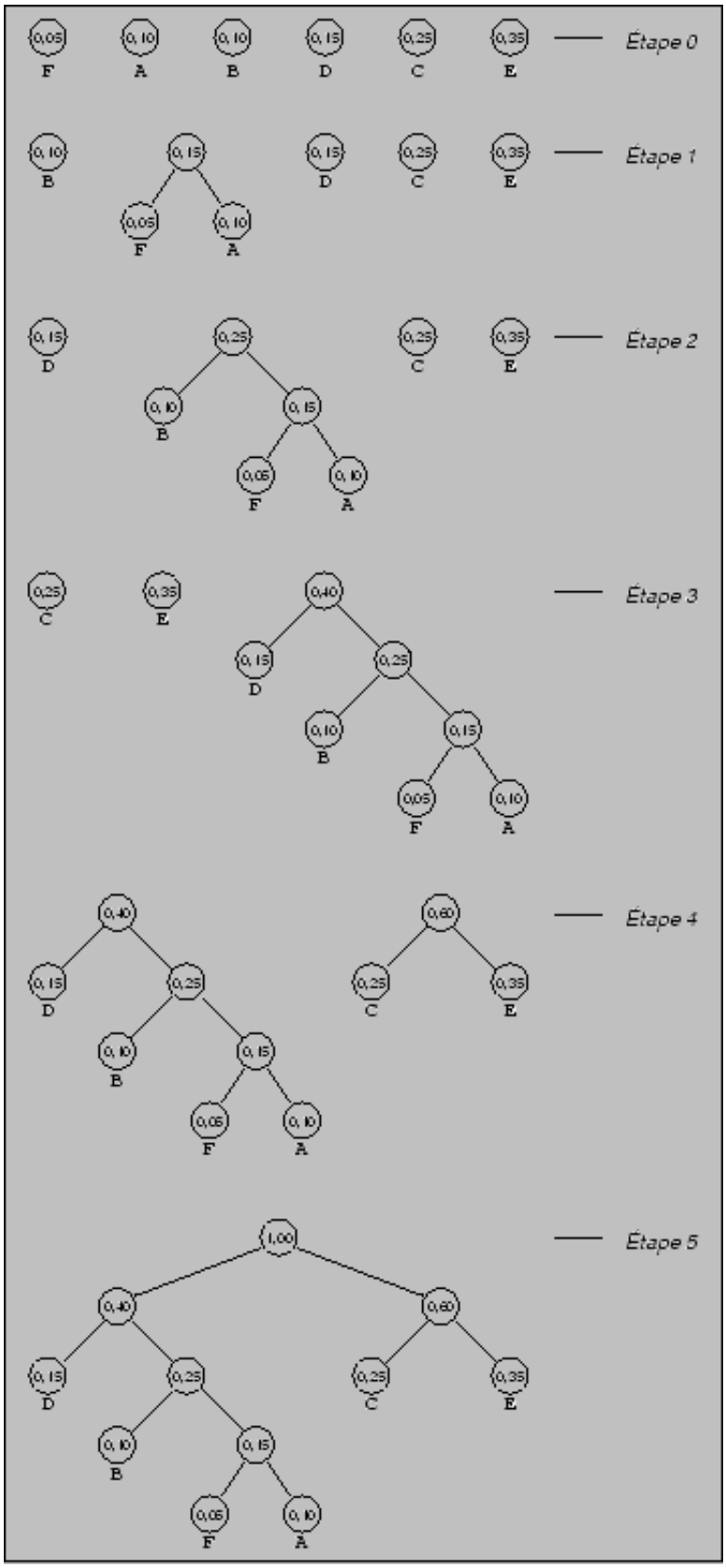
A chaque étape, on choisit les deux arbres x,y de racines minimales, et on les remplacent par l'arbre ayant pour racine r la somme des racines de x et y, et pour fils de r les arbres x et y.

La forêt finale est formée d'un unique arbre.

Le code d'une lettre est alors déterminé en suivant le chemin depuis la racine de l'arbre, jusqu'à la feuille du symbole qui nous intéresse, en concaténant successivement un 0 ou un 1 selon que la branche empruntée est à gauche ou à droite.

On dit que l'algorithme est de type glouton : il choisit à chaque étape les meilleurs choix (locaux) possibles, dans l'espoir que le résultat final sera minimal.

Voici les différentes étapes de la construction d'un code de Huffman pour l'alphabet source A, B, C, D, E, F, avec les probabilités P(A)=0.10, P(B)=0.10, P(C)=0.25, P(D)=0.15, P(E)=0.35 et P(F)=0.05.



EXEMPLE 25.31. Ainsi, sur la figure ci-contre, A=0111, B=010, C=10, D=00, E=11 et F=0110.

EXEMPLE 25.32. Par exemple le mot FADE serait codé 011001110011. Pour décoder, on lit simplement la chaîne de bits de gauche à droite. Le seul "découpage" possible, grâce à la propriété du préfixe, est 0110-0111-00-11.

IV.4.4 Applications

Malgré son ancienneté, le codage de Huffman est toujours au goût du jour, et offre encore des performances appréciables. Il est utilisé dans presque toutes les applications qui impliquent la compression et la transmission de données numériques : fax, modems, réseaux informatiques, télévision HD, *etc.*

Les premiers Macintosh utilisaient un code inspiré de Huffman pour la représentation des textes : chaque lettre faisait tantôt 4 bits, tantôt 12 bits, suivant sa fréquence d'apparition dans le texte. Cette méthode simple se révélait économiser 30% d'espace sur un texte moyen, à une époque où la mémoire vive restait encore un composant coûteux.

Le MPEG-1/2 Audio Layer 3, plus connu sous son abréviation de MP3, est constitué de deux étapes de compression :

– La première, avec perte, revient à la mise à l'écart de données inaudibles pour l'oreille humaine.
– La deuxième est un codage de Huffman.

Le logiciel de compression *gzip* (GNU zip) utilise Huffman (avec un autre algorithme : LZ77). Sa version améliorée, l'algorithme *bzip2*, utilise la transformée de Burrows-Wheeler avec le codage de Huffman.

Ce principe de compression est aussi utilisé dans le codage d'images. Il fait par exemple partie des spécifications des formats :

– TIFF (Tagged Image Format File) spécifié par Microsoft Corporation et Aldus Corporation. Le codage d'image est fait en retranscrivant exactement le contenu d'un écran (image), en utilisant Huffman.
– JPG (Join Photographic Experts Group), algorithme de compression avec perte, dont un des éléments est Huffman.

IV.4.5 Exercices

Exercice 25.33. *Construisez un codage de Huffman du message "ceciestuncodagedehuffman" (on a supprimé les espaces et la ponctuation pour simplifier la construction). Il y a plusieurs codages de Huffman possibles. Vérifiez la propriété du préfixe.*

Exercice 25.34. *Utilisez le tableau ci-dessous pour déterminer le codage de Huffman de la langue française.*

Fréquences d'apparition des lettres en français

Lettre	*Fréquence*	*Lettre*	*Fréquence*
A	*8.40 %*	*N*	*7.13 %*
B	*1.06 %*	*O*	*5.26 %*
C	*3.03 %*	*P*	*3.01 %*
D	*4.18 %*	*Q*	*0.99 %*
E	*17.26 %*	*R*	*6.55 %*
F	*1.12 %*	*S*	*8.08 %*
G	*1.27 %*	*T*	*7.07 %*
H	*0.92 %*	*U*	*5.74 %*
I	*7.34 %*	*V*	*1.32 %*
J	*0.31 %*	*W*	*0.04 %*
K	*0.05 %*	*X*	*0.45 %*
L	*6.01 %*	*Y*	*0.30 %*
M	*2.96 %*	*Z*	*0.12 %*

Fin du Chapitre

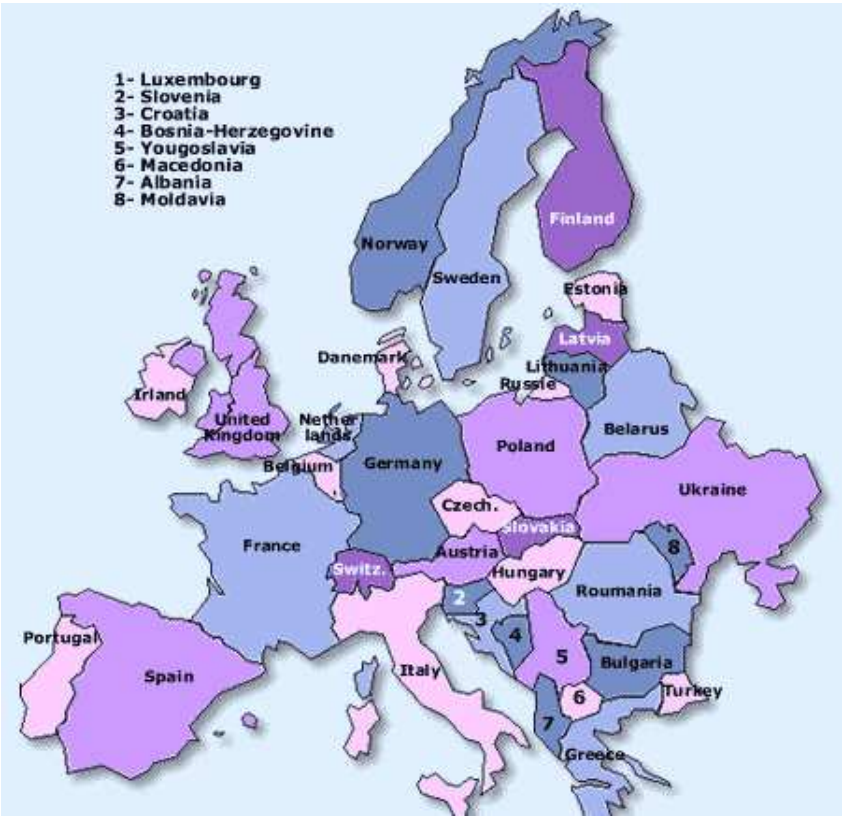
Chapitre 26

Problèmes de coloration

I Présentation du problème

I.1 Un problème historique

Une question datant de la mi-XIX^{e}, reliée à la coloration de graphes planaires, est devenue célèbre sous le nom de problème des quatre couleurs : suffit-il de quatre couleurs pour dessiner n'importe quelle carte géographique ?



Exercice 26.1. *Prenez une feuille de papier. Tracez une droite quelconque qui traverse la feuille de part en part. Recommencez l'opération n fois. Démontrez que la "carte" ainsi obtenue peut être colorée en deux couleurs.*

I.2 Formulation en théorie des graphes

Ce problème a une formulation dans le langage des graphes : y a-t-il toujours, dans un graphe planaire, une application de l'ensemble S des sommets vers un ensemble de cardinal 4, telle que deux

sommets « adjacents » admettent toujours des images distinctes ?

PROPRIÉTÉ 26.1 (THÉORÈME DES QUATRE COULEURS) : On peut colorer les sommets d'un graphe planaire (sans boucles) en utilisant au plus quatre couleurs de telle sorte que toutes les arêtes aient des extrémités de couleurs différentes.

Cette conjecture a été formulée pour la première fois par l'Écossais Francis Guthrie en 1852. Il était alors question de coloration de carte de géographie (voir exercice ci-dessous).

La preuve de ce théorème n'arriva qu'en... 1976, grâce à Kenneth Appel et Wolfgang Haken. La démonstration fit grand bruit car c'est le premier théorème de l'histoire des mathématiques qui a nécessité l'usage systématique de l'ordinateur. Le résultat de Appel et Haken est donc délicat à prouver.

La communauté mathématique se divisa alors en deux camps : ceux pour qui le théorème des quatre couleurs était définitivement démontré, et ceux pour qui tout restait à faire.

S'il n'est pas question ici de démontrer le théorème des quatre couleurs, on verra quand même, dans les prochaines sections, qu'un certain nombre de résultats sur le nombre de couleurs nécessaires à la coloration des sommets d'un graphe peuvent s'obtenir sans trop de difficultés. Le problème de la coloration des arêtes sera lui aussi introduit.

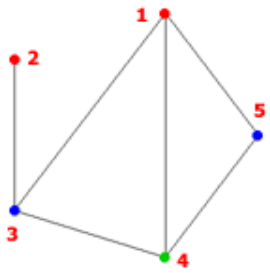
II Coloration des sommets

Afin d'obtenir des encadrements du nombre de couleurs nécessaires à la coloration des sommets d'un graphe, il nous faut exprimer rigoureusement le problème de coloration en termes issus de notre théorie des graphes...

II.1 Rappels sur la notion de stable

On rappelle que l'on appelle stable d'un graphe $G = (S, A)$ tout sous-graphe sans arête obtenu en enlevant des sommets à G (ce qui entraîne de fait la suppression de leurs arêtes incidentes).

EXEMPLE 26.2. Dans le graphe suivant, on peut obtenir un stable en enlevant les sommets $\{1, 2, 4\}$: le graphe résultant est sans arête.



DÉFINITION 26.1 (NOMBRE DE STABILITÉ). *Le cardinal du plus grand stable est le* nombre de stabilité *de* G ♢

NOTATION : On le note $a(G)$.

II.2 Le problème de coloration

DÉFINITION 26.2 (COLORATION DES SOMMETS D'UN GRAPHE). *La* coloration des sommets *d'un graphe consiste à affecter à tous les sommets de ce graphe une couleur de telle sorte que deux sommets adjacents ne portent pas la même couleur.* ♢

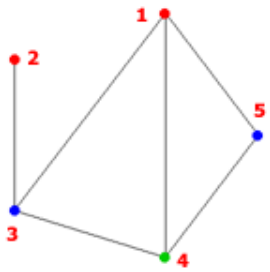
PROPRIÉTÉ 26.2 : Une coloration avec k couleurs est une partition de l'ensemble des sommets S en k stables.

DÉFINITION 26.3 (NOMBRE CHROMATIQUE). *Le* nombre chromatique *du graphe G est le plus petit entier k pour lequel il existe une partition de V en k sous-ensembles stables. C'est le nombre de couleurs minimal pour colorier un graphe.* ♢

NOTATION : Ce nombre chromatique est noté $g(G)$.

Exercice 26.3. *Dans ce qui précède, on suppose que le nombre chromatique existe toujours. Pourquoi ?*

EXEMPLE 26.4. Sur le graphe ci-dessous, on a eu besoin de trois couleurs pour colorer les sommets de sorte que deux sommets adjacents ont des couleurs différentes.



REMARQUE 26.1. La coloration minimale n'est pas forcément unique. Par exemple, ci-dessus, le sommet 2 aurait aussi pu être vert.

II.3 Encadrement du nombre chromatique

II.3.1 Majoration

On donne, dans les deux propriétés suivantes, deux majorations pour le nombre chromatique.

PROPRIÉTÉ 26.3 : $g(G) \leqslant r+1$, où r est le plus grand degré de ses sommets.

PREUVE 10 : *Soit un graphe et r le degré maximum de ses sommets. Donnons-nous une palette de $(r+1)$ couleurs. Pour chaque sommet du graphe on peut tenir le raisonnement suivant :*

- *ce sommet est adjacent à r sommets au plus,*
- *le nombre de couleurs déjà utilisées pour colorer ces sommets est donc inférieur ou égal à r.*

Il reste donc au moins une couleur non utilisée dans la palette, avec laquelle nous pouvons colorer notre sommet. †

PROPRIÉTÉ 26.4 : $g(G) \leqslant n + 1 - a(G)$

PREUVE 11 : *Considérons S un stable de V de cardinal $a(G)$: une coloration possible des sommets consiste à colorer les sommets de S d'une même couleur et les $n - a(G)$ autres sommets de couleurs toutes différentes.*

On en déduit que $g(G) \leqslant 1 + (n - a(G))$. †

II.3.2 Minoration

On a d'autres résultats, concernant la minoration...

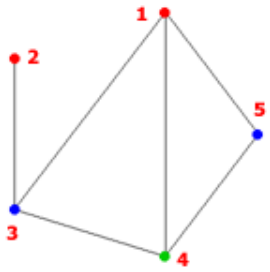
PROPRIÉTÉ 26.5 : Le nombre chromatique d'un graphe est supérieur ou égal à celui de chacun de ses sous-graphes.

PREUVE 12 : *Ce résultat découle de la définition même du nombre chromatique.* †

PROPRIÉTÉ 26.6 : $g(G) \geqslant w(G)$, où $w(G)$ désigne l'ordre de la plus grande clique de G.

PREUVE 13 : *Puisque, par définition, dans une clique d'ordre m, tous les sommets sont adjacents entre eux, il faudra m couleurs pour colorier ce sous-graphe. Donc, forcément, le nombre chromatique du graphe sera supérieur ou égal à l'ordre de sa plus grande clique.* †

EXEMPLE 26.5. Dans le graphe précédant, on a utilisé trois couleurs :



On ne peut faire mieux, à cause des cliques 1-4-5 et 1-3-4.

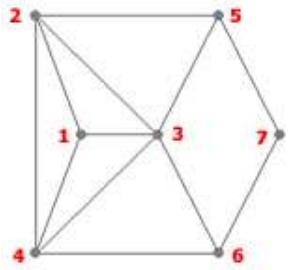
REMARQUE 26.2. On peut déduire de la propriété précédente que tout graphe possédant un triangle ne peut être colorié en moins de trois couleurs.

Exercice 26.6. *Que dire du nombre chromatique d'un graphe contenant un carré ? Un polygone régulier avec n sommets ?*

Réponse : Il faut au minimum deux couleurs quand le nombre de sommets est pair, et trois quand il est impair.

Exercice 26.7. *Soit G un graphe contenant K_n. Donner un minimum au nombre de couleurs nécessaires à la coloration de ses sommets.*

Exercice 26.8. *Déterminez un encadrement pour le nombre chromatique du graphe :*



On vérifiera que ce graphe possède une clique d'ordre 4, et on en tirera les conséquences.

II.4 Algorithme de coloration de Welsh et Powell

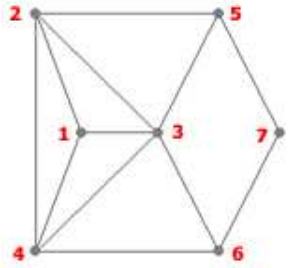
Cet algorithme couramment utilisé permet d'obtenir une assez bonne coloration d'un graphe, c'est-à-dire une coloration n'utilisant pas un trop grand nombre de couleurs. Cependant il n'assure pas que le nombre de couleurs utilisé soit minimum (et donc égal au nombre chromatique du graphe).

Étape 1 Classer les sommets du graphe dans l'ordre décroissant de leur degré.

Étape 2 En parcourant la liste dans l'ordre, attribuer une couleur libre au premier sommet s sans couleur, et attribuer cette même couleur à chaque sommet qui n'est pas adjacent à s (et qui n'est pas encore coloré).

Étape 3 S'il reste des sommets sans couleur, revenir à l'étape 2.

Exercice 26.9. *Appliquer l'algorithme de Welsh et Powell pour colorier le graphe :*



II.5 Exercices

Exercice 26.10. *On a vu que tout graphe contenant un triangle (K_3) ne peut pas être coloré en moins de trois couleurs.*

1. *Construire un graphe sans K_3 qui nécessite également trois couleurs.*
2. *Comment, à partir du graphe précédent, construire un graphe sans K_4 nécessitant 4 couleurs ?*
3. *Un graphe sans K_5 nécessitant 5 couleurs ?*

Réponse : Un carré. Puis, rajouter une diagonale contenant un sommet à ce carré. Enfin, rajouter n diagonales contenant chacune un sommet.

Exercice 26.11. *Comment ramener la résolution d'un carré latin à un problème de coloration de graphes ? L'algorithme de Welsh et Powell permet-il de le résoudre pour une taille 3 ? pour une taille 4 ?*

Exercice 26.12. *Trouver un lien entre la résolution d'un Sudoku et un problème de coloration de graphes.*

Exercice 26.13. *Sept élèves, désignés par A, B, C, D, E, F et G, se sont rendus en salle machine. Le tableau suivant, construit à partir des logs, précise «qui a joué avec qui (en réseau) ».*

l'élève	*A*	*B*	*C*	*D*	*E*	*F*	*G*
a rencontré	*D,E*	*D,E,F,G*	*E,G*	*A,B,E*	*A,B,C,D,F,G*	*B,E,G*	*B,C,E,F*

Sachant que chaque individu a trouvé un ordinateur libre pour travailler, que pouvez-vous dire du nombre minimum d'ordinateurs ?

Exercice 26.14. *A, B, C, D, E, F, G et H désignent huit poissons. Dans le tableau ci-dessous, une croix signifie que les poissons ne peuvent pas cohabiter dans un même aquarium :*

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
A		×	×	×			×	×
B	×				×	×	×	
C	×			×		×	×	×
D	×		×		×			×
E		×		×		×	×	
F		×	×		×			
G	×	×	×		×			
H	×		×	×				

Quel nombre minimum d'aquariums faut-il ?

Exercice 26.15. *Un lycée doit organiser les horaires des examens.*

On suppose qu'il y a 7 épreuves à planifier, correspondant aux cours numérotés de 1 à 7, et que les paires de cours suivantes ont des étudiants communs : 1 et 2, 1 et 3, 1 et 4, 1 et 7, 2 et 3, 2 et 4, 2 et 5, 2 et 7, 3 et 4, 3 et 6, 3 et 7, 4 et 5, 4 et 6, 5 et 6, 5 et 7 et enfin 6 et 7.

Comment organiser ces épreuves de façon qu'aucun étudiant n'ait à passer deux épreuves en même temps, et cela sur une durée miminale ?

Exercice 26.16. *Sept agences de voyage romaines proposent des visites de monuments et lieux touristiques : le Colisée, le Forum romain, le musée du Vatican et les thermes de Caracalas. Un même lieu ne peut être visité par plusieurs groupes de compagnies différentes le même jour.*

La première Compagnie fait visiter uniquement le Colisée ; la seconde le Colisée et le musée du Vatican ; la troisième les thermes de Caracalas ; la quatrième le musée du Vatican et les thermes de Caracalas ; la cinquième le Colisée et le Forum romain ; la sixième le Forum romain et les thermes de Caracalas ; la septième le musée du Vatican et le forum romain.

Ces agences peuvent-elles organiser les visites sur les trois premiers jours de la semaine ?

III Coloration des arêtes

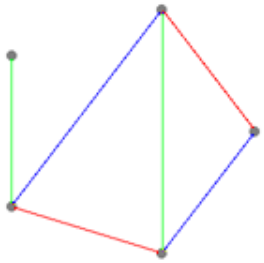
III.1 Présentation du problème

La coloration des arêtes d'un graphe consiste à affecter à toutes les arêtes de ce graphe une couleur de telle sorte que deux arêtes adjacentes ne portent pas la même couleur.

DÉFINITION 26.4 (INDICE CHROMATIQUE). *L'indice chromatique du graphe G est le plus petit entier k pour lequel il existe une coloration des arêtes.* ♢

NOTATION : On le note $c(G)$.

EXEMPLE 26.17. Sur le graphe ci-dessous, on a eu besoin de trois couleurs pour colorer les arêtes de sorte que deux arêtes adjacentes ont des couleurs différentes.



III.2 Lien avec la coloration des sommets

III.2.1 Présentation

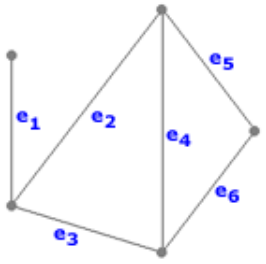
Pour colorer les arêtes d'un graphe, on peut se ramener au problème de la coloration des sommets. Il suffit pour cela de travailler non pas sur le graphe lui-même, mais sur le graphe adjoint, noté G', et que l'on définit ainsi :

- à chaque arête de G = (V, E) correspond un sommet de G' = (E, F)
- deux sommets de G' sont reliés par une arête si les deux arêtes correspondantes de G sont adjacentes.

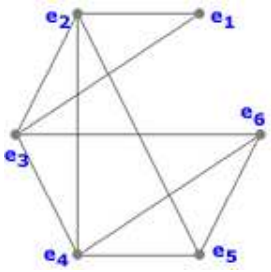
On peut ensuite appliquer par exemple l'algorithme de Welsh et Powell sur le graphe G' pour colorer ses sommets. Une fois cela fait, on colorera les arêtes de G de la même couleur que les sommets correspondants de G'.

III.2.2 Exemple de coloration d'arêtes

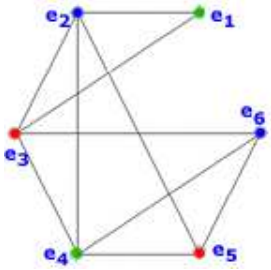
1. Un graphe G :



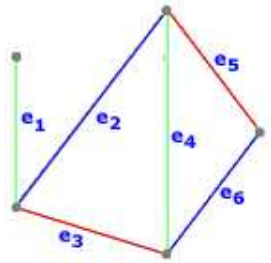
2. Son graphe adjoint G'



3. Coloration des sommets de G'



4. Coloration des arêtes de G



III.3 Exercice

Exercice 26.18. *Dans un tournoi d'échecs, chaque engagé doit rencontrer tous les autres. Chaque partie dure une heure.*

Déterminer la durée minimum du tournoi dans le cas où le nombre d'engagés est 3, 4, 5 ou 6.

Fin du Chapitre

Chapitre 27

Graphes orientés

I Définitions

I.1 Digraphe (graphe orienté), sommet, arc

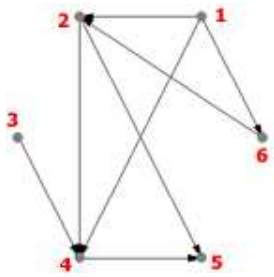
En donnant un sens aux arêtes d'un graphe, on obtient un digraphe, ou graphe orienté.

REMARQUE 27.1. De l'anglais directed graph.

DÉFINITION 27.1 (DIGRAPHE, SOMMETS, ARCS). *Un* digraphe *fini $G = (V, E)$ est défini par :*
- *l'ensemble fini $V = \{v_1, v_2, ..., v_n\}$ ($|V| = n$) dont les éléments sont appelés* sommets *,*
- *et par l'ensemble fini $E = \{e_1, e_2, ..., e_m\}$ ($|E| = m$) dont les éléments sont appelés* arcs*.*

Un arc e de l'ensemble E est défini par une paire ordonnée de sommets. Lorsque $e = (u, v)$, on dira que l'arc e va de u à v. On dit aussi que u est l'extrémité initiale et v l'extrémité finale de e. ♢

EXEMPLE 27.1. Un exemple de digraphe :



Exercice 27.2. *Construire un graphe orienté dont les sommets sont les entiers compris entre 1 et 12 et dont les arcs représentent la relation « être diviseur de ».*

Exercice 27.3. *Quel est le nombre maximal d'arêtes dans un graphe orienté d'ordre n qui ne possède pas d'arêtes parallèles ?*

EXEMPLE 27.4. Le graphe d'une relation binaire peut être assimilé à un graphe orienté.

REMARQUE 27.2. Les graphes orientés peuvent être représentés graphiquement, comme dans le cas non-orienté. On place alors des flèches au lieu de simples segments.

I.2 Degré d'un sommet d'un digraphe

Soit v un sommet d'un graphe orienté.

DÉFINITION 27.2 (DEGRÉ EXTÉRIEUR). *Le* degré extérieur *du sommet v est le nombre d'arcs ayant v comme extrémité initiale.* ♢

NOTATION : On le note $d_+(v)$.

DÉFINITION 27.3 (DEGRÉ INTÉRIEUR). *Le* degré intérieur *du sommet v est le nombre d'arcs ayant v comme extrémité finale.* ♢

NOTATION : On le note $d_-(v)$.

PROPRIÉTÉ 27.1 : On a :

$$d(v) = d_+(v) + d_-(v)$$

.

Exercice 27.5. *Soit G un graphe orienté quelconque. Démontrez que la somme des degrés entrants de tous les sommets est égal à la somme de tous les degrés sortants.*

Réponse : Chaque arête compte une fois dans la somme des degrés entrants et une fois dans la somme des degrés sortants...

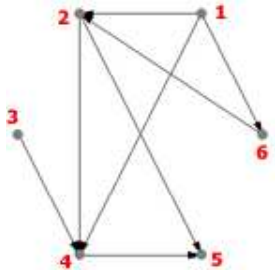
Exercice 27.6. *Soit X un ensemble de lapins, et G un graphe orienté ayant X pour ensemble de sommets. On dit que G est un «graphe de parenté» si les arcs de G codent la relation «être l'enfant de...». Quelles conditions doit nécessairement vérifier G pour pouvoir être un graphe de parenté ?*

I.3 Chemins et circuits

I.3.1 Chemin

DÉFINITION 27.4 (CHEMIN). *Un* chemin *conduisant du sommet a au sommet b est une suite de la forme $(v_0, e_1, v_1, e_2, v_2, ..., e_k, v_k)$ où les v_i sont des sommets ($v_0 = a$ et $v_k = b$) et les e_i sont des arcs tels que e_i va de v_{i-1} à v_i.* ♢

EXEMPLE 27.7. Sur le digraphe ci-dessous, on peut voir par exemple le chemin $(v_3, e_3, v_4, e_4, v_5)$.



REMARQUE 27.3. Par convention, tout chemin comporte au moins un arc.

I.3.2 Distance

DÉFINITION 27.5 (DISTANCE). *On appelle* distance *entre deux sommets d'un digraphe la longueur du plus petit chemin les reliant.*

S'il n'existe pas de chemin entre les sommets x et y, on pose $d(x, y) = +\infty$. ♢

EXEMPLE 27.8. Par exemple, sur le digraphe ci-dessus (exemple précédent), $d(v_1, v_5) = 2$, $d(v_1, v_6) = 1$, $d(v_6, v_1) = +\infty$.

Exercice 27.9. *Donnez un algorithme permettant de calculer la distance entre deux sommets x et y d'un digraphe connexe.*

I.3.3 Circuit

DÉFINITION 27.6 (CIRCUIT). *Un* circuit *est un chemin avec $u_0 = u_k$.* ♢

EXEMPLE 27.10. Le digraphe ci-dessus ne contient pas de circuit.

REMARQUE 27.4. Les notions de longueur, de chemins et de circuits sont analogues à celles des chaînes et des cycles pour le cas non orienté.

II Digraphe fortement connexe

II.1 Définitions

II.1.1 Connexité forte

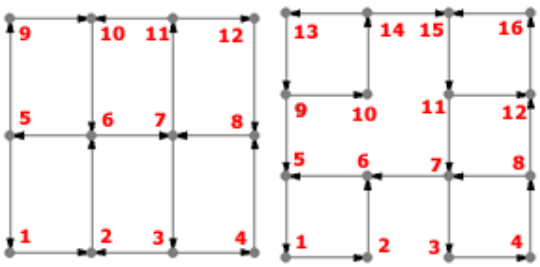
DÉFINITION 27.7 (DIGRAPHE FORTEMENT CONNEXE). *Un digraphe est* fortement connexe*, si toute paire ordonnée (a, b) de sommets distincts du graphe est reliée par au moins un chemin.* ♢

REMARQUE 27.5. En d'autres termes, tout sommet est atteignable depuis tous les autres sommets par au moins un chemin.

II.1.2 Composantes connexes

DÉFINITION 27.8 (COMPOSANTE FORTEMENT CONNEXE). *On appelle* composante fortement connexe *tout sous-graphe induit maximal fortement connexe (maximal signifie qu'il n'y a pas de sous-graphe induit connexe plus grand contenant les sommets de la composante).* ♢

Exercice 27.11. *Les graphes ci-dessous sont-il fortement connexes ? Si non, donnez leurs composantes fortement connexes.*



Exercice 27.12. *Proposez un algorithme qui détermine si un graphe est fortement connexe ou non.*

Indication : utilisez un système de marquage des sommets.

II.2 Circuits eulériens

PROPRIÉTÉ 27.2 : Dans le cas des graphes orientés, il y a équivalence entre :
- posséder un circuit eulérien,
- être fortement connexe, tel que $d_+(s) = d_-(s)$ pour tout sommet s.

III Matrice et listes d'adjacences

III.1 Matrices d'incidence

DÉFINITION 27.9 (MATRICE D'INCIDENCE ENTRANTE ET SORTANTE). *On suppose que les arêtes et les sommets ont été numérotés.*

On appelle matrice d'incidence sortante J^+ *la matrice dont l'élément* $J^+(s, \varepsilon)$ *vaut*
- *1 si le sommet* s *est le début de l'arête* ε*,*
- *0 sinon.*

On appelle de même matrice d'incidence entrante J^- *la matrice dont l'élément* (s, ε) *vaut :*
- *1 si le sommet* s *est la fin de l'arête* ε*,*
- *0 sinon.* ♢

DÉFINITION 27.10. *On suppose à nouveau que les arêtes et les sommets du graphe orienté considéré ont été numérotées, et on appelle alors* matrice d'incidence J *de ce graphe la matrice dont l'élément* (s, ε) *vaut :*
- *2 si* s *est une extrémité de* ε*, quand* ε *est une boucle,*
- *1 si* s *est une extrémité de* ε*, quand* ε *n'est pas une boucle,*
- *0 sinon.* ♢

REMARQUE 27.6. J est la matrice d'incidence du graphe non orienté sous-jacent, obtenu en remplaçant les arcs (flèches) par des arêtes.

Exercice 27.13. *Reprendre les graphes orientés dessinés dans ce chapitre, et trouver à chaque fois* J^+*,* J^- *et* J*.*

III.1.1 Résultats

On peut relier $d^+(s)$ (resp. $d^-(s)$) au nombre de 1 apparaissant dans J^+ (resp. J^-), comme suit.

PROPRIÉTÉ 27.3 : Soient $(s_1, \ldots, s_n)$ les sommets d'un graphe orienté. Alors

1. $\begin{pmatrix} d^+(s_1) \\ \vdots \\ d^+(s_n) \end{pmatrix} = J^+ \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ et $\begin{pmatrix} d^-(s_1) \\ \vdots \\ d^-(s_n) \end{pmatrix} = J^- \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$
2. $J^+ J^{-t} = \begin{pmatrix} d^+(s_1) & & 0 \\ & \ddots & \\ 0 & & d^+(s_n) \end{pmatrix}$

3. $J^- J^{+t} = \begin{pmatrix} d^-(s_1) & & 0 \\ & \ddots & \\ 0 & & d^-(s_n) \end{pmatrix}$

Exercice 27.14. *Vérifier ces résultats avec les matrices d'incidences calculées dans l'exercice précédent.*

III.2 Matrice d'adjacence

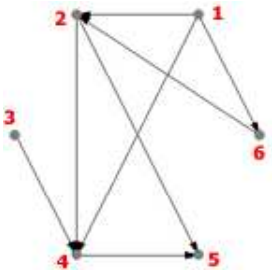
III.2.1 Définition

DÉFINITION 27.11 (MATRICE D'ADJACENCE). *On peut représenter un digraphe par une* matrice d'adjacence *:*

- *Les lignes et les colonnes représentent les sommets du graphe.*
- *Un 1 à la position (i,j) signifie qu'un arc part de i pour rejoindre j.* ♢

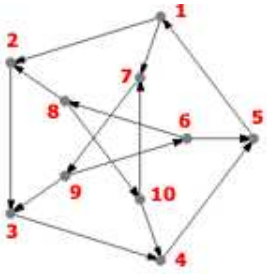
III.2.2 Exemple

Voici un digraphe et sa matrice d'adjacence :



$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Exercice 27.15. *Décrivez le digraphe G ci-contre par une matrice d'adjacences.*



REMARQUE 27.7. Se donner un graphe orienté revient à se donner sa matrice d'adjacence.

III.2.3 Propriétés

PROPRIÉTÉ 27.4 : La matrice d'adjacence à plusieurs caractéristiques :
- Elle est carrée : il y a autant de lignes que de colonnes.
- Il n'y a que des zéros sur la diagonale. Un 1 sur la diagonale indiquerait une boucle.
- Contrairement au cas non orienté, elle n'est pas symétrique.

PROPRIÉTÉ 27.5 (NOMBRE D'ARCS DE LONGUEUR K) : $A^k(s,t)$, élément à la position (s,t) de la puissance k$^{\text{ième}}$ de A, est aussi le nombre d'arcs de longueur k qui mènent de s à t.

PROPRIÉTÉ 27.6 : Soient $(s_1, \ldots, s_n)$ les sommets d'un graphe orienté. Alors

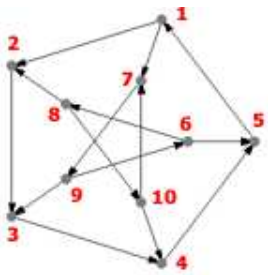
$$\begin{pmatrix} d^+(s_1) \\ \vdots \\ d^+(s_n) \end{pmatrix} = A \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \text{ et } \begin{pmatrix} d^-(s_1) \\ \vdots \\ d^-(s_n) \end{pmatrix} = A^t \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

Exercice 27.16. *Vérifier les propriétés ci-dessus sur la matrice d'adjacence calculée dans l'exercice précédent. On déterminera les chemins de longueur 2.*

PROPRIÉTÉ 27.7 (LIEN ENTRE LES MATRICES D'ADJACENCE) : Soit A la matrice d'adjacence d'un graphe orienté, et B la matrice d'adjacence du graphe non orienté qui lui est associé. Alors

$$\boxed{B = A + A^t}$$

Exercice 27.17. *Vérifier la dernière propriété sur le digraphe ci-dessous*



III.3 Lien entre matrices d'adjacences et d'incidences

Les remarques précedentes permettent de conclure que se donner (J^+, J^-) ou A, « c'est pareil ». Plus précisément, on a

$$\boxed{J^+ J^{-t} = A}$$

Exercice 27.18. *On pose*

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

1. *Dessinez le graphe orienté ayant A pour matrice d'adjacence.*
2. *Déterminez ses matrices d'incidences.*
3. *Vérifiez, sur cet exemple, les formules précédentes.*

Exercice 27.19. *A partir du graphe orienté G, on fabrique un graphe orienté H en retournant le sens de toutes les flèches.*

1. *Comment sont liées les matrices d'incidence de G et de H ?*
2. *Comment sont liées leurs matrices d'adjacence ?*

Réponse : La matrice J^+ de G est la matrice J^- de H, et réciproquement. Leurs matrices d'adjacence sont transposées l'une de l'autre.

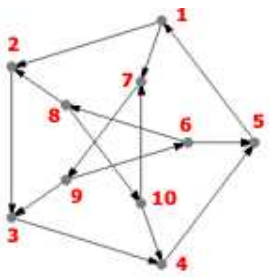
PROPRIÉTÉ 27.8 : Soient $s_1, s_2, \ldots, s_n$ les sommets d'un graphe orienté. Alors

$$\begin{pmatrix} d(s_1) \\ \vdots \\ d(s_n) \end{pmatrix} = J \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

PROPRIÉTÉ 27.9 (RELATION ENTRE J, J^+ ET J^-) : On note J^+ et J^- les matrices d'incidences d'un graphe orienté, et J la matrice d'incidence du graphe non orienté qui lui est associé. Alors

$$\boxed{J = J^+ + J^-}$$

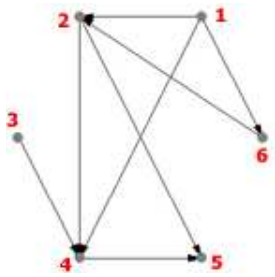
Exercice 27.20. *Vérifier ces propriété sur le digraphe ci-dessous*



III.4 Listes d'adjacence

On peut encore représenter un digraphe à l'aide de listes d'adjacences : en donnant pour chacun de ses sommets la liste des sommets qu'on peut atteindre directement en suivant un arc (dans le sens de la flèche).

EXEMPLE 27.21. Voici les listes d'adjacences du digraphe G :



1 : 2, 4, 6
2 : 4, 5
3 : 4
4 : 5
5 : -
6 : 2

Exercice 27.22. *Décrivez le digraphe G de l'exercice précédent par des listes d'adjacences.*

IV Digraphes sans circuits

IV.1 Théorème

PROPRIÉTÉ 27.10 : Le digraphe $G = (V, E)$ est sans circuit si et seulement si on peut attribuer un nombre $r(v)$, appelé le *rang* de v, à chaque sommet v de manière que pour tout arc (u, v) de G on ait $r(u) < r(v)$.

PREUVE 14 : *Si G comporte un circuit C, il n'est pas possible de trouver de tels nombres $r(i)$ car, autrement, considérant $r(j) = max\{r(i)|i \in C\}$ et l'arc $(j, k) \in C$ on aurait $r(j) \leqslant r(k)$ en contradiction avec la définition de $r()$.*

Réciproquement, si G n'a pas de circuits, il existe au moins un sommet sans prédécesseurs dans G (sans cela, en remontant successivement d'un sommet à un prédécesseur, on finirait par fermer un circuit). Ainsi, on peut attribuer séquentiellement des valeurs $r()$ aux sommets du graphe à l'aide de l'algorithme qui suit, ce qui conclura la démonstration. †

IV.2 Algorithme de calcul du rang

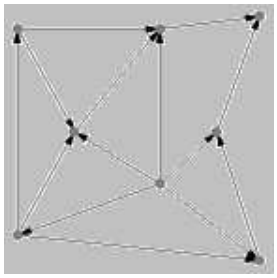
L'algorithme suivant permet de calculer $r(v)$ pour tout sommet v du digraphe. Il permet donc de savoir si un digraphe possède ou non un circuit.

Au début,
- r = 0,
- X est l'ensemble des sommets du digraphe.
- R est l'ensemble des sommets de X sans prédécesseurs dans X.

Tant que X n'est pas vide, faire
- r(v) := r pour tout sommet v de R,
- Enlever de X les sommets contenus dans R,
- Recalculer R comme ci-dessus,
- Incrémenter r.

IV.3 Exercice

Exercice 27.23. *Attribuez un rang aux sommets du digraphe ci-dessous en utilisant l'algorithme de calcul du rang :*



Fin du Chapitre

Chapitre 28

Problèmes de chemin

I Algorithme de Dijkstra

I.1 Présentation

Edgser Wybe Dijkstra (1930-2002) a proposé en 1959 un algorithme qui permet de calculer le plus court chemin entre un sommet particulier et tous les autres.

Le résultat est une arborescence.

I.2 L'algorithme

1. Numérotons les sommets du graphe $G = (V, E)$ de 1 à n.
2. Supposons que l'on s'intéresse aux chemins partant du sommet 1.
3. On construit un vecteur $l = (l(1); l(2); ...; l(n))$ ayant n composantes tel que $l(j)$ soit égal à la longueur du plus court chemin allant de 1 au sommet j.

   On initialise ce vecteur à $c_{1,j}$, c'est-à-dire à la première ligne de la matrice des coûts du graphe, définie comme indiqué ci-dessous :

$$\begin{cases} 0 & \text{si } i = j \\ +\infty & \text{si } i \neq j \text{ et } (i,j) \notin E \\ \delta(i,j) & \text{si } i \neq j \text{ et } (i,j) \in E \end{cases}$$

   où $\delta(i, j)$ est le poids (la longueur) de l'arc (i, j). Les $c_{i,j}$ doivent être strictement positifs.
4. On construit un autre vecteur p pour mémoriser le chemin pour aller du sommet 1 au sommet voulu.

   La valeur $p(i)$ donne le sommet qui précède i dans le chemin.
5. On considère ensuite deux ensembles de sommets, S initialisé à $\{1\}$ et T initialisé à $\{2, 3, ..., n\}$.

   À chaque pas de l'algorithme, on ajoute à S un sommet jusqu'à ce que $S = V$ de telle sorte que le vecteur l donne à chaque étape le coût minimal des chemins de 1 aux sommets de S.

I.3 Description de l'algorithme de Dijkstra

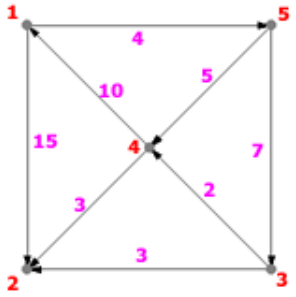
On suppose ici que le sommet de départ (qui sera la racine de l'arborescence) est le sommet 1. Notons qu'on peut toujours renuméroter les sommets pour que ce soit le cas.

Initialisations : – $l(j) = c1, jetp(j) = NIL, pour 1 \leqslant j \leqslant n$
– Pour $2 \leqslant j \leqslant n$ faire : Si $c_{1,j} < +\infty$ alors $p(j) = 1$.
– $S = \{1\}$; $T = \{2, 3, ..., n\}$.

Itérations : Tant que T n'est pas vide faire :

– Choisir i dans T tel que $l(i)$ est minimum
– Retirer i de T et l'ajouter à S
– Pour chaque successeur j de i, avec j dans T, faire : Si $l(j) > l(i) + d(i,j)$ alors
  – $l(j) = l(i) + d(i,j)$
  – $p(j) = i$

I.4 Exemple



Initialisations – $S = \{1\}$;
– $T = \{2,3,4,5\}$;
– $l = (0, 15, \infty, \infty, 4)$;
– $p = (NIL, 1, NIL, NIL, 1)$.

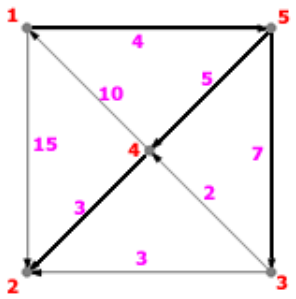
Première itération – $i = 5$ car $l(5) = min(15, \infty, \infty, 4) = 4$;
– $S = \{1,5\}$; $T = \{2,3,4\}$;
– les successeurs de 5 dans T sont 3 et 4 ;
– $l(3)$ prend la nouvelle valeur $min(\infty; l(5) + d(5;3)) = min(\infty; 4+7) = 11$; $p(3) = 5$;
– $l(4)$ prend la nouvelle valeur $min(\infty; l(5) + d(5;4)) = 9$; $p(4) = 5$;
– d'où les nouveaux vecteurs $l = (0, 15, 11, 9, 4)$ et p = $(NIL, 1, 5, 5, 1)$

Deuxième itération – $i = 4$; $l(4) = 9$;
– $S = \{1,5,4\}$; $T = \{2,3\}$;
– le seul successeur de 4 dans T est 2 ;
– $l(2)$ prend la nouvelle valeur $min(\infty; l(4) + d(4;2)) = min(15; 9+3) = 12$; $p(2) = 4$;
– d'où les nouveaux vecteurs $l = (0, 12, 11, 9, 4)$ et $p = (NIL, 4, 5, 5, 1)$

Troisième itération – $i = 3$; $l(3) = 11$;
– $S = \{1,5,4,3\}$; $T = \{2\}$;
– le seul successeur de 3 dans T est 2 ;
– $l(2)$ garde sa valeur car $min(12; l(3) + d(3;2)) = min(12; 11+3) = 12$;
– d'où les vecteurs inchangés $l = (0, 12, 11, 9, 4)$ et $p = (NIL, 4, 5, 5, 1)$

Quatrième itération – $i = 2$; $l(2) = 12$;
– $S = \{1,5,4,3,2\}$; $T = \{\}$; FIN.
– $l = (0, 12, 11, 9, 4)$;
– $p = (NIL, 4, 5, 5, 1)$.

Le chemin minimal de 1 à 4 par exemple est de coût 9. C'est le chemin 1-5-4, car $p(4) = 5$ et $p(5) = 1$.



I.5 Exercices

Exercice 28.1. *Appliquez l'algorithme de Dijkstra au graphe de l'exemple ci-dessus pour trouver tous les plus courts chemins en partant des sommets 2, 3, 4 et 5.*

Exercice 28.2. *Expliquez pourquoi des arcs avec des poids négatifs pourraient poser problème dans la recherche d'un plus court chemin dans un graphe.*

II Méthode PERT

II.1 Présentation de la méthode

Le problème du plus long chemin dans les digraphes sans circuits trouve une application dans l'ordonnancement et la planification des tâches composant un projet complexe, par exemple la construction d'une maison.

On fait correspondre à chaque tâche un arc d'un digraphe, sa durée d'exécution étant égale au poids de cet arc.

Le digraphe reflète les précédences requises dans l'exécution du projet. Ainsi, la tâche correspondant à l'arc (i, j) ne peut commencer que si toutes les tâches correspondant à des arcs (k, i) ont été complétées. Le digraphe peut contenir des tâches fictives de durée nulle afin de forcer certaines précédences.

Les sommets du digraphe représentent des événements, début (fin) des activités correspondant aux arcs dont ils sont l'extrémité initiale (finale). Le fait que le digraphe est sans circuit est garant de la faisabilité du projet. En effet, l'existence d'un circuit impliquerait une contradiction dans les précédences : une tâche devant en même temps précéder et succéder une autre !

On supposera dorénavant que les sommets ont déjà été numérotés de 1 à n de manière compatible avec leurs rangs, c'est-à-dire que $r(j) > r(i)$ implique $j > i$ (voir l'algorithme de calcul du rang).

En plus, si le digraphe possède plusieurs sommets sans prédécesseurs, on supposera avoir introduit un sommet 1 relié par un arc de durée nulle à chacun de ces sommets. Ce sommet indique le début du projet.

De même, si le digraphe possède plusieurs sommets sans successeurs, ceux-ci seront reliés par un arc de durée nulle à un dernier sommet n (fin du projet).

Enfin, on supposera éliminés les arcs parallèles par l'introduction de tâches fictives.

II.2 Algorithme du chemin critique

Données : Digraphe $G = (V, E)$, sans circuits, des activités avec leur durée d_{ik}.

Résultat :
- d_i début au plus tôt des activités correspondant aux arcs (i, k) partant de i,
- j_i fin au plus tard des activités correspondant aux arcs (k, i) arrivant à i,
- durée du chemin critique.

Début :
1. Calcul des dates de début au plus tôt (récurrence en avançant dans le projet)
   - $d_1 := 0$
   - Pour $k := 2$ à n faire $d_k := max\{d_j + d_{jk} | j \in P(k)\}$

2. Calcul des dates de fin au plus tard (récurrence en reculant dans le projet)
   - $j_n := d_n$
   - Pour $k := n-1$ à 1 faire $j_k := min\{j_j - d_{kj} | j \in S(k)\}$

Fin.

NOTATION : $P(i) = \{k \in V | (k,i) \in E\}$ est l'ensemble des sommets prédécesseurs de i.

NOTATION : $S(i) = \{k \in V | (i,k) \in E\}$ est l'ensemble des sommets successeurs de i.

II.3 Définitions

DÉFINITION 28.1 (SOMMET CRITIQUE). *Un sommet i est* critique *si $d_i = j_i$.* ♢

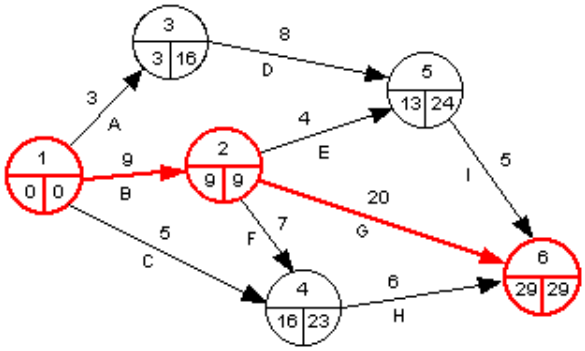
DÉFINITION 28.2 (ARC CRITIQUE). *Un arc (i,j) est* critique *si ses extrémités sont des sommets critiques et $d_{ij} = d_j - d_i$.* ♢

DÉFINITION 28.3 (CHEMIN CRITIQUE). *Un chemin* critique *est un chemin de 1 à n n'utilisant que des arcs critiques, c'est-à-dire des activités telles que tout retard dans leur exécution provoquerait un retard de la fin du projet.* ♢

DÉFINITION 28.4 (DURÉE DU CHEMIN CRITIQUE). *La* durée *du chemin critique est donnée par d_n (ou par j_n, les deux valeurs étant toujours égales). Elle correspond à la durée minimale du projet étant données les durées des tâches le composant et les précédences respectives.* ♢

II.4 Exemple

Ci-dessous le graphe des précédences obtenu avec l'algorithme du chemin critique. Les sommets et les arcs critiques sont en rouge.



Tâches	Précédences	Durée [jours]
A	-	3
B	-	9
C	-	5
D	A	8
E	B	4
F	B	7
G	B	20
H	C, F	6
I	D, E	5

II.5 Exercices

Exercice 28.3. *Refaites le graphe des précédences de l'exemple en utilisant l'algorithme du chemin critique.*

Exercice 28.4. *La rénovation du séjour d'un appartement se décompose en plusieurs tâches décrites dans le tableau ci-dessous. Ce dernier donne également les précédences à respecter lors de la planification des travaux ainsi qu'une estimation de la durée de chacune des tâches.*

	Tâches	*Précédences*	*Durée [jours]*
A	*Enlèvement des portes*	-	*1/2*
B	*Ponçage et peinture des portes*	*A*	*3*
C	*Pose des portes*	*B, J*	*1/2*
D	*Arrachage des papiers peints*	-	*1*
E	*Tirage des fils électriques*	*D*	*1*
F	*Pose des prises*	*E, H, I*	*1/2*
G	*Ragréage des murs*	*E, A*	*2*
H	*Peinture du plafond*	*G*	*2*
I	*Pose des papiers peints*	*G*	*3*
J	*Peinture des cadres*	*H, I*	*1*
K	*Arrachage de la moquette*	*H, I, J*	*1/2*
L	*Ponçage du parquet*	*K*	*1*
M	*Imprégnation et séchage du parquet*	*L, F*	*4*
N	*Peinture du balcon*	-	*2*
O	*Changement des protections solaires*	*N*	*1*

1. *Représentez le graphe des précédences de ces travaux de rénovation.*
2. *Déterminez une durée totale minimale de rénovation en exhibant un chemin critique dans le graphe précédent.*

Fin du Chapitre

Chapitre 29

Chaînes de Markov

I Généralités

I.1 Présentation

Généralement, un processus stochastique est une suite d'expériences dont le résultat dépend du hasard.

Ici, nous admettrons qu'en certains temps 0, 1, 2, ..., t, nous observons un système. Celui-ci peut se trouver dans l'un des états d'une collection finie d'états possibles. L'observation du système est ainsi considérée comme une expérience dont le résultat (aléatoire) est l'état dans lequel se trouve le système.

Nous supposons que nous connaissons pour chaque paire d'états i et j, et pour chaque instant t, la probabilité $p_{ij}(t)$ que le processus soit dans l'état j à l'instant $t+1$ étant donné qu'il se trouve dans l'état i à l'instant t. De plus, la probabilité $p_{ij}(t)$ sera supposée ne pas dépendre de t.

I.2 Définitions

DÉFINITION 29.1 (CHAÎNE DE MARKOV). *Un tel processus est appelé* chaîne de Markov *(à temps discret et avec un ensemble fini d'états), du nom de son inventeur Andrei Andreyevich Markov (1856-1922).* ♢

Avec ces hypothèses, nous pouvons décrire le système en donnant l'ensemble $\{u_1, ..., u_m\}$ des états u_i possibles et une matrice P de dimensions $m \times m$ dont le terme p_{ij} est la probabilité que le processus soit dans l'état j à l'instant $t+1$ étant donné qu'il se trouve dans l'état i à l'instant t, pour tout t.

DÉFINITION 29.2 (MATRICE DE TRANSITION). *P est appelée* matrice de transition *du système.* ♢

On représente généralement P par un graphe orienté G dont les sommets correspondent aux m états et les arcs aux couples ordonnés d'états (i,j) tels que pij>0.

I.3 Exemple

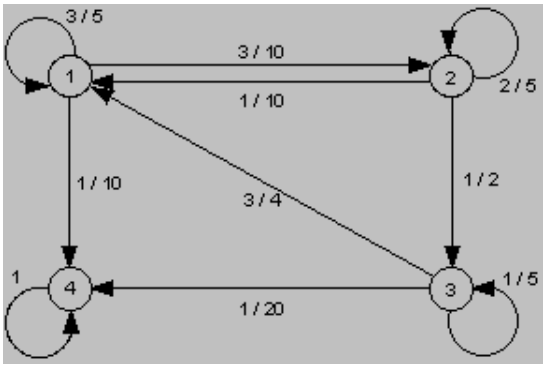
Pour représenter le passage d'une molécule de phosphore dans un écosystème, nous considérerons quatre états possibles :

1. la molécule est dans le sol,
2. la molécule est dans l'herbe,
3. la molécule a été absorbée par du bétail,
4. la molécule est sortie de l'écosystème.

La matrice de transition est la suivante :

$$P = \begin{pmatrix} \frac{3}{5} & \frac{3}{10} & 0 & \frac{1}{10} \\ \frac{1}{10} & \frac{2}{5} & \frac{1}{2} & 0 \\ \frac{3}{4} & 0 & \frac{1}{5} & \frac{1}{20} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Remarquez que la somme de chaque ligne vaut 1. Cette matrice correspond au graphe ci-dessous :



I.4 Propriétés

PROPRIÉTÉ 29.1 : La probabilité $p_{ij}(t)$ que le système soit dans l'état j au temps t sachant qu'il était dans l'état i au temps 0 est donné par $(P^t)_{i,j}$ (le terme i, j de la t-ième puissance de P).

Si on ne connaît pas l'état initial, on peut donner un vecteur de probabilité $p(0) = (p_1(0), ..., p_m(0))$ où $p_i(0)$ est la probabilité que le système se trouve dans l'état i au temps 0. Si $p(t)$ est le vecteur donnant les probabilités d'occupation des états au temps t (autrement dit la distribution), nous avons :

PROPRIÉTÉ 29.2 : $p(t) = p(0)P^t$

I.5 Exercice

Exercice 29.1. *Un individu vit dans un milieu où il est susceptible d'attraper une maladie par piqûre d'insecte. Il peut être dans l'un des trois états suivants : immunisé (I), malade (M), non malade et non immunisé (S). D'un mois à l'autre, son état peut changer selon les règles suivantes :*

- *étant immunisé, il peut le rester avec une probabilité 0,9 ou passer à l'état S avec une probabilié 0,1 ;*
- *étant dans l'état S, il peut le rester avec une probabilité 0,5 ou passer à l'état M avec une probabilité 0,5 ;*
- *étant malade, il peut le rester avec une probabilité 0,2 ou passer à l'état I avec une probabilité 0,8.*

Tracez un graphe probabiliste pour décrire cette situation et écrivez la matrice de transition. Calculez l'état de probabilité de l'individu au bout de trois mois, de six mois, d'un an, de deux ans, pour chacune des situations suivantes :

– *au départ, il est immunisé (I) ;*
– *au départ, il est non malade et non immunisé (S) ;*
– *au départ, il est malade (M).*
Pouvez-vous donner des éléments sur la proportion d'individus malades dans la population étudiée ?

II Distribution limite

II.1 Présentation

On constate souvent que la distribution $p(t)$ converge vers une distribution limite p si $t \to \infty$.

DÉFINITION 29.3. *Si tel est le cas, on dit que cette dernière définit un* régime permanent *du processus stochastique.* ♢

REMARQUE 29.1. Le régime permanent n'est pas influencé par le choix de la distribution initiale.

II.2 Existence d'une distribution limite

PROPRIÉTÉ 29.3 : Si la matrice de transition P est telle qu'une au moins de ses puissances n'a que des termes strictement positifs, alors $p(t) \longrightarrow p$ quelle que soit la distribution initiale $p(0)$ et $P^t \longrightarrow P^*$ lorsque $t \longrightarrow \infty$.

p est un vecteur de probabilité strictement positif et $P*$ une matrice dont toutes les lignes sont identiques au vecteur limite p. En plus, $pP* = p$.

REMARQUE 29.2. La démonstration de cette condition d'existence dépasse le cadre de ce cours.

II.3 Exercices

Exercice 29.2. *Soit la matrice stochastique*

$$P = \begin{pmatrix} 0,5 & 0,5 & 0 \\ 0,5 & 0 & 0,5 \\ 0 & 1 & 0 \end{pmatrix}$$

Montrez que la chaîne de Markov définie par P converge et calculez la distribution limite.

Exercice 29.3. *Soit la matrice stochastique*

$$P = \begin{pmatrix} \frac{3}{5} & \frac{3}{10} & 0 & \frac{1}{10} \\ \frac{1}{10} & \frac{2}{5} & \frac{1}{2} & 0 \\ \frac{3}{4} & 0 & \frac{1}{5} & \frac{1}{20} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Montrez que la chaîne de Markov définie par P converge et calculez la distribution limite.

Exercice 29.4. *Un service de météo a constaté après de longues années que le temps qu'il fera demain dépend essentiellement du temps qu'il faisait hier et du temps qu'il fait aujourd'hui. Les probabilités de transition ont été établies ainsi :*

Hier	*Aujourd'hui*	*Beau demain*	*Mauvais demain*
Beau	*Beau*	*0.8*	*0.2*
Beau	*Mauvais*	*0.4*	*0.6*
Mauvais	*Beau*	*0.6*	*0.4*
Mauvais	*Mauvais*	*0.1*	*0.9*

– *Modélisez ce processus à l'aide d'une chaîne de Markov.*
– *Calculez le nombre moyen de jours de beau temps par année.*

Exercice 29.5. *Un ivrogne se déplace dans les quatre bistrots d'un village, d'une manière bien personnelle : en sortant d'un bistrot, il lance une pièce de monnaie pour savoir dans lequel des deux autres bistrots les plus proches il entrera.*

Ces quatre bistrots forment les sommets d'un carré.

1. *Modélisez ce processus à l'aide d'une chaîne de Markov.*
2. *Montrez que cette chaîne de Markov n'a pas de distribution limite.*

III Chaîne absorbante

III.1 Généralités

III.1.1 Définitions

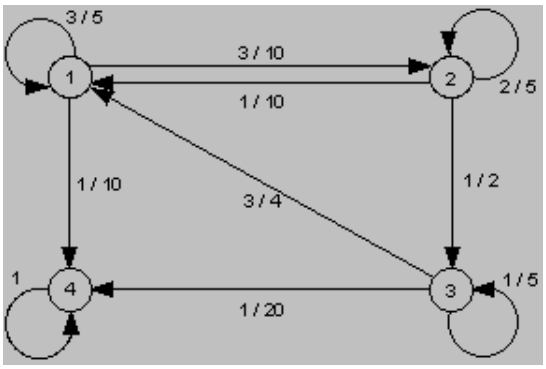
DÉFINITION 29.4 (ÉTAT ABSORBANT). *Un* état absorbant *est un état que l'on ne quitte plus lorsqu'on y pénètre.* ♢

REMARQUE 29.3. Autrement dit, l'état j est absorbant si $p_{jj} = 1$.

DÉFINITION 29.5 (CHAÎNE DE MARKOV ABSORBANTE). *Une chaîne de Markov est absorbante si et seulement si :*

1. *il y a au moins un état absorbant*
2. *de tout état non absorbant, on peut atteindre un état absorbant.* ♢

EXEMPLE 29.6. Par exemple, l'état 4 ci-dessous...



...est un état absorbant. Comme on peut atteindre cet état depuis tous les autres, la chaîne de Markov est absorbante.

III.1.2 Propriété

PROPRIÉTÉ 29.4 : Pour toute chaîne de Markov absorbante et pour tout état de départ, la probabilité de se trouver dans un état absorbant au temps t tend vers 1 lorsque t tend vers l'infini.

III.2 Délais d'absorption et probabilité d'absorption

III.2.1 Présentation

Lorsque l'on a affaire à une chaîne de Markov absorbante, on est généralement intéressé par les deux questions suivantes :

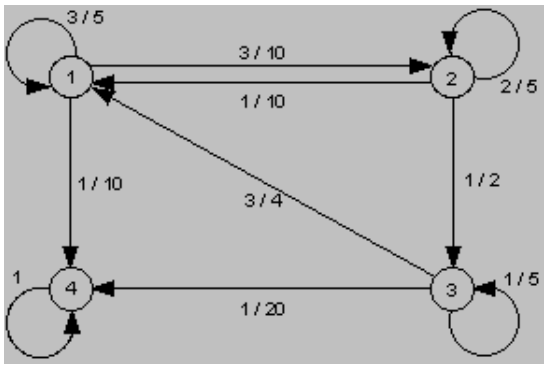
- Combien de temps faudra-t-il en moyenne pour arriver dans un état absorbant, étant donné son état initial ?
- S'il existe plusieurs états absorbants, quelle est la probabilité de tomber dans un état absorbant donné ?

III.2.2 Forme canonique de P

Si une chaîne de Markov est absorbante, on placera au début les états absorbants ; on aura alors une matrice de transition de la forme :

$$\left(\begin{array}{c|c} I & O \\ \hline R & Q \end{array}\right)$$

I est une matrice unité et 0 une matrice de 0.
Dans l'exemple du phospore vu précédemment,



nous avons (l'ordre des états est 4-1-2-3) :

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{10} & \frac{3}{5} & \frac{3}{10} & 0 \\ 0 & \frac{1}{10} & \frac{2}{5} & \frac{1}{2} \\ \frac{1}{20} & \frac{3}{4} & 0 & \frac{1}{5} \end{pmatrix}$$

III.2.3 Matrice fondamentale

DÉFINITION 29.6 (MATRICE FONDAMENTALE DE LA CHAÎNE ABSORBANTE). *La matrice* $N = (I - Q)^{-1}$ *est appelée la* matrice fondamentale de la chaîne absorbante. ♢

III.2.4 Résultats

PROPRIÉTÉ 29.5 : Le nombre moyen e_{ij} de passages à l'état j (non absorbant) avant l'absorption quand on part de l'état i (non absorbant) est donnée par $e_{ij} = (N)_{ij}$.

PROPRIÉTÉ 29.6 : Le nombre moyen d'étapes avant absorption sachant que l'on part de l'état i (non absorbant) est la somme des termes de la i-ème ligne de N.

EXEMPLE 29.7. Toujours dans l'exemple du phosphore, on a :

$$Q = \begin{pmatrix} \frac{3}{5} & \frac{3}{10} & 0 \\ \frac{1}{10} & \frac{2}{5} & \frac{1}{2} \\ \frac{3}{4} & 0 & \frac{1}{5} \end{pmatrix}, I-Q = \begin{pmatrix} \frac{2}{5} & \frac{-3}{10} & 0 \\ -\frac{1}{10} & \frac{3}{5} & -\frac{1}{2} \\ -\frac{3}{4} & 0 & \frac{4}{5} \end{pmatrix}, \text{ d'où } N = (I-Q)-1 = \begin{pmatrix} \frac{320}{37} & \frac{160}{37} & \frac{100}{37} \\ \frac{910}{111} & \frac{640}{111} & \frac{400}{111} \\ \frac{300}{37} & \frac{150}{37} & \frac{140}{37} \end{pmatrix}$$

D'où le nombre moyen d'étapes avant absorption en partant de l'état 1 : (320 + 160 + 100) / 37 = 15.67

PROPRIÉTÉ 29.7 : Dans une chaîne de Markov absorbante avec P mise sous forme canonique, le terme b_{ij} de la matrice $B = NR$ est la probabilité d'absorption par l'état absorbant j sachant que l'on part de l'état i.

EXEMPLE 29.8. Dans l'exemple du phosphore, on a :

$$R = \begin{pmatrix} \frac{1}{10} \\ 0 \\ \frac{1}{20} \end{pmatrix} \text{ d'où } B = NR = \begin{pmatrix} \frac{320}{37} & \frac{160}{37} & \frac{100}{37} \\ \frac{910}{111} & \frac{640}{111} & \frac{400}{111} \\ \frac{300}{37} & \frac{150}{37} & \frac{140}{37} \end{pmatrix} \begin{pmatrix} \frac{1}{10} \\ 0 \\ \frac{1}{20} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

La probabilité d'être absorbé par l'unique état absorbant est 1 quel que soit l'état initial !

III.3 Exercices

Exercice 29.9. *Considérons un joueur qui possède 2 francs initialement. À chaque étape du jeu il peut gagner 1 franc avec probabilité p ou perdre 1 franc avec probabilité 1-p. Il s'arrête lorsqu'il a gagné 4 francs ou lorsqu'il a tout perdu.*

1. *Représentez cette expérience par une chaîne de Markov.*
2. *Avec p = 1/3, calculez la probabilité de la ruine du joueur.*
3. *Avec p = 1/3, calculez la longueur moyenne d'une partie.*

Exercice 29.10. *Monsieur X se rend au Salon du Livre de Rigoleville dans l'espoir de trouver enfin un exemplaire du livre de Stendhal Le Rose et le Vert. Le Salon compte cinq stands et les organisateurs se sont amusés aux cours des années précédentes à construire la matrice des probabilités de transition des visteurs d'un stand à un autre :*

de à	*Stand 1*	*Stand 2*	*Stand 3*	*Stand 4*	*Stand 5*
Stand 1	*0*	*0,8*	*0*	*0*	*0,2*
Stand 2	*0,2*	*0*	*0,5*	*0,3*	*0*
Stand 3	*0*	*0*	*0*	*0,6*	*0,4*
Stand 4	*0,9*	*0*	*0,1*	*0*	*0*
Stand 5	*0,8*	*0*	*0*	*0,2*	*0*

Sachant que seuls les stands 4 et 5 disposent du livre recherché et que Monsieur X commence par visiter le stand 1, quelle est la probabilité qu'il achète son livre au stand 4 plutôt qu'au stand 5 (Monsieur X achètera le premier exemplaire qu'il trouvera) ?

Exercice 29.11. *Considérons une série de jets d'une pièce de monnaie normale. On notera P pour pile et F pour face.*

1. *Quelle est la probabilité d'obtenir une séquence PFP avant une séquence PPP ?*
2. *Quel est le nombre de jets nécessaires en moyenne pour réaliser l'une des deux séquences PFP et PPP ?*

Exercice 29.12 (La parade nuptiale des bourdons). *Une séance d'accouplement peut se décomposer en 7 phases :*

Départ (D) : *mise en contact des bourdons mâle et des reines.*

Approche (App) : *un mâle se dirige vers la reine. Il s'approche à courte distance. Il est le comportement le plus fréquent et souvent suivi d'une récidive.*

Inspection de la femelle (IF) : *le mâle suit la reine avec ses antennes tendues vers elle. Il inspecte souvent la reine au niveau de la tête (région où se trouvent les glandes produisant les phéromones sexuelles), mais parfois au niveau de l'abdomen.*

Tentative d'accouplement (T) : *le mâle s'approche de la reine, il s'accroche à elle. Il frotte de ses pattes antérieures l'extrémité de l'abdomen de la femelle. Il sort ses génitalias (appareil reproducteur) et tente de pénétrer la reine.*

Accouplement (Acc) : *lors de l'accouplement, le comportement du mâle se caractérise par des mouvements de battements des pattes sur l'extrémité de l'abdomen de la reine.*

Sortie par abandon du mâle (SA) : *lors de la séquence de 15 minutes, le bourdon mâle peut adopter un comportement indifférent vis-à-vis de la reine ; il sort de la parade nuptiale et n'y revient jamais.*

Sortie pour dépassement du temps (ST) : *l'observation est limitée à 15 minutes. Après cette durée, la probabilité d'accouplement peut être considérée comme presque nulle.*

Voici les statistiques obtenues pour 78 séances d'accouplement en laboratoire : par exemple App suivi de T = 202...

	App	*T*	*IF*	*Acc*	*ST*	*SA*	*Total*
D	*78*	*0*	*0*	*0*	*0*	*0*	*78*
App	*614*	*202*	*87*	*0*	*16*	*8*	*927*
IF	*83*	*0*	*0*	*0*	*3*	*1*	*87*
T	*152*	*0*	*0*	*35*	*7*	*8*	*202*

1. *Dessinez le graphe de transitions d'une parade nuptiale de bourdons.*
2. *Calculez les probabilités de transition d'un état à un autre et ajoutez-les au graphe.*
3. *Donnez la matrice correspondante de la chaîne de Markov.*
4. *Adaptez votre programme simulant une chaîne de Markov (voir exercice 1 sur l'introduction aux chaînes de Markov) à la situation présente. Utilisez ce programme pour simuler une parade nuptiale de bourdons.*
5. *Cette chaîne de Markov est une chaîne absorbante. Quel est le nombre moyen d'étapes avant absorption ? Trouvez le résultat théoriquement et par simulation.*

Fin du Chapitre

Sixième partie

Annexes

Chapitre 30

Programme Pédagogique National 2005 (PPN)

Voici le contenu de l'Unité de Formation Mathématiques Discrètes (TC-CCG-MATH1) du PPN actuel :

Volume horaire : 70 h

Pré-requis : aucun.

Objectifs :
- Connaître le calcul booléen.
- Calculer dans Z/nZ.
- Connaître les notions de base en théorie des graphes, des langages et des automates.

Compétences minimales :
- Mettre en œuvre des schémas de raisonnement (contraposée, absurde, récurrence, etc.).
- Mettre en œuvre des algorithmes d'arithmétique (Euclide, Bézout, etc.).
- Faire le lien entre langage usuel et langage formalisé (propositions et prédicats).

Contenu :
- Vocabulaire de la théorie des ensembles, relations, ensembles ordonnés.
- Logique : calcul propositionnel et calcul des prédicats.
- Arithmétique : nombres premiers, division euclidienne, congruences.
- Éléments de théorie des graphes : graphes orientés et non orientés.
- Éléments de langages et d'automates.

Indications de mise en œuvre : Exemples d'algorithmes de plus courts chemins, de parcours et d'arbre couvrant de poids minimum.

Prolongements possibles :
- Exemples de raisonnement par récurrence (en liaison avec les enseignements d'algorithmique).
- Développement des liens avec les enseignements d'informatique, en particulier « Architectures, Systèmes et Réseaux » et « Outils et Modèles du Génie logiciel » (algèbre relationnelle, etc.).
- Chaînage avant et chaînage arrière.
- Résolution d'équations en nombres entiers.
- Cryptographie (RSA, méthode du « sac à dos », etc.).
- Codes correcteurs et codes détecteurs d'erreurs.

Bibliographie

[CL93] René Cori and Daniel Lascar. *Logique mathématique, cours et exercices*. Masson, 1993.

[Dow07] Gilles Dowek. *Les Métamorphoses du calcul, une étonnante histoire des mathématiques*. Éditions le Pommier, 2007.

[LBDG07] Thierry Lucas, Isabelle Berlanger, and Isabelle De Greef. *Initiation à la logique formelle*. Éditions De BOECK Université, 2007.

[Lip90] Seymour Lipschutz. *Mathématiques discrètes*. McGraw-Hill, 1990.

[McC64] J. McCarthy. A basis for a mathematical theory of computation. In P. Brafford and D. Hirschberg, editors, *Computer Programming and Formal Systems*, pages 33–70. Amsterdam : North-Holland, 1964.

[Smu98] Rayomond Smullyan. *Ça y est, je suis devenu fou ! !* Dunod, 1998.

Bibliographie (suite)

Outils mathématiques pour l'informaticien, ***Michel Marchand*** **[De Boeck**] : les thèmes abordés sont proches de ce cours de mathématiques discrètes (notre première année y est plus détaillée que la partie concernant automates, graphes et langages).

On y trouve des exercices, corrigés, parfois repris dans ce support. Cependant, le formalisme choisi s'éloigne par moment de celui adopté dans notre document, ce qui pourrait en destabiliser certains.

Méthodes mathématiques pour l'informatique, ***Jacques Vélu*** **[Dunod**] : Reprend une grande partie du cours de mathématiques discrètes, et y ajoute un peu de probabilités et d'algèbres linéaires. Contient des exercices corrigés. Un peu dense par moment, mais une bonne référence quand même.

Le magazine Tangente : que l'on trouve chaque mois chez les bons marchants de journaux. Niveau lycée et plus. On y trouve fréquemment des articles sur les graphes, la logique, le cryptage, etc. Ludique et plaisant, pour ceux qui aiment les mathématiques, veulent les découvrir. Les hors-séries sur la logique, sur les codes secrets, etc. me servent à trouver des idées de TP, ou à enrichir le cours.

Introduction à la logique, ***François Rivenc*** **[Petite Bibliothèque Payot**] : Pour un public avertis, souhaitant étudier plus systématiquement la logique, sous ses aspects mathématiques et philosophiques.

http ://www.apprendre-en-ligne.net/ Site de Didier Müller, sur lequel j'ai repris la partie graphes. A noter une autre partie (très riche, bien fournie) consacrée à la cryptographie, et un blog très intéressant sur les mathématiques.