Mathématiques pour l'informatique

Christophe GUYEUX
guyeux@iut-bm.univ-fcomte.fr

21 avril 2008

Table des matières

Remerciements

Je tiens à remercier vivement *Michel Bour*, qui m'a très amicalement laissé ses supports de cours. Ce document s'en inspire très largement, au point d'en être, fréquemment, qu'une remise en forme. Sans lui, ce support de cours n'existerait pas.

Je remercie encore *Jean-François Couchot* pour ses infatigables relectures, ses conseils toujours pertinents, ses corrections de forme ou de fond, et pour ses propositions d'exercices. Il contribue fortement à l'amélioration de ce document.

Première partie

Théorie des ensembles

Chapitre 1

Introduction à la théorie des ensembles

I. Rappels de théorie des ensembles

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. *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* }

Réponses : 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.

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

Exercice 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* $\}$

Réponses : A = {2, -7}, B = {2}, et C = {1, 2, 3, 4, 6, 7, 8, 9}.

2 Règles de fonctionnement

Relation d'appartenance Il faut être capable de décider si un objet est ou non élément de l'ensemble (symbole $\in$).

Objets distincts Il faut être capable de distinguer les éléments d'un ensemble entre eux. Un ensemble ne peut pas contenir deux fois le même objet.

Ensemble vide Ensemble ne contenant aucun élément[1] (symbole : le cercle barré[2] $\varnothing$).

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

REMARQUE 1. 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 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 2. Il ne faut pas négliger l'impact d'une telle révélation. Certains allèrent jusqu'à y voir la preuve de la nom-existence de Dieu.

[1] 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.

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

3 Sous-ensembles, ensemble des parties

Sous-ensemble Ils sont définis par la relation d'inclusion : « A sous-ensemble de B ($A \subset B$) » si et seulement si tout élément de A appartient à B. On dit alors que A est sous-ensemble, ou partie, de B.
- L'ensemble vide est inclus dans n'importe quel ensemble[3].
- Tout ensemble est inclus dans lui-même.

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

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

Comme $A \subset A$, $A \in \mathcal{P}(A)$.
De plus, en général, si A possède n éléments, $\mathcal{P}(A)$ en possède 2^n.

EXEMPLE 2. Si $A = \varnothing, \mathcal{P}(A) = \{\varnothing\}, \mathcal{P}(\mathcal{P}(a)) = \{\varnothing, \{\varnothing\}\}$.

---

4 Représentation graphique

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

Exercice 4 (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.*

[3] D'après la définition d'un sous-ensemble, cela veut dire que pour tout élément x de Ø, 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 Ø appartient à A et donc que Ø est un sous-ensemble de A. Plus généralement, toute proposition commençant par « pour tout élément de Ø » est vraie.

2. *Les docteurs sont des gens heureux.*
3. *Nul ne peut être à la fois docteur et poëte.*

---

5 Exercices

Exercice 5. *Est-ce que* $\{a\} \in \{a, b, c\}$ *? Former la liste des parties de* $\{a, b, c\}$.

Exercice 6. *Montrer que* $\mathcal{P}(A) \subset \mathcal{P}(B)$ *quand* $A \subset B$.

Exercice 7. *Quels sont les éléments de* $\mathcal{P}(\varnothing)$ *? Quels sont ceux de* $\mathcal{P}(\mathcal{P}(\varnothing))$ *?*

REMARQUE 3. 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.

Exercice 8. *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

1 Égalite de deux ensembles

DÉFINITION 1. *Deux ensembles sont* égaux *si et seulement si ils ont les mêmes éléments.*$\diamondsuit$

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

Exercice 9. *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. $A = \{x \in \mathbb{N}_{20} \mid x$ *impair, non divisible par 3* $\}$ *et* $B = \{x \in \mathbb{N}_{20} \mid$ *24 divise* $x^2 - 1\}$

---

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

PROPRIÉTÉ I (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 : $A \cup \varnothing = A$

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

PROPRIÉTÉ II (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 III (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 10. *Dans chacun des cas suivants, faire la réunion des ensembles A et B.*

1. $A = \{x \in \mathbb{N} | x$ *impair* $\}$, $B = \{x \in \mathbb{N} | x$ *pas divisible par 3* $\}$
2. $A = \{x \in \mathbb{R} | 0 \leqslant x \leqslant 3\}$, $B = \{x \in \mathbb{R} | -2 < x \leqslant 1\}$
3. $A = \{(x, y) \in \mathbb{R}^2 | x + y \leqslant 2\}$, $B = \{(x, y) \in \mathbb{R}^2 | 2 < 3x - y\}$

Exercice 11. *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. $A = \{x \in \mathbb{R} | 0 \leqslant x \leqslant 3\}$, $B = \{x \in \mathbb{R} | -2 < x \leqslant 1\}$
3. $A = \{(x, y) \in \mathbb{R}^2 | x + y \leqslant 2\}$, $B = \{(x, y) \in \mathbb{R}^2 | 2 < 3x - y\}$

Exercice 12. *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.*

---

3 Complémentation

DÉFINITION 2 (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 : On note : $E \setminus A$ (« E moins A »).

REMARQUE 4. 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 IV : La complémentation a plusieurs propriétés remarquables :
- involution : $E \setminus (E \setminus A) = A$,
- loi de De Morgan : $E \setminus (A \cup B) = (E \setminus A) \cap (E \setminus B)$, et $E \setminus (A \cap B) = (E \setminus A) \cup (E \setminus B)$.

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).

III. Exercices

Exercice 13 (Ensemble des parties). *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.*

1. $A \cap (A \cup B) = A \cup (A \cap B) = ?$
2. *Montrer que $(A \cup C) \subset (A \cup B)$ et $(A \cap C) \subset (A \cap B)$ implique que $C \subset B$. Montrer que, pour qu'il soit possible de conclure à l'égalité de B et de C, les deux propositions suivantes sont nécessairement réalisées : $A \cup B = A \cup C$ et $A \cap B = A \cap C$.*
3. *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 14 (La différence symétrique). *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.

1. *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 $(A \cap B)\Delta(A \cap D)$.
2. *Vérifiez que $A\Delta B = [A \cap (E \setminus B)] \cup [(E \setminus A) \cap B]$*
3. *Calculer $A\Delta A$, $A\Delta\bar{A}$, $A\Delta E$ et $E \setminus (A\triangle B)$.*
4. *Montrer que, si $A\triangle B = C$, alors $A\triangle C = B$ et $B\triangle C = A$.*
5. *Montrer que la différence ensembliste est commutative, possède un élément neutre, est distributive, et associative.*
6. *Montrer que si $A\Delta B = A\Delta C$ alors $B = C$.*

Exercice 15. *Soit E un ensemble.*

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

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

Exercice 16. *Soit E un ensemble et A, B, C des parties de E.*

Démontrer la proposition suivante :

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

Exercice 17. *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 18 (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 19. *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.*

Fin du Chapitre

Chapitre 2

Relations binaires entre ensembles

I. Définitions

On se donne deux ensembles E et F.

1 Définition

DÉFINITION 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 $x(\in E)$ et $y(\in F)$ sont tels que $(x, y) \in G$, on dit que x est* en relation avec *y par la relation $\mathcal{R}$.* $\diamondsuit$

On utilise, pour formaliser cette proposition, la notation $x\mathcal{R}y$. Donc :

NOTATION :

$x\mathcal{R}y \iff$ [« x est en relation avec y »] $\iff (x, y) \in G$ [G : graphe de la relation $\mathcal{R}$].

Exercice 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.*
- *$x\mathcal{R}y \iff y = x^2$.*

---

2 Remarques

REMARQUE 1. Lorsque $E = F$, on parle de relation binaire définie dans l'ensemble E. Son graphe est une partie de E^2.

REMARQUE 2. Il est possible que $x\mathcal{R}y$ sans que $y\mathcal{R}x$.

3 Exercices

Exercice 2. *Sur l'ensemble des mots de la langue française, on définit la relation : « le mot M est lié au mot N s'ils coïncident après qu'on ait retourné l'ordre des lettres de M ».*

Déterminer quelques couples de mots en relation, ainsi que des mots en relation avec eux-mêmes.

Exercice 3. *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 ?

Réponse : $x\Sigma y \Leftrightarrow x + y = 2k \Leftrightarrow x + y - 2y = 2k - 2y$.
Donc $x\Sigma y \Leftrightarrow x - y = 2(k - y) = 2k' \Leftrightarrow x\Delta y$.

II. Application d'un ensemble dans un autre

1 Définition d'une application, d'une relation fonctionnelle

Application.

DÉFINITION 2 (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 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 :
- $\forall x \in E,\ \exists y \in F,\ (x, y) \in G$
- $\forall x \in E,\ \forall y \in F,\ \forall y' \in F, [(x, y) \in G \text{ et } (x, y') \in G \Longrightarrow y = y']$.

Relation fonctionnelle. Il existe un intermédiaire entre relation et application...

DÉFINITION 3 (RELATION FONCTIONNELLE). *On parle de* relation fonctionnelle *quand tout élément de l'ensemble de départ possède au plus une image.* ♢

REMARQUE 3. Une application est donc une relation fonctionnelle particulière : tout élément de l'ensemble de départ possède exactement une image.

Exercice 4. *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. $\mathcal{R} = \{(x, y) \in \mathbb{R}^2, y - x + 2 = 0\}$

Réponse : Les deux dernières sont des relations fonctionnelles, et la dernière est la seule application.

2 Image et antécédent d'un élément

2.1 Image 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 4. *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 $y = f(x)$.

NOTATION : On formalise la proposition « f est une application de E dans F » par $f : E \to F$, et la proposition « y est l'image de x par f » peut aussi être traduite par : $f : x \longmapsto y$.

Exercice 5. *Interpréter chacune des situations suivantes au moyen d'une application. Pour cela, on définira deux ensembles A et B ainsi que $f : A \to B$.*

1. *Le résultat d'une course de tiercé.*
2. *Le registre d'un hôtel qui possède 55 chambres.*
3. *Le numéro d'INSEE.*
4. *La parité d'un entier naturel.*
5. *Un emploi du temps.*

Réciproquement...

2.2 Antécédent d'un élément

Soit f une application.

DÉFINITION 5 (ANTÉCÉDENT). *Si y est l'image de x par f, alors x est appelé* antécédent *de y par f.* ♢

3 Applications injectives

3.1 Définitions

PROPRIÉTÉ I (NOMBRE D'ANTÉCÉDENTS, DÉFINITION DE L'INJECTIVITÉ) : L'antécédent de y n'est pas nécessairement unique. S'il l'est, l'application f est dite *injective* .

REMARQUE 4. Le terme injection est synonyme d'« application injective ».

On peut caractériser les applications injectives de la manière suivante :

PROPRIÉTÉ II (CARACTÉRISATION DES FONCTIONS INJECTIVES) :

« f est (une application) injective » $\iff [f(x) = f(x') \implies x = x']$

3.2 Exercices

Exercice 6. *Tracez le graphe d'une application qui est injective, et d'une application qui ne l'est pas.*

Exercice 7. *Donnez des exemples (sous forme analytique) de fonctions injectives, et de fonction qui ne le sont pas.*

Exercice 8. *On suppose gof injective. Montrer que f est injective. Est-ce que g est obligatoirement injective ?*

---

4 Applications surjectives

4.1 Définition

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

REMARQUE 5. *Surjection* est synonyme d'« application surjective ».

Exercice 9. *Tracez le graphe d'une application qui est surjective, et d'une application qui ne l'est pas.*

Exercice 10. *Donnez des exemples (sous forme analytique) de fonctions surjectives, et de fonction qui ne le sont pas.*

4.2 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 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 6. 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 III (CARACTÉRISATION DE LA SURJECTIVITÉ) :

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

EXEMPLE 1. 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 11. *On suppose gof surjective. Montrer que g est surjective. Est-ce que f est obligatoirement surjective ?*

5 Applications bijectives

5.1 Définition

DÉFINITION 8 (APPLICATIONS BIJECTIVES). *Une application qui est à la fois injective et surjective est dite* bijective *.* ♢

REMARQUE 7. Synonyme d'« application bijective » : bijection.

Exercice 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 IV : 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.

5.2 Application inverse

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

NOTATION : On la note f^{-1}

Exercice 13. *Reprendre l'exercice précédent, en trouvant l'application réciproque des applications bijectives.*

REMARQUE 8. On peut démontrer que la bijectivité est une condition nécessaire et suffisante pour qu'une application admette une inverse.

EXEMPLE 2. Soit l'application « multiplication par 2 » $f : x \longmapsto 2x$ de $\mathbb{R}$ dans $\mathbb{R}$. Elle est :

- surjective,
- injective.

Elle admet donc une application inverse, à savoir : $f^{-1} : x \longmapsto x/2$.

Exercice 14. *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 $347 = n + (-1)^n$.*

III. Cardinal et puissance d'un ensemble

Il s'agit ici de proposer une réflexion sur la notion intuitive de « nombre d'éléments d'un ensemble » , nécessaire pour pouvoir aborder ultérieurement les notions de dénombrabilité et de calculabilité, fondamentaux en informatique.

1 Cas des ensembles finis

On commence par prendre deux ensembles $E = \{a, b, c, d\}$ et $F = \{1, 2, 3\}$ et à remarquer que :

- il est possible de définir une injection de F dans E, mais pas une surjection,
- de définir une surjection de E sur F, mais pas d'injection,

...tout simplement parce qu'il n'y a « pas assez » d'éléments dans F (ou « trop » dans E).

Si l'on veut pouvoir définir une bijection entre deux ensembles, il semble nécessaire et suffisant qu'ils aient le même « nombre d'éléments » (on se limite ici au domaine fini).

Cette notion intuitive, résultat immédiat d'une simple opération de comptage, est reliée à la notion mathématique de mise en bijection avec une partie de $\mathbb{N}^*$ de la forme $\{1\,,\,2\,,\,\ldots\,,\,n\}$.

Ainsi, pour $n \in \mathbb{N}^*$ donné, les ensembles à n éléments sont tous ceux qui peuvent être mis en bijection avec $\{1\,,\,2\,,\,\ldots\,,\,n\}$ (et, par convention, $\varnothing$ a 0 élément).

DÉFINITION 10 (CARDINAL D'UN ENSEMBLE FINI). *L'élément maximum n de la partie finie $P = \{1\,,\,2\,,\,\ldots\,,\,n\}$ de $\mathbb{N}^*$ est un représentant du* cardinal *de tout ensemble en bijection avec P.* ♢

Pas de problème donc lorsque l'on s'en tient aux ensembles finis, dont la définition mathématique est :

DÉFINITION 11 (ENSEMBLE FINI). *Un ensemble est dit fini s'il ne peut pas être mis en bijection avec une partie stricte de lui-même.* ♢

2 Cas des ensembles infinis

Lorsque l'on se pose la question du « nombre d'éléments » d'un ensemble infini, on peut être tenté, dans un premier temps, de répondre par « une infinité », ce qui n'est pas satisfaisant pour au moins deux raisons :

- jusqu'à présent, le « nombre d'éléments » était un entier, et l'infini n'est pas un nombre entier,
- cela laisserait supposer que tous les ensemble infinis ont le même « nombre d'éléments »

Une réflexion plus appronfondie apparaît comme nécessaire. On décide donc de se calquer sur le domaine fini...

DÉFINITION 12 (PUISSANCE D'UN ENSEMBLE INFINI). *Deux ensembles ont* même puissance *(« même nombre d'éléments ») si l'on peut les mettre en bijection.* ♢

EXEMPLE 3. Il existe des bijections entre $\mathbb{N}$ et l'ensemble des entiers pairs ($2\mathbb{N}$) ou impairs ($2\mathbb{N}+1$), qui conduisent à des formulations du style « il y a autant d'entiers que d'entiers pairs ».

REMARQUE 9. $(2\mathbb{N}) \cup (2\mathbb{N}+1) = \mathbb{N}$, bien que ces trois ensembles « on le même nombre d'éléments », les deux premiers étants disjoints... Á rapprocher de $\infty+\infty = \infty$.

REMARQUE 10. Pour ce genre de raisons, on a tendance à abandonner ce vocabulaire (sur les nombres d'individus) pour dire simplement que ces ensembles ont même puissance.

3 Nombre d'infinis

3.1 Notion de puissance d'un ensemble

Le résultat fondamental est...

PROPRIÉTÉ V : Il est impossible de définir une surjection de E sur $\mathcal{P}(E)$.

PREUVE Admis ■

Ce résultat montre que, même si E est un ensemble infini, il ne peut être mis en bijection avec $\mathcal{P}(E)$, qui a donc « strictement plus d'éléments » que E.

Conséquemment, $\mathbb{N}$ et ses parties infinies, $\mathbb{Z}$, et même $\mathbb{Q}$ ont tous la même puissance, dite *puissance du dénombrable* . Mais...

DÉFINITION 13 (PUISSANCE DU CONTINU). *$\mathcal{P}(\mathbb{N})$ n'est pas dénombrable, et est de puissance strictement supérieure, dite* puissance du continu. ♢

EXEMPLE 4. C'est la puissance de $\mathbb{R}$.

De même, $\mathcal{P}(\mathbb{R})$ est de puissance strictement supérieure à $\mathbb{R}$, et ainsi de suite.....

3.2 $\mathbb{R}$ est indénombrable : une démonstration de Cantor.

Supposons que $\mathbb{R}$ soit dénombrable.

On pourrait alors écrire la liste de TOUS les réels, à partir du premier, en écriture décimale r_1, r_2, r_3, etc.

REMARQUE 11. On choisit pour les nombres décimaux l'écriture qui ne comporte que des zéros à partir d'un certain rang, s'il y a ambiguïté.

EXEMPLE 5. Ainsi, l'on préférera l'écriture 1 à 0,9999999..... (ces nombres sont égaux ; c'est-à-dire que l'on a affaire ici à deux représentations décimales d'un même nombre).

Construisons alors le nombre S de la manière suivante :

1. sa partie entière est 0,
2. et pour sa partie décimale,
   - le premier chiffre est différent du premier chiffre après la virgule de r_1,
   - le deuxième chiffre est différent du deuxième chiffre après la virgule de r_2,
   - *etc.*

Le nombre S ne fait alors pas partie de la liste des réels, ce qui est contradictoire : l'hypothèse $\mathbb{R}$ dénombrable (on peut numéroter tous les réels avec $\mathbb{N}$) est erronée.

IV. Relations d'ordre

Dans ce paragraphe, on se place dans le cas où $E = F$.

1 Définition

Soit $\mathcal{R}$ une relation binaire définie dans un ensemble E, de graphe G.

1.1 Réflexivité, antisymétrie, transitivité

DÉFINITION 14 (RÉFLEXIVITÉ). *$\mathcal{R}$ est dite* réflexive *quand tout élément de E est en relation avec lui-même :*

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

REMARQUE 12. C'est-à-dire $\forall x \in E,\ x\mathcal{R}x$, ou encore : la diagonale de E^2 est incluse dans G.

DÉFINITION 15 (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 $x = y$) :*

$$\forall x \in E,\ \forall y \in E,\ (x, y) \in G \text{ et } (y, x) \in G \implies x = y$$

REMARQUE 13. C'est-à-dire $\forall (x, y) \in E^2,\ x\mathcal{R}y \text{ et } y\mathcal{R}x \implies x = y$.

DÉFINITION 16 (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 :*

$$\forall x \in E, \forall y \in E,\ \forall z \in E,\ (x, y) \in G \text{ et } (y, z) \in G \implies (x, z) \in G$$

REMARQUE 14. C'est-à-dire : $\forall x \in E,\ \forall y \in E, \forall z \in E,\ x\mathcal{R}y \text{ et } y\mathcal{R}z \implies x\mathcal{R}z$.

Exercice 15. *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 $sin^2x + cos^2y = 1$.*
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.*

1.2 Relation d'ordre

DÉFINITION 17 (RELATION D'ORDRE). *$\mathcal{R}$ est une* relation d'ordre *lorsqu'elle est réflexive, antisymétrique et transitive.* ♢

EXEMPLE 6 (EXEMPLES DE RELATIONS D'ORDRE). Quelques relations d'ordre :
- $(\mathbb{R}, \leqslant)$
- $(\mathcal{P}(E), \subset)$

EXEMPLE 7 (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*.

2 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.

Cela signifie que, si l'on choisit deux éléments x et y quelconques dans E, x est en relation avec y, ou y est en relation avec x : $\forall x \in E, \forall y \in E, (x, y) \in G \text{ ou } (y, x) \in G$.

DÉFINITION 18 (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 15. 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 19 (RELATION D'ORDRE PARTIEL). *Dans le cas contraire, il existe des éléments qui ne sont pas comparables : on parle alors d'*ordre partiel *.* ♢

EXEMPLE 8. $\leqslant$ est une relation d'ordre totale dans $\mathbb{R}$.

EXEMPLE 9. $\subset$ dans $\mathcal{P}(E)$, et $|$ dans $\mathbb{N}$ sont des relations d'ordre partiels.

---

3 Exercices

Exercice 16. *Parmi les relations suivantes sur l'ensemble E, repérez les relations d'ordre, les relations d'ordre total :*

1. $E = \mathbb{Z}$, $x\mathcal{R}y \Leftrightarrow \exists k \in \mathbb{N} : x = y^k$.
2. $E = \mathbb{N}$, $x\mathcal{R}y \Leftrightarrow x < y$.
3. $E = \mathbb{R}$, $x\mathcal{R}y \Leftrightarrow x^2 = y^2$.
4. $E = \mathbb{R}^2$, $(x_1, x_2)\mathcal{R}(y_1, y_2) \Leftrightarrow (x_1 \leqslant y_2) \wedge (x_2 \leqslant y_1)$.

Exercice 17. *Tenter de caractériser par son graphe une relation d'ordre (partiel total).*

Exercice 18. *Définir, par leurs graphes, les relations d'ordre dans E qui comportent respectivement le moins et le plus possible de points ; que peut-on dire de ces relations ?*

Exercice 19 (Relations d'ordre en Algèbre de Boole). *Soit $\mathcal{A}$ une algèbre de Boole. On considère la relation binaire, de symbole $<$, définie par*

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

1. *Montrer qu'il s'agit d'une relation d'ordre.*
2. *Montrer que $a < b \iff a \cdot b = a$.*
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, comparer avec les résultats précédents.*
5. *En utilisant l'une ou l'autre des définitions ci-dessus pour la relation d'ordre, trouver, lorsqu'ils existent, les éléments $\mathrm{Sup}\{a, b\}$ et $\mathrm{Inf}\{a, b\}$. Trouver $\mathrm{Max}\,\mathcal{A}$ et $\mathrm{Min}\,\mathcal{A}$.*

---

4 Éléments maximaux

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

DÉFINITION 20 (MAJORANT). *On appelle* majorant *de A tout élément M de E tel que, quel que soit $a \in A$, $a\mathcal{R}M$.* ♢

DÉFINITION 21 (PARTIE MAJORÉE). *La partie A de E est dite* majorée *s'il existe un majorant de A.* ♢

Exercice 20. *Trouvez des exemples de majorants et de parties majorées sur $\mathbb{N}$ et $\mathcal{R}$.*

REMARQUE 16. Il existe des parties non majorées ($\mathcal{R}^+$ pour $\leqslant$ dans $\mathbb{R}$)

REMARQUE 17. Il peut exister une infinité de majorants pour une partie majorée.

DÉFINITION 22 (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. ♢

Exercice 21. *Trouvez des exemples de minorants et de parties minorées sur $\mathbb{Z}$ et $\mathcal{Q}$.*

DÉFINITION 23 (ÉLÉMENT MAXIMUM). *On appelle* élément maximum *de A un élément de A qui est majorant de A.* ♢

Exercice 22. *Trouvez des exemples d'élément maximum sur $\mathbb{N}$ et $\mathcal{R}$.*

NOTATION : $\operatorname{Max} A$.

REMARQUE 18. Si A est non majorée, il est exclu qu'elle admette un élément maximum.

REMARQUE 19. 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 24 (ÉLÉMENT MINIMUM). *On appelle* élément minimum *de A un élément de A qui est minorant de A.* ♢

NOTATION : $\mathrm{Min}\, A$.

Exercice 23. *Etant donné $B = \{1, 2, 3, 4, 5\}$ ordonné selon la relation $4 < 2$, $5 < 2$, $5 < 3$, $2 < 1$, $3 < 1$. Trouver $\mathrm{Min}\, A$ et $\mathrm{Max}\, A$.*

DÉFINITION 25 (BORNE SUPÉRIEURE). *On appelle* borne supérieure *de A l'élément minimum, s'il existe, de l'ensemble des majorants de A.* ♢

NOTATION : $\mathrm{Sup}\, A$.

DÉFINITION 26 (BORNE INFÉRIEURE). *On appelle* borne inférieure *de A l'élément maximum de l'ensemble des minorants de A.* ♢

NOTATION : $\mathrm{Inf}\, A$.

Exercice 24. *Trouvez des exemples de bornes sup et de bornes inf sur $\mathcal{R}$.*

Exercice 25 (Une relation d'ordre). *On considère l'ensemble des points d'un plan affine euclidien, et on y définit une relation binaire (symbole $\leqslant$) par $P_1 \leqslant P_2 \iff (x_1 \leqslant x_2 \text{ et } y_1 \leqslant y_2)$.*

1. *Définir, lorsqu'ils existent, les points $\mathrm{Sup}\{P_1, P_2\}$ et $\mathrm{Inf}\{P_1, P_2\}$.*
2. *Existent-ils toujours, quels que soient les points P_1 et P_2 ?*

PROPRIÉTÉ VI : Il est clair que :
- dans certains cas, les éléments définis ici n'existent pas,
- que l'élément maximum est aussi borne supérieure.

Et finalement, pour une partie A d'un ensemble ordonné E :
- A peut ne pas être majorée.
- Si A est majorée, elle peut ne pas admettre de borne supérieure.
- Si $SupA$ existe (A est majorée), A peut ne pas admettre d'élément maximum.
- Si $MaxA$ existe, alors $SupA = MaxA$.

...et on a les mêmes résultats pour les parties minorées.

EXEMPLE 10. Cas de $E = \{x \in \mathbb{Q} \mid x \leqslant 32\}$.
- ensemble majoré de nombres réels
- 56 , 32 sont majorants
- ensemble E' des majorants : $E' = \{y \in \mathbb{Q} \mid y \geqslant 32\}$
- $\mathrm{Max}\, E = 32$
- $\mathrm{Min}\, E' = 32$, donc $\mathrm{Sup}\, E = 32$.

EXEMPLE 11. Cas de $E = \{x \in \mathbb{Q} \mid x < 32\}$.
- ensemble majoré de nombres réels
- 56, 32 sont majorants
- ensemble E' des majorants : $E' = \{y \in \mathbb{Q} \mid y \geqslant 32\}$
- E n'a pas d'élément maximum
- $\mathrm{Min}\, E' = 32$, donc $\mathrm{Sup}\, E = 32$.

---

5 Treillis

5.1 Cas des ensembles totalement ordonnés

Dans un ensemble totalement ordonné, si l'on choisit une paire d'éléments quelconques (x, y) il est possible de décider lequel est le plus petit et lequel est le plus grand.

Et, comme cette partie $\{x\,,\,y\}$ admet un élément minimum et un élément maximum, ces deux éléments sont aussi (respectivement) borne inférieure et supérieure, on peut écrire $\mathrm{Inf}\{x\,,\,y\} = x$ et $\mathrm{Sup}\{x\,,\,y\} = y$.

L'existence de ces deux éléments est assurée, ce qui n'est pas le cas dans un ensemble qui n'est que partiellement ordonné.

5.2 Cas des ensembles partiellement ordonnés

Il existe alors des paires d'éléments (x, y) qui ne sont pas comparables et, pour une telle paire, les éléments $\mathrm{Min}\{x\,,\,y\}$ et $\mathrm{Max}\{x\,,\,y\}$ ne sont pas définis.

Il se peut, cependant, que les éléments $\mathrm{Inf}\{x\,,\,y\}$ et $\mathrm{Sup}\{x\,,\,y\}$ soient, eux, définis.

Si cette propriété est vérifiée pour tous les couples d'éléments (x, y), alors l'ensemble est un treillis :

DÉFINITION 27 (TREILLIS). *Un ensemble ordonné est un* treillis *lorsque toute partie finie admet une borne sup et une borne inf.* ♢

REMARQUE 20. Il suffit que la propriété soit vraie pour deux éléments distincts (*i.e.* une partie à deux éléments) pour qu'elle soit vrai pour toutes les parties finies.

Exercice 26. *Démontrez l'assertion de la remarque précédente.*

DÉFINITION 28 (TREILLIS COMPLET). *Si la propriété est vrai pour toute partie, alors le treillis est dit* complet. ♢

EXEMPLE 12. Un ensemble totalement ordonné est toujours un treillis.

REMARQUE 21. Cette notion n'offre un intérêt que dans le cas d'un ensemble partiellement ordonné, car l'existence d'une borne inférieure et d'une borne supérieure pour tout couple (et donc pour toute partie finie) permet de les comparer aux autres par l'intermédiaire de ces bornes, même si on ne peut pas les comparer entre eux.

En effet, on a toujours $\mathrm{Inf}\{x\,,\,y\} \leqslant x \leqslant \mathrm{Sup}\{x\,,\,y\}$ et $\mathrm{Inf}\{x\,,\,y\} \leqslant y \leqslant \mathrm{Sup}\{x\,,\,y\}$ (on reprend les exemples vus plus haut).

Exercice 27 (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 ? est-il un treillis ?*
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)}.*

V. Relations d'équivalence

On se place encore dans ce paragraphe dans le cas où $E = F$.

1 Définition

Soit $\mathcal{R}$ une relation binaire définie dans un ensemble (non vide) E, de graphe G.

DÉFINITION 29 (RELATION SYMÉTRIQUE). $\mathcal{R}$ *est dite* symétrique *si, dès que* x *est en relation avec* y*, alors* y *est en relation avec* x

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

REMARQUE 22. Ou encore : $\forall x \in E, \forall y \in E,\ x\mathcal{R}y \Longrightarrow y\mathcal{R}x$.

Exercice 28. *Est-ce qu'une relation sur un ensemble A dont le graphe est constitué uniquement de couples (x,x) est symétrique ? transitive ?*

DÉFINITION 30 (RELATION D'ÉQUIVALENCE). $\mathcal{R}$ *est une relation d'équivalence lorsqu'elle est réflexive, symétrique et transitive.* ♢

EXEMPLE 13. L'égalité est une relation d'équivalence.

EXEMPLE 14 (RELATION DE CONGRUENCE MODULO n DANS $\mathbb{Z}$). Par définition :

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

.

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

Exercice 30. *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.

2 Classes d'équivalence

2.1 Définition

DÉFINITION 31 (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 31. *Trouvez les classes d'équivalences des deux exercices précédents.*

Exercice 32. *On définit une relation sur l'ensemble des mots de la langue française de la façon suivante : « le mot M est lié au mot N si N est une anagramme*[1] *de M. »*
Quelle est la classe d'équivalence du mot chien ?

2.2 Propriétés des classes d'équivalence

PROPRIÉTÉ VII : Une classe d'équivalence n'est jamais vide.

PREUVE En effet, la classe de x contient toujours au moins l'élément x lui-même, par réflexivité. ■

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

REMARQUE 23. On dit aussi que les classes sont deux à deux disjointes.

PREUVE 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 32 (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.*♢

[1] Mot obtenu par transposition des lettres d'un autre mot. Une anagramme intéressante : aimer - Marie. Le pseudonyme de Alcofribas Nasier est, à peu près, l'anagramme de François Rabelais.

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

PREUVE 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 15. 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 33 (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 :*

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

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

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

---

3 Ensemble-quotient

DÉFINITION 33 (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 16 (CONGRUENCE MODULO 4). On choisit pour représentants les entiers < 4, donc 0, 1, 2 et 3.

L'ensemble-quotient est $\mathbb{Z}/4\mathbb{Z} = \{\dot{0}, \dot{1}, \dot{2}, \dot{3}\}$.

---

4 Exercices

Exercice 34. *Sur un ensemble à n éléments, combien y a-t-il de relations...*

1. *réflexives ?*
2. *symétriques ?*

Exercice 35. *Déterminer quand une relation $\mathcal{R}$ dans un ensemble A est*

1. *non réflexive,*
2. *non symétrique,*
3. *non transitive,*
4. *non antisymétrique.*

Exercice 36. *Donner des exemples de relation $\mathcal{R}$ dans $\{1, 2, 3\}$ ayant les propriétés suivantes :*

1. *$\mathcal{R}$ est symétrique,*
2. *$\mathcal{R}$ n'est ni symétrique ni antisymétrique,*
3. *$\mathcal{R}$ est transitive mais $\mathcal{R} \cup \mathcal{R}^{-1}$ n'est pas transitive.*

Exercice 37. *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.*

Exercice 38. *Soit $\mathcal{R}$ la relation d'équivalence suivante dans l'ensemble $A = \{A, 2, 3, 4, 5, 6\}$:*

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

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

Exercice 39. *Tenter de caractériser par son graphe une relation d'équivalence.*

Exercice 40. *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 ?

VI. Compatibilité entre une opération et une relation binaire

DÉFINITION 34. *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 $(x \circ y)\mathcal{R}(x' \circ y')$* $\diamondsuit$

Autrement dit, l'opération conserve la relation.

EXEMPLE 17. 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 $6 \leqslant -1$... 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 41 (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

Relations n-aires

I. Définitions

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 1 (RELATION n-AIRE). *Cette partie définit une relation n-aire entre ces ensembles.* ♢

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.

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

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

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

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.

2 Relations équivalentes, relations égales

DÉFINITION 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 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 $\mathcal{R}$

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 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.

---

4 SGBD

DÉFINITION 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

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 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 :*

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

REMARQUE 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.

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 I (THÉORÈME DES PROJECTIONS) :

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

PREUVE (Démonstration immédiate). ■

REMARQUE 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

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

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 $(\mathcal{R} \cap \mathcal{S})\,[A]$

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

V. Dépendances fonctionnelles et clés

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 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 $D_1\,,\,D_2\,,\,\ldots\,,\,D_p$,
– D_Y pour $D_{p+1}\,,\,\ldots\,,\,D_q$,
– D_Z pour $D_{q+1}\,,\,\ldots\,,\,D_n$,

– x pour $(x_1\ ,\ \ldots\ ,\ x_p)$,
– y pour $(x_{p+1}\ ,\ \ldots\ ,\ x_q)$
– et z pour $(x_{q+1}\ ,\ \ldots\ ,\ x_n)$.

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. 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} $\longrightarrow$ {Age}
– {Groupe , Niveau} $\longrightarrow$ {Age}

---

2 Théorème des dépendances fonctionnelles

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

EXEMPLE 5. La relation précédente est le produit de ses deux projections $\mathcal{R}[\{\text{Nom}\ ,\ \text{Age}\}]$ et $\mathcal{R}[\{\text{Nom}\ ,\ \text{Groupe}\ ,\ \text{Niveau}\}]$.

---

3 Clés

DÉFINITION 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 $C \longrightarrow A \setminus K$.*

(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 $K' \longrightarrow A \setminus K'$). ♢

REMARQUE 3. Cette « minimalité » n'entraîne en aucune manière l'unicité de la clé pour une relation donnée.

PROPRIÉTÉ III : 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 4

Ensembles de nombres entiers

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

1 Définition

DÉFINITION 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 I (PRINCIPE DE RÉCURRENCE) : Il s'énonce ainsi : « Si M contient 0 et le sucesseur 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* $[n \in M \Longrightarrow s(n) \in M]$*, alors* $M = \mathbb{N}$*.* ♢

REMARQUE 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 »

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 II (RÉCURRENCE RESTREINTE) : Soit $M = \{n \in \mathbb{N} \mid p(n)\}$.
Si $p \in M$ et si, $[n \in M \Longrightarrow s(n) \in M]$, 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 III (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 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 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.

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

PROPRIÉTÉ IV : 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.

3 Nombres premiers

3.1 Définitions

DÉFINITION 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 1. *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 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 2. Ainsi, le plus petit nombre premier (et le seul qui soit pair) est 2.

PROPRIÉTÉ V : Il existe une infinité de nombres premiers.

REMARQUE 4. Le problème de la primalité d'un nombre (très grand, évidemment) est difficile.

3.2 Décomposition en facteurs premiers

DÉFINITION 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, ...) $\diamondsuit$

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

Exercice 2. *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 3 (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.*

---

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

Exercice 4. *Calculez* $ppcm(102, 138)$.

Réponse : 2346.

PROPRIÉTÉ VII : $\mathbb{N}^*$ 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 6. *Deux nombres entiers strictement positifs a et b sont dits* premiers entre eux *lorsque* $a \wedge b = 1$. ♢

Exercice 5. *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.

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

1 Définition

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

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

EXEMPLE 6. 0 est divisible par tout nombre entier non nul $(0 = 0 \times b + 0)$.

Exercice 6. *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 7 (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.*

---

2 Représentation des nombres entiers

2.1 Définition

DÉFINITION 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 $(\overline{n_p n_{p-1} \dots n_0})_b$.

REMARQUE 5. En informatique, on utilise couramment les bases 2, 8 et 16.

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.

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 $(.....(((n_p b + n_{p-1})b + n_{p-2})b + \cdots + n_1)b + n_0)$

2.4 Exercices

Exercice 8 (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 9 (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 $x = \dfrac{p}{q}$, 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 $x = \dfrac{p}{q}$, 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.*

---

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 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 :*

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

PROPRIÉTÉ IX : 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 7. 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 $\dot{(n-1)}$, c'est-à-dire, il faut le remarquer, un nombre fini.

PROPRIÉTÉ X : 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 8. $\mathbb{Z}/3\mathbb{Z} = \{\dot{0}, \dot{1}, \dot{2}\}$.

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

PREUVE En effet, on suppose que :
- $x \equiv x'[n] \Longleftrightarrow \exists k \in \mathbb{Z},\ x - x' = k \cdot n$ et que
- $y \equiv y'[n] \Longleftrightarrow \exists l \in \mathbb{Z}, y - y' = l \cdot n$.
- 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 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 10. *Par définition, on pose* $\dot{x} + \dot{y} = \dot{(x + y)}$ *et* $\dot{x} \cdot \dot{y} = \dot{(xy)}$. ♢

EXEMPLE 9. 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 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 10. *Calculez :*

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

Réponses : 5 et 630.

Exercice 11 (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 12. *Résolvez modulo 18 les équations suivantes :*

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

Réponses : {8,17}, { } et {15}.

Exercice 13. *Si m est un entier naturel plus grand que 2, quel est l'inverse de $m - 1$ modulo m ?*

Réponse : $m - 1$.

Exercice 14. *Un nombre « pseudo-premier de base b » est un entier naturel non premier p tel que $(b^p - b)modp = 0$.*

Vérifier que 561 est pseudo-premier de base 3 et que 341 est pseudo-premier de base 2.

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...

5 Arithmétique modulo 2^n dans les ordinateurs

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.

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.

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 10. Premiers résultats, corrects :

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

EXEMPLE 11. 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 12. 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 13. 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 14. 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 15. 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

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.

2 Algorithme d'Euclide

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

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 );
}
```

3 Théorème de Bézout

On considère deux nombres entiers strictement positifs a et b.

PROPRIÉTÉ XII (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 $ax = ax' \Longrightarrow x = x'$, 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 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 15. *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 16. *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.

4 Algorithme d'Euclide généralisé

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 $(u - q_{i+1} \cdot v) \cdot r_{i+1} + v \cdot r_i = d$.

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 1 :
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 $r_0 \cdot u_{i+2} + r_1 \cdot v_{i+2} = r_0 \cdot (u_i - q_{i+1} \cdot u_{i+1}) + r_1 \cdot (v_i - q_{i+1} \cdot v_{i+1}) = r_0 \cdot u_i + r_1 \cdot v_i - q_{i+1} \cdot (r_0 \cdot u_{i+1} + r_1 \cdot v_{i+1}) = r_i - q_{i+1} \cdot r_{i+1} = r_{i+2}$. ■

4.3 Exemple.

Illustrons la mise en œuvre de cet algorithme...

EXEMPLE 16. 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 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 17. *Exprimer $pgcd(1330, 602)$ comme combinaison à coefficients entiers des nombres 1330 et 602.*

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

Fin du Chapitre

Chapitre 5

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 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 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

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 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 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 »).

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.

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

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)

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 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.

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 « co-processeur 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 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 $3,403 \cdot 10^{38}$.

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. 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 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 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 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 6

Cryptologie et arithmétique

I. Méthodes de cryptage « à clé publique »

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 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.

2 Utilisation de l'indicatrice d'Euler

Exercice 1 (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, $a^{\varphi(n)} \equiv 1\ [n]$.*

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 I : 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.

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 2. 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.

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.

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 7

Tests de primalité

I. Théorème de Fermat

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

Exercice 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 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 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 II : 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 III (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 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 IV (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 8

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 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)

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 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.

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.

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 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 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)

1 Introduction aux courbes elliptiques

DÉFINITION 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). ♢

REMARQUE 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 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 $P' = (x, -y)$.* ♢

Les coordonnées de P_3 sont obtenues comme suit :

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

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 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 9

Algèbre de Boole

I. Propriétés générales

1 Définition

DÉFINITION 1 (ALGÈBRE DE BOOLE). *On appelle* algèbre de Boole *la structure algébrique définie par un ensemble (non vide) $\mathcal{A}$ et trois opérations :*

- *la somme booléenne : $+$,*
- *le produit booléen : $.$,*
- *la négation booléenne (unaire) : $\overline{\ }$ (ex. $\overline{a}$)*

Ces opérations doivent de plus posséder les propriétés de la page suivante, pour que l'on puisse alors dire que le quadruplet $(\mathcal{A}, +, ., \overline{\ })$ est une algèbre de Boole.

(On a mit en parallèle les expressions obtenues :

- *en utilisant les notations générales d'une algèbre de Boole,*
- *et celles qui sont obtenues dans l'ensemble des parties d'un ensemble : c'est une algèbre de Boole particulière, bien connue, et qui possède des notations spécifiques.)* ♢

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

REMARQUE 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 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 2 (Opérateurs de Sheffer et de Peirce). *Soit $(E, +, \cdot, \overline{\ })$ une algèbre de Boole.*

1. *On définit l'opération de Sheffer*[1] *par : $a|b = \overline{a} + \overline{b}$ (c'est le NAND des informaticiens).*
   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 : $a \downarrow b = \overline{a} \cdot \overline{b}$ (c'est le NOR). Mêmes questions.*

REMARQUE 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 absence totale de lisibilité, ces connecteurs ne sont pas utilisés en logique.

2 Règles de calcul dans une algèbre de Boole

2.1 Particularités du calcul booléen

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 $a = b = 1$ (*c.f.* $A \cap B = E \Leftrightarrow A = E$ 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 faut s'habituer aussi à celle des deux distributivités qui n'est pas habituelle, celle de la somme sur le produit (booléens), et, par exemple, ne pas hésiter à écrire directement $(a + b)(a + c)(a + d)(a + e)(a + f) = a + bcdef$!

[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 !

2.2 Règles de « redondance »

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

REMARQUE 3. L'application de cette troisième règle peut être combinée avec celle des autres, comme par exemple dans le calcul suivant :

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

PREUVE $a \cdot b + \overline{a} \cdot c + b \cdot c = a \cdot b + \overline{a} \cdot c + (a + \overline{a}) \cdot b \cdot c = a \cdot b + \overline{a} \cdot c + a \cdot b \cdot c + \overline{a} \cdot b \cdot c$; $a \cdot b$ absorbe $a \cdot b \cdot c$ et $\overline{a} \cdot c$ absorbe $\overline{a} \cdot b \cdot c$, d'où le résultat. ■

Exercice 3 (Somme disjonctive). *Montrez que l'on a $a = b$ si et seulement si $a \oplus b = 0$).*

Exercice 4 (Calcul booléen élémentaire). *Effectuez les calculs suivants :*

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

Exercice 5 (Calcul booléen). *Simplifier les expressions suivantes.*
(Il n'est pas interdit de s'aider éventuellement d'un diagramme de Karnaugh – c.f. ci-dessous – mais toutes les simplifications proposées doivent être justifiées par calcul algébrique.)

- $A=(\overline{a}+b)(\overline{c}+\overline{a}\cdot\overline{b}+a\cdot b)$.
- $B=(a+\overline{b}+\overline{c})\cdot(\overline{a}+b)\cdot(\overline{b}+c)$.
- $C=(a+\overline{b}+c+b\cdot\overline{d})\cdot(\overline{b}+c)$.
- $D=\overline{\overline{a}+\overline{b}+c}+\overline{\overline{a}+b}+\overline{a}+c$.
- $E=\overline{(a+\overline{b}+c)\cdot(\overline{a+b}+c+d)}+\overline{a+\overline{b}+d}\cdot\overline{\overline{a+b}+a+d}$.
- $F=\overline{[(\overline{a}+c)+(\overline{b}+d)]\cdot(\overline{c}+\overline{d})}+\overline{a}+\overline{b}$.
- $G=a\cdot(\overline{b}+c)\cdot(\overline{a\cdot b}+a\cdot c)+\overline{a\cdot(\overline{b}+c)}\cdot\overline{\overline{a\cdot b}+a\cdot c}$.
- $H=\overline{(\overline{a\cdot b\cdot c+a\cdot b\cdot d})\cdot(\overline{a}+\overline{b}+\overline{c+d})}+a\cdot b\cdot(c+d)\cdot(\overline{a}+\overline{b}+\overline{c+d})$.
- $I=\overline{\overline{a}\cdot\overline{b}+a\cdot b}+\overline{b}\cdot\overline{c}+b\cdot c$.
- $J=\overline{\overline{a}+b+\overline{\overline{a}\cdot b}}+a\cdot\overline{b}+c$.
- $K=\overline{\overline{a\cdot\overline{b}}+\overline{c}+d\cdot(\overline{a\cdot c}+b+d)}+(\overline{a\cdot\overline{b}+\overline{c}+d)\cdot\overline{\overline{a\cdot c}+b+d}}$.
- $L=(a+c)\cdot(\overline{a}+d)\cdot(\overline{b}+\overline{e})\cdot(\overline{b}\cdot\overline{c}+b\cdot c)\cdot(\overline{d}+c\cdot e)\cdot(\overline{c}+d)$.
- $M=(\overline{a}\cdot\overline{a\cdot(\overline{b}+\overline{c})}+a\cdot(\overline{b}+\overline{c}))\cdot(\overline{b}\cdot\overline{\overline{a}+c}+(\overline{a}+c)\cdot b)\cdot(a\cdot\overline{b}\cdot c+\overline{a\cdot\overline{b}}\cdot\overline{c})$.

II. Fonctions booléennes

1 Définitions

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

DÉFINITION 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 2. $f(a, b, c) = a \cdot \overline{b} + c$.

REMARQUE 4. 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 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 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 $f(x_1, x_2, \ldots, x_n) = 1$. ♢

2 Fonctions booléennes élémentaires

DÉFINITION 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 variables.*

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

DÉFINITION 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 3 (MINTERME À TROIS VARIABLES). $a \cdot \overline{b} \cdot c$

EXEMPLE 4 (MAXTERME À TROIS VARIABLES). $\overline{a} + b + \overline{c}$.

Exercice 6. *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 7. *Dressez la liste des mintermes et des maxtermes pour deux variables a et b.*

PROPRIÉTÉ I (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) : On les représente par $m_i^{(n)}$ pour un minterme, et $M_i^{(n)}$ pour un maxterme.

L'indice i varie entre 0 et $2^n - 1$, et fait l'objet d'une convention de numérotation des mintermes (et maxtermes).

Pour que cette numérotation ait un sens, il est indispensable d'adopter un ordre d'énumération des variables, et, une fois que celui-ci a été défini, de s'y tenir une fois pour toutes.

EXEMPLE 5. 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 $a \cdot b \cdot c \cdot d$.

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.

En écrivant ces chiffres les uns à côté des autres, et dans le même ordre que les variables correspondantes, on obtient une expression qu'on peut considérer comme l'écriture d'un entier positif en base 2.

DÉFINITION 7 (INDICE D'UN MINTERME). *L'*indice du minterme *(ou du maxterme) est la valeur décimale de cet entier.* ♢

EXEMPLE 6. 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 8. *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 $\overline{a} \cdot b \cdot c$.*

3 Correspondance entre maxtermes et mintermes

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

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

EXEMPLE 7. $\overline{\overline{a} \cdot b \cdot \overline{c}} = a + \overline{b} + c$

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

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

4 Principaux résultats concernant mintermes et maxtermes

PROPRIÉTÉ IV : Les mintermes à n variables sont disjoints

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

Exercice 9. *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 5. 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 V : 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.

On les ordonne, dans cette somme, 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. *Le vérifier dans le cas de deux variables.*

REMARQUE 6. Par négation (booléenne) de cette propriété, on obtient : *le produit de tous les maxtermes à n variables est nul*.

5 Formes canoniques d'une fonction booléenne

5.1 Définition et théorème

DÉFINITION 8 (MONÔMES). *Un* monôme *est une fonction booléenne dans l'expression de laquelle n'interviennent que le produit et la négation booléennes.* ♢

PROPRIÉTÉ VI : 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.

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.

Puis on développe les produits qui portent sur des sommes, en utilisant la distributivité du produit sur la somme.

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

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

PREUVE 2 :
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 VIII (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 7. L'unicité de cette FCD permet la comparaison des fonctions booléennes entre elles.

Par négation booléenne de ce résultat, on obtient :

PROPRIÉTÉ IX (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).

5.2 Obtention des formes canoniques

La méthode algébrique consiste à :
- tout développer pour mettre l'expression sous la forme d'une somme,
- dans chaque terme de cette somme, faire apparaître les valeurs qui n'y figurent pas.

EXEMPLE 8. 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 9. $f(a,b,c) = a + bc = (a+b)(a+c) = (a+b+\overline{c}c)(a+\overline{b}b+c)$
$= (a+b+\overline{c}) \cdot (a+b+c) \cdot (a+\overline{b}+c) \cdot (a+b+c) = M_5 M_6 M_7$

REMARQUE 8. 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 11. *Obtenir la FCC de* $x + \overline{y}z$.

Il existe une autre méthode pour obtenir ces formes canoniques : la méthode des diagrammes.

III. Représentation et simplification des fonctions

1 Diagrammes de Karnaugh

1.1 Présentation

La représentation des fonctions booléennes par diagrammes de Karnaugh-Veitch :
- est fondée sur les propriétés des mintermes (ils réalisent une partition de l'unité),
- et est copiée de la représentation des ensembles par diagrammes d'Euler-Venn (les fameuses « patates »).

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. 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 11. 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 12. 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 12. *Faire un diagramme à cinq variables.*

Réponse :

de \ abc	000	001	011	010	110	111	101	100
00	0	4	8	12	28	24	20	16
01	1	5	9	13	29	25	21	17
11	3	7	11	15	31	27	23	19
10	2	6	10	14	30	26	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.

1.2 Utilisation

Les diagrammes peuvent être utilisés « en réunion » comme « en intersection ».

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 et en gras 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 et gras : 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 13 (Fonctions booléennes). *Donner la forme canonique disjonctive de la fonction booléeene dont l'expression est*

$$f(a,b,c,d,e) = \overline{a} \cdot [\overline{b} \cdot \overline{e} \cdot (c+d) + b \cdot (\overline{c} \cdot \overline{d} \cdot \overline{e} + c \cdot \overline{d \cdot e})].$$

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.

Il existe diverses méthodes, nous n'en exposerons ici qu'une seule, la méthode de Quine-Mac Cluskey, dite aussi *méthode des consensus*.

2 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...

2.1 Les consensus

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 des deux.

EXEMPLE 13. 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 14. 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 14. *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 X (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, soit m et m' deux termes de la somme ne contenant pas la variable a et ne présentant pas de consensus.

Alors le consensus de $a \cdot m$ et de $\overline{a} \cdot m'$ est $m \cdot m'$, et on peut constater que $a{\cdot}m{+}\overline{a}{\cdot}m'{+}m{\cdot}m' = a{\cdot}m{+}\overline{a}{\cdot}m'{+}(a{+}\overline{a}){\cdot}m{\cdot}m' = a{\cdot}m{+}\overline{a}{\cdot}m'{+}a{\cdot}m{\cdot}m'{+}\overline{a}{\cdot}m{\cdot}m' = a \cdot m + \overline{a} \cdot m'$. ■

2.2 La méthode des consensus

Venons-en à la méthode proprement dite. Elle se déroule en trois étapes...

Étape préliminaire Développer l'expression pour qu'elle soit sous forme de somme de monômes, et en suppprimer les termes redondants (par idempotence et application de la règle n°1) : on obtient ainsi l'expression de départ.

Toute autre tentative de simplification est, dans cette étape, parfaitement inutile.

Obtention d'une forme stable par consensus Répéter les deux phases suivantes jusqu'à ce que l'expression obtenue ne soit plus modifiée :

1. Rajouter tous les consensus des termes qui en présentent.
2. Supprimer les termes redondants introduits (idempotence et règle n°1 exclusivement).

REMARQUE 9. L'introduction des consensus fait parfois apparaître des redondances, la suppression de celles-ci fait parfois apparaître de nouvelles possibilités de consensus, etc.

DÉFINITION 9 (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 *(pour l'expression de départ).* ♢

La somme des monômes principaux d'une expression booléenne est généralement plus longue que l'expression de départ, quelquefois 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 15. 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.

EXEMPLE 16. 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.

Exercice 15. *Trouvez la forme stable par consensus de*

$$f(a, b, c, d) = \overline{a}\overline{b}c + \overline{a}c\overline{d} + a\overline{b}\overline{c}\overline{d} + a\overline{c}d + bcd$$

Choix d'un nombre minimal de monômes principaux On calcule la FCD de la fonction booléenne à simplifier.

Pour chacun des mintermes de cette forme, on dresse la liste des monômes principaux qui le contiennent (dont ce minterme est un multiple).

Toute forme égale à l'expression donnée doit contenir tous les mintermes (c'est-à-dire, pour chacun d'entre eux, au moins un des monômes principaux qui le contiennent).

On exprime cette condition en associant à chaque monôme principal une variable booléenne binaire (lorsque celle-ci prend la valeur 1, cela signifie que ce monôme doit figurer dans toute forme représentant la fonction à simplifier) et en reprenant alors les mintermes l'un après l'autre.

Exprimer que ce minterme doit être contenu dans toute forme représentant la fonction booléenne à simplifier se fait en posant l'équation booléenne dont le premier membre est la somme des variables booléennes associées aux monômes principaux qui contiennent ce minterme et dont le second membre est 1.

Ensuite, on forme le produit des premiers membres de ces diverses équations (en algèbre de Boole, un produit ne peut être égal à 1 que si tous ses facteurs sont égaux à 1).

On développe ce produit, et on choisit le terme qui comporte le moins de facteurs (ou l'un de ceux qui comportent le moins de facteurs) : il correspond à la (ou une) forme minimale de l'expression de départ.

Exercice 16. *On reconsidère*

$$f(a,b,c,d) = \overline{a}\overline{b}c + \overline{a}c\overline{d} + a\overline{b}\overline{c}\overline{d} + a\overline{c}d + bcd$$

- *Trouvez sa FCD.*
- *En déduire ses formes minimales.*

2.3 Exercices corrigés

Premier exercice :

Exercice 17. *Appliquez la méthode des consensus à*

$$f(a,b,c) = (a+b)(\overline{a} + \overline{b} + c)$$

1. On développe : $f(a,b,c) = a\overline{b} + ac + \overline{a}b + bc$.
   Il n'y a pas de simplification possible par idempotence et règle 1 : c'est donc notre expression de départ.
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 monomes principaux.
3. La FCD...

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

   D'où la FCD : $m_2 + m_3 + m_4 + m_5 + m_7$.
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.

Deuxième exercice :

Exercice 18. *Appliquez 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 $a \cdot b + \overline{a} \cdot c$.

Troisième exercice :

Exercice 19. *Appliquez 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 :

$$\left\{ \begin{array}{c} \overline{b} \cdot \overline{c} + b \cdot d + c \cdot \overline{d} \\ \text{ou} \\ \overline{c} \cdot d + \overline{b} \cdot \overline{d} + b \cdot c \end{array} \right\} \quad + \quad \left\{ \begin{array}{c} \overline{a} \cdot \overline{c} \\ \text{ou} \\ \overline{a} \cdot b \\ \text{ou} \\ \overline{a} \cdot \overline{d} \end{array} \right\}$$

IV. Complément : Résolution d'équations booléennes

1 Présentation de la méthode

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,
$$x_1 \cdot s = 0 \iff \forall y_1 \in E, x_1 = y_1 \cdot \overline{s}$$
.

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 :

$$\forall z_1 \in E, y_1 = \overline{z_1} + r$$

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.

2 Exercice

Exercice 20. *Résoudre l'équation :* $x + y = x + z$.

Exercice 21. *Résoudre l'équation :* $x \cdot y + \overline{x} \cdot z = 0$

Exercice 22. *Résoudre le système d'équations :* $\begin{cases} x + y = x + z \\ x \cdot y = x \cdot z \end{cases}$

V. Exercices

Exercice 23 (Méthode des consensus). *Utiliser la méthode des consensus pour obtenir toutes les formes minimales des fonctions booléennes suivantes :*

1. *Celles des précédents exemples et exercices.*
2. $\overline{d} \cdot e + \overline{a} \cdot c + b \cdot \overline{c} + a \cdot \overline{b} + a \cdot d \cdot e + \overline{a} \cdot d \cdot \overline{e}$

Exercice 24. *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 25 (Implicants premiers). *Un monôme de P est un implicant premier de l'expression booléenne E si $P + E = E$ et tout autre monôme inclus P n'a pas cette proprité.*

Etant donné $E = x\overline{y} + xy\overline{z} + \overline{x}y\overline{z}$, montrer que $x\overline{z}$ est un implicant premier de E.

Réponse : $x\overline{z} + E = E$, $x + E \neq E$ et $\overline{z} + E \neq E$.

Exercice 26 (Application de la méthode des consensus). *Utiliser la méthode du consensus pour trouver une forme minimale de E et F avec $E = x\overline{y} + xy\overline{z} + \overline{x}y\overline{z}$ et $F = xy + \overline{y}t + \overline{x}y\overline{z} + x\overline{y}z\overline{t}$.*

Exercice 27 (Application de la méthode de Karnaugh). *Trouver une forme minimale de $E = x\overline{y} + xyz + \overline{xyz} + \overline{x}yz\overline{t}$.*

Exercice 28 (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 29 (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* $f_A : E \longrightarrow \{0,1\}$*, 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 30. *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.*

Exercice 31 (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* $x \cdot y$*.*

Fin du Chapitre

Chapitre 10

Calcul Propositionnel

I. Introduction

1 Objets de la logique

L'objet de la Logique est d'analyser le raisonnement humain et, si possible, de le formaliser de manière à le rendre aussi indépendant que possible des diverses contingences matérielles qui pourraient influer sur lui.

C'est une discipline philosophique, dont on étudiera les aspects mathématiques.

Cette formalisation poursuit aussi d'autres buts :
- essai d'unification de diverses théories qui présentent, du point de vue des raisonnements qui sont utilisés, des analogies,
- essai de production automatique de résultats dans certains domaines.

REMARQUE 1. Indépendamment de la « beauté théorique » de la chose, le premier point présente un intérêt pratique : faire profiter telle branche de la science des résultats qui pourraient avoir été obtenus ailleurs.

C'est surtout le deuxième aspect qui nous occupera dans ce chapitre (et les suivants).

2 Production automatique

Chacun sait que, si un ordinateur calcule vite et bien, il est parfaitement incapable de produire de lui-même quelque « raisonnement » que ce soit.

Mais s'il était possible de programmer les raisonnements étudiés par la logique sous forme de calculs, précisément, il serait raisonnable de penser que l'ordinateur serait alors, de ce point de vue aussi, bien plus rapide et efficace que l'esprit humain.

Les divers domaines de recherche de ce que l'on appelle de nos jours l'« Intelligence Artificielle » convergent presque tous vers ce but.

Même si certains espoirs ont été déçus, comme dans le domaine de la traduction automatique (on arrive tout juste à traduire quelques textes techniques « standard », et il s'avère très difficile de « donner le sens des nuances » à un algorithme...), de nombreux résultats intéressants et prometteurs ont été obtenus dans divers autres domaines (diagnostic médical, etc.)

3 Des problèmes de l'évidence

Avant de rechercher à « programmer un raisonnement », il convient de savoir précisément de quoi il s'agit.

Il faut apprendre à disséquer nos démarches intellectuelles, et, avant tout, ne pas se laisser paralyser par l'« évidence » : rien ne l'est pour un ordinateur. Le plus difficile est peut-être de pourchasser, pour les expliciter clairement, tous les sous-entendus qui nous permettent de produire des raisonnements intelligibles.

Finalement, il convient toujours d'adapter son discours au niveau de son auditoire : le niveau de l'ordinateur, de ce point de vue, peut être considéré comme égal à zéro.

II. Les fondements de la logique des Propositions

1 Les Propositions

1.1 Articulation d'un raisonnement

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.

1.2 Les propositions, intuitivement

DÉFINITION 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 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).

REMARQUE 2. Ce n'est pas la logique qui décide de ce qui est vrai et de ce qui est faux : les valeurs de vérité sont attribués en-dehors de la logique.

Exercice 1. *Si je dis : « la présente affirmation est vraie », cette affirmation possède-t-elle une valeur de vérité ?*

Exercice 2. *Les affirmations suivantes sont-elles des propositions ?*

1. *« l'affirmation qui suit est vraie »,*
2. *« l'affirmation qui précède est fausse ».*

1.3 Des énoncés n'étant pas des propositions

Un énoncé « hypothétique » comme « Dans six mois, il y aura une exceptionnelle période de beau temps » est exclu du domaine des propositions et donc de notre étude.

Il en est de même pour ce qu'on appelle les « paradoxes » de la logique comme, par exemple, celui du menteur.

En effet, il est impossible d'attribuer une valeur de vérité à ces énoncés : nous les rejetons en dehors du cadre des propositions.

1.4 Les propositions, plus rigoureusement

D'une manière générale et plus précise, les « propositions » qu'étudie la logique des valeurs de vérité répondent aux axiomes suivants :

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).

Notons à nouveau que l'objet de la logique n'est pas de savoir qui attribue les valeurs de vérité, ni sur quels critères, mais seulement de manipuler ces valeurs de vérité.

1.5 D’autres logiques

Il existe, bien entendu, d’autres logiques :

– fondées sur d’autres axiomes,
– qui admettent une troisième « valeur de vérité » : le « possible »,
– qui attribuent des « coefficients de vraisemblance » aux énoncés,
– etc.

Ces logiques sortent du cadre de notre étude.

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 », de la manière suivante...

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 ce qu’en grammaire française on appelle 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.

Ces propositions, au sens grammatical du terme, sont reliées entre elles par « parce que » et par « ou », qui sont –grammaticalement parlant– des conjonctions (respectivement de subordination et de coordination) :

parce que : introduit habituellement un lien de « cause à effet »,

ou : se contente de juxtaposer les propositions (au même niveau).

2.2 Vers une formalisation

Bref, cette proposition exprime que « ma mauvaise note est conséquence de l’une (au moins) des deux causes suivantes :

– mon manque de travail,
– l'excessive difficulté du cours ».

Autrement posé (il s'agit d'un début de formalisation) :

((manque de travail) ou (cours trop difficile)) entraîne (ma mauvaise note).

REMARQUE 3. Il ne faut pas sous-estimer la difficulté de ce travail d'analyse :
– le langage courant est souvent imprécis ou ambigu,
– il faut souvent se livrer à une véritable interprétation pour parvenir à formaliser une phrase.

REMARQUE 4. L'analyse en logique des propositions s'arrête au niveau des connecteurs logiques (qui vont être présentés).

Elle ne permet pas de prendre en compte certaines nuances, la concordance des temps, ou d'autres liens qui peuvent exister entre des propositions.

EXEMPLE 2. En logique des propositions, les propositions « il y a des gens qui font ceci ou cela » et « les gens font ceci ou cela » sont simplement différentes, et les connecteurs logiques ne permettent pas d'établir un rapport sémantique (au niveau du sens) entre les deux.

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 »).

Ils sont définis (axiomatiquement) à partir de leurs tables de vérité.

2.3 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 en fonction de celles de P et de Q par la table de vérité :

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

Le « ou » dont il s'agit, comme l'indique la table de vérité, est le « ou » inclusif. C'est la définition, qui est conforme à l'usage général qui en est fait dans la langue française.

REMARQUE 5. Même si l'on conteste cette dernière assertion, par exemple en produisant le fameux exemple du « fromage ou dessert » (dans lequel le « ou » semble bien être exclusif dans l'esprit du restaurateur)... IL FAUT traduire un « ou » par une disjonction logique.

Cette règle de traduction vient de ce que :

- cette interprétation ne risque pas de conduire à des fautes de raisonnement (alors que le traduire par un « ou exclusif » peut mener à une telle erreur),
- et aussi parce qu'il y aura toujours un plus grand nombre de solutions en utilisant un « ou » inclusif et qu'il est plus facile d'éliminer une solution qui vous déplaît que d'en inventer une que le système n'a pas proposé.

Conclusion : on ne traduira un « ou » par un « ou exclusif » que dans le cas où mention explicite est faite de l'exclusion mutuelle des possibilités.

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 en fonction de celles de P et de Q par la table de vérité :

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

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 en fonction de celle de P par la table de vérité :

P	$\neg P$
F	V
V	F

Implication logique : Connecteur « si...alors », symbole $\longrightarrow$.

À partir de deux propositions P et Q, ce connecteur permet la construction de la proposition (Si P, alors Q) [notée $P \longrightarrow Q$], dont la valeur de vérité est définie en fonction de celles de P et de Q par la table de vérité :

P	Q	$P \longrightarrow Q$
F	F	V
F	V	V
V	F	F
V	V	V

REMARQUE 6. La proposition « Si P, alors Q » est vraie, quelle que soit la valeur de vérité de la proposition Q, lorsque la proposition P est fausse.

EXEMPLE 3. La proposition : « Si le pôle Nord est l'endroit le plus chaud de la planète, alors les poules ont des dents » doit être considérée comme ayant la valeur de vérité « vrai ».

Le connecteur logique « $\longrightarrow$ » ne contient aucune idée de raisonnement, et ne s'occupe nullement (on l'a dit) des liens sémantiques qui peuvent exister entre la température qui règne au pôle Nord et la dentition des poules.

Il s'agit simplement d'une proposition, qui forme un tout, et qui a la valeur de vérité « vrai » lorsque, notamment, P et Q ont toutes les deux la valeur de vérité « faux ».

Exercice 3. *Trouver un raisonnement qui donne la valeur de vérité de la proposition : « Si Napoléon et Jules César sont une seule et même personne alors 5 = 0 ».*

On se souviendra que Napoléon a vécu 52 ans et Jules César 57...

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 : $\longleftrightarrow$.

À partir de deux propositions P et Q, ce connecteur permet la construction de la nouvelle proposition (P si et seulement si Q) [notée $P \longleftrightarrow Q$], dont la valeur de vérité est donnée par la table de vérité :

P	Q	$P \longleftrightarrow Q$
F	F	V
F	V	F
V	F	F
V	V	V

REMARQUE 7. Même remarque que pour l'implication logique : l'équivalence logique de deux propositions fausses est une proposition vraie.

2.4 Exercices

Exercices corrigés

Exercice 4. *En notant P et Q les affirmations suivantes :*
- *P = « Jean est fort en Maths »,*
- *Q = « Jean est fort en Chimie »,*

représenter les affirmations qui suivent sous forme symbolique, à l'aide des lettres P et Q 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 : $P \wedge (\neg Q)$; $(\neg P) \wedge (\neg Q)$; $P \vee (\neg P \wedge Q)$; $Q \in P$; $(P \wedge Q) \vee (\neg P \wedge Q)$.

Exercice 5. *En notant P, Q et R les trois affirmations suivantes :*
- *P = « Pierre fait des Maths »*
- *Q = « Pierre fait de la Chimie »*
- *R = « Pierre fait de l'Anglais »*

représenter les affirmations qui suivent sous forme symbolique, à l'aide des lettres P, Q, R 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 »*

Réponses : $P \wedge R \wedge (\neg Q)$; $(P \wedge Q) \wedge (\neg(Q \wedge R))$; $\neg(R \wedge (\neg P))$; $\neg((\neg P) \wedge Q)$; $(\neg R) \wedge (\neg Q) \wedge P$.

Exercice 6. *Énoncer la négation des affirmations suivantes en évitant d'employer l'expression : « il est faux que »*

1. *« S'il pleut demain ou s'il fait froid je ne sortirai 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° »*

Réponses :

1. S'il pleut demain ou s'il fait froid je sortirai
2. Le nombre 522 est divisible par 3 ou il n'est pas divisible par 7
3. Ce quadrilatère est un rectangle ou un losange
4. Paul n'ira pas travailler ce matin mais il ne perdra pas son emploi
5. Il existe un nombre entier impair divisible par 2
6. Il existe un triangle équilatéral dont les angles ne sont pas égaux à 60°

Exercice 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. *Si $531^{617}+1$ est divisible par 7 alors $531^{617}-13$ est divisible par 43*
11. *La décimale de π qui porte le numéro 10^{400} est 3 implique que si ce n'est pas 3 alors c'est 3.*

Réponses : F ; V ; V ; F ; V ; F ; V ; V ; V ; F ; V.

Exercice 8. *Partant des deux affirmations P et Q, on peut en construire une autre, notée $P \downarrow Q$, bâtie sur le modèle : « ni P, ni Q ».*

Cette opération est-elle une connexion ? Si oui, quelle est sa table de vérité ?

Réponse : c'est une connexion, puisque $P \downarrow Q = (\neg P) \wedge (\neg Q)$.

Exercices sans correction

Exercice 9. *A et B sont des variables propositionnelles, susceptibles de représenter n'importe quelles propositions.*

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 10. *Les variables propositionnelles N et T serviront, dans cet exercice, à représenter (respectivement) les propositions « Un étudiant a de bonnes notes » et « Un étudiants 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.*

3 Variables et Formes Propositionnelles

3.1 Définitions

Comme le calcul propositionnel ne s'occupe que des valeurs de vérité, il est possible, dans une expression logique, de remplacer une proposition donnée par un symbole (en général, une lettre de l'alphabet majuscule), ou *variable propositionnelle*.

DÉFINITION 2 (FORMES PROPOSITIONNELLES). *Les expressions ainsi obtenues sont appelées* formes propositionnelles . ♢

REMARQUE 8. 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 forme propositionnelle considérée.

Exercice 11. *Combien de lignes contient la table de vérité d'une forme propositionnelle qui dépend de n variables ?*

Réponse : 2^n.

3.2 Règles de formation

PROPRIÉTÉ I : Les règles (de syntaxe) qui permettent de former des formes propositionnelles correctes sont les suivantes :
- Toute variable propositionnelle est une forme propositionnelle
- Si F et G sont des formes propositionnelles, alors $\neg(F)$, $(F) \vee (G)$, $(F) \wedge (G)$, $(F) \longrightarrow (G)$ et $(F) \longleftrightarrow (G)$ sont des formes propositionnelles.

Lorsqu'on remplace, dans une forme propositionnelle, les variables propositionnelles par des propositions, l'assemblage obtenu est une proposition.

Cependant, une forme propositionnelle n'est pas une proposition : $A \longrightarrow B$ n'est ni vrai ni faux.

PROPRIÉTÉ II (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 4. $\neg A \wedge B \longrightarrow C$ doit être interprété par $((\neg A) \wedge B) \longrightarrow C$ et $A \vee B \wedge C$ n'a pas de sens, car les deux connecteurs ont même niveau de priorité.

PROPRIÉTÉ III (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 $\wedge$: $(A \vee C) \wedge C \neq A \vee (B \wedge C)$.

REMARQUE 9. L'implication n'est pas associative : $A \longrightarrow (B \longrightarrow C) \neq (A \longrightarrow B) \longrightarrow C$. Donc les parenthèses sont obligatoires.

Il en est de même pour eqv, et a fortiori quand ces deux opérateurs sont mélangés dans une même proposition.

Exercice 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 forme propositionnelle ? Déterminer la table de vérité de chacune des formes obtenues. Que remarque-t-on quand les trois propositions sont fausses ?*

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

On remarque que la proposition obtenue est vraie quelle que soit la façon de placer les parenthèses.

3.3 Exercices

Exercices corrigés

Exercice 13. *Construire les tables de vérité des formes propositionnelles suivantes :*

1. $(\neg p) \wedge q$
2. $(\neg p) \longrightarrow (p \vee q)$
3. $\neg((\neg p) \wedge (\neg q))$
4. $(p \wedge q) \longrightarrow (\neg q)$
5. $(p \longrightarrow q) \vee (q \longrightarrow p)$
6. $(p \longrightarrow (\neg q)) \vee (q \longrightarrow (\neg p))$

7. $(p \vee (\neg q)) \wedge ((\neg p) \vee q)$
8. $p \longrightarrow ((\neg p) \longrightarrow p)$

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 14. *Faire de même avec*

1. $(p \vee q) \vee (\neg r)$
2. $p \vee (\neg(q \wedge r))$
3. $(\neg p) \longrightarrow ((\neg q) \vee r)$
4. $(p \vee r) \longrightarrow (r \vee (\neg p))$
5. $(p \longrightarrow (\neg q)) \vee (q \longrightarrow r)$
6. $(p \vee (\neg q)) \longrightarrow ((\neg p) \vee r)$
7. $(p \longrightarrow (\neg r)) \vee (q \wedge (\neg r))$
8. $(p \longrightarrow q) \longrightarrow ((q \longrightarrow r) \longrightarrow (p \longrightarrow r))$

Réponse :

p	q	r	1	2	3	4	5	6	7	8
V	V	V	V	V	V	V	V	V	V	V
V	V	F	V	V	V	F	F	F	V	V
V	F	V	V	V	V	V	V	V	F	V
V	F	F	V	V	V	F	V	F	V	V
F	V	V	V	F	V	V	V	V	V	V
F	V	F	V	V	F	V	V	V	V	V
F	F	V	F	V	V	V	V	V	V	V
F	F	F	V	V	V	V	V	V	V	V

Exercices sans correction

Exercice 15 (Les animaux de la maison). *Formalisez, en logique des propositions :*

1. *Les seuls animaux de cette maison sont des chats.*
2. *Tout animal qui aime contempler la lune est apte à devenir un animal familier.*
3. *Quand je déteste un animal, je l'évite soigneusement.*
4. *Aucun animal n'est carnivore, à moins qu'il n'aille rôder dehors la nuit.*
5. *Aucun chat ne manque jamais de tuer les souris.*
6. *Aucun animal ne s'attache jamais à moi, sauf ceux qui sont dans cette maison.*
7. *Les panthères ne sont pas aptes à devenir des animaux familiers.*
8. *Aucun animal non carnivore ne tue de souris.*
9. *Je déteste les animaux qui ne s'attachent pas à moi.*
10. *Les animaux qui vont rôder dehors la nuit aiment toujours contempler la lune.*

Exercice 16 (Les exercices de Logique). *Faire de même avec*

1. *Quand un étudiant résout un exercice de logique sans soupirer, vous pouvez être sûr qu'il le comprend.*
2. *Ces exercices de logique ne se présentent pas sous la forme habituelle.*
3. *Aucun exercice de logique facile ne donne mal à la tête.*
4. *Les étudiants ne comprennent pas les exercices de logique qui ne se présentent pas sous la forme habituelle.*
5. *Les étudiants ne soupirent jamais devant un exercice de logique, à moins qu'il ne leur donne mal à la tête.*

Exercice 17 (Mes idées sur les chaussons aux pommes). *Toujours pareil avec*

1. *Toute idée de moi qui ne peut s'exprimer sous forme de syllogisme est vraiment ridicule.*
2. *Aucune de mes idées sur les chaussons aux pommes ne mérite d'être notée par écrit.*
3. *Aucune idée de moi que je ne parviens pas à vérifier ne peut être exprimée sous forme de syllogisme.*

4. *Je n'ai jamais d'idée vraiment ridicule sans la soumettre sur le champ à mon avocat.*
5. *Mes rêves ont tous trait aux chaussons aux pommes.*
6. *Je ne soumets aucune de mes idées à mon avocat si elle ne mérite pas d'être notée par écrit.*

Exercice 18 (Les matières enseignées à l'IUT). *Encore pareil avec*

1. *Aucune matière n'est primordiale, sauf l'ACSI.*
2. *Toute matière enseignée par des professeurs dynamiques est susceptible de plaire aux étudiants.*
3. *Je ne travaille pas les matières que je n'aime pas.*
4. *Les seules matières intéressantes sont les matières informatiques.*
5. *Aucune matière informatique n'évite l'abstraction.*
6. *Aucune matière ne me réussit, excepté les matières intéressantes.*
7. *Les mathématiques ne sont pas susceptibles de plaire aux étudiants.*
8. *Aucune matière non primordiale ne tombe dans l'abstraction.*
9. *Je n'aime pas les matières qui ne me réussissent pas.*
10. *L'ACSI est enseignée par des professeurs dynamiques.*

III. Premier point de vue : la Logique des valeurs de vérité

1 Fonctions de vérité

Soit F une forme propositionnelle, dans l'expression de laquelle interviennent les variables propositionnelles $P_1, P_2, P_3, \ldots, P_n$.

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 3 (FONCTION DE VÉRITÉ DE F). *La fonction de vérité de F est la fonction booléenne Φ_F des n variables binaires concernées, obtenue de la manière suivante :*

– *Si F est de la forme P, où P est une variable propositionnelle, alors $\Phi_F(p) = p$.*

– *Si F est de la forme $\neg G$, où G est une forme propositionnelle, alors $\Phi_F = \overline{\Phi_G}$.*

– *Si F est de la forme $G \vee H$, où G et H sont des formes propositionnelles, alors $\Phi_F = \Phi_G + \Phi_H$.*

– *Si F est de la forme $G \wedge H$, où G et H sont des formes propositionnelles, alors $\Phi_F = \Phi_G \cdot \Phi_H$.*

– *Si F est de la forme $G \longrightarrow H$, où G et H sont des formes propositionnelles, alors $\Phi_F = \overline{\Phi_G} + \Phi_H$.*

– *Si F est de la forme $G \longleftrightarrow H$, où G et H sont des formes propositionnelles, alors $\Phi_F = \overline{\Phi_G} \cdot \overline{\Phi_H} + \Phi_G \cdot \Phi_H$.*

EXEMPLE 5. Si $F = P \longrightarrow Q$, alors $\Phi_F = \overline{p} + q$.

EXEMPLE 6. Si $G = \neg Q \longrightarrow \neg P$, alors $\Phi_G = \overline{\Phi_{\neg Q}} + \Phi_{\neg P} = \overline{\overline{q}} + \overline{p}$.

REMARQUE 10. On remarque que les deux fonctions de vérités des exemples précédents sont identiques. On en déduit que F et G sont logiquement équivalentes. G est appelée implication contraposée de l'implication F.

EXEMPLE 7. Soit $F = A \vee \neg B \longleftrightarrow (B \longrightarrow C)$. 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. Il est clair que les « tables de vérité » des connecteurs logiques sont les mêmes que les tables des opérations booléennes sur {faux,vrai}...
– de la négation booléenne (pour la négation logique),
– de la somme booléenne (pour la disjonction logique),

– du produit booléen (pour la conjonction logique),
– de la fonction booléenne de deux variables appelée « implication » (pour l'implication logique)
– de la fonction booléenne de deux variables appelée « équivalence » (pour l'équivalence 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 forme propositionnelle associée.

2 Tautologies, antilogies, conséquences logiques

2.1 Tautologies

DÉFINITION 4 (TAUTOLOGIE). *Toute forme propositionnelle dont la fonction de vérité est la fonction référentiel est appelée* tautologie . ♢

Ainsi, une tautologie est une forme 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 8. Soit $F = A \longrightarrow A$.
$\Phi_F(a) = \overline{a} + a = 1$, donc : $\models F$ (F est une tautologie).
Par exemple : « Si un étudiant est sérieux, alors il est sérieux ».

EXEMPLE 9. $F = (A \longrightarrow C) \longrightarrow ((B \longrightarrow C) \longrightarrow (A \vee B \longrightarrow C))$.
$\Phi_F = \overline{\Phi_{A \longrightarrow C}} + \overline{\Phi_{B \longrightarrow C}} + \Phi_{A \vee B \longrightarrow C} = \overline{\overline{a} + c} + \overline{\overline{b} + c} + \overline{a + b} + c$
$= a\overline{c} + b\overline{c} + \overline{a}\overline{b}c = a + b + \overline{a}\overline{b} + c = 1 + c = 1$

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 méthématique, tous les théorèmes sont des tautologies ; la reconnaissance de cette propriété n'est cependant pas toujours complètement évidente...

Exercice 19. *Les formes propositionnelles suivantes sont-elles des tautologies ?*

1. $(p \wedge q) \longrightarrow p$
2. $(p \vee q) \longrightarrow (p \wedge q)$
3. $(p \wedge q) \longrightarrow (p \vee q)$
4. $p \longrightarrow (p \vee q)$
5. $p \longrightarrow ((\neg p) \longrightarrow p)$
6. $p \longrightarrow (p \longrightarrow q)$
7. $p \longrightarrow (p \longrightarrow p)$
8. $(p \longrightarrow q) \longrightarrow ((q \longrightarrow r) \longrightarrow (p \longrightarrow r))$

Réponses : 1, 3, 4, 5, 7 et 8 sont des tautologies.

2.2 Antilogies

DÉFINITION 5 (ANTILOGIE). *Toute forme 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 10. Soit $F = A \wedge \neg A$.
$\Phi_F(a) = a \cdot \overline{a} = 0$. Donc F est bien une antilogie.

REMARQUE 12. Le caractère d'antilogie d'une forme propositionnelle n'est pas toujours aussi évident.

2.3 Conséquences logiques

Soit $\mathcal{F}$ un ensemble de formes propositionnelles.

DÉFINITION 6 (CONSÉQUENCE LOGIQUE). *On dit que la forme propositionnelle A est* conséquence logique *des formules de l'ensemble $\mathcal{F}$ lorsque, chaque fois que les fonctions de vérité des formes de l'ensemble $\mathcal{F}$ 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 : $\mathcal{F} \models \mathcal{A}$ (se lit : A est conséquence logique de $\mathcal{F}$).

EXEMPLE 11. $\{P \longrightarrow Q, P\} \models Q$. En effet :

P	Q	$P \longrightarrow Q$
F	F	V
F	V	V
V	F	F
V	V	V

Il n'y a qu'un seul cas dans lequel $P \longrightarrow Q$ et P sont simultanément vrai. Dans ce cas, Q est vrai.

Exercice 20. *Dans chacun des cas suivants, que peut-on dire d'une forme propositionnelle :*

1. *qui a pour conséquence une antilogie,*
2. *qui a pour conséquence une tautologie,*
3. *qui est conséquence d'une antilogie,*
4. *qui est conséquence d'une tautologie.*

Réponses : 1) c'est une antilogie, 2) rien, 3) rien, 4) c'est une tautologie.

EXEMPLE 12. On reconsidère l'ensemble des deux formes propositionnelles

$$\{P, P \longrightarrow Q\}$$

et on va montrer autrement que Q est conséquence logique de ces deux formes.

Autrement dit, on va remontrer que : $\{P, P \longrightarrow Q\} \models Q$.

- $\Phi_P(p) = p$: prend la valeur 1 lorsque p prend la valeur 1.
- $\Phi_{P \longrightarrow Q}(p, q) = \overline{p} + q$: 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 $\Phi_{P \longrightarrow Q}(p, q)$ 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 $\{P, P \longrightarrow Q\}$.

Exercice 21. *Dans chacun des cas suivants, déterminer si la première forme a pour conséquence logique la deuxième forme (celle qui est sur la même ligne) :*

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

Réponse : oui pour 1, 2, 3, 5, 7, 9.

2.4 Formes équivalentes

DÉFINITION 7 (FORMES ÉQUIVALENTES). *Si la forme propositionnelle G est conséquence logique de la forme propositionnelle F et si F est aussi conséquence logique de G,*

Alors ces deux formes sont dites équivalentes*, soit :*

$$\{F\} \models G \text{ et } \{G\} \models F \Longleftrightarrow F \approx G.$$

C'est cette notion de formes équivalentes qui autorise le remplacement d'une expression par une autre (équivalente, bien sûr) dans une forme propositionnelle.

EXEMPLE 13. On est autorisé à remplacer $\neg\neg A$ par A, puisque ces formes sont équivalentes.

2.5 Exercices

Exercice 22. *Dans chacun des cas suivants, dire si les deux formes propositionnelles inscrites sur la même ligne sont équivalentes :*

1	$\neg(\neg p)$	p
2	$p \wedge (p \longrightarrow q)$	$p \wedge q$
3	$p \longrightarrow q$	$(\neg p) \vee (p \wedge q)$
4	$p \longrightarrow q$	$(\neg p) \longrightarrow (\neg q)$
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	$p \longrightarrow (q \longrightarrow r)$	$(p \longrightarrow q) \longrightarrow r$
9	$p \longrightarrow (q \wedge r)$	$(p \longrightarrow q) \wedge (p \longrightarrow r)$
10	$p \longrightarrow (q \vee r)$	$(p \longrightarrow q) \vee (p \longrightarrow r)$
11	$(p \longrightarrow q) \wedge (q \longrightarrow p)$	$(p \wedge q) \longrightarrow (p \wedge q)$
12	$(p \wedge q) \vee (q \wedge r) \vee (r \wedge p)$	$(p \vee q) \wedge (q \vee r) \wedge (p \vee r)$

Réponse : oui pour 1, 2, 3, 5, 6, 7, 9, 10, 12.

Exercice 23. *La forme propositionnelle f étant fixée, que peut-on dire d'une forme propositionnelle g qui possède chacune des deux propriétés :*

- *$f \vee g$ est une tautologie,*
- *$f \wedge g$ est une antilogie.*

Réponse : la forme g est équivalente à la négation de la forme f.

Exercice 24. *Soit f une forme 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 $f(p, q, r)$ cha nge 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
V	F	F	V
F	V	V	F
F	V	F	V
F	F	V	V
F	F	F	F

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

Exercice 25. *Déterminer des formes 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

Réponses :
$f = (p \wedge q \wedge r) \vee (p \wedge (\neg q) \wedge r) \vee ((\neg p) \wedge q \wedge (\neg r)) \vee ((\neg p) \wedge (\neg q) \wedge r) \vee ((\neg p) \wedge (\neg q) \wedge (\neg r))$
$g = (p \wedge q \wedge (\neg r)) \vee (p \wedge (\neg q) \wedge r) \vee ((\neg p) \wedge q \wedge (\neg r)) \vee ((\neg p) \wedge (\neg q) \wedge r)$
$h = (p \wedge q \wedge r) \vee (p \wedge q \wedge (\neg r)) \vee (p \wedge (\neg q) \wedge r) \vee ((\neg p) \wedge q \wedge (\neg r)) \vee ((\neg p) \wedge (\neg q) \wedge r) \vee ((\neg p) \wedge (\neg q) \wedge (\neg r))$

3 Simplification du calcul des fonctions de vérité

3.1 Théorème de substitution

PROPRIÉTÉ IV (THÉORÈME DE SUBSTITUTION) : Soit F une forme propositionnelle dans laquelle interviennent les variables propositionnelles P_1, P_2, P_3,..., P_n.

Supposons que l'on remplace ces variables par des formes propositionnelles G_1 , G_2 , G_3 ,... , G_n ; la nouvelle forme 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 propositionnelles, qui peuvent donc être remplacées par n'importe quelle fonction booléenne. ■

Attention, la réciproque n'est pas vraie...,

EXEMPLE 14. Soit $F^* = P \wedge \neg P \longrightarrow Q$.

La fonction de vérité est $\Phi_{F^*}(p,q) = \overline{p \cdot \overline{p}} + q = \overline{0} + q = 1 + q = 1$: il s'agit d'une tautologie.

Mais, si $F = A \longrightarrow B$, F n'est pas une tautologie, et F^* est obtenue à partir de F en remplaçant A par $P \wedge \neg P$ et B par Q...

Exemple d'utilisation de ce résultat :

EXEMPLE 15. Soit

$$F^* = ((P \longrightarrow Q \wedge \neg R) \vee (\neg S \longleftrightarrow T)) \longrightarrow ((P \longrightarrow Q \wedge \neg R) \vee (\neg S \longleftrightarrow T)),$$

c'est-à-dire une forme compliquée à 5 variables ; il y a donc 32 lignes à calculer pour obtenir les valeurs de la fonction de vérité.

Il suffit de remarquer que F^* est obtenue à partir de $F = A \longrightarrow A$, qui est une tautologie ; donc F^* en est une aussi.

Ce résultat peut évidemment être appliqué aussi à des parties de formes propositionnelles, pour accélérer le calcul de leurs fonctions de vérité :

Si une partie d'une forme propositionnelle constitue à elle seule une tautologie, la partie correspondante de la fonction de vérité peut être avantageusement remplacée par 1.

3.2 Théorème de la validité

PROPRIÉTÉ V (THÉORÈME DE LA VALIDITÉ) : Soit $\{G_1, G_2, \ldots, G_n\}$ un ensemble de formes propositionnelles et H une forme propositionnelle ; alors :

$\{G_1, G_2, \ldots, G_n\} \models H$ si et seulement si $\{G_1, G_2, \ldots, G_{n-1}\} \models G_n \longrightarrow H$

PREUVE Hypothèse $\{G_1, G_2, \ldots, G_n\} \models H$.

(C'est à dire, chaque fois que les formes de $\{G_1, G_2, \ldots, G_n\}$ sont vraies, H l'est aussi).

Supposons que les formes de $\{G_1, G_2, \ldots, G_{n-1}\}$ soient vraies :

- Alors, si G_n est vraie, toutes les formes 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), $G_n \longrightarrow H$ est vraie.
- Et si G_n n'est pas vraie, alors $G_n \longrightarrow H$ est vraie.

Donc, dans tous les cas, chaque fois que les formes de $\{G_1, G_2, \ldots, G_{n-1}\}$ sont vraies, $G_n \longrightarrow H$ est vraie. C'est à dire : $\{G_1, G_2, \ldots, G_{n-1}\} \models G_n \longrightarrow H$.

Hypothèse : Réciproquement, supposons $\{G_1, G_2, \ldots, G_{n-1}\} \models G_n \longrightarrow H$. Chaque fois que les formes de $\{G_1, G_2, \ldots, G_{n-1}\}$ sont vraies, $G_n \longrightarrow H$ est vraie.

Et $G_n \longrightarrow H$ est vraie dans deux cas :

- 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 formes 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 formes 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 16 (EXEMPLE D'APPLICATION). Soit à montrer que :

$$\models (A \longrightarrow (B \longrightarrow C)) \longrightarrow ((A \longrightarrow B) \longrightarrow (A \longrightarrow C)).$$

On pourrait bien entendu déterminer la fonction de vérité de cette forme.

Mais, d'après le théorème précédent, la démonstration du résultat demandé est équivalente à celle de :

$$\{A \longrightarrow (B \longrightarrow C)\} \models (A \longrightarrow B) \longrightarrow (A \longrightarrow C).$$

Une nouvelle application de ce même théorème nous montre que la démonstration demandée est encore équivalente à celle de :

$$\{A \longrightarrow (B \longrightarrow C), (A \longrightarrow B)\} \models (A \longrightarrow C).$$

Et enfin à celle de :

$$\{A \longrightarrow (B \longrightarrow C), (A \longrightarrow B), A\} \models C.$$

Or, dire que les formes de $\{A \longrightarrow (B \longrightarrow C), (A \longrightarrow B), A\}$ sont simultanément vraies revient à dire que A est vraie.

Dans ce cas, $(A \longrightarrow B)$ ne peut être aussi vraie que si B est vraie et, de même, $A \longrightarrow (B \longrightarrow C)$ ne peut être vraie que si $(B \longrightarrow C)$ est vraie. B étant vraie, $(B \longrightarrow C)$ ne peut être vraie que si C est vraie.

Donc, chaque fois que $\{A \longrightarrow (B \longrightarrow C), (A \longrightarrow B), A\}$ sont simultanément vraies, C est nécessairement vraie. Donc

$$\{A \longrightarrow (B \longrightarrow C), (A \longrightarrow B), A\} \models C,$$

ce qui, d'après le théorème énoncé ci-dessus, est équivalent à

$$\models (A \longrightarrow (B \longrightarrow C)) \longrightarrow ((A \longrightarrow B) \longrightarrow (A \longrightarrow C)).$$

La propriété est établie.

---

4 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 forme 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 paragraphe suivant.

5 Exercices

5.1 Sur les fonctions de vérité

Exercice 26. *Prouver les tautologies suivantes*

1. $\models A \longrightarrow (B \longrightarrow A)$
2. $\models (A \longrightarrow B) \longrightarrow ((A \longrightarrow (B \longrightarrow C)) \longrightarrow (A \longrightarrow C))$
3. $\models A \longrightarrow (B \longrightarrow A \wedge B)$
4. $\models A \wedge B \longrightarrow A$ $\qquad\qquad$ $\models A \wedge B \longrightarrow B$
5. $\models A \longrightarrow A \vee B$
6. $\models B \longrightarrow A \vee B$
7. $\models (A \longrightarrow C) \longrightarrow ((B \longrightarrow C) \longrightarrow (A \vee B \longrightarrow C))$
8. $\models (A \longrightarrow B) \longrightarrow ((A \longrightarrow \neg B) \longrightarrow \neg A)$
9. $\models \neg\neg A \longrightarrow A$
10. $\models (A \longrightarrow B) \longrightarrow ((B \longrightarrow A) \longrightarrow (A \longleftrightarrow B))$
11. $\models (A \longleftrightarrow B) \longrightarrow (A \longrightarrow B)$ $\quad$ $\models (A \longleftrightarrow B) \longrightarrow (B \longrightarrow A)$

Exercice 27. *Calculer les fonctions de vérité des formes propositionnelles suivantes, et dire s'il s'agit éventuellement de tautologies ou d'antilogies :*

– $(A \longrightarrow B) \wedge (A \vee B) \longrightarrow B$
– $(A \longrightarrow C) \wedge (B \longrightarrow D) \wedge (A \vee B) \longrightarrow C \vee D$
– $\neg(A \wedge B) \vee \neg A \vee \neg B \longrightarrow C$
– $(A \longrightarrow C) \vee (B \longrightarrow D) \longrightarrow (A \vee B \longrightarrow C \vee D)$
– $(A \longrightarrow C) \wedge (B \longrightarrow D) \longrightarrow (A \wedge B \longrightarrow C \wedge D)$
– $(A \wedge B) \vee (\neg A \wedge \neg C) \longrightarrow (B \longrightarrow C)$
– $(\neg(B \wedge C) \wedge \neg\neg(C \wedge A) \longrightarrow \neg(B \wedge C) \wedge (C \wedge A)) \longrightarrow (\neg(A \vee \neg C) \longleftrightarrow (A \longrightarrow B))$
– $(\neg A \vee B) \wedge (C \longrightarrow (A \longleftrightarrow B))$
– $A \wedge \neg A \longrightarrow (B \vee C \longrightarrow (C \longrightarrow \neg A))$
– $((A \longrightarrow B) \longrightarrow A) \longrightarrow A$
– $(A \longrightarrow C) \wedge (B \longrightarrow D) \wedge (\neg C \vee \neg D) \longrightarrow \neg A \vee \neg B$
– $A \wedge (A \vee B) \longleftrightarrow A$
– $(\neg A \vee B \longrightarrow (A \longrightarrow \neg A \vee B)) \longleftrightarrow (\neg A \vee B \longrightarrow (A \longrightarrow (A \longrightarrow B)))$
– $(A \longrightarrow B) \wedge (A \vee C) \longrightarrow B \vee C$
– $(A \longrightarrow B) \wedge (A \vee C) \longrightarrow (A \longrightarrow C)$

Exercice 28 (Le club des Écossais). *Pour constituer un club, on a énoncé le règlement suivant :*

– *Article 1 : Tout membre non Écossais porte des chaussettes oranges.*
– *Article 2 : Tout membre porte une jupe ou ne porte pas de chaussettes oranges.*
– *Article 3 : Les membres mariés ne sortent pas le dimanche.*
– *Article 4 : Un membre sort le dimanche si et seulement s'il est Écossais.*
– *Article 5 : Tout membre qui porte une jupe est Écossais et marié.*
– *Article 6 : Tout membre Écossais porte une jupe.*

Déterminer le nombre de membres de ce club.

5.2 Application des fonctions de vérité

Exercice 29. *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".*

Ces trois témoignages sont-ils compatibles ? En supposant qu'ils sont tous les trois innocents, lequel a menti ? En supposant que chacun dit la vérité, qui est innocent et qui est coupable ? En supposant que les innocents disent la vérité et que les coupables mentent, qui est innocent et qui est coupable ?

Exercice 30. *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 31. *Un inspecteur des services de santé visite un hôpital psychiâtrique 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.

- *Après une intense réflexion, l'inspecteur fait sortir l'un des deux de l'hôpital. Lequel (et pourquoi ?)*
- *Peut-il dire quelque chose au sujet de l'autre ?*

Exercice 32. *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 lorqu'on énonce une proposition vraie, et l'autre si on énonce une proposition fausse.*

Comment peut-il délivrer la princesse ?

IV. Deuxième point de vue : théorie de la démonstration

1 Présentation

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.

Nous disposons en fait d'une « base de connaissances » qui fait que nous savons que, dans telles ou telles circonstances, certaines causes produisent certains effets. Pour obtenir ces effets, nous essayons de nous replacer dans les conditions de leurs réalisations.

1.1 Le système formel, ses règles d'inférences

Le système formel qui est chargé de reproduire mécaniquement ce fonctionnement est constitué d'*axiomes logiques*, qui jouent le rôle de « connaissances de base » ; il s'agit de formules de logique qui servent de « points de départ » aux déductions ultérieures.

Puis, le système est muni de « règles d'inférence », qui sont chargées de simuler les divers modes de raisonnement que nous utilisons. Elles se présentent sous la forme de règles qui, lors de la rencontre de formules d'un type donné, autorisent la production de nouvelles formules auxquelles on attachera le même crédit que celui qui est attribué aux formules dont elle sont issues. Ces formules auront le statut de nouveaux résultats considérés comme établis, et pourront venir enrichir la « base de connaissances », on les appellera des *théorèmes logiques*.

1.2 Status des axiomes logiques

Il convient de distinguer les axiomes logiques des autres axiomes qui peuvent être posés dans divers domaines, par exemple, en géométrie, l'axiome d'Euclide ; ces axiomes n'appartiennent pas à la logique et autorisent la construction d'une théorie (la géométrie euclidienne en l'occurence).

Il faut aussi les distinguer des axiomes posés au sujet de la logique elle-même, comme, par exemple, celui que nous avons appelé le principe de non-contradiction.

Ces axiomes énoncés *au sujet de* la logique, ne sont pas non plus des axiomes logiques.

Ils jouent le même rôle, pour la logique, que l'axiome d'Euclide pour la géométrie : ils conduisent à la construction d'une certaine logique, étant bien entendu que d'autres axiomes peuvent conduire à la construction d'autres logiques que celle qui est l'objet du présent chapitre.
Plus précisément :

Théorème logique

DÉFINITION 8 (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. ♢

DÉFINITION 9 (DÉMONSTRATION). *La chaîne de déductions qui conduit à un théorème logique est appelée* démonstration *de ce résultat.* ♢

DÉFINITION 10 (AXIOME LOGIQUE). *Cette chaîne peut éventuellement ne comporter qu'un seul élément.*
Dans ce cas, le « résultat » est un axiome logique. ♢

Un axiome logique est donc un théorème logique, et il ne se démontre pas.

NOTATION : On exprime que la formule F est un théorème par la notation : $\vdash F$, qui se lit « F est un théorème ».

Déduction sous hypothèse 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).

DÉFINITION 11. *On parle alors, non plus de démonstration, mais de déduction sous hypothèses*. ♢

NOTATION : L'expression : « la formule logique H est obtenue par déduction sous les hypothèses $G_1, G_2, \ldots, G_n$ » est notée : $\{G_1, G_2, \ldots, G_n\} \vdash H$.

2 Les axiomes logiques

Il existe plusieurs systèmes d'axiomes qui permettent de construire la Logique propositionnelle.

Nous nous en tiendrons aux axiomes et règles d'inférence suivants.

Axiomes relatifs à l'implication logique :
- Axiome 1 : $P \longrightarrow (Q \longrightarrow P)$
- Axiome 2 : $(P \longrightarrow Q) \longrightarrow ((P \longrightarrow (Q \longrightarrow R)) \longrightarrow (P \longrightarrow R))$

Axiomes relatifs à la conjonction logique :
- Axiome 3 : $P \longrightarrow (Q \longrightarrow P \wedge Q)$
- Axiome 4 : $P \wedge Q \longrightarrow P$
- Axiome 5 : $P \wedge Q \longrightarrow Q$

Axiomes relatifs à la disjonction logique :
- Axiome 6 : $P \longrightarrow P \vee Q$
- Axiome 7 : $Q \longrightarrow P \vee Q$
- Axiome 8 : $(P \longrightarrow R) \longrightarrow ((Q \longrightarrow R) \longrightarrow (P \vee Q \longrightarrow R))$

Axiomes relatifs à la négation logique :
- $A_9 = \neg\neg P \longrightarrow P$
- $A_{10} = (P \longrightarrow Q) \longrightarrow ((P \longrightarrow \neg Q) \longrightarrow \neg P)$

Axiomes relatifs à l'équivalence logique :
- Axiome 11 : $(P \longrightarrow Q) \longrightarrow ((Q \longrightarrow P) \longrightarrow (P \longleftrightarrow Q))$
- Axiome 12 : $(P \longleftrightarrow Q) \longrightarrow (P \longrightarrow Q)$
- Axiome 13 : $(P \longleftrightarrow Q) \longrightarrow (Q \longrightarrow P)$

REMARQUE 13. Il ne s'agit pas là d'un système d'axiomes minimal, mais il n'est pas contradictoire.

3 Les règles d'inférence

La logique classique utilise les règles d'inférence suivantes :
- le « modus ponendo ponens » (le mode « en posant, on pose ») :
  $\{P, P \longrightarrow Q\} \vdash Q$.
  « Des formules P et $P \longrightarrow Q$, on peut déduire par modus ponens la formule Q ».
- le « modus tollendo tollens » (le mode « en supprimant, on supprime ») :
  $\{P \longrightarrow Q, \neg Q\} \vdash \neg P$.
- le » modus ponendo tollens » (le mode « en posant, on supprime ») :
  $\{\neg(P \wedge Q), P\} \vdash \neg Q$.
- le « modus tollendo ponens » (le mode « en supprimant, on pose ») :
  $\{P \vee Q, \neg P\} \vdash Q$.

REMARQUE 14. Les noms sont traditionnellement des noms latins, utilisés depuis les philosophes du XVI$^{\text{ème}}$ siècle.

Ces noms utilisent une forme verbale - le gérondif - qui n'a pas d'équivalent en français, et constituent donc des latinismes intraduisibles.

Une brève étude montrerait que ces quatre règles sont équivalentes.

Il est donc possible de n'en conserver qu'une seule, qu'on appelle le « modus (sous-entendu : ponendo) ponens », ou *règle de détachement* et qui est la première.

REMARQUE 15. Sinon, il serait possible de supprimer quelques axiomes. Mais il vaut mieux, dans un but ultime de programmation, diminuer le nombre de règles d'inférence que celui d'axiomes.

Cependant, dans le but de raccourcir au maximum les démonstrations ou déductions à faire à la main, nous nous conformerons à l'usage et conserverons les quatre règles classiques.

4 Démonstrations et déductions sous hypothèses

Le raisonnement logique peut prendre deux formes : la démonstration, et la déduction sous hypothèses.

4.1 Cas de la démonstration

Une démonstration (la démonstration d'un théorème) est constituée :

1. d'un en-tête, portant l'indication « Démonstration » (de manière à l'isoler totalement du contexte),
2. puis d'un certain nombre de lignes, numérotées (pour pouvoir être référencées dans la suite). Chacune d'entre elles doit comporter deux champs :
   – Le premier indique pourquoi le résultat suivant peut être avancé (il est indispensable pour pouvoir juger de la correction de la démonstration).
   – Le second consiste en une formule, qui est le « résultat » de la ligne courante.
3. La conclusion de la démonstration est la formule écrite sur la dernière ligne.
4. Elle est répétée dans une dernière ligne, non numérotée, qui porte l'en-tête « conclusion ».

Dans une ligne, on peut avancer :
– un axiome,
– un théorème considéré comme connu (dont la démonstration a été vue par ailleurs),
– le résultat de l'application d'une règle d'inférence sur des formules qui sont écrites dans les lignes précédentes.

4.2 Cas de la déduction sous hypothèses

Une déduction sous hypothèses...

1. Commence par une première ligne qui comporte les mots « Déduction sous les hypothèses ».
2. Cette première ligne est suivie de l'écriture de l'ensemble des hypothèses utilisées...
3. Puis, comme dans une démonstration, de lignes numérotées dans lesquelles peuvent figurer les mêmes éléments, auxquels il faut rajouter les hypothèses, dont on a le droit de se servir comme s'il s'agissait de résultats établis.
4. La ligne indiquant la conclusion doit rappeler les hypothèses.

Dans la pratique, dans un but d'économie de place (la moindre déduction peut très vite prendre de nombreuses lignes), on s'autorisera d'autres éléments (par exemple, l'utilisation de la technique de l'hypothèse supplémentaire), qui seront vus dans la suite.

4.3 Exemples

Voici, par exemple, comment on peut obtenir un « modus (tollendo) tollens » à l'aide du « modus ponens » et des axiomes (il s'agit donc de déduire $\neg P$ sous les hypothèses $P \longrightarrow Q$ et $\neg Q$) :

EXEMPLE 17 (MODUS (TOLLENDO) TOLLENS). Déduction sous les hypothèses

$$\{P \longrightarrow Q\,,\ \neg Q\}$$

:

1	Axiome 10	$(P \longrightarrow Q) \longrightarrow ((P \longrightarrow \neg Q) \longrightarrow \neg P)$
2	Hypothèse	$P \longrightarrow Q$
3	m.p. sur 1, 2	$(P \longrightarrow \neg Q) \longrightarrow \neg P$
4	Axiome 1	$\neg Q \longrightarrow (P \longrightarrow \neg Q)$
5	Hypothèse	$\neg Q$
6	m.p. sur 4, 5	$(P \longrightarrow \neg Q)$
7	m.p. sur 3, 6	$\neg P$

Conclusion : $\neg P$.

Autre exemple : la démonstration du théorème appelé *théorème d'idempotence*

$$P \longrightarrow P$$

Ou encore :

$$\vdash (P \longrightarrow P)$$

EXEMPLE 18 (THÉORÈME D'IDEMPOTENCE). Démonstration :

1	Axiome 2	$(P \longrightarrow (Q \longrightarrow P)) \longrightarrow ((P \longrightarrow ((Q \longrightarrow P) \longrightarrow P)) \longrightarrow (P \longrightarrow P))$ (en remplaçant Q par $Q \longrightarrow P$ et R par P
2	Axiome 1	$P \longrightarrow (Q \longrightarrow P)$
3	m.p. sur 1, 2	$(P \longrightarrow ((Q \longrightarrow P) \longrightarrow P)) \longrightarrow (P \longrightarrow P)$
4	Axiome 1	$P \longrightarrow ((Q \longrightarrow P) \longrightarrow P)$ (en remplaçant Q par $Q \longrightarrow P$)
5	m.p. sur 3, 4	$(P \longrightarrow P)$

Conclusion : $\vdash (P \longrightarrow P)$

Les démonstrations sont souvent considérablement simplifiées par l'utilisation du théorème de la déduction (mais la démonstration de ce dernier utilise le théorème d'idempotence).

Exercice 33. *En théorie de la démonstration, démontrer les règles d'inférence :*

1. *« modus tollendo ponens » :* $\{\neg A, A \vee B\} \vdash B$
2. *« modus ponendo tollens » :* $\{A, \neg(A \wedge B)\} \vdash \neg B$.

Pour la première, on pourra utiliser le théorème de la contradiction et, pour la seconde, le théorème de la contraposée.

---

5 Théorème de la déduction

5.1 Le théorème

Il s'agit du résultat qui équivaut, en théorie de la démonstration, au théorème de la validité en théorie des valeurs de vérité.

PROPRIÉTÉ VI (THÉORÈME DE LA DÉDUCTION) : Ce théorème s'énonce par :
$\{G_1, G_2, \ldots, G_n\} \vdash H$ si et seulement si $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$

5.2 Démonstration

La démonstration s'effectue par récurrence sur la longueur de la déduction.
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 :

    Déduction sous les hypothèses $\{G_1, G_2, \ldots, G_{n-1}\}$:

1	Axiome 1	$H \longrightarrow (G_n \longrightarrow H)$
2	Axiome j	H
3	m.p. sur 1, 2	$G_n \longrightarrow H$

    Conclusion : $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$

    Dans ce premier cas : $\{G_1, G_2, \ldots, G_n\} \vdash H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$ (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 $H = G_i$ $(0 < i < n)$:

    Déduction sous les hypothèses $\{G_1, G_2, \ldots, G_{n-1}\}$:

1	Axiome 1	$G_i \longrightarrow (G_n \longrightarrow G_i)$
2	Hypothèse	G_i
3	m.p. sur 1, 2	$G_n \longrightarrow G_i$

    Conclusion : $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$

    Dans ce deuxième cas : $\{G_1, G_2, \ldots, G_n\} \vdash H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$ (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 : $\vdash G_n \longrightarrow G_n$ (voir paragraphe précédent).

    Dans ce troisième cas : $\{G_1, G_2, \ldots, G_n\} \vdash H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$.

  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 $P \longrightarrow H$. 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 $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow P$ et que $\{G_1, G_2, \ldots, G_n\} \vdash P \longrightarrow H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow$

$(P \longrightarrow H)$.

Déduction sous les hypothèses $\{G_1, G_2, \ldots, G_{n-1}\}$:

1	Résultat intermédiaire 1	$G_n \longrightarrow P$
2	Résultat intermédiaire 2	$G_n \longrightarrow (P \longrightarrow H)$
3	Axiome 2	$(G_n \longrightarrow P) \longrightarrow ((G_n \longrightarrow (P \longrightarrow H)) \longrightarrow (G_n \longrightarrow H))$
4	m.p. sur 1, 3	$(G_n \longrightarrow (P \longrightarrow H)) \longrightarrow (G_n \longrightarrow H)$
5	m.p. sur 2, 4	$G_n \longrightarrow H$

Conclusion $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$,

et donc : $\{G_1, G_2, \ldots, G_n\} \vdash H$ implique $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$,

lorsque la déduction est de longueur $p + 1$.

Le résultat est établi.

Hypothèse : Réciproquement, supposons $\{G_1, G_2, \ldots, G_{n-1}\} \vdash (G_n \longrightarrow H)$. Alors,

Déduction sous les hypothèses $\{G_1, G_2, \ldots, G_n\}$

1	Résultat obtenu sous les hyp $\{G_1, G_2, \ldots, G_{n-1}\}$	$G_n \longrightarrow H$
2	Hypothèse n	G_n
3	m.p. sur 1, 2	H

Conclusion $\{G_1, G_2, \ldots, G_n\} \vdash H$

Donc : $\{G_1, G_2, \ldots, G_{n-1}\} \vdash G_n \longrightarrow H$ entraîne $\{G_1, G_2, \ldots, G_n\} \vdash H$.

Le théorème est établi.

5.3 Exemples d'utilisation du théorème de la déduction

EXEMPLE 19. Le « modus tollens » $\{P \longrightarrow Q, \neg Q\} \vdash$

EXEMPLE 20. Soit à démontrer : $\vdash (A \longrightarrow (B \longrightarrow C)) \longrightarrow (B \longrightarrow (A \longrightarrow C))$.

La démonstration de ce théorème équivaut à la déduction

$$\{A \longrightarrow (B \longrightarrow C)\} \vdash (B \longrightarrow (A \longrightarrow C))$$

Cette dernière est équivalente à la déduction

$$\{A \longrightarrow (B \longrightarrow C), B\} \vdash (A \longrightarrow C)$$

Elle-même équivalente à la déduction :

$$\{A \longrightarrow (B \longrightarrow C), B, A\} \vdash C$$

établissons ce résultat :

Déduction sous les hypothèses $\{A \longrightarrow (B \longrightarrow C), B, A\}$

1	Hypothèse 3	A
2	Hypothèse 1	$A \longrightarrow (B \longrightarrow C)$
3	m.p. sur 1, 2	$B \longrightarrow C$
4	Hypothèse 2	B
5	m.p. sur 3, 4	C

Conclusion $\{A \longrightarrow (B \longrightarrow C), B, A\} \vdash C$.

Déduction équivalente à la démonstration du *théorème d'échange des prémisses* :

$$\vdash (A \longrightarrow (B \longrightarrow C)) \longrightarrow (B \longrightarrow (A \longrightarrow C)).$$

REMARQUE 16. 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.

EXEMPLE 21. On a démontré la règle d'inférence du « modus (tollendo) tollens » :

$$\{P \longrightarrow Q, \neg Q\} \vdash \neg P.$$

D'après le théorème de la déduction, ce dernier résultat est équivalent à :

$$\{P \longrightarrow Q\} \vdash (\neg Q \longrightarrow \neg P).$$

Une nouvelle application de ce même théorème donne :

$$\vdash (P \longrightarrow Q) \longrightarrow (\neg Q \longrightarrow \neg P).$$

DÉFINITION 12 (CONTRAPOSÉE). *L'implication* $\neg Q \longrightarrow \neg P$ *est appelée* contraposée *de l'implication* $P \longrightarrow Q$. ♢

PROPRIÉTÉ VII (THÉORÈME DE LA CONTRAPOSÉE) : Le théorème

$$\vdash (P \longrightarrow Q) \longrightarrow (\neg Q \longrightarrow \neg P)$$

sera utilisé dans la suite sous le nom de *théorème de la contraposée*.

REMARQUE 17. L'utilisation principale du théorème de la déduction consiste à remplacer la démonstration d'implication par des déductions sous hypothèses.

6 Quelques théorèmes classiques et quelques règles d'inférence annexes

Nous avons déjà évoqué le théorème d'idempotence ($\vdash P \longrightarrow P$), le théorème de la contraposée, et le théorème d'échange des prémisses. Citons encore...

6.1 Théorème de transitivité de l'implication

PROPRIÉTÉ VIII (THÉORÈME DE TRANSITIVITÉ DE L'IMPLICATION) : On a
$$\vdash (A \longrightarrow B) \longrightarrow ((B \longrightarrow C) \longrightarrow (A \longrightarrow C)).$$

PREUVE $\vdash (A \longrightarrow B) \longrightarrow ((B \longrightarrow C) \longrightarrow (A \longrightarrow C))$
équivalent $\{A \longrightarrow B\} \vdash (B \longrightarrow C) \longrightarrow (A \longrightarrow C)$
équivalent $\{A \longrightarrow B, B \longrightarrow C\} \vdash A \longrightarrow C$
équivalent $\{A \longrightarrow B, B \longrightarrow C, A\} \vdash C$
Cette déduction est évidente avec le modus ponens, avec :

1. $A \longrightarrow B$ et B,
2. $B \longrightarrow C$ et B obtenu en 1,
3. on obtient C en 2. ■

6.2 Théorème de la contradiction

Déduction sous les hypothèses $\{A\,,\,\neg A\}$

1	Hypothèse 1	A
2	Axiome 1	$A \longrightarrow (\neg B \longrightarrow A)$
3	m.p. sur 1, 2	$\neg B \longrightarrow A$
4	Hypothèse 2	$\neg A$
5	Axiome 1	$\neg A \longrightarrow (\neg B \longrightarrow \neg A)$
6	m.p. sur 4, 5	$\neg B \longrightarrow \neg A$
7	Axiome 10	$(\neg B \longrightarrow A) \longrightarrow ((\neg B \longrightarrow \neg A) \longrightarrow \neg\neg B)$
8	m.p. sur 4, 7	$(\neg B \longrightarrow \neg A) \longrightarrow \neg\neg B$
9	m.p. sur 6, 8	$\neg\neg B$
10	Axiome 9	$\neg\neg B \longrightarrow B$
11	m.p. sur 9, 10	B

Conclusion $\{A\,,\,\neg A\} \vdash B$.
Ce qui est équivalent, d'après le théorème de la déduction, au théorème
$$\vdash A \longrightarrow (\neg A \longrightarrow B)$$
ou encore...

PROPRIÉTÉ IX (THÉORÈME DE LA CONTRADICTION) : On a
$$\vdash \neg A \longrightarrow (A \longrightarrow B)$$

6.3 Règle de disjonction des cas

Considérons l'axiome 8

$$\vdash (P \longrightarrow R) \longrightarrow ((Q \longrightarrow R) \longrightarrow (P \vee Q \longrightarrow R))$$

En appliquant deux fois de suite le théorème de la déduction, il est équivalent à la déduction

$$\{P \longrightarrow R\,,\, Q \longrightarrow R\} \;\vdash\; P \vee Q \longrightarrow R$$

que l'on peut utiliser sous cette forme comme règle d'inférence annexe : elle s'appelle *règle de disjonction des cas*.

PROPRIÉTÉ X (RÈGLE DE DISJONCTION DES CAS) : On a

$$\{P \longrightarrow R\,,\, Q \longrightarrow R\} \;\vdash\; P \vee Q \longrightarrow R$$

6.4 Règle de réduction à l'absurde

Pour finir, l'axiome 10

$$\vdash (P \longrightarrow Q) \longrightarrow ((P \longrightarrow \neg Q) \longrightarrow \neg P)$$

En appliquant deux fois de suite le théorème de la déduction, il est équivalent à la déduction

$$\{P \longrightarrow Q\,,\, P \longrightarrow \neg Q\} \vdash\; \neg P$$

que l'on peut utiliser sous cette forme comme règle d'inférence annexe : elle s'appelle *règle de réduction à l'absurde*.

PROPRIÉTÉ XI (RÈGLE DE RÉDUCTION À L'ABSURDE) : On a

$$\{P \longrightarrow Q\,,\, P \longrightarrow \neg Q\} \vdash\; \neg P$$

Exercice 34 (Démonstrations et déductions sous hypothèses). *Démontrer les théorèmes logiques suivants (seuls les axiomes, régles d'inférence, régles d'inférence annexes et théorèmes du cours sont autorisés)*

1. $\vdash (A \longrightarrow B) \longrightarrow ((B \longrightarrow C) \longrightarrow (A \longrightarrow C))$
2. $\vdash (A \longrightarrow (B \longrightarrow C)) \longleftrightarrow (B \longrightarrow (A \longrightarrow C))$
3. $\vdash (A \longrightarrow (B \longrightarrow C)) \longleftrightarrow (A \wedge B \longrightarrow C)$
4. $\vdash (A \longrightarrow B) \longleftrightarrow (\neg B \longrightarrow \neg A)$
5. $\vdash A \longleftrightarrow \neg\neg A$

6. $\vdash (A \longleftrightarrow B) \wedge (B \longleftrightarrow C) \longrightarrow (A \longleftrightarrow C)$
7. $\vdash A \wedge (B \vee C) \longleftrightarrow (A \wedge B) \vee (A \wedge C)$
8. $\vdash A \vee (B \wedge C) \longleftrightarrow (A \vee B) \wedge (A \vee C)$
9. $\vdash \neg(A \wedge \neg A)$
10. $\vdash \neg(A \vee B) \longleftrightarrow \neg A \wedge \neg B$
11. $\vdash \neg(A \wedge B) \longleftrightarrow \neg A \vee \neg B$
12. $\vdash A \vee B \longleftrightarrow \neg(\neg A \wedge \neg B)$
13. $\vdash A \wedge B \longleftrightarrow \neg(\neg A \vee \neg B)$
14. $\vdash (A \longrightarrow B) \longleftrightarrow \neg A \vee B$
15. $\vdash (A \longrightarrow B) \longleftrightarrow \neg(A \wedge \neg B)$
16. $\vdash \neg(A \longrightarrow B) \longleftrightarrow A \wedge \neg B$

---

7 Technique de l'hypothèse supplémentaire

7.1 Introduction et suppression d'une hypothèse supplémentaire

Supposons qu'au cours d'une démonstration ou déduction sous hypothèses, il arrive que l'on ne sache plus quoi écrire pour progresser en direction du résultat recherché, mais que l'utilisation d'une hypothèse qui ne figure pas dans l'ensemble déclaré au départ semble pouvoir débloquer la situation.

On admettra alors l'écriture d'une ligne contenant la déclaration de cette hypothèse, appelons-la H, avec la justification « Hypothèse supplémentaire ».

Cette hypothèse pourra alors être utilisée pour progresser, et pour atteindre, quelques lignes plus loin, un certain résultat, disons R.

La déduction a alors l'allure suivante :

Déduction sous les hypothèses $\{H_1\,,\,H_2\,,\,\ldots\,,\,H_n\}$

1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$
i	Hypothèse supplémentaire	H
$\vdots$	$\vdots$	$\vdots$
j	$\cdots$	R

Dans une telle déduction, le résultat obtenu est celui qui est écrit sur la dernière ligne écrite (même si ce n'est pas le résultat recherché !).

Autrement dit, si la déduction s'arrêtait à cet endroit, le résultat obtenu serait R mais, pour l'obtenir, une hypothèse non prévue au départ a été utilisée.

Le véritable résultat obtenu est en fait
Conclusion $\{H_1\,,\,H_2\,,\,\ldots\,,\,H_n\,,\,H\} \vdash R$.
(en rajoutant H à l'ensemble des hypothèses de départ).

Mais on sait que ce résultat est équivalent à :
Conclusion $\{H_1\,,\,H_2\,,\,\ldots\,,\,H_n\} \vdash H \longrightarrow R$.
C'est-à-dire un résultat établi en utilisant les seules hypothèses de départ.

On pourra donc rajouter une ligne contenant la justification : « Suppression de l'hypothèse supplémentaire » et comme résultat $H \longrightarrow R$ et, comme ce résultat est obtenu avec les seules hypothèses de départ, il est utilisable dans la suite (à la différence de ceux qui sont écrits dans les lignes i à j incluses).

On obtient
Déduction sous les hypothèses $\{H_1\,,\,H_2\,,\,\ldots\,,\,H_n\}$

1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$
i	Hypothèse supplémentaire	H
$\vdots$	$\vdots$	$\vdots$
j	$\cdots$	R
j+1	Suppression hyp. suppl. H	$H \longrightarrow R$
$\vdots$	$\vdots$	$\vdots$

On espère évidemment que ce dernier résultat a rendu le résultat recherché plus accessible.

Cependant, les lignes écrites dans le domaine de validité de l'hypothèse suplémentaire (de $i+1$ à j) ne peuvent pas être utilisées dans la suite de la déduction.

Dans certains cas complexes, il est possible de faire successivement plusieurs hypothèses supplémentaires.

Il faut dans ce cas respecter la régle selon laquelle, lorsque plusieurs hypothèses supplémentaires coexistent, elles doivent être supprimées dans l'ordre inverse de leur introduction.

7.2 Utilisation de cette technique dans une disjonction des cas

Lorsque le résulat à établir a la forme $\{A \vee B\} \vdash C$, où C est une formule quelconque, le raisonnement « par disjonction des cas » prend la forme suivante, lorsque le

résultat n'est pas immédiat.

Déduction sous l' hypothèse $\{A \vee B\}$

1	Hypothèse supplémentaire	A
⋮	⋮	⋮
i	...	C
i+1	Suppression hyp. suppl. A	$A \longrightarrow C$
i+2	Hypothèse supplémentaire	B
⋮	⋮	⋮
j	...	C
j+1	Suppression hyp. suppl. B	$B \longrightarrow C$
j+2	Disjonction des cas sur i+1, j+1	$A \vee B \longrightarrow C$
j+3	Hypothèse	$A \vee B$
j+4	m.p. sur j+2, j+3	C

Conclusion $\{A \vee B\} \vdash C$.

REMARQUE 18. Il faut veiller que ce soit bien le résultat final attendu qui soit obtenu avant la suppression des hypothèses supplémentaires.

8 Méthodes de démonstration

Il peut sembler *a priori* difficile de trouver les idées qui permettent de mener à bien une déduction sous hypothèses.

On pourra s'inspirer des principes suivants qui sont souvent d'une aide précieuse bien qu'on ne puisse affirmer qu'ils fournissent dans tous les cas la solution la plus courte, ni même la solution elle-même.

La première chose à faire est de faire intervenir au maximum le théorème de la déduction. On conseille ensuite d'observer...

Les hypothèses. Une hypothèse (ou un résultat intermédiaire) qui se présente sous la forme d'une

conjonction : s'utilise par application des axiomes 4 et 5

disjonction : s'utilise par une disjonction des cas (voir paragraphe précédent)

négation : s'utilise par application du théorème de la contraposée

implication logique : souvent la technique de l'hypothèse supplémentaire (utilisée aussi dans le cas précédent) permet de l'utiliser.

équivalence logique : s'utilise par application des axiomes 12 et 13

L'application de ces quelques principes permet en général de bien démarrer les déductions, par leurs premières lignes. On observe ensuite...

La conclusion. Si elle se présente sous la forme d'une

conjonction : elle peut s'obtenir par application de l'axiome 3.

disjonction : elle peut s'obtenir par application de l'axiome 6 ou de l'axiome 7.

négation : elle peut s'obtenir par une réduction à l'absurde.

implication : vous n'avez pas suivi les conseils d'utiliser le théorème de la déduction au maximum !

équivalence logique : elle peut s'obtenir par application de l'axiome 11 (concrètement, on effectue en général séparément les démonstrations des deux implications et on peut s'abstenir de « reconstruire » l'équivalence logique à partir des deux implications, de l'axiome 11 et de deux « modus ponens »).

L'application de ces quelques principes permet en général de découvrir un bon chemin vers le résultat, c'est-à-dire les dernières lignes de la déduction.

Au total, sauf dans des cas très compliqués, on devrait avoir ainsi l'intégralité de la déduction.

9 Exercices

Exercice 35 (Validité de raisonnements). *Formaliser, puis étudier la validité des raisonnements suivants :*

1. *Un étudiant ne peut résoudre un exercice si on ne lui dit pas comment faire. On ne lui dit pas comment faire s'il ne pose pas de questions. Il a trouvé, donc il a posé des questions.*
2. *Ce qui est compréhensible ne m'intrigue jamais ; la logique m'intrigue. Donc, la logique est incompréhensible.*
3. *Aucun professeur n'est ignorant ; les gens ignorants sont ennuyeux. Donc, aucun professeur n'est ennuyeux.*
4. *Tout professeur de mathématiques est logique ; un homme illogique est toujours têtu. Donc, aucun professeur de mathématiques n'est têtu.*
5. *Un parapluie est utile en voyage ; toute chose inutile en voyage doit être laissée à la maison. Il faut donc emporter son parapluie en voyage.*
6. *Aucun problème ne m'intéresse s'il est possible de le résoudre ; tous ces problèmes m'intéressent. Ils sont donc impossibles à résoudre.*

Exercice 36. *Étude de raisonnements concrets :*

1. *Peut-on conclure quelque chose au sujet de la réussite de l'attaque envisagée dans les conditions suivantes ?*

   L'attaque réussira seulement si l'ennemi est surpris ou si la position est peu défendue. L'ennemi ne sera pas surpris, à moins qu'il ne soit téméraire. L'ennemi n'est pas téméraire lorsque la position est peu défendue.

2. *Peut-on conclure quelque chose au sujet de la culpabilité du suspect décrit dans les propositions suivantes ?*

   Si le suspect a commis le vol, celui-ci a été minutieusement préparé ou le suspect disposait d'un complice dans la place. Si le vol a été minutieusement préparé, alors, si le suspect avait un complice dans la place, le butin aurait été beaucoup plus important.

3. *La conclusion du raisonnement suivant est-elle valide ? :*

   À moins que nous ne continuions la politique de soutien des prix, nous perdrons les voix des agriculteurs. Si nous continuons cette politique, la surproduction se poursuivra, sauf si nous contingentons la production. Sans les voix des agriculteurs, nous ne serons pas réélus. Donc, si nous sommes réélus et que nous ne contingentons pas la production, la surproduction continuera.

4. *Quelles sont les conclusions que l'on peut tirer des renseignements suivants :*

   J'aime les tomates à la provençale, ou je suis né un 29 février, ou je sais jouer du cornet à pistons. Si je sais jouer du cornet à pistons, alors je suis né un 29 février ou j'aime les tomates à la provençale. Si je n'aime pas les tomates à la provençale et si je suis né un 29 février, alors je ne sais pas jouer du cornet à pistons. Je n'aime pas les tomates à la provençale ou je ne sais pas jouer du conet à pistons.

Exercice 37 (Un problème de logique). *Trois artisans coiffeurs (Aristide, Barnabé et Clotaire) tiennent un salon de coiffure. Celui-ci ne peut rester sans surveillance pendant les heures d'ouverture, donc l'un au moins des trois artisans est obligatoirement présent. On sait, de plus, qu'Aristide ne peut sortir seul ; lorsqu'il s'absente, il se fait obligatoirement accompagner par Barnabé.*

Oncle Jim et oncle Joe, deux sympathiques logiciens, se dirigent vers le salon, en échangeant les propos suivants :

– J'espère bien que Clotaire est là. Barnabé est très maladroit, et la main d'Aristide tremble constamment depuis qu'il a eu cet accès de fièvre qui le handicape, dit oncle Jim.

– Clotaire est sûrement là, affirme oncle Joe.

– Qu'en sais-tu ? Je te parie cent sous que non, reprend oncle Jim.

– Je peux te le prouver logiquement, rétorque oncle Joe, qui poursuit : Prenons comme hypothèse de travail que Clotaire est absent, et voyons ce qu'il résulte de cette supposition. Pour cela, je vais utiliser le principe de réduction à l'absurde.

– Je m'en doutais, grommelle oncle Jim. Je ne t'ai encore jamais entendu discuter quelque chose sans que cela se termine par quelque absurdité !

– Sans me laisser démonter par tes propos venimeux, dit noblement oncle Joe, je continue ; Clotaire étant absent, tu m'accordes que, si Aristide est également absent, Barnabé est obligatoirement présent ?

– Évidemment, dit oncle Jim en haussant les épaules, sinon il n'y aurait plus personne pour garder le salon.

– Nous voyons par conséquent que l'absence de Clotaire fait intervenir une implication logique, « si Aristide est absent, Barnabé est présent », et que cette implication reste vraie tant que Clotaire est absent.

– Admettons, fait Oncle Jim, et après ?

– Il nous faut à présent tenir compte d'une autre hypothétique, que je formule par « Si Aristide est absent, alors Barnabé est absent », laquelle est `toujours` *vraie, indépendamment de la présence de Clotaire, n'est-ce pas ?*

– Sans doute, dit oncle Jim, qui semble gagné par l'inquiétude, je confirme qu'Aristide, s'il s'absente, se fait obligatoirement accompagner par Barnabé.

– Nous somme donc placés en présence de deux implications. La première est vraie tant que Clotaire est absent. La seconde, qui est toujours vraie, l'est en particulier lorsque Clotaire est absent. Or ces deux implications me paraissent parfaitement incompatibles. Donc, par une très belle réduction à l'absurde, je conclus qu'il est impossible que Clotaire soit absent, tu vas donc le voir tout de suite.

– Il y a quand même quelque chose qui me paraît douteux, dans ton « incompatibilité », dit oncle Jim, qui semble totalement décontenancé. Il réfléchit, puis reprend : Pourquoi, en effet, seraient-elles contradictoires ? Tu ne prouves en fait qu'une seule chose : c'est que c'est Aristide qui est présent !

– Très cher et très illogique frère, fait oncle Joe, ne vois-tu pas que tu es en train de décomposer une implication logique en prémisse et conclusion ? Qui te permet d'affirmer que l'implication « si Aristide est absent, Barnabé est présent » est vraie ? N'as-tu jamais appris que, de la valeur de vérité d'une implication, on ne peut rien déduire sur celles de ses éléments ?

À l'aide des variables propositionnelles A (pour : « Aristide est présent »), B (pour : « Barnabé est présent ») et C (pour : « Clotaire est présent »), formaliser les raisonnements d'oncle Joe (dont la conclusion est C) et d'oncle Jim (dont la conclusion est A). Qui a raison et qui a tort ?

10 Tableaux de Beth

Il s'agit d'une autre méthode de démonstration formelle, ignorant tout système d'axiomes, mais comportant de nombreuses règles d'inférence.

Elle est particulièrement adaptée à l'automatisation des démonstrations, et constitue une ouverture sur les algorithmes étudiés en seconde année (utilisation systématique du « théorème de falsification » qui sera envisagé plus loin).

10.1 Principe de la méthode

Il s'agit de montrer qu'une formule F est un théorème($\vdash F$) ; le résultat est obtenu en montrant que la « réfutation » de F conduit à une contradiction.

La « réfutation » se conduit dans un arbre (appelé tableau de Beth (logicien néerlandais). « Réfuter » F, c'est le « nier », selon sa forme, en suivant les règles d'inférences qui sont données dans le tableau :

Règles d'affirmation	Règles de réfutation
$\neg A$ $\overline{A}$	$\overline{\neg A}$ A
$A \vee B$ $A \quad B$	$\overline{A \vee B}$ $\overline{A}$ $\overline{B}$
$A \wedge B$ A B	$\overline{A \wedge B}$ $\overline{A} \quad \overline{B}$
$A \longrightarrow B$ $\overline{A} \quad B$	$\overline{A \longrightarrow B}$ A $\overline{B}$

La règle de contradiction est $\boxed{\begin{array}{c} A \\ \overline{A} \\ \square \end{array}}$.

Écrire, dans un tableau de Beth, A signifie « j'affirme A » et $\overline{A}$ signifie « je nie A » (ne pas confondre avec $\neg A$).

Une branche d'un tableau de Beth est considérée comme terminée lorsque plus aucune règle ne peut s'appliquer et elle est considérée comme close lorsqu'elle contient le symbole de contradiction $\square$.

Il n'est pas nécessaire, pour l'application de la règle de contradiction, que A et $\overline{A}$ figurent dans cet ordre, ni même consécutivement, dans l'arbre : il suffit qu'ils soient dans la même branche.

10.2 Justification de la méthode

Le théorème suivant fonde cette méthode :

PROPRIÉTÉ XII (THÉORÈME DE BETH) : La formule F est un théorème si et seulement si $\overline{F}$ est la racine d'un arbre dont toutes les branches sont closes.

10.3 Exemple

Démonstration de $\vdash ((P \longrightarrow Q) \longrightarrow P) \longrightarrow P$

<table>
<tr><td colspan="2">$\overline{((P \longrightarrow Q) \longrightarrow P) \longrightarrow P}$</td></tr>
<tr><td colspan="2">$\overline{P}$</td></tr>
<tr><td colspan="2">$(P \longrightarrow Q) \longrightarrow Q$</td></tr>
<tr><td>$\overline{P \longrightarrow Q}$</td><td>$P$</td></tr>
<tr><td>P</td><td>$\square$</td></tr>
<tr><td>$\overline{Q}$</td><td></td></tr>
<tr><td>$\square$</td><td></td></tr>
</table>

V. Complétude du calcul propositionnel

1 Théorème de complétude

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 démonstration, avec ses
   - axiomes,

– règles d'inférence,
– démonstrations (ou déductions sous hypothèses).

On peut se demander si les résultats obtenus dans chacune des deux théories sont identiques, c'est-à-dire faire l'étude de la complétude du calcul propositionnel...

1.1 Théorème

Le théorème de complétude du calcul propositionnel s'énonce ainsi :

PROPRIÉTÉ XIII (THÉORÈME DE COMPLÉTUDE) : Tout théorème est une tautologie et réciproquement, soit :

$$\vdash F \text{ si et seulement si } \models F.$$

Autrement dit, les deux points de vue (théorie des valeurs de vérité, théorie de la démonstration) sont équivalents pour les théorèmes et les tautologies.

1.2 Démonstration

Hypothèse : $\vdash F$ (c'est-à-dire : F est un théorème) : La démonstration utilise les résultats suivants :

– Tous les axiomes du calcul propositionnel sont des tautologies (simple calcul sur leurs fonctions de vérité, qui ne sera pas développé ici).
– La règle de détachement est valide (voir exemple du paragraphe 3.3.2 : $\{P, P \longrightarrow Q\} \models Q$.

Il résulte de ces considérations que, lorsqu'on effectue une démonstration (donc : une déduction sans hypothèses), cette démonstration utilise des axiomes, qui sont des tautologies, c'est-à-dire des formes propositionnelles « toujours vraies », et en déduit d'autres formes propositionnelles ; d'après la validité du « modus ponens », les formes propositionnelles qui ont été déduites sont « vraies » chaque fois que les formes d'origine le sont aussi ; or, ces dernières le sont toujours, donc les formes déduites le sont aussi toujours ; autrement dit, les formules déduites en théorie de la démonstration sont des tautologies. Conclusion : $\models F$.

Hypothèse : $\models F$ (c'est-à-dire : F est une tautologie) : Plus compliqué : il faut exhiber une démonstration d'une tautologie. La démonstration se fait en trois étapes :

– étape 1 : On montre que, si $\models F$, alors $\{P_1 \vee \neg P_1, P_2 \vee \neg P_2, \ldots, P_n \vee \neg P_n\} \vdash F$ [Dans cette expression, $P_1, P_2, \ldots, P_n$ sont les variables propositionnelles qui interviennent dans l'expression de F, à l'exclusion de toute autre].
– étape 2 : On montre que $\vdash A \vee \neg A$.

– étape 3 : On montre que, si, $\forall i \in \{1, 2, ..., p\},\ \Gamma \vdash B_i$ et si $\{B_1, B_2, \ldots, B_p\} \vdash C$, alors $\Gamma \vdash C$ ou transitivité de la déductibilité [Γ est un ensemble, éventuellement vide, d'hypothèses].
– Conclusion : L'étape 3, appliquée à $\vdash P_1 \vee \neg P_1, \vdash P_2 \vee \neg P_2, \ldots, \vdash P_n \vee \neg P_n$ [l'ensemble d'hypothèses Γ est ici vide] permet d'obtenir : $\vdash F$.

Voici le détail de la démonstration des trois étapes :

– étape 1 : Toute ligne d'une table de vérité concernant une forme propositionnelle peut être interprétée comme introduisant une « règle de déductibilité ». Exemple : la ligne

P	Q	$P \vee Q$
V	F	V

de la table de vérité de la disjonction logique peut être interprétée comme représentant une déduction : $\{P, \neg Q\} \vdash (P \vee Q)$. Cette « règle de déductibilité » résulte immédiatement des considérations suivantes :

Déduction sous les hypothèses $\{P\ ,\ \neg Q\}$

1	Hypothèse 1	P
2	Axiome 6	$P \longrightarrow P \vee Q$
3	m.p. sur 2 , 3	$P \vee Q$

Conclusion $(P, \neg Q) \vdash P \vee Q$.

[remarque : l'hypothèse $\neg Q$ ne sert en fait pas, mais peu importe]. Il suffit alors d'établir ces « règles de déductibilité » pour toutes les lignes des tables de vérité des connecteurs logique (ce qui est un peu long -il y en a 18 - mais ne présente pas de véritable caractère de difficulté, elles sont proposées en TD) ; comme la table de vérité de F est obtenue à partir de celles des connecteurs logiques, chacune de ses lignes peut aussi être considérée comme une règle de déductibilité ; comme F est une tautologie, sa dernière colonne est uniquement composée de « V », autrement dit, les règles de déductibilité qui la concernent pourront s'écrire :

$$\begin{array}{ccccccccccccc}
\{ & \neg P_1 & , & \neg P_2 & , & \ldots & , & \neg P_{n-1} & , & \neg P_n & \} & \vdash & F \\
\{ & \neg P_1 & , & \neg P_2 & , & \ldots & , & \neg P_{n-1} & , & P_n & \} & \vdash & F \\
. & \ldots & & \ldots & & .. & & ..... & & .... & . & . & . \\
\{ & P_1 & , & P_2 & , & \ldots & , & P_{n-1} & , & \neg P_n & \} & \vdash & F \\
\{ & P_1 & , & P_2 & , & \ldots & , & P_{n-1} & , & P_n & \} & \vdash & F
\end{array}$$

[Le tableau comporte 2^n lignes]. Il ne reste plus qu'à établir que, si Γ est un ensemble quelconque de formes propositionnelles, et si on a $\Gamma \cup \{A\} \vdash C$ et $\Gamma \cup \{B\} \vdash C$, alors on a $\Gamma \cup \{A \vee B\} \vdash C$, ce qui se fait de la manière suivante (il s'agit en fait d'une disjonction des cas) :

– $\Gamma \cup \{A\} \vdash C$ est équivalent à $\Gamma \vdash A \longrightarrow C$ [d'après le théorème de la déduction].
– $\Gamma \cup \{B\} \vdash C$ est équivalent à $\Gamma \vdash B \longrightarrow C$ [d'après le théorème de la déduction].

Donc, dans une

Déduction sous les hypothèses Γ :

1	Résultat intermédiaire 1	$A \longrightarrow C$
2	Résultat intermédiaire 2	$B \longrightarrow C$
3	Axiome 8	$(A \longrightarrow C) \longrightarrow ((B \longrightarrow C) \longrightarrow (A \vee B \longrightarrow C))$
4	m.p. sur 1, 3	$(B \longrightarrow C) \longrightarrow (A \vee B \longrightarrow C)$
5	m.p. sur 2, 4	$A \vee B \longrightarrow C$

Conclusion $\Gamma \vdash A \vee B \longrightarrow C$

Donc, si on suppose $\Gamma \cup \{A\} \vdash C$ et $\Gamma \cup \{B\} \vdash C$, il existe une déduction sous les hypothèses Γ dont la conclusion est $A \vee B \longrightarrow C$. C'est à dire : $\Gamma \vdash A \vee B \longrightarrow C$. D'après le théorème de la déduction, ce dernier résultat est équivalent à : $\Gamma \cup \{A \vee B\} \vdash C$. C'est la conclusion qui était recherchée.

Ce résultat est à présent appliqué n fois de suite aux 2^n « règles de déductibilité », que l'on prend deux par deux, pour obtenir enfin : $\{P_1 \vee \neg P_1, P_2 \vee \neg P_2, \ldots, P_n \vee \neg P_n\} \vdash F$.

– étape 2 : Il faut montrer que : $\vdash A \vee \neg A$ (*théorème du tiers-exclu*).

Démonstration :

1	Axiome 6	$A \longrightarrow A \vee \neg A$
2	Th. de la contrap.	$(A \longrightarrow A \vee \neg A) \longrightarrow (\neg(A \vee \neg A) \longrightarrow \neg A)$
3	m.p. sur 1, 2	$\neg(A \vee \neg A) \longrightarrow \neg A$
4	Axiome 7	$\neg A \longrightarrow A \vee \neg A$
5	Th. de la contrap.	$(\neg A \longrightarrow A \vee \neg A) \longrightarrow (\neg(A \vee \neg A) \longrightarrow \neg\neg A)$
6	m.p. sur 4, 5	$\neg(A \vee \neg A) \longrightarrow \neg\neg A$
7	Axiome 10	$(\neg(A \vee \neg A) \longrightarrow \neg A) \longrightarrow ((\neg(A \vee \neg A) \longrightarrow \neg\neg A) \longrightarrow \neg\neg(A \vee \neg A)$
8	m.p. sur 3, 7	$(\neg(A \vee \neg A) \longrightarrow \neg\neg A) \longrightarrow \neg\neg(A \vee \neg A)$
9	m.p. sur 6, 8	$\neg\neg(A \vee \neg A)$
10	Axiome 9	$\neg\neg(A \vee \neg A) \longrightarrow A \vee \neg A$
11	m.p. sur 9, 10	$A \vee \neg A$

Conclusion : $\vdash A \vee \neg A$.

– étape 3 : Il faut montrer que, si, $\forall i \in \{1, 2, ..., p\}$, $\Gamma \vdash B_i$ et si $\{B_1, B_2, \ldots, B_p\} \vdash C$, alors $\Gamma \vdash C$. Pour cela, il suffit de remarquer qu'une « conclusion partielle », c'est à dire une formule obtenue au cours d'une déduction, est dûment établie, et peut donc servir pour des inférences dans la suite (cette remarque a déjà été utilisée dans plusieurs des démonstrations de ce cours). Ainsi, il suffit de regrouper au sein d'une même déduction (sous les hypothèses de Γ), les déductions qui mènent aux B_i (pour $1 \leqslant i \leqslant p$). Ceux-ci seront considérés, dans la suite, comme « conclusions partielles » ou « résultats intermédiaires ». Et ils seront donc réutilisés en tant que tels dans la suite de la déduction, qui consistera à recopier la déduction de C (sous les hypothèses B_i). Dans cette déduction, les références aux B_i, en tant qu'hypothèses, seront remplacées par les références aux lignes qui précèdent dans lesquelles les B_i ont été obtenus comme « conclu-

sions partielles ». Cette nouvelle déduction sera bien une déduction de C sous les hypothèses de Γ.

Le théorème de complétude est établi.

2 Théorème de complétude généralisé

Le résultat énoncé dans le théorème de complétude au sujet des théorèmes (donc des déductions sans hypothèses) et des tautologies est étendu aux déductions sous hypothèses et aux conséquences logiques ; c'est-à-dire :

PROPRIÉTÉ XIV (THÉORÈME DE COMPLÉTUDE GÉNÉRALISÉ) : On a

$\{P_1, P_2, \ldots, P_n\} \vdash Q$ si et seulement si $\{P_1, P_2, \ldots, P_n\} \models Q$

La démonstration de ce théorème est obtenue aisément en appliquant, pour la partie « théorie de la démonstration », le théorème de la déduction, et, pour la partie « théorie des valeurs de vérité », le théorème de validité. Il ne reste plus alors qu'à appliquer le théorème de complétude.

Ainsi, $\{P_1, P_2, \ldots, P_n\} \vdash Q$ est remplacé par $\vdash (P_1 \longrightarrow (P_2 \longrightarrow \cdots (P_n \longrightarrow Q) \cdots))$

et $\{P_1, P_2, \ldots, P_n\} \models Q$ est remplacé par $\models (P_1 \longrightarrow (P_2 \longrightarrow \cdots (P_n \longrightarrow Q) \cdots))$.

Fin du Chapitre

Chapitre 11

Calcul des prédicats

I. Introduction

1 Insuffisances de la formalisation en Calcul Propositionnel

La Logique des propositions ne s'occupe, on le sait, que des liens entre les propositions réalisés à l'aide des connecteurs logiques.

Autrement dit, une fois qu'une proposition « complexe » a été analysée suivant ces connecteurs, il subsiste des « atomes » (non sécables en Logique des Propositions) et l'analyse ne peut pas être poussée plus profondément... alors qu'il existe cependant encore des « relations » entre ces atomes, relations que la Logique des Propositions ne permet pas de prendre en compte.

1.1 Introduction des « prédicats »

EXEMPLE 1. La proposition « Pierre est le père de Marc » ne comporte aucun connecteur logique, elle n'est plus analysable en Logique des Propositions, on ne peut la formaliser que par une variable propositionnelle, par exemple : A.

Alors que la proposition « Jean est le père de Sylvie » entretient des rapports évidents avec la précédente, elle ne peut être formalisée en Logique des Propositions, elle aussi, que par une (autre) variable propositionnelle, par exemple : B.

Après la formalisation, donc, 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 (la paternité), simplement entre des individus différents.

Ce premier exemple suggère la nécessité, pour poursuivre l'analyse des propositions, de la création d'un langage qui permettrait de décrire des propriétés accordées (ou non) à des individus, ou les reliant.

Cette préoccupation est aussi celle des Mathématiciens, qui s'occupent, par exemple, de décrire les propriétés des entiers naturels.

Il faudra donc envisager des « variables » , dans un certain domaine, et, pour certaines « valeurs » de ces variables, certaines propriétés seront « vraies » ou « fausses ».

1.2 Introduction de l'« univers du discours »

EXEMPLE 2. Soit la proposition « 7 n'a pas de racine carrée ».

En calcul propositionnel, elle ne peut être formalisée que par une variable propositionnelle.

De plus, il semble même difficile de lui accorder une « valeur de vérité » ; en effet, si l'on se place dans l'ensemble des entiers naturels, cette proposition est « vraie », alors que si l'on se place dans l'ensemble des nombres réels, elle est évidemment fausse.

Dans ce second exemple, comme dans le premier, la proposition évoque une propriété, accordée ou refusée, à un objet bien précis (7).

Bien entendu, la valeur de vérité dépend de la valeur de cet objet, mais aussi de l'ensemble dans lequel ces valeurs peuvent être choisies.

On pourra envisager de formaliser (partiellement) une telle proposition par une expression du type « x a une racine carrée », et la valeur de vérité de la proposition obtenue en « donnant à x une valeur » dépend, non seulement de cette valeur, mais aussi de l'ensemble dans lequel cette valeur peut être prise.

D'une manière générale, et avant de donner une définition précise de ces concepts, on dira qu'une « propriété » possédée (ou non) par un objet est un prédicat.

La valeur de vérité de la proposition obtenue en appliquant ce prédicat à un (ou plusieurs) objets dépend de l'univers du discours, c'est à dire de l'ensemble des valeurs reconnues comme possibles (par exemple, dans l'exemple précédent, $\mathbb{N}$ ou $\mathbb{R}$), et aussi, bien entendu, de cette valeur elle-même.

1.3 Introduction de la « quantification »

EXEMPLE 3. Considérons les propositions « Tous les étudiants sont sérieux » et « certains étudiants sont sérieux ».

La Logique des Propositions est incapable de faire apparaître le lien manifeste qui existe entre elles. Elle ne pourra les formaliser que par A et B, parce qu'elles sont différentes et que l'une n'est pas la négation de l'autre.

La généralisation du calcul propositionnel suggérée ici est encore plus grande que dans les exemples précédents, puisque 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 même à toute une catégorie d'individus, sans qu'aucun ne soit nommément désigné.

Ce troisième exemple suggère la notion de quantification d'une variable (tous les..., certains...).

L'objet du calcul des prédicats est d'étudier et de formaliser les notions qui viennent d'être brièvement évoquées, et que le calcul propositionnel ne permet pas de faire apparaître.

2 Univers du discours, sujets et individus

Le calcul des prédicats fait intervenir des variables, qui prennent des valeurs dans un certain ensemble appelé, on l'a dit, univers du discours.

Les éléments de cet univers sont appelés sujets.
Un sujet distingué ou individu est une « valeur particulière » d'une variable (d'un sujet) dans l'univers du discours.

EXEMPLE 4. 7 est un individu, ou un sujet distingué, dans l'univers du discours des entiers naturels. Le nom de cet individu est « $sept$ » , il est représenté par le symbole 7.

---

3 Groupes opératoires et termes

Il existe deux manières de distinguer un sujet de l'univers du discours.

– le nommer (donner explicitement son nom, ou le symbole qui le représente), comme dans l'exemple précédent, pour l'entier 7.

– le désigner indirectement par une « propriété » caractéristique (qu'il doit donc être le seul à posséder, de manière à ce qu'il soit identifié sans aucune ambiguïté).

EXEMPLE 5. Par exemple, si $Pierre$ est le père de $Jean$, il est possible de distinguer $Pierre$ sans le nommer explicitement par la locution « $le\ père\ de\ Jean$ ».

EXEMPLE 6. L'entier 7 peut être distingué sans être nommé explicitement par l'expression $5 + 2$ qui désigne un et un seul entier positif.

De manière tout à fait précise :

DÉFINITION 1 (OPÉRATION N-AIRE). *Soit $\mathcal{U}$ l'univers du discours.*
On appelle opération n-aire *définie sur $\mathcal{U}$ toute application de $\mathcal{U}^n$ dans $\mathcal{U}$.* ♢

Soit f une telle opération n-aire.

DÉFINITION 2 (GROUPE OPÉRATOIRE). *L'expression $f(x_1, x_2, \ldots, x_n)$ est appelée* groupe opératoire *à n places.*
f (le symbole de l'application) est plutôt appelé, en calcul des prédicats, symbole opératoire ou symbole fonctionnel. ♢

Comme toute application donne une image et une seule de tout élément de son domaine, un groupe opératoire à n places représente bien, sans ambiguïté, un individu de l'univers du discours.

EXEMPLE 7. L'addition ordinaire des entiers est une opération binaire sur $\mathbb{N}$ et $+(x, y)$.
Le groupe opératoire correspondant désigne, de manière unique, un élément de cet ensemble, dès que les variables x et y ont été remplacées dans son expression par des sujets distingués (par une méthode ou une autre).
Ainsi, $+(5, 2)$ est l'individu qui peut aussi être représenté par 7.

EXEMPLE 8. Le groupe opératoire à 1 place $père(x)$ désigne sans ambiguïté un et un seul individu dans l'univers des êtres humains, dès que la variable x a été remplacée par un sujet distingué.

Si $Jean$ est bien un nom d'individu (c'est à dire, désigne bien un individu unique), s'il en est de même de $Pierre$, et si $Pierre$ est le père de $Jean$, alors $père(Jean)$ distingue l'individu $Pierre$.

Par extension, et par convention, un nom d'individu est considéré comme groupe opératoire zéro-aire (sans variable).

DÉFINITION 3 (TERME). *En calcul des prédicats, on fait intervenir des groupes opératoires à n places. Le résultat est un terme du calcul des prédicats :*

- *une variable,*
- *une « constante » (un groupe opératoire zéro-aire),*
- *un groupe opératoire n-aire $(n \geqslant 1)$ $f(t_1, t_2, \ldots, t_n)$, dans lequel $t_1, t_2, \ldots, t_n$ sont des termes et f un symbole opératoire (ou fonctionnel).* ♢

REMARQUE 1. C'est une définition récursive.

4 Groupes relationnels et atomes

Il s'agit ici de faire intervenir les relations qui peuvent lier des individus de l'univers du discours.

EXEMPLE 9. « 22 est le double de 11 », « 46 est le double de 45 » sont des propositions (vraies ou fausses !) qui évoquent la relation « être le double de » .

Très précisément, soit $\mathcal{U}$ l'univers du discours.

DÉFINITION 4 (RELATION N-AIRE). *On appelle* relation n-aire *définie sur $\mathcal{U}$ toute application de $\mathcal{U}^n$ dans $\{0, 1\}$ (ou $\{$faux,vrai$\}$).* ♢

DÉFINITION 5 (GROUPE RELATIONNEL, PRÉDICAT). *Si r est une telle relation n-aire, on appelle* groupe relationnel à n places*, ou* prédicat de poids n *l'expression $r(t_1, t_2, \ldots, t_n)$ dans laquelle $t_1, t_2, \ldots, t_n$ sont des termes.* ♢

REMARQUE 2. r (le symbole de l'application) est encore appelé *symbole relationnel*, ou *symbole de prédicat*.

Par extension et par convention, un groupe relationnel zéro-aire (sans « variables ») est une proposition (qui a donc une valeur de vérité).

REMARQUE 3. On notera que, lorsque les termes qui interviennent dans un groupe relationnel sont des individus, ce groupe devient une proposition (vraie ou fausse) :

- $double\_de(x, y)$ est un prédicat, ou groupe relationnel à deux places défini sur $\mathcal{U} = \mathbb{N}$.
- $double\_de(22, 11)$ est une proposition (vraie).
- $double\_de(+(23, 23), +(40, 5))$ en est une autre (fausse !).

Dans une formule du calcul des prédicats, un groupe relationnel est un atome.

5 Les quantificateurs

Considérons la proposition « Tous les étudiants travaillent les mathématiques » pour l'analyser du point de vue du calcul des prédicats.

Aucun connecteur logique n'apparaît. Par contre, on peut remarquer l'intervention d'un prédicat (de poids deux) qui peut être formalisé par $travaille(x, y)$, et qui signifie qu'un individu x travaille une certaine matière y.

Il faut préciser l'univers du discours, il s'agit ici d'un ensemble produit cartésien de deux ensembles :
- le premier est l'ensemble des étudiants,
- le second un ensemble de noms de matières (celles qui sont susceptibles d'être enseignées).

Il est bien clair qu'il s'agit d'une proposition. Or, si la seconde variable (y) est bien remplacée par un symbole d'individu ($maths$), aucun nom, explicite ou implicite, n'est donné pour désigner un étudiant particulier : on a « $travaille(x, maths)$ » sans que l'on sache « qui est x » .

Il s'agit d'une construction particulière, qui permet d'obtenir des propositions à partir de groupes relationnels sans qu'une variable soit distinguée, et qui porte le nom de quantification : le sujet x n'est pas distingué, il est quantifié.

NOTATION : La notation utilisée pour cette quantification est « $\forall x\, travaille(x, maths)$ ».

DÉFINITION 6 (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 4. Dans une expression quantifiée, un symbole de variable (x) subsiste, bien que le résultat de l'expression quantifiée soit une proposition.

Mais ce symbole n'a plus la même signification que dans l'expression (non quantifiée) « $travaille(x, maths)$ » qui, elle, n'est pas une proposition.

$travaille(x, maths)$ **n'est pas une proposition :** impossible de lui attribuer une valeur de vérité si on ne sait pas « qui est x ».

DÉFINITION 7 (VARIABLE LIBRE). *La variable représentée par le symbole* x *est dite libre, ce qui signifie qu'elle est susceptible de recevoir n'importe quelle valeur (de l'univers du discours).* ♢

Le fait de remplacer ce symbole par un terme qui distingue un individu transforme l'expression en proposition, qui a alors une valeur de vérité.

EXEMPLE 10. C'est ainsi que $travaille(Jean, maths)$ est $vrai$ ou $faux$, et que $travaille(fils(Pierre), m$ est aussi $vrai$ ou $faux$.

$\forall x\ travaille(x, maths)$ **est une proposition :** la variable représentée par le symbole x n'est plus libre (on dit qu'elle est liée par le quantificateur $\forall x$).

En effet, il est devenu impossible de lui attribuer une valeur quelconque de l'univers du discours, que ce soit par un nom ou par un terme, sous peine de perte totale de sens.

REMARQUE 5. Il n'est pas possible de substituer quoi que ce soit à x dans cette expression, si ce n'est un autre symbole de variable : en effet, $\forall y\ travaille(y, maths)$ a très exactement le même sens, et la même valeur de vérité.

EXEMPLE 11. Les « variables liées » (on dit aussi variables muettes) sont d'un emploi courant en Mathématiques.

Par exemple, considérons l'expression $\sum_{i=1}^{3} i = 1 + 2 + 3 = 6$.

Bien qu'un symbole de variable (i) intervienne dans cette expression, le résultat ne fait pas intervenir de variable : c'est une simple valeur numérique.

On pourrait, sans changer cette valeur, attribuer un autre symbole à la variable muette, par exemple, on a aussi $\sum_{j=1}^{3} j = 6$.

Bref, la variable n'est en fait présente que « fictivement », elle n'intervient pas dans le résultat. Il est par conséquent impossible de lui attribuer quelque valeur que ce soit, alors que c'est possible pour une variable libre.

Par exemple dans l'expression $(i + j)$ (qui n'a pas de valeur), on peut « donner à i la valeur 3 » pour obtenir $(3+j)$ et à j la valeur 5 pour obtenir $(3+5)$, qui a maintenant une valeur.

Dans l'exemple prédédent, la proposition était obtenue à partir du groupe relationnel par substitution de la seconde variable par un individu et par quantification de la première.

Toutes les variables d'un groupe relationnel peuvent être quantifiées, pour obtenir toujours une proposition.

EXEMPLE 12. Dans la proposition « Tous les étudiants travaillent une matière » , intervient le même groupe relationnel à deux places $travaille(x, y)$; on remarque qu'aucun étudiant n'est explicitement nommé, pas plus que la matière qu'il travaille.

Il faut formaliser cette proposition en calcul des prédicats par « pour tout étudiant, il y a une matière, et cet étudiant travaille cette matière » , soit $\forall x \exists y\ travaille(x, y)$: la variable x est liée par le quantificateur universel $\forall x$ et la variable y est liée par le quantificateur existentiel $\exists y$

REMARQUE 6. Dans d'autres Logiques, on fait intervenir d'autres symboles de quantificateurs.

Ces autres Logiques se situent en dehors du cadre de notre étude.

ATTENTION : *Il est interdit d'intervertir des quantificateurs de symboles différents* (si l'on désire que la proposition conserve le même sens, bien entendu).

EXEMPLE 13. – La proposition $\forall x \exists y\ travaille(x, y)$ signifie que tout étudiant travaille une matière (au moins).

Autrement dit, la matière travaillée est fonction de l'étudiant, elle dépend de cet étudiant, elle n'est éventuellement pas la même pour tous les étudiants.

La liaison des variables x et y par des quantificateurs $\forall x \exists y$ (dans cet ordre) introduit donc une relation entre x et y.
Selon la valeur de x, y pourra prendre telle ou telle valeur, a priori pas la même pour chaque valeur de x.

– La proposition $\exists y \forall x \ travaille(x, y)$ signifie : il y a une matière que tous les étudiants travaillent.

  La différence fondamentale avec le cas précédent est que, dans cette dernière proposition, on affirme que tous les étudiants travaillent la même matière.

EXEMPLE 14. Un autre exemple, plus mathématique :

– $\forall x \exists y \, (x + y = x)$, soit : « pour tout x, on peut trouver un y, tel que $x + y = x$ » . Si l'univers du discours est une algèbre de Boole, il est clair que la proposition est vraie : il suffit pour s'en convaincre de choisir $y = x$, et on a bien, en algèbre de Boole, $x + x = x$: la valeur de y dépend bien de celle de x (c'est la même).

– $\exists y \forall x \, (x + y = x)$, soit : « il existe une valeur de y telle que, pour tout x, $x + y = x$ » . Si l'univers du discours est une algèbre de Boole, il est clair que la proposition est vraie : la valeur 0 pour y convient, et, si $y = 0$, on a bien, quelle que soit la valeur de x, $x + 0 = x$.

EXEMPLE 15. En supposant que (a, b, c) prenne ses valeurs dans un univers du discours qui est une partie de $\mathbb{R}^3$ telle que $b^2 - 4ac > 0$, la proposition :

– $\forall a \forall b \forall c \exists x \, (ax^2 + bx + c = 0)$ est une proposition vraie (une équation du second degré dont le discriminant est positif admet des racines réelles, et tout le monde sait que les valeurs de ces racines dépendent de celles de a, de b et de c.

– $\exists x \forall a \forall b \forall c \, (ax^2 + bx + c = 0)$ est une proposition fausse : il n'y a pas de réel qui soit solution de n'importe quelle équation du second degré !

---

6 Formules du calcul des prédicats

Comme les groupes relationnels produisent des propositions
– soit quand les variables sont remplacées par des individus,
– soit quand elles sont liées par quantification,
il est possible de relier de telles expressions par les connecteurs logiques habituels, comme en calcul propositionnel.

On obtient ainsi une formule du calcul des prédicats...

DÉFINITION 8 (ATOME). *On appelle* atome *un groupe relationnel n-aire, dans l'expression duquel les éventuelles variables sont des termes.* ♢

DÉFINITION 9 (FORMULE). *La définition d'une* formule *est alors :*
– *Une formule est un atome,*
– *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 $(P) \vee (Q)$, $(P) \wedge (Q)$, $(P) \longrightarrow (Q)$, $(P) \longleftrightarrow (Q)$ sont des formules.*

7 Champ d'un quantificateur

DÉFINITION 10 (CHAMP D'UN QUANTIFICATEUR). *Le* champ 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, son champ est donc généralement clairement délimité par une paire de parenthèses (d'après la règle de priorité, s'il n'y a pas de parenthèses, son champ est limité au premier connecteur rencontré).

EXEMPLE 16. Ainsi, dans la formule : $\forall x\, p(x, y) \vee q(x)$, le champ du quantificateur $\forall x$ est strictement limité à l'atome $p(x, y)$.

Autrement dit, $q(x)$ se trouve en dehors de ce champ.

On remarquera que, dans $q(x)$, x est donc libre, alors qu'elle est liée dans $p(x, y)$, ce qui signifie donc que le symbole x n'a pas la même signification dans l'ensemble de la formule :
– dans $q(x)$, elle peut être remplacée par n'importe quel terme ou individu,
– ce n'est pas le cas dans $p(x, y)$.

REMARQUE 7. Une telle expression serait considérée comme incorrecte en calcul algébrique, par exemple, de même que dans tout langage de programmation.

En effet, un même symbole ou identificateur ne peut désigner qu'un seul objet à la fois et, pour des objets différents, dans une même formule, il faut utiliser des symboles différents.

Par contre, en calcul des prédicats, une telle expression est parfaitement licite. Il vaut évidemment mieux s'abstenir d'expressions qui entretiennent de telles ambiguïtés, mais elles sont admises parce qu'elles peuvent intervenir « indépendamment de notre volonté » lors de substitutions (voir plus loin).

Toujours est-il que, si le champ du quantificateur $\forall x$ est l'ensemble de la formule précédente, l'écriture de celle-ci doit être : $\forall x\,(p(x,y) \vee q(x))$.

II. Théorie de la validité en calcul des prédicats

1 Extension des valeurs de vérité au calcul des prédicats

1.1 Valeurs de vérité en calcul des prédicats

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 $P \longrightarrow Q$.

   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 $p(x,y) \longrightarrow q(x,z)$.

   Les atomes sont ici des prédicats de poids deux qui, eux aussi, ne peuvent prendre que les valeurs « vrai » ou « faux », mais pas indépendamment du prédicat

considéré.

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 groupe relationnel.

– si $f(x, y)$ est la fonction booléenne associée au prédicat $p(x, y)$
– si $g(x, z)$ est celle qui est associée à $q(x, z)$,

alors la fonction de vérité de la formule est : $\overline{f(x, y)} + g(x, z)$.

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...).

1.2 Définitions

Les définitions de tautologie et de conséquence logique (en calcul des prédicats, on dira plutôt conséquence valide) sont les mêmes...

DÉFINITION 11 (TAUTOLOGIE, CONSÉQUENCE VALIDE, 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 peuvent intervenir 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 valide 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 faut 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 (liste effective lorsque c'est possible, ou démonstration). 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 $faux$ (liste effective ou démonstration), 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 $p(x, y, z)$ (mais aussi pour toute formule P), on a : $\models p(x, y, z) \longrightarrow p(x, y, z)$.

Mais, pour donner un exemple de la complexité introduite en théorie des valeurs de vérité par le calcul des prédicats, étudions la formule $p(x) \longrightarrow p(y)$ pour savoir s'il s'agit d'une tautologie ($p(x)$ est évidemment un prédicat unaire : difficile de proposer un exemple plus simple). Pour prouver qu'il s'agit d'une tautologie, il faut prouver que :

– pour tout prédicat unaire p,
– pour tout univers du discours $\mathcal{U}$,
– pour toutes valeurs des variables x et y dans $\mathcal{U}$,

la valeur de vérité de la formule est $vrai$. « Pour tout univers du discours » : il faut envisager des univers à 1, puis 2,... éléments et essayer d'en déduire un résultat général, si c'est possible.

– Dans un univers à un seul élément, x, comme y, ne peut prendre qu'une seule valeur ; on a donc $x = y$, et donc la valeurs de vérité de $p(x)$ est la même que celle de $p(y)$, et donc $p(x) \longrightarrow p(y)$ est $vraie$ [table de l'implication].
– Dans un univers à deux éléments, pour que la valeur de vérité de $p(x)$ soit définie, il faut se la donner pour chacune des deux valeurs de x ; or, il faut le faire « pour tout prédicat unaire p » , donc envisager toutes les fonctions booléennes de deux variables : il y en a quatre, que nous noterons z, i, n, et r. Désignons les deux éléments de $\mathcal{U}$ par 1 et 2 (pour ne pas les confondre avec des variables booléennes) :

x	$z(x)$	$i(x)$	$n(x)$	$r(x)$
1	$faux$	$faux$	$vrai$	$vrai$
2	$faux$	$vrai$	$faux$	$vrai$

Puis il faut faire le tableau des valeurs de vérité de $p(x)$, $p(y)$ et de $p(x) \longrightarrow p(y)$

en fonction des valeurs de x et de y et de la fonction booléenne associée à p (z, i, n, ou r)

fonction booléenne de p	x	y	$p(x)$	$p(y)$	$p(x) \longrightarrow p(y)$
z	1	1	$faux$	$faux$	$vrai$
	1	2	$faux$	$faux$	$vrai$
	2	1	$faux$	$faux$	$vrai$
	2	2	$faux$	$faux$	$vrai$
i	1	1	$faux$	$faux$	$vrai$
	1	2	$faux$	$vrai$	$vrai$
	2	1	$vrai$	$faux$	$faux$
	2	2	.....	.....	.....
n	.	.	.....	.....	.....

Inutile de construire le reste du tableau : on a trouvé $faux$ dans une ligne, donc $p(x) \longrightarrow p(y)$ n'est pas une tautologie [en effet, il existe un prédicat unaire, un univers à deux éléments, et, dans cet univers, une valeur de x et une valeur de y pour lesquels la formule est fausse...].

Exemples Donnons enfin deux exemples pour lesquels la construction d'une table de vérité n'est pas nécessaire :

EXEMPLE 17 (MONTRER QUE $\models \forall x\, p(x,x) \longrightarrow \forall x \exists y\, p(x,y)$.). Soit $\mathcal{U}$ un univers du discours, p une relation binaire sur $\mathcal{U}$.

- Premier cas : dans $\mathcal{U}$, $\forall x\, p(x,x)$ est $faux$; alors l'implication est vraie.
- Deuxième cas : dans $\mathcal{U}$, $\forall x\, p(x,x)$ est $vrai$; donc, pour chaque valeur de x dans $\mathcal{U}$, $p(x,x)$ est vrai. Alors $\forall x \exists y\, p(x,y)$ est $vrai$ car, pour toute valeur de x, il suffit de prendre pour y la même valeur que celle de x pour que $p(x,y)$ soit $vrai$.

EXEMPLE 18 (MONTRER QUE $\forall x\,(r(x) \longrightarrow s(x)) \wedge \exists x\,(\neg r(x) \wedge s(x))$ N'EST NI UNE ANTILOGIE, NI U
Pour montrer que cette formule n'est pas une antilogie, il suffit d'exhiber un exemple dans lequel elle est vraie ; pour montrer qu'elle n'est pas une tautologie, il suffit d'exhiber un exemple dans lequel elle est fausse.

- $\mathcal{U}$ est l'ensemble des habitants de la France ; $r(x)$ est le prédicat « x habite Belfort » et $s(x)$ est le prédicat « x habite la France » . Puisque Belfort est en France,

tout habitant de Belfort habite la France, donc $\forall x\,(r(x) \longrightarrow s(x))$ est $vrai$. Mais tous les habitants de la France ne sont pas concentrés à Belfort, autrement dit : il existe des habitants de la France qui n'habitent pas Belfort, donc : $\exists x\,(\neg r(x) \wedge s(x))$ est $vrai$; enfin, d'après la table de vérité de la conjonction logique, la formule est vraie.

– $\mathcal{U}$ est un ensemble E, dans lequel on a défini deux parties R et S, telles que $S \subset R$; $r(x)$ est le prédicat « $x \in R$ » et $s(x)$ est le prédicat « $x \in S$ » . Comme $S \subset R$, tout élément extérieur à R ne peut pas être dans S, donc $\exists x\,(\neg r(x) \wedge s(x))$ est $faux$, et donc la formule est fausse.

---

2 Équivalences classiques entre formules

De la définition de la valeur de vérité d'une formule quantifiée, on peut déduire :

– $\forall x\,p(x) \approx \neg\exists x\,\neg p(x)$

– $\exists x\,p(x) \approx \neg\forall x\,\neg p(x)$.

À titre d'illustration de cette propriété, voici un exemple ; considérons la proposition : « L'ensemble des entiers naturels admet un élément maximum » (cette proposition est, bien sûr, fausse !). Pour formaliser cette proposition en calcul des prédicats, il faut introduire le prédicat classique d'inégalité qu'il faudrait, en toute rigueur, noter $\leqslant (x, y)$, qui signifie que x est inférieur ou égal à y, mais pour lequel nous conserverons la notation usuelle $x \leqslant y$. L'univers du discours est évidemment $\mathbb{N}$, et, pour exprimer que $\mathbb{N}$ admet un élément maximum, on exprime que tout élément de $\mathbb{N}$ est inférieur ou égal à cet élément maximum. Soit : $\exists M \forall n\,(n \leqslant M)$ (en calcul des prédicats, on ne précise jamais l'univers du discours dans la formule, comme on le fait habituellement en mathématiques : $\exists M \in \mathbb{N},\ \forall n \in \mathbb{N},\ n \leqslant M$). Cette proposition étant fausse, sa négation est vraie, soit $\neg(\exists M \forall n\,(n \leqslant M))$. D'après ce que nous venons de voir, cette négation a même valeur de vérité que : $\forall M \exists n\,\neg(n \leqslant M)$, ou encore $\forall M \exists n\,(M < n)$; c'est à dire, pour tout entier M, on peut trouver un entier n qui est plus grand (autrement dit, on ne pourra jamais en trouver un plus grand que tous les autres, ce qui exprime bien la propriété que $\mathbb{N}$ n'a pas d'élément maximum).

Voici d'autres résultats d'équivalences entre formules :

– $\forall x\,(p(x) \wedge q(x)) \approx \forall x\,p(x) \wedge \forall x\,q(x)$

– $\exists x\,(p(x) \vee q(x)) \approx \exists x\,p(x) \vee \exists x\,q(x)$

– $\exists x\,(p(x) \longrightarrow q(x)) \approx \forall x\,p(x) \longrightarrow \exists x\,q(x)$

En effet, 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 x, $q(x)$ est vrai. Réciproquement, si, pour toute valeur de x, $p(x)$ est vrai, et, si, pour toute valeur de x, $q(x)$ est aussi vrai, il est bien évident que, pour toute valeur de x, $p(x)$ et $q(x)$ sont simultanément vrais. 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 $q(x)$ est vraie ; la réciproque est aussi évidente.

Mais attention : dans l'univers du discours $\mathcal{U} = \mathbb{R}$, $\exists x\,(x < 0)$ est vraie, $\exists x\,(x \geqslant 0)$ est vraie aussi, mais $\exists x\,((x < 0) \wedge (x \geqslant 0))$ n'est pas vraie ! Dans le même univers, $\forall x\,((x < 0) \vee (x \geqslant 0))$ est évidemment vraie, alors que $\forall x\,(x < 0) \vee \forall x\,(x \geqslant 0)$ n'est, évidemment aussi, pas vraie !

D'après les tables de vérité des connecteurs logiques, $\exists x\,(p(x) \longrightarrow q(x))$ est équivalente à $\exists x\,(\neg p(x) \vee q(x))$, qui est équivalente à $\exists x\,\neg p(x) \vee \exists x\,q(x)$, or $\exists x\,\neg p(x)$ est équivalente à $\neg\forall x\,p(x)$ (voir ci-dessus), donc notre formule est encore équivalente à $\neg\forall x\,p(x) \vee \exists x\,q(x)$, soit, finalement, à $\forall x\,p(x) \longrightarrow \exists x\,q(x)$.

On remarquera bien que dans, par exemple, $\forall x\,p(x) \wedge \forall x\,q(x)$, les deux occurrences de x ne représentent pas la même variable, mais deux variables liées ou muettes, tandis que dans $\forall x\,(p(x) \wedge q(x))$, les deux occurrences de x représentent bien la même variable (toujours muette).

3 Substitutions libres

On notera par $A(x)$ une formule du calcul des prédicats (éventuellement compliquée !) dans laquelle la variable x présente des occurrences ; ces occurrences peuvent être libres ou liées, et A peut présenter des occurrences libres et des occurrences liées de x ; bien entendu, il peut aussi n'y avoir aucune occurrence libre, ou aucune occurrence liée ; enfin, d'autres variables peuvent aussi présenter des occurrences dans A. Soit ensuite t un terme du calcul des prédicats, qui présente des occurrences de diverses variables (toutes libres, dans un terme, évidemment ; un terme ne comporte pas de quantificateurs).

On dit que ce terme t est libre pour x dans A lorsque les occurrences libres de x dans A ne rentrent dans le champ d'aucun quantificateur portant sur une variable de t. Exemple : Soit la formule $A(x) = \exists x\,r(x,y) \longrightarrow \forall y\,s(x,y,z)$ [Cette formule présente une occurrence liée et une occurrence libre de x].

- Examinons le terme y (une variable est un terme) par rapport à la variable x dans A. Comme l'occurrence libre de x dans A (la seconde) entre dans le champ du quantificateur $\forall y$, et que y est une variable (et même la seule...) du terme y, le terme y n'est pas libre pour x dans A.
- Examinons le terme $t = f(x,z)$ [où f est un symbole opératoire, t est bien un terme] par rapport à la variable x dans A. Comme l'occurrence libre de x dans A n'entre pas dans le champ d'un quantificateur portant sur x (et pour cause...), ni dans celui d'un quantificateur portant sur z (il n'y en a pas...), ce terme t est libre pour x dans A.

On appelle substitution libre de la variable x par le terme t dans la formule A le remplacement de x, dans toutes ses occurrences libres dans A (et uniquement celles-ci...) par le terme t, à condition que ce terme soit libre pour x dans A. La nouvelle formule obtenue est notée $(x \mid t)A$.

- il est impossible d'effectuer la substitution libre de x par y dans la formule de l'exemple ci-dessus (y n'est pas libre pour x dans A).

– il est possible d'effectuer la substitution libre de x par $t = f(x, z)$ dans A, car ce terme est libre pour x dans A. Le résultat est $(x \mid f(x, z))A = \exists x\, r(x, y) \longrightarrow \forall y\, s(f(x, z), y, z)$.

Par convention, si x ne présente pas d'occurrences libres dans A, la substitution libre de x par n'importe quoi est toujours possible, et elle laisse la formule A inchangée.

4 Élimination et introduction des quantificateurs

Soit $A(x)$ une formule répondant à la définition du paragraphe précédent et $A(r)$ le résultat de la substitution libre de x par r dans A (r libre pour x dans A). Alors :

– Tautologie 1 : $\models \forall x\, A(x) \longrightarrow A(r)$
– Tautologie 2 : $\models A(r) \longrightarrow \exists x\, A(x)$

La première est le « principe de particularisation » : si, pour toute valeur de x, $A(x)$ est vrai, alors $A(r)$ est vrai, r étant une « valeur particulière » de x. La seconde est le principe d'« exhibition d'une valeur particulière » : si on peut trouver une valeur r pour laquelle $A(r)$ est vraie alors on peut affirmer qu'il existe une valeur de x pour laquelle $A(x)$ est vraie (c'est r !).

Exemples (mathématiques) :

– Si tout nombre premier n'est divisible que par 1 et par lui-même, alors 17 (qui est un nombre premier) n'est divisible que par 1 et par 17.
– Puisque 2 divise 6, il existe des nombres qui sont divisibles par 2.

L'extrême simplicité de ces exemples ne doit pas faire oublier que le terme r peut être bien autre chose qu'une constante, en fait, tout terme libre pour x dans A.

Soit $A(x)$ une formule répondant à la définition du paragraphe précédent et B une formule ne contenant pas d'occurrences libres de x. Alors :

– Si $\models B \longrightarrow A(x)$, alors : $\models B \longrightarrow \forall x\, A(x)$
– Si $\models A(x) \longrightarrow B$, alors : $\models \exists x\, A(x) \longrightarrow B$.

Le premier (méta)-résultat est le « principe de généralisation » ; il est très souvent utilisé en mathématiques : Si on conduit une démonstration qui a pour conclusion un résultat dont l'expression fait intervenir une variable x, et si cette démonstration n'a fait intervenir aucune hypothèse sur cette variable, alors le résultat est vrai pour toute valeur de x. On appelle aussi cette règle (il s'agira ultérieurement, en théorie de la démonstration, d'une règle d'inférence) la règle d'introduction du quantificateur universel.

Le second est l'introduction du quantificateur existentiel. Autant le premier est facilement utilisé (souvent abusivement...) autant le second est malaisé à saisir. Donnons-en une illustration simple : On vous apprend qu'un chat qui vole met les oiseaux en grand danger ; vous n'y croyez guère (au chat qui vole), mais vous êtes obligé d'admettre que s'il existe un chat volant, alors les oiseaux sont en grand danger.

III. Théorie de la démonstration en calcul des prédicats

1 Axiomes et règles d'inférence

Pour élaborer la théorie de la démonstration en calcul des prédicats, on reprend celle qui a été donnée en calcul propositionnel en ajoutant les axiomes et les règles d'inférence qui permettent de traiter les nouveaux symboles introduits (les quantificateurs).

- Nouveaux axiomes :
  - $A_{14} = \forall x\, A(x) \longrightarrow A(r)$
  - $A_{15} = A(r) \longrightarrow \exists x\, A(x)$

  [Dans les conditions indiquées au paragraphe précédent : substitution libre de x par r dans A, r étant libre pour x dans A].
- Nouvelles règles d'inférence :
  - $\{C \longrightarrow A(x)\} \vdash C \longrightarrow \forall x\, A(x)$ [règle d'introduction de $\forall$ ou règle-$\forall$]
  - $\{A(x) \longrightarrow C\} \vdash \exists x\, A(x) \longrightarrow C$ [règle d'introduction de $\exists$ ou règle-$\exists$].

  [La formule C ne contient pas d'occurrences libres de x].

2 Validité des résultats établis en calcul propositionnel

D'une manière générale, on peut dire (puisque, en quelque sorte, le calcul propositionnel est un sous-ensemble du calcul des prédicats), que tout ce qui a été établi en calcul propositionnel demeure établi en calcul des prédicats ; par exemple, comme, en calcul propositionnel, $\vdash (A \longrightarrow A)$, [en calcul propositionnel, A est une forme propositionnelle quelconque], on a aussi, en calcul des prédicats, $\vdash (A \longrightarrow A)$ [en calcul des prédicats, A est une formule quelconque]. La démonstration est immédiate [utilisation des mêmes axiomes qu'en calcul propositionnel, puisqu'ils sont repris sans modification en calcul des prédicats].

Bref, tant que l'on n'utilise pas les symboles (quantificateurs), axiomes (14 et 15) et règles d'inférence (introductions de quantificateurs) propres au calcul des prédicats, les démonstrations et déductions du calcul des propositions s'étendent sans difficultés au calcul des prédicats.

Il ne faudrait pas en conclure hâtivement que tout ce qui est vrai en calcul propositionnel est vrai en calcul des prédicats, du moins sans nouvelle démonstration : en particulier tout méta-théorème du calcul propositionnel doit être redémontré en calcul des prédicats (si c'est possible !). On va étudier le cas du méta-théorème de la déduction.

3 Le (méta-) théorème de la déduction

On peut énoncer les résultats suivants : En calcul des prédicats, on a :

Si $(B_1, B_2, \ldots, B_{n-1}) \vdash B_n \longrightarrow C$, alors $(B_1, B_2, \ldots, B_n) \vdash C$ (c'est un « modus ponens). La réciproque n'est en général pas vraie, sauf (en particulier) dans les cas suivants :

– la déduction est faite « à variables constantes » (cette expression malheureuse, mais consacrée, signifie simplement que la déduction n'utilise aucune des règles d'introduction des quantificateurs, donc est faite comme en calcul propositionnel).
– les hypothèses de la déduction sont toutes des formules closes (c'est à dire sans variables libres).

IV. Le système formel « PR »

Il s'agit évidemment du système formel construit pour la formalisation du calcul des prédicats. Comme pour « LP » , le nombre de symboles est réduit par rapport à celui du calcul des prédicats tel qu'il a été décrit ci-dessus, dans le but d'obtenir plus d'efficacité en « démonstration automatique » .

1 Définition

– L'alphabet Σ_{PR} est celui de « LP » , auquel on adjoint :
  – le symbole de quantificateur universel
  – des symboles de variables et d'individus (les « constantes »)
  – des symboles opératoires et fonctionnels
– Les formules sont celles du calcul des prédicat (limitées au vocabulaire de « PR »)
– Les (schémas d') axiomes sont au nombre de 5 :
  – Axiome 1 : $A \longrightarrow (B \longrightarrow A)$
  – Axiome 2 : $(A \longrightarrow (B \longrightarrow C)) \longrightarrow ((A \longrightarrow B) \longrightarrow (A \longrightarrow C))$
  – Axiome 3 : $(\neg B \longrightarrow \neg A) \longrightarrow (A \longrightarrow B)$
  – Axiome 4 : $\forall x\, A(x) \longrightarrow A(t)$
  – Axiome 5 : $(C \longrightarrow B(x)) \longrightarrow (C \longrightarrow \forall x\, B(x))$ [x n'étant pas variable libre de C].
– Il y a 2 règles d'inférence :
  – Le « modus ponens » $(A, A \longrightarrow B) \vdash B$
  – La généralisation $A \vdash \forall x\, A$.

2 Calcul des prédicats égalitaire

Dans de nombreuses applications, notamment mathématiques, on a besoin de l'égalité. La « vraie » égalité, c'est à dire la capacité de reconnaissance de l'identité de deux objets, n'est pas une notion de logique du premier ordre (celle dont nous nous occupons ici). En effet, deux objets sont identiques lorsque toute propriété qui est vraie pour l'un est aussi vraie pour l'autre, et que toute propriété fausse pour l'un est aussi fausse pour l'autre. Autrement dit, pour définir l'égalité, il faudrait quantifier universellement un symbole de prédicat, écrire une formule qui pourrait ressembler à :

$$(x = y) \text{ si et seulement si } \forall p\,(p(x) \longleftrightarrow p(y)).$$

Or, quantifier un symbole de prédicat n'est pas possible en calcul des prédicats du premier ordre. Donc, lorsqu'on a besoin d'une égalité, on se contente d'une égalité « affaiblie » comme simple relation d'équivalence (réflexive, symétrique, transitive), donc comme un prédicat binaire, qu'il faudrait noter $\equiv (x, y)$ mais que l'on note $(x \equiv y)$. Bien entendu, il faut ajouter les axiomes qui fixent les propriétés de cette égalité, pour obtenir un calcul des prédicats égalitaire.

3 Interprétations de « PR »

Soit E la base d'une interprétation de « PR » .

- une formule de « PR » est dite satisfaite dans E chaque fois qu'il existe un ensemble de valeurs, choisies dans E, pour les variables libres qui interviennent dans cette formule tel que la proposition qui en résulte est « vraie » .
- une formule de « PR » est dite valide dans E si, pour tout ensemble de valeurs, choisies dans E, pour les variables libres qui interviennent dans E, la proposition qui en résulte est vraie.
- une formule de « PR » est universellement valide ou est une thèse lorsqu'elle est valide dans toute interprétation. L'usage veut cependant que l'on conserve terme de tautologie (qui devrait être réservé au calcul propositionnel).

4 (Méta-)Théorème de complétude

La démonstration est longue et repose sur un raisonnement par récurrence sur la complexité de la formule (en gros, le nombre de connecteurs ou quantificateurs qui interviennent dans celle-ci), après avoir démontré la propriété individuellement pour chaque formule de complexité 1 (un seul connecteur ou quantificateur). Toute la difficulté provient de la cardinalité (du « nombre d'éléments ») de la base de l'interprétation. Le résultat est connu sous le nom de théorème de complétude de Gödel (1930). Bref, le (méta-) théorème suivant peut être énoncé :

$$\boxed{\text{Le système formel « PR » est complet } (\vdash B \text{ si et seulement si } \models B)}$$

Le théorème de complétude généralisé pour « PR » a aussi été obtenu par Gödel :

$$\boxed{F \models B \text{ si et seulement si } F \vdash B}$$

5 Satisfiabilité et insatisfiabilité

Soit F un ensemble de formules de « PR » ; F est dit satisfiable s'il est possible de trouver une interprétation dans laquelle les formules de F sont simultanément satisfaites. La base d'une telle interprétation est appelée modèle de F. Évidemment, s'il n'est pas possible de trouver un modèle pour l'ensemble de formules F, celui-ci est dit insatisfiable. Ce sont ces notions d'insatisfiabilité qui sont à l'origine des méthodes de démonstration dites de falsification (et qui servent dans les algorithmes de « chaînage

arrière »). Il s'agit d'obtenir une déduction d'une formule B de « PR » à partir d'un ensemble d'hypothèses F. On énonce le (méta-) théorème suivant, appelé Théorème de falsification :

Soit F un ensemble de formules closes et B une formule close de « PR » . Alors, $F \vdash B$ si et seulement si $F \cup \{\neg B\}$ est insatisfiable

Ce théorème peut être démontré à l'aide du théorème de complétude.

V. Traitement des formules de « PR »

1 Forme prénexe

1.1 Définition

La mise sous forme prénexe d'une formule de « PR » est la première étape à lui faire subir dans le traitement nécessaire à la présenter à un algorithme de résolution (démonstrations automatisées de théorèmes) Une formule de « PR » est dite sous forme prénexe lorsque tous les quantificateurs sont en tête de cette formule. Elle est alors de la forme $Q_1x_1Q_2x_2 \dots Q_nx_nB$, où B est une formule sans quantificateurs.

1.2 Méthode

La méthode à suivre est la suivante :

– Réduire les connecteurs. En théorie, il faudrait ne conserver que les connecteurs $\longrightarrow$ et $\neg$, en fait, on ne conserve que $\wedge$, $\vee$, $\neg$. Cette réduction est obtenue par les équivalences entre formules classiques

$$A \longrightarrow B \approx \neg A \vee B$$

$$A \longleftrightarrow B \approx (\neg A \wedge \neg B) \vee (A \wedge B)$$

– 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 :

$$\forall x A(x) = \forall y A(y)$$

$$\exists x A(x) = \exists y A(y)$$

– Faire « remonter » les quantificateurs en tête, par les équivalences suivantes :

$$\neg\neg A \approx A$$

$$\neg(\forall x A(x)) \approx \exists x \neg A(x)$$

$$\neg(\exists x A(x)) \approx \forall x \neg A(x)$$

et, si x n'est pas variable libre de C (ce qui doit être le cas si on a correctement renommé les variables...]

$$C \vee \forall x A(x) \approx \forall x(C \vee A(x))$$

$$C \vee \exists x A(x) \approx \exists x(C \vee A(x))$$

$$C \wedge \forall x A(x) \approx \forall x(C \wedge A(x))$$

$$C \wedge \exists x A(x) \approx \exists x(C \wedge A(x))$$

1.3 Exemple

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 $P_6 \approx \neg Q$ (si on montre que cet ensemble de propositions est « insatisfiable » , on aura montré que Q est conséquence de P_1 à P_5).
Pour cela, on va considérer les prédicats

- $\mathrm{ar}(x)$: la personne x est arrêtée
- $\mathrm{mal}(x)$: la personne x est malhonnête
- $\mathrm{co}(x, y)$: la personne x commet l'action y
- $\mathrm{cr}(y)$: l'action y est un crime

On obtient

F_1 $\forall y(\mathrm{cr}(y) \longrightarrow \exists x \mathrm{co}(x, y))$
F_2 $\forall x \forall y(\mathrm{cr}(y) \wedge \mathrm{co}(x, y) \longrightarrow \mathrm{mal}(x))$
F_3 $\forall x(\mathrm{ar}(x) \longrightarrow \mathrm{mal}(x))$
F_4 $\forall x(\mathrm{mal}(x) \wedge \mathrm{ar}(x) \longrightarrow \neg(\exists y(\mathrm{cr}(y) \wedge \mathrm{co}(x, y))))$
F_5 $\exists x \mathrm{cr}(x)$
F_6 $\neg(\exists x(\mathrm{mal}(x) \wedge \neg \mathrm{ar}(x)))$

qui deviennent, sous forme prénexe,

F'_1 $\forall y \exists x(\neg \mathrm{cr}(y) \vee \mathrm{co}(x, y))$
F'_2 $\forall x \forall y(\neg \mathrm{cr}(y) \vee \neg \mathrm{co}(x, y) \vee \mathrm{mal}(x))$
F'_3 $\forall x(\neg \mathrm{ar}(x) \vee \mathrm{mal}(x))$
F'_4 $\forall x \forall y(\neg \mathrm{mal}(x) \vee \neg \mathrm{ar}(x) \vee \neg \mathrm{cr}(y) \vee \neg \mathrm{co}(x, y))$
F'_5 $\exists x \mathrm{cr}(x)$
F'_6 $\forall x(\neg \mathrm{mal}(x) \vee \mathrm{ar}(x))$

2 Forme de Skolem

2.1 Définition

On appelle forme de Skolem d'une formule de « PR » , mise préalablement sous forme prénexe, la formule obtenue en éliminant tous les quantificateurs de symbole $\exists$, de la manière suivante :

- Soit $\exists x_j$ un quantificateur existentiel qui figure après les quantificateurs universels $\forall x_{j_1} \forall x_{j_2} \ldots \forall x_{j_n}$: alors x_j est remplacé par $f(x_{j_1}, x_{j_2}..., x_{j_n})$ (un terme, groupe opératoire à n places).
- 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 (qui n'est autre qu'un groupe opératoire à zéro place).

Cette transformation est autorisée par le théorème suivant :

2.2 Théorème de Skolem

Soit $\mathcal{A}$ un ensemble fini de formules de « PR » , et $\mathcal{A}_S$ l'ensemble des formes de Skolem de ces formules. Alors, $\mathcal{A}$ admet un modèle (de base E) si et seulement si $\mathcal{A}_S$ admet un modèle de base E.

On va vérifier ce théorème pour un ensemble réduit à une seule formule du type

$$A = \forall x \exists y p(x, y)$$

Sa forme de Skolem est

$$A_S = \forall x p(x, f(x))$$

- Soit E un modèle de A. Ce modèle admet une relation binaire p. E étant un modèle de A, par définition, A y est satisfaite. Donc la formule $\exists y p(x, y)$ est valide dans E. Ceci signifie que, pour chaque valeur α de x, il existe (au moins) une valeur β de y telle que $p(\alpha, \beta)$ [qui est une proposition] a la valeur de vérité « vrai » . On peut définir une fonction f de E dans lui-même en posant, pour chaque valeur de α, $\beta = f(\alpha)$. Le problème se complique s'il existe plusieurs, voire une infinité de valeurs de y possibles pour une valeur donnée de x. Autrement dit, se pose le problème de l'axiome du choix. Mais c'est le théorème de Löwenheim-Skolem qui intervient ici ; ce théorème affirme en effet que, si un ensemble de formules de « PR » admet un modèle, alors il admet un modèle de base dénombrable. En se plaçant dans ce modéle, le choix auquel il est fait allusion est possible, sans recours à l'axiome du choix. Ayant ainsi défini une fonction f, E, cette fois muni de p et de f, constitue un modèle de la formule $\forall x p(x, f(x))$, c'est à dire de A_S.
- Réciproquement, si (E,p, f) est un modèle de A_S, alors, évidemment, $\exists y p(x, y)$ est valide dans E, puisqu'il suffit d'exhiber $y = f(x)$ pour le montrer, et donc, (E,p) est un modèle de A.

2.3 Exemple

Les formes de Skolem de nos formules sont donc :

F_1'' $\forall y(\neg \mathrm{cr}(y) \vee \mathrm{co}(f(y), y))$
F_2'' $\forall x \forall y(\neg \mathrm{cr}(y) \vee \neg \mathrm{co}(x, y) \vee \mathrm{mal}(x))$
F_3'' $\forall x(\neg \mathrm{ar}(x) \vee \mathrm{mal}(x))$
F_4'' $\forall x \forall y(\neg \mathrm{mal}(x) \vee \neg \mathrm{ar}(x) \vee \neg \mathrm{cr}(y) \vee \neg \mathrm{co}(x, y))$
F_5'' $\mathrm{cr}(a)$
F_6'' $\forall x(\neg \mathrm{mal}(x) \vee \mathrm{ar}(x))$

3 Forme clausale

3.1 Suppression des quantificateurs universels

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 résolution utilisée (et n'est possible que si c'est cette méthode qui est effectivement 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.

3.2 Clauses

On dit qu'une formule de « PR » est mise sous forme de clause lorsque son expression n'utilise plus de quantificateurs, et que les seuls connecteurs qui y subsistent sont les connecteurs $\neg$ et $\vee$. Si une formule, sans quantificateur, se présente sous la forme $A \wedge B$, elle est remplacée par deux formules : A et B. Il est bien clair que si l'on satisfait, d'une part A, et d'autre part B, on aura satisfait $A \wedge B$. Une clause (avec ou sans variables) est donc de la forme : $A \vee B \vee \neg C \vee \ldots$. A, B, C sont les littéraux de la clause, ils apparaissent sous forme positive (A) ou négative ($\neg C$). Enfin, la clause vide (sans littéraux) est représentée par $\square$. Elle est réputée insatisfiable dans toute interprétation. Dans notre exemple, il n'y a plus rien à faire, et les formules de départ mises sous formes de clauses sont les formules de F_1'' à F_6'', sans les quantificateurs universels.

VI. Système de Herbrand

1 Introduction

Pour reconnaître si, oui ou non, un ensemble de formules admet un modèle, il faut essayer diverses bases possibles : à un élément, à deux éléments, ..., à une infinité d'éléments. Chacun de ces essais conduit lui-même à de nombreux cas ; l'intérêt du théorème de Herbrand est qu'il affirme que tous ces essais peuvent se ramener à un seul, qui n'utilise que le « vocabulaire » des formules qui sont en question, un « modèle syntaxique » unique. Par ailleurs la méthode des « tables de vérité » , ou, ce qui revient au même, la méthode des fonctions de vérité booléennes, conduit elle aussi à un nombre d'essais très rapidement beaucoup trop grand (dès que les formules sont un peu compliquées ou un peu nombreuses et comportent beaucoup de variables. Le nombre d'essais croît comme 2^n, et devient très vite trop important).

2 Univers, atomes, système de Herbrand

Pour un ensemble de formules (mis sous forme de clauses), l'univers de Herbrand associé est l'ensemble des termes sans variables qui peuvent être construits à partir du vocabulaire de ces formules (symboles de constantes et symboles opératoires ou fonctionnels). Pour que cet univers ne soit pas vide, s'il n'y a pas de constante, on en introduit une. Les atomes de Herbrand sont les atomes qui interviennent dans ces formules, lorsqu'on a remplacé les variables par des éléments de l'univers de Herbrand.

Reprenons l'exemple de nos criminels

F_1''' $\quad \neg\mathrm{cr}(y) \vee \mathrm{co}(f(y), y)$
F_2''' $\quad \neg\mathrm{cr}(y) \vee \neg\mathrm{co}(x, y) \vee \mathrm{mal}(x)$
F_3''' $\quad \neg\mathrm{ar}(x) \vee \mathrm{mal}(x)$
F_4''' $\quad \neg\mathrm{mal}(x) \vee \neg\mathrm{ar}(x) \vee \neg\mathrm{cr}(y) \vee \neg\mathrm{co}(x, y)$
F_5''' $\quad \mathrm{cr}(a)$
F_6''' $\quad \neg\mathrm{mal}(x) \vee \mathrm{ar}(x)$

Les éléments du vocabulaire de ces clauses sont :

- un symbole de constante (a)
- un symbole opératoire de poids 1 (f)

L'univers de Herbrand est donc

$$\{a, f(a), f(f(a)), f(f(f(a))), \ldots\}$$

Un atome de Herbrand est

$\mathrm{ar}(a)$, ou $\mathrm{mal}(f(f(a)))$, ou, etc...

Le système de Herbrand est obtenu en remplaçant successivement les variables de l'ensemble de formules considéré par les éléments de l'univers de Herbrand.

3 Théorème de Herbrand

Un ensemble de clauses admet un modèle si et seulement s'il admet un modèle de base l'univers de Herbrand. De plus, pour que cet ensemble de clauses soit insatisfiable, il suffit que l'une des formules du système de Herbrand ne soit pas satisfiable.
Application :

- 1ère génération : avec $x = a$, $y = a$
  $\{\neg\mathrm{cr}(a)\vee\mathrm{co}(f(a),a)$, $\neg\mathrm{cr}(a)\vee\neg\mathrm{co}(a,a)\vee\mathrm{mal}(a)$, $\neg\mathrm{ar}(a)\vee\mathrm{mal}(a)$, $\neg\mathrm{mal}(a)\vee$ $\neg\mathrm{ar}(a)\vee\neg\mathrm{cr}(a)\vee\neg\mathrm{co}(a,a)$, $\mathrm{cr}(a)$, $\neg\mathrm{mal}(a)\vee\mathrm{ar}(a)\}$

  satisfiables :
  $\mathrm{cr}(a)$ $\mathrm{co}(f(a),a)$ $\neg\mathrm{co}(a,a)$ $\neg\mathrm{ar}(a)$ $\neg\mathrm{mal}(a)$
  ou
  $\mathrm{cr}(a)$ $\mathrm{co}(f(a),a)$ $\neg\mathrm{co}(a,a)$ $\mathrm{ar}(a)$ $\mathrm{mal}(a)$
- 2ème génération : avec $x = f(a)$, $y = a$
  $\{\neg\mathrm{cr}(a)\vee\neg\mathrm{co}(f(a),a)\vee\mathrm{mal}(f(a))$, $\neg\mathrm{ar}(f(a))\vee\mathrm{mal}(f(a))$, $\neg\mathrm{mal}(f(a))\vee$ $\neg\mathrm{ar}(f(a))\vee\neg\mathrm{cr}(a)\vee\neg\mathrm{co}(f(a),a)$, $\neg\mathrm{mal}(f(a))\vee\mathrm{ar}(f(a))\}$

  la première exige $\mathrm{mal}(f(a))$, donc la dernière exige $\mathrm{ar}(f(a))$, donc la troisième ne peut être satisfaite : c'est terminé.

L'ensemble de clauses n'admet pas de modèle ayant pour base l'univers de Herbrand, donc il n'admet aucun modèle, donc la formule F_6 n'est pas conséquence des cinq premières, donc $\neg F_6$ est un théorème, et donc « il y a bien des personnes malhonnêtes en liberté » .

4 Algorithme de Herbrand

Il s'agit du premier algorithme de « démonstration automatique » . D'un point de vue théorique, il ne fait rien de plus que celui qui consisterait à produire toutes les conséquences possibles (à l'aide des règles d'inférence) des formules de l'ensemble de départ, pour essayer de trouver la formule finale comme conséquence de cet ensemble. Il est cependant plus facile à mettre en œuvre, et donne souvent le résultat plus rapidement que la méthode des tables de vérité. Cependant, il doit être clair qu'il s'agit d'un algorithme qui s'arrête au bout d'un temps fini (éventuellement long !) si l'ensemble de clauses est insatisfiable, mais qui ne s'arrête pas dans le cas contraire (conséquence désagréable de l'indécidabilité du calcul des prédicats...).

Fin du Chapitre

Chapitre 12

Algorithme de résolution

I. Résolution sans variables

1 Cadre

Il n'y a ici pas de variables autres que les variables propositionnelles, et on se place en logique des propositions.

2 Le système formel RSV

Le système formel RSV (Résolution sans variable) est le système formel choisi pour traiter les problèmes posés en calcul propositionnel.

3 Principes généraux

Quelques principes généraux de la résolution sans variable (RSV) :

- Les seules formules correctes admises sont les clauses, c'est-à-dire des disjonctions de littéraux, positifs ou négatifs (précédé de $\neg$).
- Les seuls connecteurs admis sont $\neg$ et $\vee$.
- Une clause est satisfaite quand l'un de ses littéraux est vrai.
- La clause vide (ne contenant aucun littéral) ne vérifie pas cette condition, elle est dite insatisfaite.

NOTATION : La clause vide est représentée par $\square$.

4 Le système formel RSV

Ces principes généraux se formalisent ainsi :

– alphabet : les variables propositionnelles, $\neg$ et $\vee$.

– axiomes : pas d'axiomes.

– règles d'inférence : $\{F \vee G, \neg F \vee G\} \vdash (RSV)G \vee H$

REMARQUE 1. Cette dernière règle est une généralisation du « modus ponens » :

$$\{F, F \rightarrow G\} \vdash H,\ \{F \vee B, \neg F \vee H\} \vdash B \vee H$$

Par rapport à la logique propositionnelle, la suppression des axiomes et la généralisation de la règle d'inférence ont pour conséquence la perte de la complétude (*i.e.* équivalence tautologie/théorème).

REMARQUE 2. En logique propositionnelle, on avait $\{F\} \vdash F \vee G$. Cette déduction ne peut plus être obtenue dans RSV.

5 Quelques indications sur les algorithmes de résolution

5.1 Généralités

Il y a trois classes d'algorithmes de résolution :

Les algorithmes de chaînage avant : on part des données (l'ensemble des clauses de départ), on en déduit de nouvelles clauses, et ainsi de suite, en espérant arriver à la conclusion.

Les algorithmes de chaînage arrière on part de la conclusion, on construit les clauses qui par déduction permettraient de les obtenir, et ainsi de suite, en espérant obtenir les clauses de départ.

Les algorithmes mixtes

PROPRIÉTÉ I (THÉORÈME DE FALSIFICATION) : $\mathcal{F} \vdash A$ si et seulement si $\mathcal{F} \cup \{\neg A\} \vdash \square$, où $\mathcal{F}$ est un ensemble de clauses, et A une clause.

5.2 Un exemple de chaînage avant : algorithme de saturation

On produit systématiquement toutes les clauses possibles à partir des clauses de départ, et on recommence jusqu'à ce qu'il n'y ait plus production de nouvelles clauses.

Cet algorithme n'est cependant pas efficace. On peut l'améliorer en « saturation avec simplification », les simplifications étant :

– la suppression de tautologie,
– la suppression des clauses subsumées (C est subsumée par C' si tous les littéraux de C' sont dans C : $A \vee B \vee C$ est subsumée par $A \vee B$).

EXEMPLE 1. Ensemble de clauses de départ :

$a \rightarrow b \quad C_1 = \neg a \vee b$
$b \rightarrow c \quad C_2 = \neg b \vee c$
$c \rightarrow a \quad C_3 = \neg c \vee a$
$a \vee b \vee c \quad C_4 = a \vee b \vee c$

Génération 1 $\{C_1, C_2\} \vdash C_5 = \neg a \vee c$
$\{C_1, C_3\} \vdash C_6 = b \vee \neg c$
$\{C_1, C_4\} \vdash C_7 = b \vee c$
$\{C_2, C_3\} \vdash C_8 = a \vee \neg b$
$\{C_2, C_4\} \vdash C_9 = a \vee c$
$\{C_3, C_4\} \vdash C_{10} = a \vee b$

C_4 est subsumé par C_7.

Génération 2 Les clauses des deux générations précédentes entre elles :
$\{C_1, C_8\} \vdash C_{11} = b \vee \neg b$
$\{C_1, C_9\} \vdash C_{12} = b \vee c$
$\{C_1, C_{10}\} \vdash C_{13} = b$
$\{C_2, C_6\} \vdash C_{14} = c \vee \neg c$
$\{C_2, C_{10}\} \vdash C_{15} = b$
etc.
C_1, C_6, C_7, C_{10} sont subsumées par C_{13}.
C_{12} est C_7, C_{15} est C_{13}.
C_{11} et C_{14} sont des tautologies.

REMARQUE 3. Il est préférable de retirer les clauses subsumées au fur et à mesure.

5.3 Exemple d'algorithme de chaînage arrière

Présentation Le problème est de savoir si $\mathcal{F} \vdash C$, où $\mathcal{F}$ est un ensemble de clauses, et C est une clause.

On sait que c'est équivalent à

$\mathcal{F} \vdash \{\neg C\}$ insatisfiable à condition que $\mathcal{F}$ soit satisfiable.

On étudie donc la clause pour essayer de formuler des diagnostiques.

On essaye, par exemple, si C serait conséquence de $\mathcal{F}$, puis en cas de réponse positive, si $\neg C$ ne serait pas aussi une conclusion possible.

Si c'est le cas, cela signifie que $\mathcal{F}$ est contradictoire : le problème est mal posé.

Le but de « savoir si C (un littéral) se déduit de $\mathcal{F}$ » se traduit par : « Effacer $\neg C$ dans l'ensemble des clauses de $\mathcal{F}$, la liste de ses ancêtres étant vide. »

Les règles de l'algorithme SL résolution

1. Un littéral qui figure dans sa propre liste d'ancètres ne s'efface pas ($\Rightarrow$ arrêt de l'algo)
2. Un littéral dont la négation figure dans la liste des ancêtres s'efface.
3. Si on est dans aucun des cas précédent, on doit chercher une résolvante, c'est-à-dire une clause de $\mathcal{F}$ qui contienne la négation du littéral que l'on cherche à effacer qui permet donc une déduction (par RSV).

   Le but « effacer le littéral dans l'ensemble des clauses $\mathcal{F}$ et avec la liste d'ancêtre N » est alors remplacé par « effacer tous les littéraux de la résolvante dans $\mathcal{F}$, en rajoutant le littéral sur lequel se porte la déduction à la liste des ancêtres ».

EXEMPLE 2.

$$\begin{array}{ll} a \vee b & \leftarrow C_1 \\ \neg b \vee c & \leftarrow C_2 \\ a \vee \neg c & \leftarrow C_3 \\ a \vee b \vee c & \leftarrow C_4 \end{array}$$

On va demander à SL résolution un diagnostique sur a (a est déductible de cet ensemble de clauses).

Le but est « effacer $\neg a$ dans l'ensemble des clauses, liste d'ancêtres [] (vide) ».

Avec la clause C_4, on obtient la résolvante $\{\neg a, a \vee b \vee c\} \vdash b \vee c$.

Le but est remplacé par « effacer les littéraux de $b \vee c$ dans l'ensemble des clauses, liste des ancêtres [$\neg a$].

– Effacer b « ... », liste ancêtre [$\neg a$].
  Avec $C_2\{b, \neg b \vee c\} \vdash c$.
  « Effacer c, liste ancêtre [$\neg a, b$]. »
  Avec $C_3\{c, \neg c \vee a\} \vdash a$.
  « Effacer a, liste ancêtre [$\neg a, b, c$]. »
  $\Rightarrow$ Succès car $\neg a$ dans liste des ancêtres.
– Effacer c « ... », liste ancêtre [$\neg a$].
  Avec $C_3\{c, \neg c \vee a\} \vdash a$.
  « Effacer a, liste ancêtre [$\neg a$]. »
  $\Rightarrow$ Succès car $\neg a$ dans la liste des ancêtres.

Succès pour effacer a.

Il reste à vérifier si on peut effacer $\neg a$.

---

6 Exemples complets de résolution

Ces divers exemples sont indépendants...

6.1 Méthode de résolution

...La méthode de résolution est cependant la même. Elle est exposée ici, et elle est valable pour chacun d'entre eux sauf le dernier, qui ne se satisfait pas d'une déduction linéaire.

1. Formaliser en logique des propositions (pas en calcul des prédicats) les propositions, puis
2. les mettre sous forme de clauses (sans variables). Ces clauses seront numérotées, dans l'ordre de leur obtention.
3. Appliquer la règle d'inférence de « RSV » pour obtenir la conclusion que l'on peut tirer de l'ensemble de propositions.

   Les nouvelles clauses déduites recevront un numéro.

   Une déduction sera présentée sous la forme $(C_i, C_j) \vdash C_k = ...$

   La conclusion demandée est une clause qui ne contient comme littéraux que ceux qui ne figurent qu'une seule fois dans l'ensemble de clauses de départ.

   Utiliser une déduction linéaire dont la racine est une clause qui contient l'un de ces littéraux.
4. Traduire la conclusion obtenue sous forme de phrase française.

6.2 Les exemples complets

EXEMPLE 3. Les animaux de la maison...

1. Les seuls animaux de cette maison sont des chats.
2. Tout animal qui aime contempler la lune est apte à devenir un animal familier.
3. Quand je déteste un animal, je l'évite soigneusement.
4. Aucun animal n'est carnivore, à moins qu'il n'aille rôder dehors la nuit.
5. Aucun chat ne manque jamais de tuer les souris.
6. Aucun animal ne s'attache jamais à moi, sauf ceux qui sont dans cette maison.

7. Les panthères ne sont pas aptes à devenir des animaux familiers.
8. Aucun animal non carnivore ne tue de souris.
9. Je déteste les animaux qui ne s'attachent pas à moi.
10. Les animaux qui vont rôder dehors la nuit aiment toujours contempler la lune.

Réponse : L'univers est { Les animaux}.

$$
\begin{array}{lll}
Maison \longrightarrow Chats & C_1 : & \neg Maison \vee Chats \\
Lune \longrightarrow Familier & C_2 : & \neg Lune \vee Familier \\
Deteste \longrightarrow Evite & C_3 : & \neg Deteste \vee Evite \\
Rode \vee \neg Carnivore & C_4 : & Rode \vee \neg Carnivore \\
Chats \longrightarrow Tuer & C_5 : & \neg Chats \vee Tuer \\
Attache \longrightarrow Maison & C_6 : & \neg Attache \vee Maison \\
Panthere \longrightarrow \neg Familier & C_7 : & \neg Panthere \vee \neg Familier \\
\neg Carnivore \longrightarrow \neg Tuer & C_8 : & Carnivore \vee \neg Tuer \\
\neg Attache \longrightarrow Deteste & C_9 : & Attache \vee Deteste \\
Rode \longrightarrow Lune & C_{10} : & \neg Rode \vee Lune
\end{array}
$$

$$
\begin{array}{lll}
\{C_2, C_7\} & \vdash \neg Panthere \vee \neg Lune & : C_{11} \\
\{C_{11}, C_2\} & \vdash \neg Panthere \vee \neg Rode & : C_{12} \\
\{C_{12}, C_{10}\} & \vdash \neg Panthere \vee \neg Carnivore & : C_{13} \\
\{C_{13}, C_4\} & \vdash \neg Panthere \vee \neg Tuer & : C_{14} \\
\{C_{14}, C_8\} & \vdash \neg Panthere \vee \neg Chats & : C_{15} \\
\{C_{15}, C_5\} & \vdash \neg Panthere \vee \neg Maison & : C_{16} \\
\{C_{16}, C_1\} & \vdash \neg Panthere \vee \neg Attache & : C_{17} \\
\{C_{17}, C_6\} & \vdash \neg Panthere \vee \neg Deteste & : C_{18} \\
\{C_{18}, C_9\} & \vdash \neg Panthere \vee Deteste & : C_{19} \\
\{C_{19}, C_3\} & \vdash \neg Panthere \vee Evite &
\end{array}
$$

Soit $Panthere \longrightarrow Evite$: J'évite les panthères.

EXEMPLE 4. Les exercices de logique...

1. Quand un étudiant résout un exercice de logique sans soupirer, vous pouvez être sûr qu'il le comprend.
2. Ces exercices ne se présentent pas sous la forme habituelle.
3. Aucun exercice facile ne donne mal à la tête.
4. Les étudiants ne comprennent pas les exercices qui ne se présentent pas sous la forme habituelle.
5. Les étudiants ne soupirent jamais devant un exercice de logique, à moins qu'il ne leur donne mal à la tête.

Réponse : L'univers est { Exercices }.

$$
\begin{array}{ll}
\neg Soupirent \longrightarrow Comprend & C_1 : \ Soupirent \vee Comprend \\
Ces\ exercices \longrightarrow \neg Habituelle & C_2 : \ \neg Ces\ exercices \vee \neg Habituelle \\
Facile \longrightarrow \neg Mal & C_3 : \ \neg Facile \vee \neg Mal \\
\neg Habituelle \longrightarrow \neg Comprend & C_4 : \ Habituelle \vee \neg Comprend \\
\neg Soupirent \vee Mal & C_5 : \ \neg Soupirent \vee Mal
\end{array}
$$

$$
\begin{array}{lll}
\{C_2, C_4\} & \vdash \neg Ces\ exercices \vee \neg Comprend & : C_6 \\
\{C_6, C_1\} & \vdash \neg Ces\ exercices \vee Soupirent & : C_7 \\
\{C_7, C_5\} & \vdash \neg Ces\ exercices \vee Mal & : C_8 \\
\{C_8, C_3\} & \vdash \neg Ces\ exercices \vee \neg Faciles &
\end{array}
$$

Soit $Ces\ exercices \longrightarrow \neg facile$: Ces exercices ne sont pas faciles.

EXEMPLE 5. Mes idées sur les chaussons aux pommes...

1. Toute idée de moi qui ne peut s'exprimer sous forme de syllogisme est vraiment ridicule.
2. Aucune de mes idées sur les chaussons aux pommes ne mérite d'être notée par écrit.
3. Aucune idée de moi que je ne parviens pas à vérifier ne peut être exprimée sous forme de syllogisme.
4. Je n'ai jamais d'idée vraiment ridicule sans la soumettre sous le champ à mon avocat.
5. Mes rêves (idées) ont tous trait aux chaussons aux pommes.
6. Je ne soumets aucune de mes idées à mon avocat si elle ne mérite pas d'être notée par écrit.

Réponse : L'univers est { Idée }.

$$
\begin{array}{ll}
Ridicule \longrightarrow Sullogisme & C_1 : \ \neg Ridicule \vee Syllogisme \\
Pomme \longrightarrow \neg Ecrit & C_2 : \ \neg Pomme \vee \neg Ecrit \\
\neg Verifier \longrightarrow \neg Syllogisme & C_3 : \ \neg Verifier \vee \neg Syllogisme \\
\neg Ridicule \longrightarrow Avocat & C_4 : \ Ridicule \vee Avocat \\
Reves \longrightarrow Pomme & C_5 : \ \neg Reves \vee Pomme \\
Avocat \longrightarrow Ecrit & C_5 : \ \neg Avocat \vee Ecrit
\end{array}
$$

$$
\begin{array}{lll}
\{C_3, C_1\} & \vdash \neg Verifier \vee \neg Ridicule & : C_7 \\
\{C_7, C_4\} & \vdash \neg Verifier \vee Avocat & : C_8 \\
\{C_8, C_6\} & \vdash \neg Verifier \vee Ecrit & : C_9 \\
\{C_9, C_2\} & \vdash \neg Verifier \vee \neg Pomme & C_{10} \\
\{C_{10}, C_5\} & \vdash \neg Verifier \vee \neg Reves &
\end{array}
$$

Soit $Reves \longrightarrow \neg Verifier$: Mes rêves ne sont pas vérifiés.

EXEMPLE 6. Les matières enseignées à l'IUT

1. Aucune matière n'est primordiale, sauf l'ACSI.
2. Toute matière enseignée par des professeurs dynamiques est susceptible de plaire aux étudiants.
3. Je ne travaille pas les matières que je n'aime pas.
4. Les seules matières intéressantes sont les matières informatiques.
5. Aucune matière informatique n'évite l'abstraction.
6. Aucune matière ne me réussit, excepté les matières intéressantes.
7. Les mathématiques ne sont pas susceptibles de plaire aux étudiants.
8. Aucune matière non primordiale ne tombe dans l'abstraction.
9. Je n'aime pas les matières qui ne me réussissent pas.
10. L'ACSI est enseignée par des professeurs dynamiques.

Réponse : L'univers est { Matieres }.

$$
\begin{array}{lll}
Primordiale \longrightarrow ACSI & C_1 : & \neg Primordiale \vee ACSI \\
Dynamique \longrightarrow Plait & C_2 : & \neg Dynamique \vee Plait \\
\neg Aime \longrightarrow \neg Travaille & C_3 : & Aime \vee \neg Travaille \\
Interessante \longrightarrow Informatique & C_4 : & \neg Interessante \vee Informatique \\
Informatique \longrightarrow Abstraction & C_5 : & \neg Informatique \vee Abstraction \\
Reussit \longrightarrow Interessant & C_6 : & \neg Reussit \vee Interessant \\
Maths \longrightarrow \neg Plait & C_7 : & \neg Maths \vee \neg Plait \\
\neg Primordial \longrightarrow \neg Abstraction & C_8 : & Primordiale \vee \neg Abstraction \\
\neg Reussit \longrightarrow \neg Aime & C_9 : & Reussit \vee \neg Aime \\
ACSI \longrightarrow Dynamique & C_{10} : & \neg ACSI \vee Dynamique
\end{array}
$$

$$
\begin{array}{lll}
\{C_7, C_2\} & \vdash \neg Maths \vee \neg Dynamique & : C_{11} \\
\{C_{11}, C_{10}\} & \vdash \neg Maths \vee \neg ACSI & : C_{12} \\
\{C_{12}, C_1\} & \vdash \neg Maths \vee \neg Primordiale & : C_{13} \\
\{C_{13}, C_8\} & \vdash \neg Maths \vee \neg Abstraction & : C_{14} \\
\{C_{14}, C_5\} & \vdash \neg Maths \vee \neg Informatique & : C_{15} \\
\{C_{15}, C_4\} & \vdash \neg Maths \vee \neg Interessant & : C_{16} \\
\{C_{16}, C_6\} & \vdash \neg Maths \vee Reussit & : C_{17} \\
\{C_{17}, C_9\} & \vdash \neg Maths \vee \neg Aime & : C_{18} \\
\{C_{18}, C_3\} & \vdash \neg Maths \vee \neg Travaille &
\end{array}
$$

Soit $Maths \longrightarrow \neg Travaille$: Je ne travaille pas les maths.

7 Exercices

Exercice 1. *Les logiciens ne dorment pas beaucoup.*

1. *Un logicien qui dîne de deux côtelettes de porc risque de se ruiner.*
2. *Un joueur dont l'appétit n'est pas féroce risque de se ruiner.*
3. *Un homme qui est déprimé parce qu'il est ruiné et qu'il risque de se ruiner à nouveau se lèbe toujours à cinq heures du matin.*
4. *Un homme qui ne joue pas et qui ne dîne pas de deux côtelettes de porc est assuré d'avoir un appétit féroce.*
5. *Un homme dynamique qui se couche avant quatre heures du matin aurait intérêt à devenir conducteur de taxi.*
6. *Un homme doué d'un appétit féroce, qui ne s'est pas ruiné et qui ne se lève pas à cinq heures du matin, dîne toujours de deux côtelettes de porc.*
7. *Un logicien qui risque de se ruiner aurait intérêt à devenir conducteur de taxi.*
8. *Un joueur convaincu, qui est déprimé bien qu'il ne soit pas ruiné, ne court aucun risque de se ruiner.*
9. *Un homme qui ne joue pas et dont l'appétit n'est pas féroce est toujours dynamique.*
10. *Un logicien dynamique, qui est véritablement convaincu, ne risque pas de se ruiner.*
11. *Un homme doté d'un appétit féroce n'a nul besoin de devenir conducteur de taxi, s'il est véritablement convaincu.*
12. *Un joueur qui est déprimé et qui ne risque pas de se ruiner reste debout jusqu'à quatre heures du matin.*
13. *Un homme qui est ruiné et qui ne dîne pas de deux côtelettes de porc aurait intérêt à devenir conducteur de taxi ou à se lever à cinq heures du matin.*

14. *Un joueur qui se couche avant quatre heures du matin n'a nul besoin de devenir conducteur de taxi, à moins qu'il ne soit doué d'un appétit féroce.*
15. *Un homme doué d'un appétit féroce, déprimé et qui ne risque nullement de se ruiner, est un joueur.*

Exercice 2. *Les gastronomes...*

1. *Tous les agents de police du secteur aiment dîner avec notre cuisinière.*
2. *Aucun homme aux cheveux longs ne peut être autre chose que poète.*
3. *Je n'ai jamais fait de séjour en prison.*
4. *Les cousins de notre cuisinière aiment tous le gigot froid.*
5. *Seuls les agents de police du secteur sont poètes.*
6. *Les hommes aux cheveux courts ont tous fait un séjour en prison.*

Exercice 3. *Les programmeurs...*

1. *Les étudiants se sentent abandonnés lorque les professeurs ne s'intéressent pas à eux*
2. *Les seuls étudiants en informatique se trouvent dans cet IUT.*
3. *Aucun étudiant ne peut écrire correctement un programme s'il n'a pas reçu une formation convenable.*
4. *Aucun étudiant de cet IUT n'est complètement nul.*
5. *Quand un étudiant se sent abandonné, il risque de sombrer dans la dépression.*
6. *Les professeurs ne s'intéressent pas aux étudiants que n'apprennent pas l'informatique.*

II. Résolution avec variable

On est ici en calcul des prédicats. On ne s'occupe que de clauses :
– symbole de prédicat, de variables, de constantes,
– $\neg$ et $\vee$ seulement,
– pas de quantificateurs.

1 Unification

1.1 Composants de substitutions et substitutions

DÉFINITION 1 (COMPOSANT DE SUBSTITUTION). *Si x est un symbole de variable et t un terme du calcul des prédicats, $(x|t)$ est un* composant de substitution *et indique la substitution de x par t, dans une certaine formule.* ♢

EXEMPLE 7. $A = p(x, y, z) \vee r(x, u)$, où x, y, z, u sont des symboles de variables, et p, r des symboles de prédicats.

Substitution de x par a (symbole de constante) dans A :

$$(x|a)A = p(a, y, z) \vee r(a, u)$$

On peut composer les composants de substitution (par exemple considérer $(y|b)(x|t)A$.

REMARQUE 4. L'ordre a de l'importance, la composition des composants de substitution n'est pas commutative.

REMARQUE 5. On ne peut substituer que des variables.

NOTATION : La substitution identique est notée $[]$.

EXEMPLE 8. Si x, y sont des symboles de variable, alors

$$\left\{ \begin{array}{ll} p(x) & (x|y)p(x) = p(y) \\ \neg p(y) & \neg p(y) \end{array} \right\} \vdash \square$$

mais si a, b sont des symboles de constante, alors :

$$\left\{ \begin{array}{l} p(a) \\ \neg p(b) \end{array} \right.$$

... on ne peut rien conclure.

Par ailleurs, comme il n'y a plus de quantificateurs, toutes les variables sont libres et toutes les substitutions licites (c'est-à-dire une variable par un terme quelconque : constante, variable, groupe opératoire) sont possibles.

1.2 Unificateurs

Étant donnés deux clauses A et B, on appelle unificateur de ces deux clauses toute substitution s telle que $sA = sB$.

- Pour deux formules quelconques, il n'existe pas forcément d'unificateur. Ainsi, pour $A = p(x)$ et $B = q(y)$: comme les deux symboles sont différents, elles ne sont pas unifiables.
- L'unificateur, s'il existe, n'est pas unique. Ainsi, si s est un unificateur de A et B ($sA = sB$) et si s' est une substitution quelconque, alors $s's$ est aussi unificateur.
- Si s est un unificateur de A et B, et que, pour tout unificateur s' de A et B, s' peut se mettre sous la forme $s' = s''s$, où s'' est une substitution, alors s est appelé *le plus grand unificateur* de A et de B (ce dernier n'existe pas forcément).

2 Résolution

Dans le système « RAV »(résolution avec variable) :
- Les formules sont des clauses du calcul des prédicats.
- Il n'y a pas d'axiomes.
- Deux règles d'inférences :

  résolution $\{a \vee f, \neg b \vee g\} \vdash \theta(\sigma f \vee g)\}$, avec σ une substitution de renommage, destinée à faire qu'il n'y ait aucune variable commune entre $a \vee f$ et $\neg b \vee g$, et θ un unificateur de (σa) et de b.

  diminution $\{a \vee b \vee f\} \vdash \sigma(b \vee f)$, où σ : unificateur de a et b.

EXEMPLE 9. $\{p(x) \vee p(y) \vee q(z)\} \vdash p(y) \vee q(z)$
$(x|y)$ est un unificateur de $p(x)$ et $p(y)$

2.1 Quelques exemples d'applications (en RAV)

EXEMPLE 10. $\{p(x, c) \vee r(x), \neg p(c, c) \vee q(x)\}$, avec :
- p, q, r : symboles de prédicats.
- x : symbole de variable.
- c : symbole de constante.

La substitution de x par c :
$\{p(c, c) \vee r(c), \neg p(c, c) \vee q(c)\} \vdash r(c) \vee q(c)$
est incorrect, car le renommage n'a pas été fait.

On commence donc par un renommage dans l'une des deux clauses (qui ne doivent plus avoir de variable commune) : par exemple dans C_2, $(x|y)$.

$$\{p(x,c) \vee r(x), \neg p(c,c) \vee q(y)\}$$

puis on substitue $(x|c)$ dans les deux clauses pour unifier $p(x,c)$ et $p(c,c)$.

$$\{p(c,c) \vee r(x), \neg p(c,c) \vee q(y)\} \vdash r(c) \vee q(y)$$

EXEMPLE 11. $\{p(x,f(a)), \neg p(a,y) \vee r(f(y))\}$, avec
- p, r : symbole de prédicat
- f : symbole opératoire
- x, y : symbole de variable
- a : symbole de constante

Pas de renommage, car pas de variable communes.

$$(y|f(a))(x|a)\{p(a,f(a)); \neg p(a,f(a)) \vee r(f(f(a)))\} \vdash r(f(f(a)))$$

EXEMPLE 12. $p(x,g(y)) \vee p(f(c),x) \vee r(x,y,z)$
p, r : symbole de prédicat
f, g : symbole opératoire
x, y, z : symbole de variable
c : symbole de constante

$(z|g(y))(x|f(c))\{p(x,g(y)) \vee p(f(c),x) \vee r(x,y,z)\} = p(f(c),g(y)) \vee idem \vee r(f(c),y,g(y)) \vdash p(f(c),g(y$

EXEMPLE 13 (UN EXEMPLE COMPLET). $C_1 = \neg cr(y) \vee cv(f(y),y)$
$C_2 = \neg cr(y) \vee \neg cv(x,y) \vee ma(x)$
$C_3 = \neg ar(x) \vee ma(x)$
$C_4 = \neg ma(x) \vee \neg ar(x) \vee \neg cr(y) \vee \neg cv(x,y)$
$C_5 = cr(a)$
$C_6 = \neg ma(x) \vee ar(x)$: mise sous forme de clause de la négation de la conclusion.

La conclusion est valide ssi on peut déduire la clause vide de cet ensemble de clauses.

$\{C_5, (y|a)C_1\} \vdash C_7 = cv(f(a), a)$
$\{C_5, (y|a)C_2\} \vdash C_8 = \neg cv(x, a) \vee ma(x)$
$\{C_5, (y|a)C_4\} \vdash C_9 = \neg ma(x) \vee \neg ar(x) \vee \neg cv(x, a)$
$\{C_7, (x|f(a))C_8\} \vdash C_{10} = ma(f(a))$
$\{C_7, (x|f(a))C_9\} \vdash C_{11} = \neg ma(f(a)) \vee \neg ar(f(a))$
$\{C_{10}, C_{11}\} \vdash C_{12} = \neg ar(f(a))$
$\{C_{10}, (x|f(a))C_6\} \vdash C_{13} = ar(f(a))$
$\{C_{12}, C_{13}\} \vdash \square$: la conclusion était valide.

Exercice 4. *On considère les formules suivantes de « PR ».*

- $A_1 : \forall x\,(p(x) \longrightarrow q(f(x)))$
- $A_2 : \forall x\,(q(x) \longrightarrow p(f(x)))$
- $B : \forall x\,(\neg p(x) \longrightarrow \vee \neg p(f(f(f(f(x))))))$

(p et q sont des symboles de prédicats, f est un symbole opératoire et x est un symbole de variable.)

Il s'agit de prouver que la formule B est la conclusion des formules A_1 et A_2. On mettra ces formules, ainsi que la négation de B, sous forme de clauses. On utilisera ensuite les règles d'inférence de « RAV » pour déduire, si possible, la clause vide de cet ensemble de clauses.

Exercice 5. *On considère l'ensemble de deux clauses $\{p(x) \vee p(y), \neg p(x) \vee \neg p(y)\}$ (p est un symbole de prédicat, x et y sont des symboles de variables).*

Peut-on en déduire quelque chose et, si oui, quoi ?

---

3 Stratégie de résolution

correction Une stratégie de résolution est correcte quand elle ne trouve dans l'arbre des déductions la clause vide que lorsqu'elle est effective.

complétude Une stratégie de résolution est complète quand elle conduit effectivement à la découverte de la clause vide (si elle figure dans l'arbre de déduction).

REMARQUE 6. Aucune stratégie à ce jour ne peut détecter l'absence de la clause vide dans l'arbre des déductions lorsqu'il est infini. Ce résultat est dû à l'indécidabilité du calcul des prédicats.

Deux problèmes :

1. gestion de l'ensemble des clauses,
2. parcours de l'arbre de déduction.

Gestions de l'ensemble des clauses Saturation (avec simplification).
Préférences des clauses simples : permet d'avancer plus rapidement.
Stratégies de résolution :
– gestion de l'ensemble des clauses
– gestion de l'arbre des déductions
Parcours de l'arbre :
– En profondeur d'abord : rapide, facile à programmer, se perd dans des branches infinies (arrête au bout d'un certain temps
– En largeur d'abord : lent, difficile à programmer, nécessite une mémoire importante.

Choix d'un sous-arbre

déduction linéaire On appelle déduction linéaire de racine C toute suite de clause : $C_0, \ldots C_n$ telle que $C_0 = C$ et $\forall i, C_{i+1}$ est obtenue par déduction entre C_1 et une autre clause. C'est une stratégie correcte, mais incomplète. Si F est un ensemble de clauses satisfiables, et si $F \cup \{C\}$ (une autre clause) est un ensemble de clauses insatisfiables, alors il existe une déduction linéaire de racine C menant à la clause vide.

déduction limitée aux entrées (« entrées » : clauses de départ) On appelle déduction limitée aux entrées de racine C toute suite de clauses ($C_0, C_1, \ldots C_n$) telle que $C_0 = C$ et que pour $i > 0, C_{i+1}$ est obtenue par déduction entre des clauses dont l'une est une entrée. Il n'y a pas de résultat analogue à celui énoncé pour les déductions linéaires, sauf dans un cas très particulier.

4 Exemples complets de résolution

4.1 La méthode de résolution

Pour ces divers exercices, on demande le travail suivant :

1. Mettre les diverses propositions sous forme de « PR » (on rappelle que cette traduction doit être littérale, c'est-à-dire qu'elle ne doit pas être l'occasion de commencer à « tirer des conclusions »).
2. Mettre les formules obtenues, ainsi que la négation de la conclusion proposée, sous forme de clauses (numérotées)

3. Utiliser la stratégie de « préférence des clauses simples » pour décider si la conclusion est valide ou non.

   On présentera la résolution de manière très claire : des numéros seront affectés aux clauses, les déductions seront explicites, par exemple :

   $$((x|a)C_i, C_k) \vdash C_n, \text{ avec } C_n = \ldots$$

   On supprimera au fur et à mesure les clauses subsumées, en indiquant, par exemple,

   La clause C_i est subsumée par la clause C_k.

4.2 Les exemples complets

Ils vont venir...

5 Exercices

Exercice 6. *On considère les propositions suivantes :*

- *Il arrive que des programmes tournent.*
- *Tout programme qui tourne est écrit par un programmeur.*
- *Seuls les programmeurs compétents écrivent des programmes qui tournent.*
- *Les programmeurs compétents qui ne sont pas engagés n'ont pas l'occasion d'écrire des programmes qui tournent.*
- *On n'engage que des programmeurs compétents.*

On désire vérifier que la conclusion suivante est correcte :

- *Certains programmeurs compétents sont engagés.*

Exercice 7. *Les exercices de logique...*

- *Aucun exercice de logique n'est intéressant.*
- *Certains exercices de logique ne sont pas intéressants.*

On désire vérifier que la conclusion suivante est valide :

- *Quelques exercices de logique ne sont pas mathématiques.*

Exercice 8. *La logique est difficile...*

– *Tous ceux qui comprennent la logique peuvent se passer de dormir.*
– *Certains étudiants ne peuvent pas se passer de dormir.*

On désire vérifier que la conclusion suivante est valide :

– *Certains étudiants ne comprennent pas la logique.*

Exercice 9. *Encore les exercices...*

– *Certains exercices sont franchement de mauvais goût.*
– *Les étudiants aiment tout ce qui est de bon goût.*

On désire vérifier que la conclusion suivante est valide :

– *Il y a certains exercices que les étudiants n'aiment pas.*

Exercice 10. *Les professeurs...*

– *Certaines personnes instruites manquent de générosité.*
– *Tous les professeurs sont généreux.*

On souhaite vérifier que la conclusion suivante est valide :

– *Les professeurs ne sont pas des personnes instruites.*

Exercice 11. *Les étudiants de première année d'informatique sont répartis en six groupes, du groupe A au groupe F.*

Chacun de ces groupes est subdivisé en deux sous-groupes (A_1, A_2, B_1, B_2, etc.)

Pour l'élaboration de l'emploi du temps du second semestre, les contraintes suivantes doivent être respectées :

– *Si A_1 et A_2 suivent la même matière, s'il en est de même pour B_1 et B_2, et si E_1 suit la même que F_2, C_1 doit suivre la même que D_2.*
– *Si A_1 et A_2 suivent la même matière, s'il en est de même pour F_1 et F_2, et si B_1 suit la même que C_2, D_1 ne peut pas suivre la même que E_2.*
– *Si C_1, C_2, D_1 et D_2 suivent la même matière, et si A_1 ne suit pas la même matière que B_2, E_1 ne peut pas suivre la même que F_2.*
– *Si A_1 et A_2 suivent la même matière, s'il en est de même pour D_1 et D_2, et si B_1 ne suit pas la même que C_2, E_1 doit suivre la même que F_2.*
– *Si E_1 et E_2 suivent la même matière, s'il en est de même pour F_1 et F_2, et si C_1 suit la même que D_2, A_1 doit suivre la même que B_2.*

– *Si B_1, B_2, C_1 et C_2 suivent la même matière, et si E_1 ne suit pas la même matière que F_2, D_1 doit suivre la même que E_2.*

On souhaite vérifier que la conclusion suivante est valide :

– *A tout instant de la journée, il y aura un groupe au moins dont les deux sous-groupes ne suivront pas la même matière.*

On pourra utiliser le prédicat $mm(x, y)$ qui signifie que le sous-groupe x_1 suit la même matière que le sous groupe y_2, et les symboles de CONSTANTES a, b, c, d, e et f qui représentent les groupes.

Exercice 12. *Les intellectuels achètent des livres...*

– *Il y a des livres.*
– *Certains intellectuels jouent en bourse.*
– *Seuls les intellectuels lisent des livres.*
– *Seuls les intellectuels se ruinent en jouant en bourse.*
– *Tout livre a un lecteur.*
– *Les intellectuels ruinés ne lisent pas de livres.*

On souhaite vérifier que la conclusion suivante est valide :

– *Certains intellectuels ne sont pas ruinés.*

Exercice 13. *Les châtelains sont riches...*

– *On ne prête qu'aux riches.*
– *Il y a des châteaux.*
– *Les gens riches qui ont contracté un prêt ne possèdent pas de châteaux.*
– *Seuls les gens riches sont propriétaires de châteaux.*
– *Tous les châteaux ont des propriétaires.*

On souhaite vérifier que la conclusion suivante est valide :

– *Tous les gens riches ne recourent pas à l'emprunt.*

III. Clauses de Horn

1 Définition

Une clause de Horn est une clause ne comportant qu'un seul littéral positif.

REMARQUE 7. ...donc de la forme $p \vee \neg q_1 \vee \ldots \vee \neg q_n = p \vee \neg(q_1 \wedge \ldots \wedge q_n)$: de la forme $\neg QouP$, c'est-à-dire $P \rightarrow Q$.

2 Déduction ordonnée

Une déduction est dite ordonnée quand elle porte sur le littéral de tête (entre deux clauses).

3 Le résultat évoqué dans le paragraphe précédent

Si F est un ensemble de clauses de Horn ordonnées satisfiables, si C est une clause ne comportant que des littéraux négatifs, et que $T \cup \{C\}$ est insatisfiable, alors il existe une déduction limitée aux entiers et ordonnée de racine C qui conduit à la clause vide.

Fin du Chapitre

Chapitre 13

Exercices sur la logique

Exercice 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 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 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

Quatrième partie

Langages, grammaires et automates

Chapitre 14

Compilation, langages et grammaires

I. Introduction à la compilation

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.

2 Les diverses phases d'une compilation

Détaillons succintement les différentes phases d'une compilation...

2.1 L'analyse lexicale

On analyse le flux d'entrée de manière à le découper en unités lexicales, ou *lexèmes*.

EXEMPLE 1. Dans *if (temps == beau)* { etc., les unités lexicales sont « if », « (», « temps », « == », « beau », «) ».

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) ?

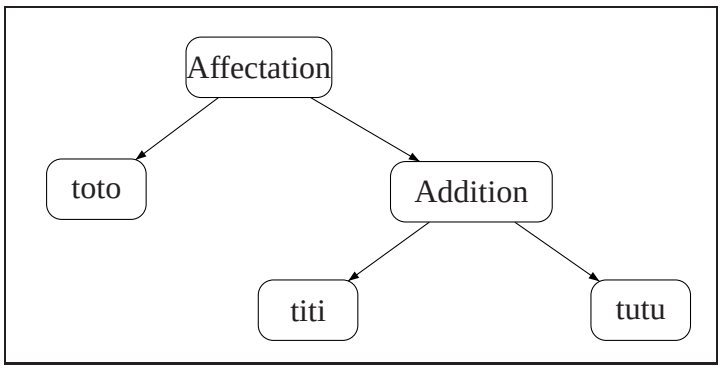
2.3 L'analyse sémantique

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 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 1. Certains compilateurs utilisent des structures arborescentes :



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 2. En général, ces différentes phases sont menées en parallèle.

II. Les grammaires

1 Définition de la notion de grammaire

DÉFINITION 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).

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).

3 Les symboles terminaux

DÉFINITION 2 (SYMBOLE TERMINAL). *Un* symbole terminal *est un symbole qui peut effectivement intervenir dans le texte analysé.* ♢

EXEMPLE 3. Dans *if (temps == beau), if* est un symbole terminal (on trouve le mot dans le programme).

NOTATION : Les symboles terminaux sont entourés par : « ».

4 Les symboles non terminaux

DÉFINITION 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 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 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. $\diamondsuit$

EXEMPLE 5. $< programme\ en\ C >::=< entete >< suite\ de\ fct >$

La grammaire est terminée quand tous les SNT ont reçu au moins une définition.

5 Exercices

Exercice 1. *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 2. *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 3. *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 4. *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 5. *Le langage considéré est le prototypage des fonctions en C.*

Donner une grammaire BNF de ce langage.

Exercice 6. *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. »

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 7. *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.*

---

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.

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 3. Toujours commencer par l'analyseur syntaxique pur.

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);
}
```

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, express
      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 15

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 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 1. *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 2. *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 3. *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 $>$ $\mid$ $<$ 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 4. *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 I : 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 5. *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 2. 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 3. 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 16

Automates Finis

I. Automates finis

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 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 1. *Donnez des exemples de machines.*

REMARQUE 1. À chaque instant, la condition interne de la machine, y compris la mémoire, constitue son état.

2 Mécanismes

C'est le type le plus simple de machine :

DÉFINITION 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 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 I (EXISTENCE D'UN CYCLE) : Un mécanisme entre nécessairement dans une boucle infinie (on dit : *un cycle*).

En effet, le nombre d'états est fini.

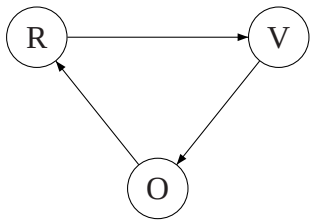
DÉFINITION 3 (ÉTAT-REPOS). *S'il existe un état $e \in E$ tel que $t(e) = e$, cet état est appelé* état-repos . ♢

REMARQUE 2. Un mécanisme qui entre dans un tel état n'en sort évidemment plus.

EXEMPLE 1. 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 3. *L'exemple précédent possède-t-il un cycle ? un état-repos ? Sinon, le modifier pour.*

II. Automates finis à comportement déterminé

1 Définition

DÉFINITION 4 (AUTOMATE FINI À COMPORTEMENT DÉTERMINÉ). *On appelle* automate fini à comportement déterminé*(AFD) tout triplet* (E, I, t)*, où*

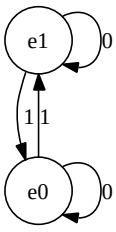
- E *est un ensemble fini (l'ensemble des* états*),*
- I *est le* vocabulare *de l'automate : c'est l'ensemble fini des symboles admis en entrée,*
- $t : E \times I \to E$ *est la* fonction de transition d'états *: si l'automate se trouve dans l'état* $e \in E$*, il réagit à l'entrée* $i \in I$ *en passant 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 2. 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...



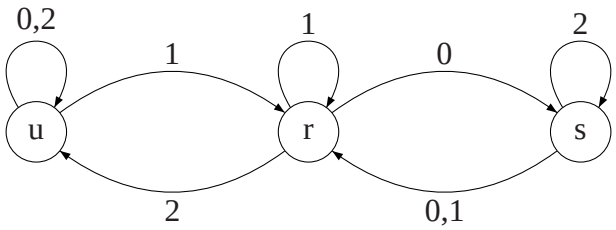
La table qui donne les valeurs de la fonction t est appelée *table de transition d'états* de l'automate considéré :

t	0	1
e_0	e_0	e_1
e_1	e_1	e_0

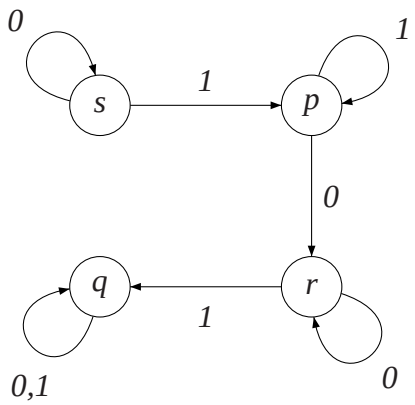
Exercice 4. *Représenter le graphe de l'automate fini M dont la table de transition des états est*

t	0	1	2
r	s	r	u
s	r	r	s
u	u	r	u

Réponse :



Exercice 5. *Écrire la table de transition d'états de l'automate dont le graphe est représenté dans la figure suivante :*



Réponse :

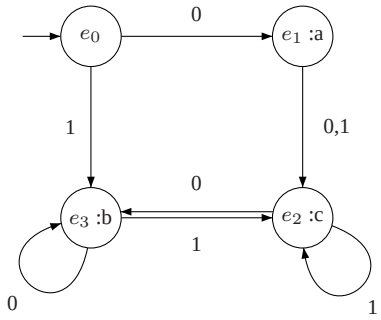
	0	1
p	r	p
q	q	q
r	r	q
s	s	p

2 Automates finis avec sorties (machines de Moore et de Mealy)

DÉFINITION 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* $g(e) \in V$*.* $\diamondsuit$

EXEMPLE 3. 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 3. Une telle machine est aussi appelée *traducteur* (elle « traduit » l'entrée 0001001 en une sortie $acbcbbc$).

DÉFINITION 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 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.

3 Automates de Moore

DÉFINITION 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 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

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 8 (LANGAGE). *Ce système formel constitue le* langage associé à l'automate M*.* ♢

NOTATION : $\mathcal{L}(M)$

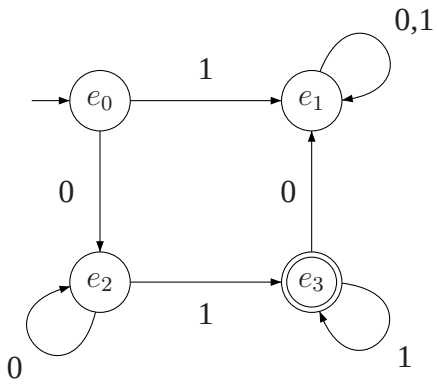
Réciproquement, étant donné un langage $\mathcal{L}$, on peut éventuellement construire un automate de Moore M tel que le langage associé à M soit $\mathcal{L}$: $\mathcal{L} = \mathcal{L}(M)$.

REMARQUE 6. Cela n'est pas possible pour tous les langages. Quand c'est possible, cet automate analyse les mots du langage.

2 Exemple et exercices

EXEMPLE 4. 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 6. *Décrire le langage $\mathcal{L}(M)$ de l'automate de Moore M dont la table de transition des états est :*

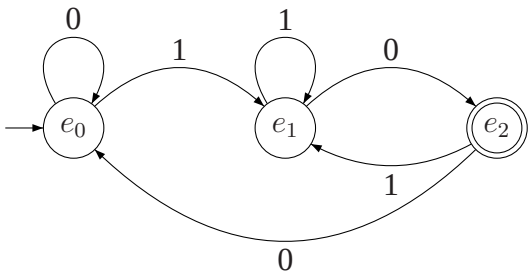
t	0	1
e_0	e_1	e_2
e_1	e_1	e_2
e_2	e_2	e_1

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.

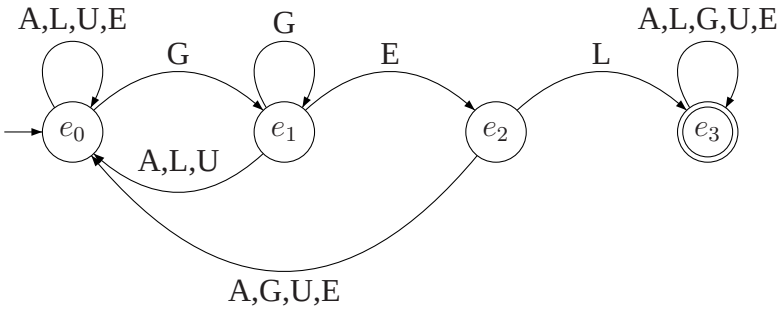
Exercice 7. *Sur l'alphabet $I = \{0, 1\}$, construire l'automate de Moore dont le langage est l'ensemble de tous les mots sur I se terminant par 10.*

Réponse :



Exercice 8. *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

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 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 7. Il se peut donc qu'une entrée puisse conduire un automate vers plusieurs états possibles ou qu'elle laisse l'automate indifférent.

EXEMPLE 5. 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.

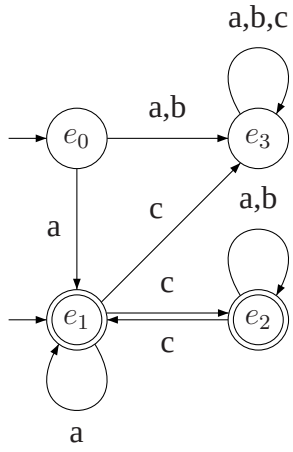


Table de transitions

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 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 6. 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 9. *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.

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.

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 7. 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 $\mathcal{T} : \mathcal{E} \times \mathcal{I} \to \mathcal{E}, (Y, x) \longmapsto \mathcal{T}(Y, x) = Y_x$.

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 $\mathcal{T}(S, x) = S_x$.

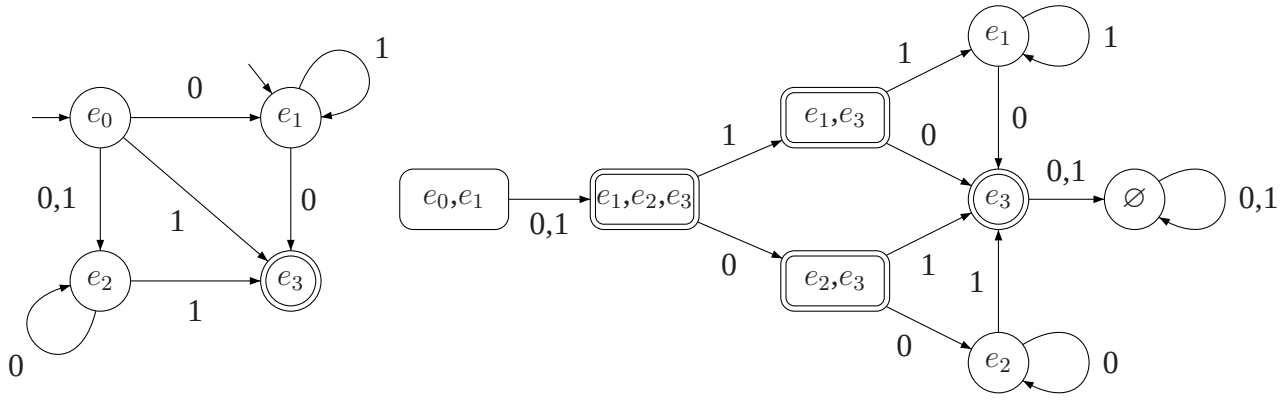
On recommence l'opération pour chacun des états S_x ainsi obtenus (pour les diverses entrées x), etc.

REMARQUE 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..

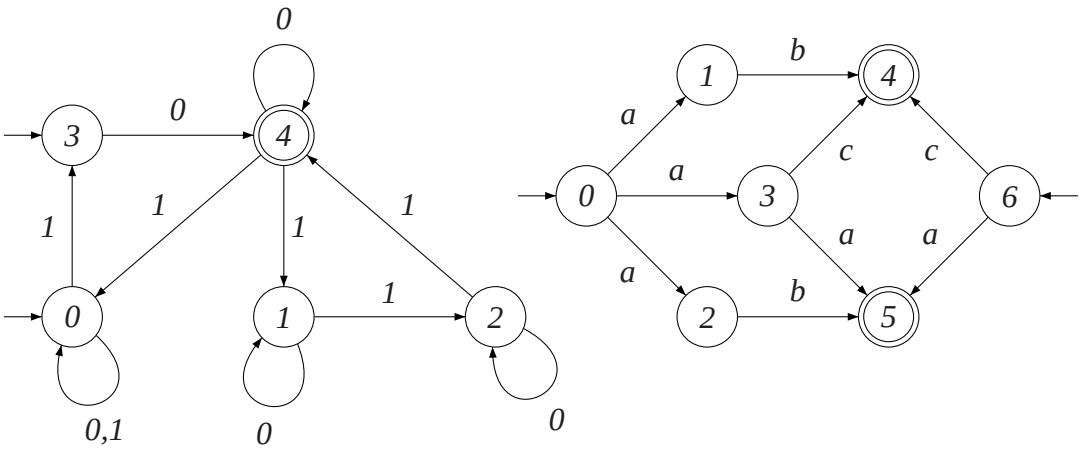
REMARQUE 9. Il se peut qu'aucun état ne soit accessible depuis l'un quelconque des états d'un état Y_x de $\mathcal{M}$, sur une entrée y.

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 8. Un exemple...



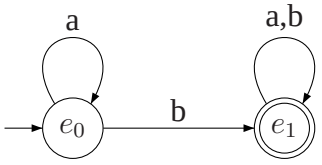
Exercice 10. *Construire des automates de Moore équivalents aux AFND ci-dessous :*



Exercice 11. *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 12. *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.*

---

1 Propriétés d'un automate à n états

Exercice 13. *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.

---

2 Les palindromes

Exercice 14 (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 15 (Suite palindrome). *Soit $\Sigma = \{a, b\}$. Construire un AFD qui accepte les palindromes de longueur 3.*

Exercice 16. *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 17

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

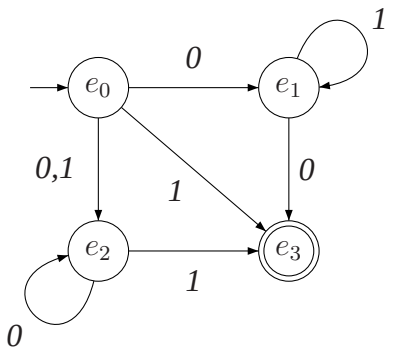
Soit (E, I, t) un AFD et $\mathcal{R}$ une relation d'équivalence sur E.

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 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 $t(e, x)$.

2 Définition

DÉFINITION 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 2. *La relation d'équivalence $\mathcal{R}$ de l'exercice précédent est-elle une congruence d'automates ?*

---

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 3. *Représenter le graphe de l'automate M dont la table de transition des états est*

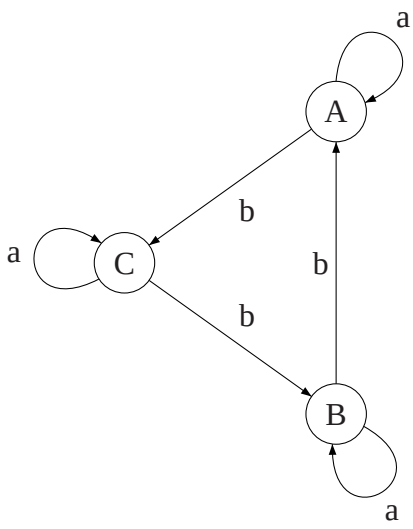
t	a	b
e_0	e_0	e_4
e_1	e_1	e_0
e_2	e_2	e_4
e_3	e_5	e_2
e_4	e_4	e_3
e_5	e_3	e_2

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
$A = \{e_0, e_2\}$	$\{e_0, e_2\}$	$\{e_4\}$
$B = \{e_1, e_3, e_5\}$	$\{e_1, e_3, e_5\}$	$\{e_0, e_2\}$
$C = \{e_4\}$	$\{e_4\}$	$\{e_1, e_3, e_5\}$



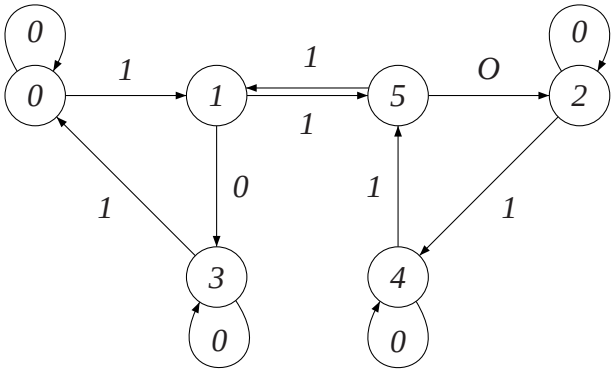
Exercice 4. *Soit M l'automate fini dont la table de transition des états est*

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 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 $E = \{0, 1, 2, 3, 4, 5\}$:

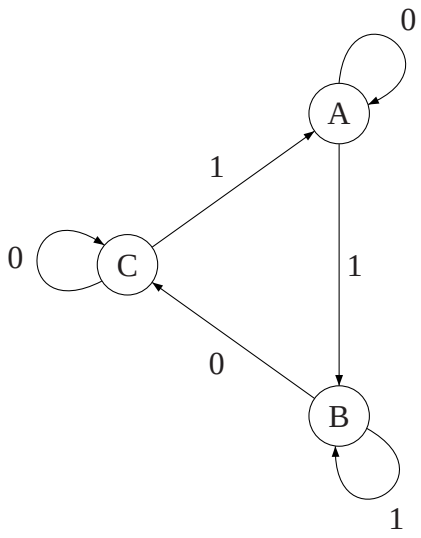
	0	1	2	3	4	5
0	1	0	0	0	1	0
1	0	1	0	0	0	1
2	0	0	1	1	0	0
3	0	0	1	1	0	0
4	1	0	0	0	1	0
5	0	1	0	0	0	1

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 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

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 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 3 (ÉQUIVALENCE DE NÉRODE). *Cette relation est manifestement une relation d'équivalence, elle est appelée* équivalence de Nérode *associée à M.* $\diamondsuit$

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 ».

2 L'algorithme

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.

2.2 La pratique

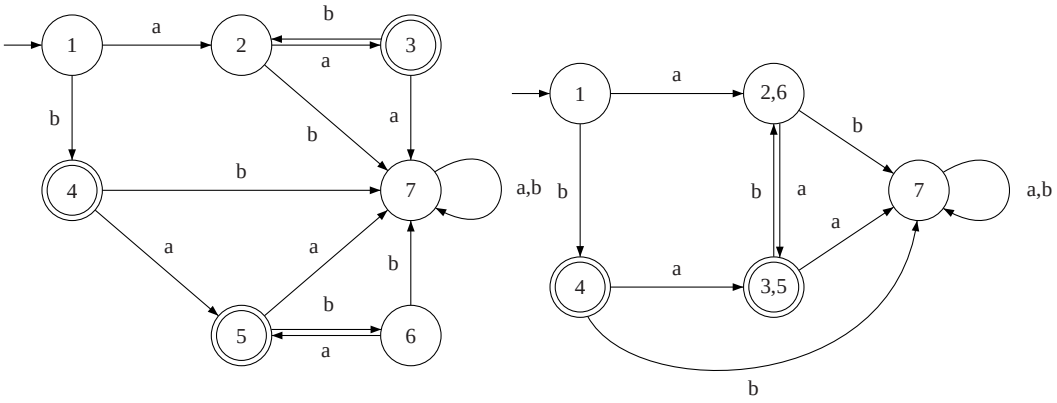
Cet algorithme n'est pas utilisable dans la pratique. On fait plutôt...

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 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 1. Un automate de Moore et l'automate simplifié.



Exercice 6. *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 7. *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

1 Dual d'un automate

Soit $M = (E, I, t, S, A)$ un automate quelconque (AFD ou AFND).

DÉFINITION 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.

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 8. *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 9. *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 10. *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

1 Outils

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).

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.

2 Méthodes d'optimisation

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.

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 18

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 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...

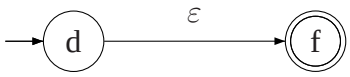
III. Algorithme

1. Décomposer l'expression en ses sous-expressions.

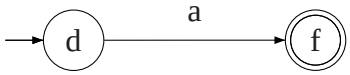
2. En utilisant les règles (a) et (b) ci-dessous, construire un AFND pour les symboles terminaux de la grammaire ou la chaîne vide (si un même symbole a apparaît plusieurs fois dans l'expression rationnelle, un AFND séparé est construit pour chacune de ses occurences).

3. Combiner ensuite récursivement les AFND de base en utilisant la règle (c) jusqu'à obtenir l'AFND pour l'expression rationnelle toute entière.

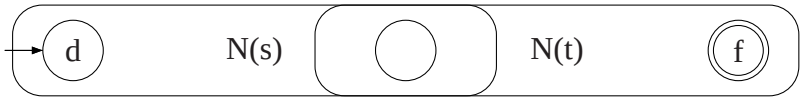
   (a) Pour $<>$, construire l'AFND :



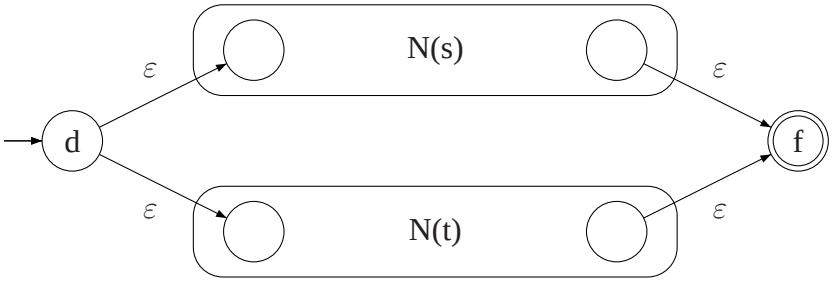
   (b) Pour $a \in \Sigma$, construire l'AFND :



   (c) Si $N(s)$ et $N(t)$ sont les AFND pour les expressions rationnelles s et t,
   – Pour st, on construit l'AFND :

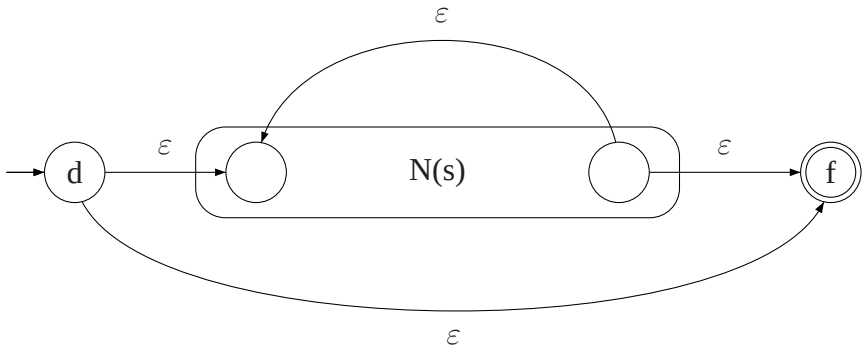


   L'état initial de $N(t)$, qui est état d'acceptation pour $N(s)$, perd ce double caractère dans la nouvelle construction.
   – Pour $s|t$, on construit l'AFND



   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 $N(s^*)$:

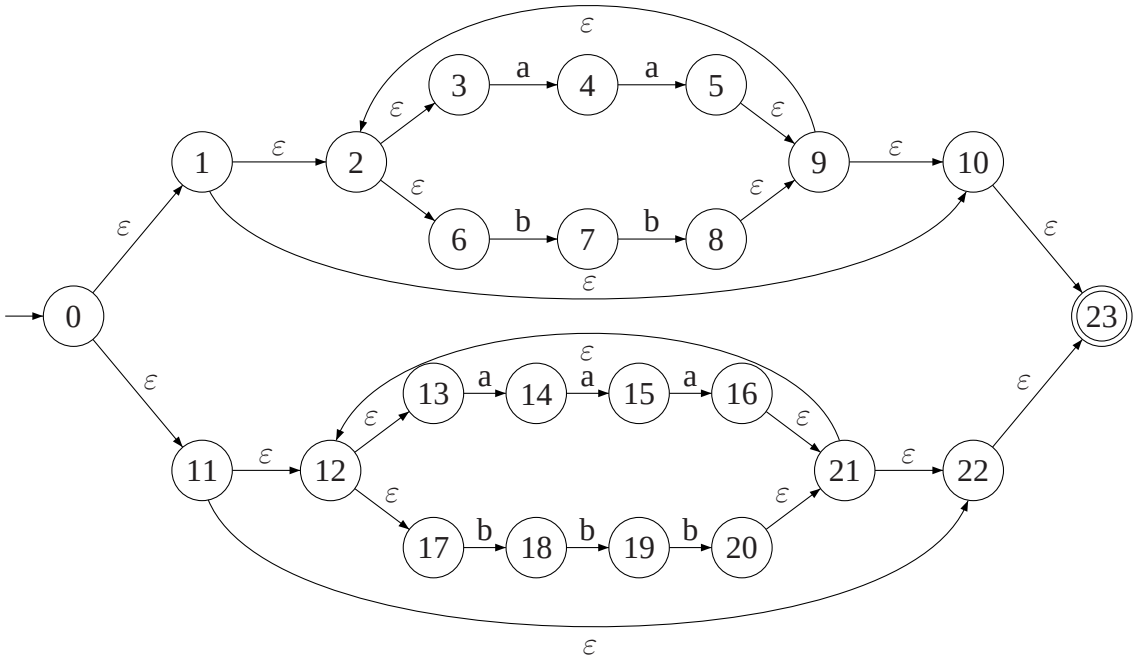


Les états initiaux et les états d'acceptation de $N(s)$ perdent leurs qualités.

– 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 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 1. 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 2. *Finalisez l'automate de l'exercice précédent. Le déterminiser, puis obtenir l'automate minimal reconnaissant le même langage.*

Exercice 3 (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 4. *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 19

Automates à pile

On l'a dit, les expressions rationnelles ne permettent pas de représenter tous les langages.

EXEMPLE 1. Le langage défini par $\{a^n b^n | n \in \mathbb{N}^*\}$ n'est pas représentable par une expression rationnelle.

Exercice 1. *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 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.

1 Automate à pile non déterministe

1.1 Définition

DÉFINITION 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 2. Le symbole de pile initial n'est pas toujours noté p_0.

1.2 Transition

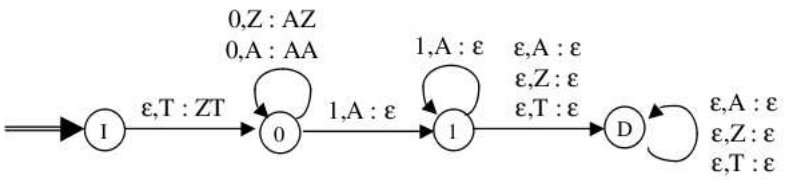
DÉFINITION 2 (TRANSITION). *Lorsque $((e, x, p), (e', q))$ appartient au graphe de la relation t, on parle de la transition $(e, x, p) \longmapsto (e', q)$.* ♢

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 2 (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 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.

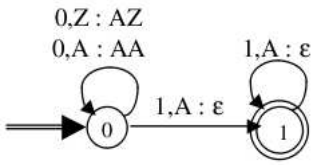
2 Automate à pile déterministe

2.1 Définition

DÉFINITION 3 (AUTOMATE À PILE DÉTERMINISTE). *Dans le cas d'un* automate à pile déterministe*, t est une fonction sur son domaine de définition, et* $(e', q) = t(e, x, p)$.$\diamondsuit$

REMARQUE 4. En particulier si t est définie pour le triplet (e, x, p), $t(e, x, p)$ est unique.

EXEMPLE 3 (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 5. Cet automate à pile déterministe reconnaît, par état final, le même langage que l'exemple précédent.

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 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 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

1 Encore quelques définitions...

DÉFINITION 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).* ♢

DÉFINITION 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. ♢

DÉFINITION 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) ♢

NOTATION : On peut noter cela : $(e_1, w_1, q_1) \overset{*}{\vdash} (e_n, w_n, q_n)$.

DÉFINITION 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*,* $\diamondsuit$

REMARQUE 8. On peut envisager des reconnaissances par combinaison de deux de ces conditions, voire les trois simultanément.

On démontre que...

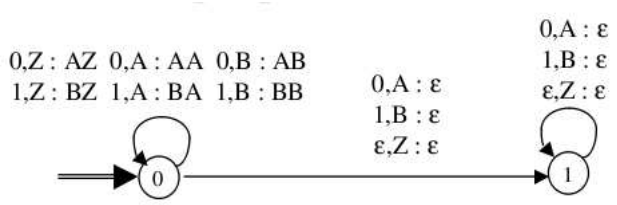
PROPRIÉTÉ 1 : Tous ces types de reconnaissance peuvent se ramener à la seule reconnaissance par pile vide (éventuellement avec un automate beaucoup plus compliqué).

C'est pourquoi nous n'envisagerons plus que cette dernière dans la suite. Enfin,

DÉFINITION 8 (LANGAGE RECONNU). *Le* langage reconnu *par automate est l'ensemble des mots reconnus par cet automate (par le même mode de reconnaissance).* $\diamondsuit$

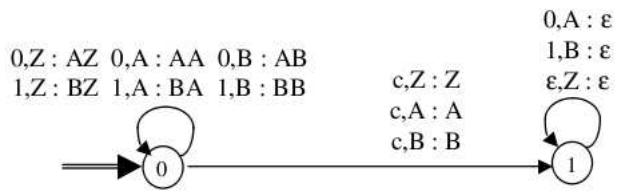
2 Premiers exemples

EXEMPLE 4. 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 5. 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 $\{Z, A, B\}$.)

---

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 :
- $(e_0, 0, p_0) \rightarrow (e_0, 0)$
- $(e_0, 0, 0) \rightarrow (e_0, 00)$
- $(e_0, 1, 0) \rightarrow (e_1, \varepsilon)$
- $(e_1, 1, 0) \rightarrow (e_1, \varepsilon)$

Exercice 2. *Représentez cet automate.*

III. Construction d'un automate à pile

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.

2 Utilisation d'un symbolisme

On introduit un nouveau symbolisme (développé dans la suite) pour cette même grammaire ; il se comprend aisément :

$$\begin{array}{lll} E & -> & T \\ E & -> & T + E \\ T & -> & F \\ T & -> & F * T \\ F & -> & (E) \\ F & -> & a \\ F & -> & b \\ & \ldots & \\ F & -> & z \end{array}$$

...en admettant comme symboles de variables les caractères alphabétiques minuscules.

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 $A \rightarrow x$ 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 :

$$\begin{array}{lllll}
(1) & (e_0, \varepsilon, p_0) & \longmapsto & (e_0, E) \\
(2) & (e_0, \varepsilon, E) & \longmapsto & (e_0, T) \\
(3) & (e_0, \varepsilon, E) & \longmapsto & (e_0, T + E) \\
(4) & (e_0, \varepsilon, T) & \longmapsto & (e_0, F) \\
(5) & (e_0, \varepsilon, T) & \longmapsto & (e_0, F * T) \\
(6) & (e_0, \varepsilon, F) & \longmapsto & (e_0, (E)) \\
(7) & (e_0, a, F) & \longmapsto & (e_0, \varepsilon) \\
\vdots & \vdots & \vdots & \vdots \\
(8) & (e_0, z, F) & \longmapsto & (e_0, \varepsilon) \\
(9) & (e_0, +, +) & \longmapsto & (e_0, \varepsilon) \\
(10) & (e_0, *, *) & \longmapsto & (e_0, \varepsilon) \\
(11) & (e_0, (, () & \longmapsto & (e_0, \varepsilon) \\
(12) & (e_0,),)) & \longmapsto & (e_0, \varepsilon)
\end{array}$$

4 Exercices

Exercice 3. *Soit l'automate à pile défini par* $\Sigma = \{a, b\}$, $E = \{q_0, q_1, q_2\}$, $P = \{p, A\}$ *et les transitions suivantes*

$$\begin{array}{rcl}
(q_0, a, p_0) & \mapsto & (q_1, A) \\
(q_1, b, A) & \mapsto & (q_2, \epsilon) \\
(q_1, a, p) & \mapsto & (q_1, Ap) \\
(q_2, b, A) & \mapsto & (q_2, \epsilon)
\end{array}$$

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 4. *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 5. *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 6. *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 7. *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 8. *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 9. *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 20

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 I : 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 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

1 Définitions

DÉFINITION 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 $\alpha \rightarrow \beta$, qui se lit « α se réécrit en β ».

REMARQUE 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 « : := »).

2 Types de grammaires de Chomsky

On distingue divers types de grammaires de Chomsky :

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.

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.

2.3 Les grammaires algébriques, ou de type 2

Toute production est de la forme $A \rightarrow \alpha$, 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).

2.4 Les grammaires régulières, ou de type 3

Chaque production est de l'une des formes $A \rightarrow xB$ ou $A \rightarrow x$, 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).

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 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. $S \rightarrow aSBC$
2. $S \rightarrow aBC$
3. $CB \rightarrow BC$
4. $aB \rightarrow ab$
5. $bB \rightarrow bb$
6. $bC \rightarrow bc$
7. $cC \rightarrow cc$

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 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 21

Exercices sur les grammaires, langages et automates

Exercice 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 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 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 22

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

1 Définitions

DÉFINITION 1 (GRAPHE NON ORIENTÉ, SOMMET, ARÊTE). *Un* graphe non orienté $G = (V, E)$ *est défini par l'ensemble fini* $V = \{v_1, v_2, ..., v_n\}$ *dont les éléments sont appelés* sommets*, et par l'ensemble fini* $E = \{e_1, e_2, ..., e_m\}$ *dont les éléments sont appelés* arêtes*.*

Une arête e *de l'ensemble* E *est définie par une paire non-ordonnée de sommets, appelés les* extrémités *de* e*. Si les extrémités coïncident, on parle de* boucle*.*

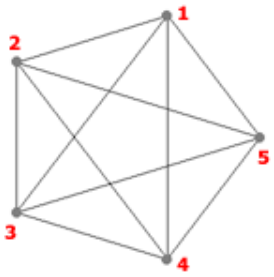
Si l'arête e *relie les sommets* a *et* b*, on dira que ces sommets sont* adjacents*, ou incidents avec* e*, ou encore que l'arête* e *est incidente avec les sommets* a *et* b*.*

On notera qu'un graphe a au moins un sommet ; on notera par la suite ordre *d'un graphe son nombre de sommets.* ♢

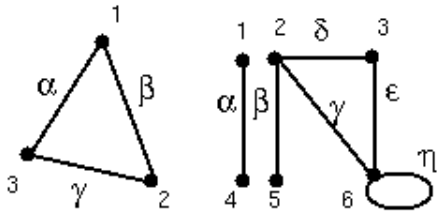
REMARQUE 1. Dans le présent chapitre, et ses proches successeurs, graphe signifie graphe non orienté (même quand cela n'est pas spécifié). Il existe aussi des graphes orientés ; ils seront étudiés plus loin.

2 Exemples

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 :



3 Degré, chaîne

3.1 Degré d'un sommet, d'un graphe

DÉFINITION 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 I (LEMME DES POIGNÉES DE MAINS) : La somme des degrés des sommets d'un graphe est égale à deux fois le nombre d'arêtes.

DÉFINITION 3 (DEGRÉ D'UN GRAPHE). *Le degré d'un graphe est le degré maximum de tous ses sommets.* ♢

Exercice 1. *Calculez les degrés des sommets, et le degré des graphes ci-dessus.*

DÉFINITION 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 2. *Les graphes précédent sont-ils réguliers ?*

Exercice 3. *Représentez un graphe 3-régulier.*

3.2 Chaîne

DÉFINITION 5 (CHAÎNE). *Une chaîne dans G, est une suite de la forme*

$$(v_0, e_1, v_1, e_2, ..., v_{k-1}, e_k, v_k)$$

- *ayant pour éléments alternativement des sommets (v_i) et des arêtes (e_i),*
- *commençant et se terminant par un sommet,*
- *et telle que les extrémités de e_i soient v_{i-1} et v_i, $i = 1, ..., k$.* ♢

v_0 est appelé le *départ* de la chaîne et v_k l'*arrivée*.

REMARQUE 2. On a choisi ici de réserver le terme de *chemin* aux graphes orientés.

DÉFINITION 6 (SOMMET ACCESSIBLE). *Dans un graphe (orienté ou non), on dit que le sommet t est* accessible *à partir du sommet s s'il existe une chaîne menant de s à t.* ♢

REMARQUE 3. On dit aussi qu'on peut *atteindre* t à partir de s.

DÉFINITION 7 (CHAÎNE ÉLÉMENTAIRE). *Une chaîne dans lequel tous les sommets sont différents s'appelle une* chaîne élémentaire *.* ♢

REMARQUE 4. On parle aussi de chaîne *simple* .

REMARQUE 5. Une chaîne simple a forcément toutes ses arêtes différentes, et ne contient évidemment pas de boucle.

PROPRIÉTÉ II (EXISTENCE DE CHAÎNES ÉLÉMENTAIRES) : Étant donné une chaîne qui joint s et t (différents), on peut toujours lui enlever arêtes et sommets pour obtenir une chaîne *élémentaire* joignant s à t.

4 circuit-cycle

DÉFINITION 8 (CIRCUIT). *Une chaîne de longueur n dont le départ et l'arrivée coïncident s'appelle un* circuit *de longueur n.* ♢

EXEMPLE 1. Une boucle est un circuit de longueur 1.

DÉFINITION 9 (CYCLE). *Un circuit dont tous les sommets et toutes les arêtes sont différentes, s'appelle un* cycle. ♢

Exercice 4. *Représentez un graphe qui admet :*

1. *un circuit,*
2. *un cycle.*

REMARQUE 6. Dans le cas non orienté, un circuit qui à tous ses sommets différents n'a pas forcément toutes ses arêtes différentes.

DÉFINITION 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 5. *Représentez un graphe simple (resp. qui n'est pas simple).*

---

5 Exercices

Exercice 6. *On s'intéresse aux graphes 3-réguliers.*

1. *Construisez de tels graphes ayant 4 sommets, 5 sommets, 6 sommets, 7 sommets.*
2. *Qu'en déduisez-vous ?*

Exercice 7. *Montrez qu'un graphe simple a un nombre pair de sommets de degré impair.*

Exercice 8. *Est-il possible de relier 15 ordinateurs de sorte que chaque appareil soit relié avec exactement trois autres ?*

Exercice 9. *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 7 autres. Notons I l'ensemble de ces 7 sommets. Il reste 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

1 Graphes planaires

DÉFINITION 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 10. *Représentez un graphe planaire.*

Exercice 11. *Représentez un graphe non planaire.*

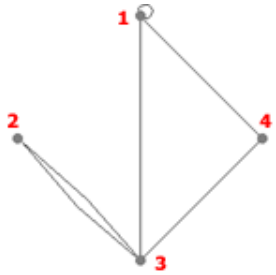
REMARQUE 7. Les graphes planaires seront plus systématiquement étudiés au chapitre suivant.

2 Multigraphes

Les graphes étudiés ici sont simples, mais on peut imaginer des graphes avec une arête qui relie un sommet à lui-même (une boucle), ou plusieurs arêtes reliant les deux mêmes sommets (voir ci-dessous, à gauche).

DÉFINITION 12 (MULTIGRAPHE). *Dans ce cas, on parle de* multigraphe. ♢

EXEMPLE 2 (MULTIGRAPHE). Un exemple de multigraphe :



V = {1, 2, 3, 4}
E = {(1,1), (1,3), (1,4), (2,3), (2,3), (3,4)}

---

3 Graphes connexes

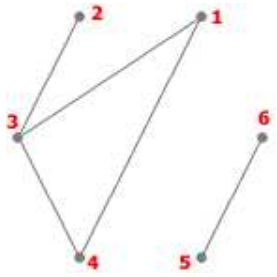
DÉFINITION 13 (GRAPHE CONNEXE). *Un graphe est* connexe *s'il est possible, à partir de n'importe quel sommet, de rejoindre tous les autres en suivant les arêtes.* ♢

REMARQUE 8. C'est en particulier le cas lorsqu'à partir d'un sommet on peut atteindre tous les autres sommets.

Exercice 12. *Représenter un graphe non orienté connexe, et un graphe non connexe.*

DÉFINITION 14 (COMPOSANTES CONNEXES). *Un graphe non connexe se décompose en* composantes connexes. ♢

EXEMPLE 3 (GRAPHE NON CONNEXE). Exemple d'un graphe n'étant pas connexe :



V = {1, 2, 3, 4, 5, 6}
E = {(1,3), (1,4), (2,3), (3,4), (5,6)}

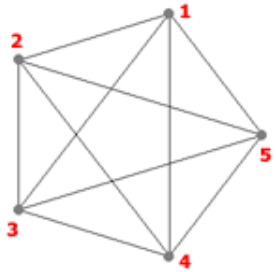
Ici, les composantes connexes sont {1, 2, 3, 4} et {5, 6}.

---

4 Graphes complets

DÉFINITION 15 (GRAPHE COMPLET). *Un graphe est* complet *si chaque sommet du graphe est relié directement à tous les autres sommets* ♢

EXEMPLE 4 (GRAPHE COMPLET K_5). Exemple d'un graphe complet :



V = {1, 2, 3, 4, 5}
E = {(1,2), (1,3), (1,4), (1,5), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5)}

Exercice 13. *Un tournoi d'échecs oppose 6 personnes. Chaque joueur doit affronter tous les autres.*

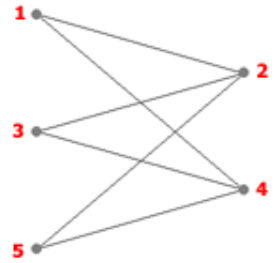
1. *Construisez un graphe représentant toutes les parties possibles.*
2. *Quel type de graphe obtenez-vous ?*
3. *Si chaque joueur ne joue qu'un match par jour, combien de jours faudra-t-il pour terminer le tournoi ?*
4. *Aidez-vous du graphe pour proposer un calendrier des matches.*

---

5 Graphes biparti

DÉFINITION 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.* ♢

EXEMPLE 5 (GRAPHE BIPARTI). Exemple d'un graphe biparti :



V = $\{1, 2, 3, 4, 5\}$
E = $\{(1,2), (1,4), (2,3), (2,5), (3,4), (4,5)\}$
Avec les notations de la définition, on a X=$\{1, 3, 5\}$ et Y=$\{2, 4\}$, ou vice versa.

Un tel graphe se note $K_{3,2}$. Plus généralement,

NOTATION : On note $K_{m,n}$ un graphe biparti liant m sommets à n sommets.

PROPRIÉTÉ III : Ces graphes $K_{m,n}$ possèdent mn arêtes.

6 Exercices

Exercice 14. *Trois professeurs P1, P2, P3 doivent donner ce lundi un certain nombre d'heures de cours à trois classes C1, C2, C3 :*

- *P1 doit donner deux heures de cours à C1 et une heure à C2.*
- *P2 doit donner une heure de cours à C1, une heure à C2 et une heure à C3 ;*
- *P3 doit donner une heure de cours à C1, une heure à C2 et deux heures à C3.*

On demande...

1. *Comment représenter cette situation par un graphe ?*
2. *Quel type de graphe obtenez-vous ?*
3. *Combien faudra-t-il de plages horaires au minimum ?*

Aidez-vous du graphe pour proposer un horaire du lundi pour ces professeurs.

Exercice 15. *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.*

Exercice 16. *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 : $\frac{n(n+1)}{2}$.

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...

1 Matrice d'incidence

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 17 (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 17. *Représentez le graphe non orienté dont la matrice d'incidence est :*

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

1.2 Résultat

PROPRIÉTÉ IV : Si $s_1, \ldots, s_n$ sont les sommets d'un graphe non orienté, alors :

$$\begin{pmatrix} d(s_1) \\ d(s_2) \\ \vdots \\ d(s_n) \end{pmatrix} = J \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

2 Matrice d'adjacence

2.1 Présentation

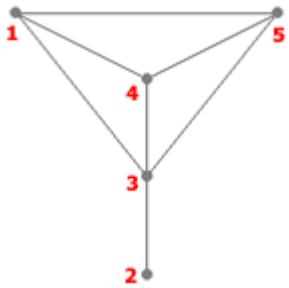
On peut représenter un graphe non orienté par une matrice d'adjacences.

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.

2.2 Exemple

Considérons le graphe G :



Voici la matrice d'adjacences du graphe G :

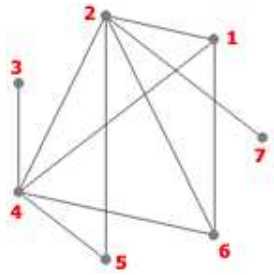
$$M = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

2.3 Propriétés de la matrice d'adjacence

Cette matrice a plusieurs caractéristiques :

PROPRIÉTÉ V : 1. Elle est carrée : il y a autant de lignes que de colonnes.
2. Il n'y a que des zéros sur la diagonale. Un 1 sur la diagonale indiquerait une boucle.
3. Elle est symétrique : $m_{ij} = m_{ji}$.

Exercice 18. *Décrivez le graphe G ci-dessous par une matrice d'adjacences.*



Exercice 19. *Calculez M^2 et M^3 pour la matrice d'adjacence de l'exemple précédent. Que représentent-ils ?*

PROPRIÉTÉ VI : 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 de s à t.

Exercice 20. *Démontrez ce résultat, par récurrence.*

Exercice 21. *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.*

2.4 Avantages et inconvénients de cette représentation

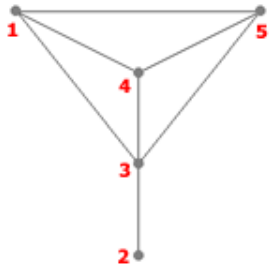
Avantages – Représentation si beaucoup d'arêtes : cardinalité de l'ensemble d'arc proche de cardinalité de l'ensemble des sommets au carré.
– Accès facile à la relation d'adjacence.

Inconvénients – Moins adapté aux graphes pondérés.
– Nécessite une grande quantité de mémoire.

3 Listes d'adjacence

On peut encore représenter un graphe en donnant pour chacun de ses sommets la liste des sommets auxquels il est adjacent.

EXEMPLE 6. 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

3.1 Avantages et inconvénients de cette représentation

Avantages – Représentation adaptée aux graphes peu denses.
- Représentation des graphes pondérés.
- Faible occupation mémoire.

Inconvénients – Difficulté de savoir si deux sommets sont effectivement connectés.
- Difficulté de mise en œuvre.

Fin du Chapitre

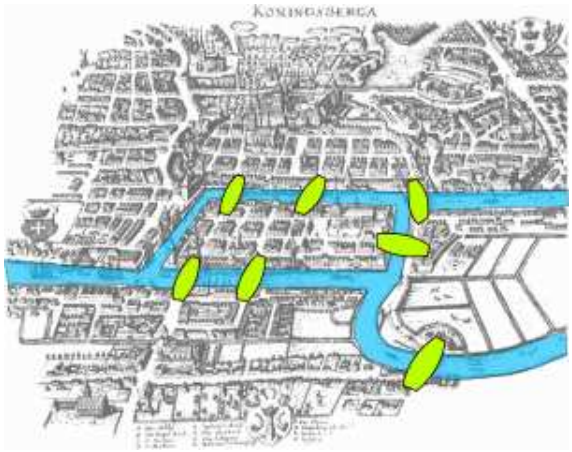
Chapitre 23

Graphes eulériens, planaires et hamiltoniens

I. Circuits eulériens

1 Introduction : les ponts de Königsberg

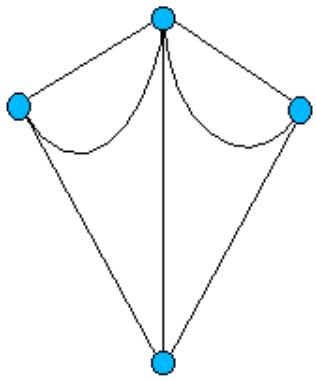
La question à l'origine de la théorie des graphes est due à Euler[1], en 1736 : dans cette partie de la ville de Königsberg :



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 :

[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...



Le problème de départ se ramène alors à la question suivante : peut-on trouver un chemin permettant d'emprunter une, et une seule fois chaque arête ? Un raisonnement élémentaire sur ce graphe en donne la réponse.

2 Définitions

DÉFINITION 1 (CHEMIN EULÉRIEN). *On appelle* chemin eulérien *un chemin contenant une et une seule fois toutes les arêtes du graphe.* ♢

DÉFINITION 2 (CIRCUIT EULÉRIEN). *On appelle* circuit eulérien *un circuit contenant une et une seule fois toutes les arêtes du graphe.* ♢

REMARQUE 1. C'est un circuit : le point de départ et celui d'arrivée coïncident.

DÉFINITION 3 (GRAPHE EULÉRIEN). *Un* graphe eulérien *est un graphe possédant un circuit eulérien.* ♢

REMARQUE 2. On peut passer plusieurs fois par un sommet donné.

Exercice 1. *Donnez des exemples de graphes possèdant des circuits eulériens, et d'autres exemples de graphes n'en possédant pas.*

Exercice 2. *Soit G un graphe non eulérien. Est-il toujours possible de rendre G eulérien en lui rajoutant un sommet et quelques arêtes ?*

---

3 Résultat d'Euler

PROPRIÉTÉ 1 : Dans le cas des graphes non orientés, il y a équivalence entre :
- posséder un circuit eulérien,
- être connexe, avec $d(s)$ pair pour tout sommet s.

et entre :
- posséder un chemin eulérien,
- être connexe, avec au plus deux sommets d'ordre impair.

PREUVE 3 :
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. ■

4 Exercice : les dominos

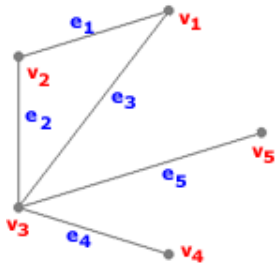
Exercice 3. *On considère des dominos dont les faces sont numérotées 1, 2, 3, 4 ou 5.*

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 ?*

II. Graphes planaires

1 Graphes partiels et sous-graphes

EXEMPLE 1 (LE GRAPHE G). On considère, dans ce paragraphe, pour illustrer les définitions à venir, le graphe



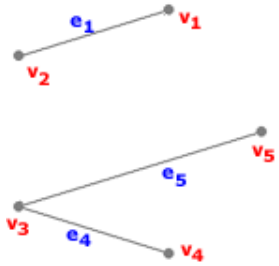
Soit :

- G=(V, E),
- V=$\{v_1, v_2, v_3, v_4, v_5\}$,
- E=$\{e_1 = (v_1, v_2), e_2 = (v_2, v_3), e_3 = (v_1, v_3), e_4 = (v_3, v_4), e_5 = (v_3, v_5)\}$

DÉFINITION 4 (GRAPHE PARTIEL). *Soit G = (V, E) un graphe. Le graphe G' = (V, E') est un* graphe partiel *de G, si E' est inclus dans E.* ♢

REMARQUE 3. Autrement dit, on obtient G' en enlevant une ou plusieurs arêtes au graphe G.

EXEMPLE 2 (GRAPHE PARTIEL DE G). Voici un exemple de graphe partiel de G



Ici,

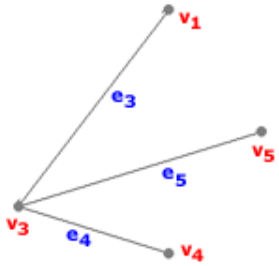
- V'=V,
- E'=$\{e_3, e_4, e_5\}$.

DÉFINITION 5 (SOUS-GRAPHE). *On dit que le graphe* (V', E') *est un* sous-graphe *du graphe* (V, E) *si*

1. $V' \subset V$,
2. $E' \subset E$,
3. $E' = \{(x, y) \mid (x, y) \in E \wedge x \in V' \wedge y \in V'\}$ ♢

REMARQUE 4. Un sous-graphe d'un graphe donné est donc obtenu en y enlevant des sommets et toutes les arêtes incidentes à ces sommets.

EXEMPLE 3 (SOUS-GRAPHE DE G). Voici un exemple de sous-graphe de G :

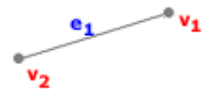


Ici,

- V'=$\{v_1, v_3, v_4, v_5\}$,
- E'=$\{e_3, e_4, e_5\}$.

DÉFINITION 6 (SOUS-GRAPHE PARTIEL). *Un graphe partiel d'un sous-graphe est un* sous-graphe partiel *de G.* ♢

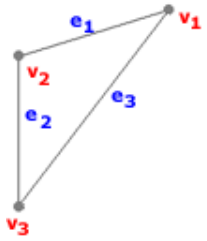
EXEMPLE 4 (SOUS-GRAPHE PARTIEL DE G). Voici un sous-graphe partiel de G :



Ici,
- V'=$\{v_1, v_2, v_3, v_4\}$,
- E'=$\{e_1, e_4\}$.

DÉFINITION 7 (CLIQUE). *On appelle* clique *un sous-graphe complet de G.* ♢

EXEMPLE 5 (UNE CLIQUE DE G). Exemple de clique de G :



Ici,
- V'=$\{v_1, v_2, v_3\}$,
- E'=$\{e_1, e_2, e_3\}$.

DÉFINITION 8 (STABLE). *On appelle* stable *un sous-graphe de G sans arêtes.* ♢

EXEMPLE 6 (UN STABLE DE G). Voici un stable de G :

Ici,
- V'=$\{v_1, v_4, v_5\}$,
- E'=$\{\}$.

Exercice 4. *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 : Un graphe ayant six sommets possède soit une clique à trois sommets, soit un stable à trois sommets.

Cela n'est plus valable pour les graphes à cinq sommets.

Exercice 5. *Combien un graphe d'ordre n possède-t-il de sous-graphes ?*

Réponse : $2^n - 1$.

2 Graphe planaire

On rappelle la définition...

DÉFINITION 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 5. Rappelons que les arêtes ne sont pas forcément rectilignes.

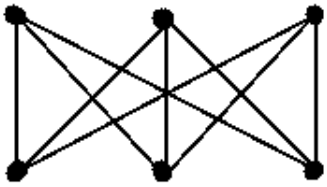
Exercice 6. *Un graphe peut-il être planaire s'il possède un sous-graphe qui ne l'est pas ?*

Réponse : Non : une fois dessiné dans un plan, un graphe planaire a forcément tous ses sous-graphes qui le sont aussi.

3 Exemples

EXEMPLE 7 (K_n). On rappelle que l'on note K_n tout graphe non orienté d'ordre n, tel que toute paire de sommets est reliée par une unique arête. Alors K_3 est planaire, et K_5 ne l'est pas :

EXEMPLE 8 ($K_{3,3}$). Il est non planaire



Exercice 7. *Dessiner K_1, K_2, K_3 et K_4. Sont-ils planaires ? Et qu'en est-il de $K_n, n \geqslant 6$?*

Réponse : Oui.

Exercice 8. *Combien d'arêtes possède le graphe K_n ?*

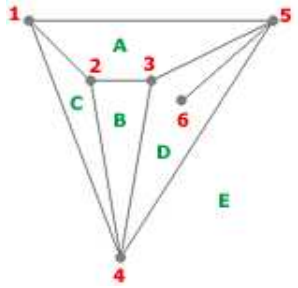
Réponse : $\dfrac{n(n-1)}{2}$.

4 Problèmes de dénombrement

4.1 Cartes, régions

DÉFINITION 10 (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 9. 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.

DÉFINITION 11 (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 10. Dans le graphe ci-dessus, d(A)=4, d(B)=3, d(C)=3, d(D)=5, d(E)=3.

REMARQUE 6. 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...

4.2 Lemme des régions

PROPRIÉTÉ II (LEMME DES RÉGIONS) : La somme des degrés des régions d'une carte connexe est égale à deux fois le nombre d'arêtes.

4.3 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 III (EULER) :

$$S - A + R = 2$$

Exercice 9. *Utilisez ce résultat pour prouver que $K_{3,3}$ n'est pas planaire.*

REMARQUE 7. Cette formule est applicable aux polyèdres convexes :
- le nombre de sommets,
- moins le nombre d'arêtes,
- plus le nombre de faces

...est toujours égal à 2.

5 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.

5.1 Graphes homéomorphes

DÉFINITION 12 (GRAPHES HOMÉOMORPHES). *Deux graphes sont* homéomorphes *s'ils peuvent tous les deux être obtenus à partir d'un graphe commun en remplaçant les arêtes par des chaînes simples.* ♢

Exercice 10. *Représentez deux graphes homéomorphes.*

5.2 Théorème de Kuratowski

La réponse au problème de caractérisation des graphes planaires est...

PROPRIÉTÉ IV (KURATOWSKI) : Un graphe est non planaire si et seulement si il contient un sous-graphe homéomorphe à $K_{3,3}$ ou K_5.

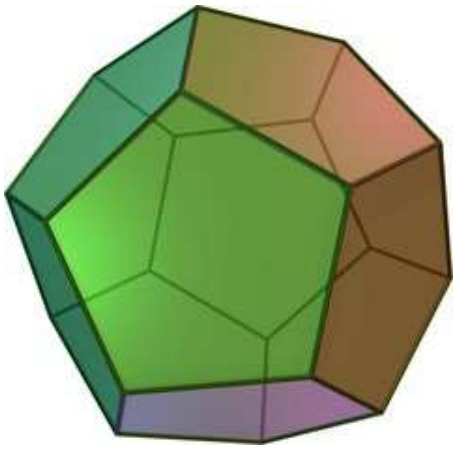
Exercice 11. *En admettant, comme nous y invite le résultat précédent, que K_5 n'est pas planaire, déterminer les valeurs de n pour lesquelles K_n est planaire.*

Réponse : On a vu (exercice précédent) que K_n est planaire pour $n \leqslant 4$. K_5 n'est pas planaire. Enfin, pour $n \geqslant 5$, K_n contient des sous-graphes K_5...

III. Circuit hamiltonien

1 Les dodécaèdres de Hamilton

Le dodécaètre 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.

2 Définition

DÉFINITION 13 (CIRCUIT HAMILTONIEN). *C'est un circuit qui passe par tous les sommets du graphe, une et une seule fois.* ♢

REMARQUE 8. Dans le cas des graphes orientés (voir plus loin), on parle de graphe *hamiltonien* s'il est possible de trouver un cycle passant une et une seule fois par tous les sommets.

3 Résultat

Contrairement aux graphes eulériens, il n'existe pas de caractérisation simple des graphes hamiltoniens ou semi-hamiltoniens. On peut cependant énoncer quelques propriétés et conditions suffisantes :

PROPRIÉTÉ V :
1. Un graphe possédant un sommet de degré 1 ne peut être hamiltonien.
2. Si un sommet dans un graphe est de degré 2, alors les deux arêtes incidentes à ce sommet doivent faire partie du cycle hamiltonien.
3. Les graphes complets K_n sont hamiltoniens.

Exercice 12. *Cherchez les raisons (élémentaires) justifiant les trois points précédents.*

PROPRIÉTÉ VI (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 VII (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 4 :
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 13. *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.*

---

4 Le problème du voyageur de commerce

Ajoutons à cette présentation des graphes hamiltoniens le problème dit *du voyageur de commerce* : on munit notre graphe d'une application de l'ensemble des arêtes vers un ensemble donné (correspondant aux longueurs des arêtes, en kilomètres, pour le problème original), et on cherche à visiter tous les sommets (i.e. toutes les villes) en minimisant la distance parcourue.

Fin du Chapitre

Chapitre 24

Arbres et arborescence

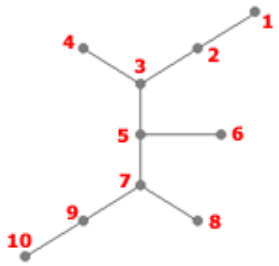
I. Présentation générale

1 Définitions

DÉFINITION 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*

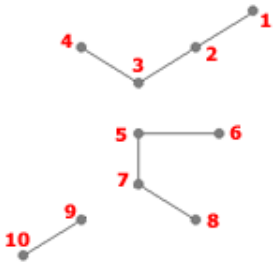
- *les* feuilles *dont le degré est 1 ;*
- *les* nœuds internes *dont le degré est supérieur à 1.* ♢

EXEMPLE 1. Un exemple d'arbre :



DÉFINITION 2 (FORÊT). *Un graphe sans cycles mais non connexe est appelé une* forêt. ♢

EXEMPLE 2. Un exemple de forêt :



2 Caractérisation des arbres

PROPRIÉTÉ I : 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).

3 Nombre minimal de feuilles

PROPRIÉTÉ II : Tout arbre fini avec au moins deux sommets comporte au moins deux feuilles.

4 Exercices

Exercice 1. *Démontrez la propriété précédente.*

Indication : Récurrence.

Exercice 2. *Combien d'arbres différents existe-t-il avec 3, 4, 5 sommets ?*

Indication : Réfléchir sur le nombre de feuilles.

Exercice 3. *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 4. *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

1 Présentation

Le codage de Prüfer est une manière très compacte de décrire un arbre.

2 Codage

2.1 La méthode

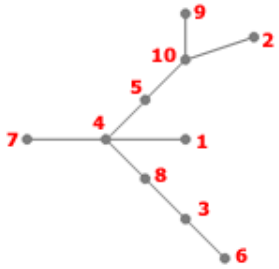
Soit l'arbre $T = (V, E)$ et supposons $V = 1, 2, ..., n$. L'algorithme ci-dessous fournira le code de T, c'est-à-dire une suite S de $n-2$ termes employant (éventuellement plusieurs fois) des nombres choisis parmi $\{1, ..., n\}$.

PROPRIÉTÉ III (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.

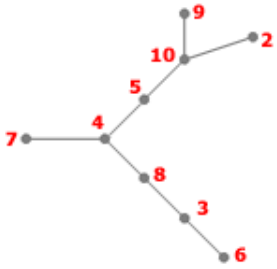
2.2 Exemple de codage

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

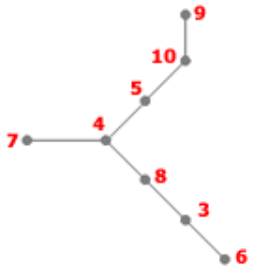
Étape 1 :



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

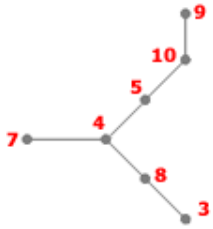
S={4}



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

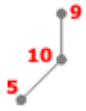
S = {4,10,3,8,4}

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.

3 Décodage

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.

3.2 Exemple de décodage

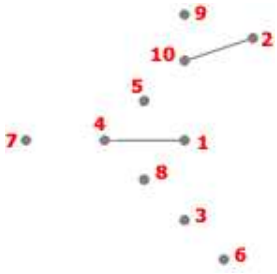
Étape 0 : arbre à décoder

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}

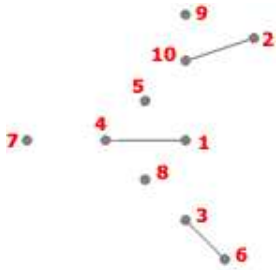


Étape 2 :

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 :

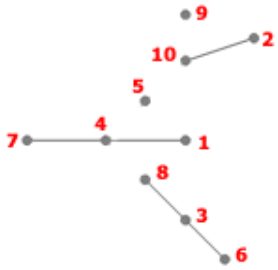
1 : 4
2 : 10
3 : 6
4 : 1
6 : 3
10 : 2

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

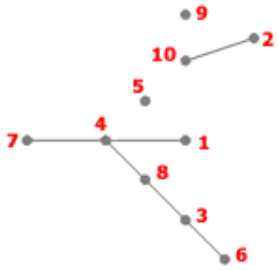
I={4,5,7,8,9,10}
S={4,4,5,10}



Étape 5 :

1 : 4
2 : 10
3 : 6, 8
4 : 1, 7
6 : 3
7 : 4
8 : 3
10 : 2

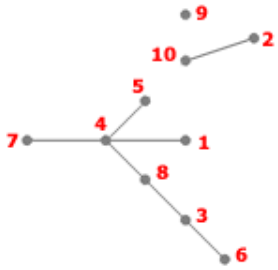
I={4,5,8,9,10}
S={4,5,10}



Étape 6 :

1 : 4
2 : 10
3 : 6, 8
4 : 1, 7, 8
6 : 3
7 : 4
8 : 3, 4
10 : 2

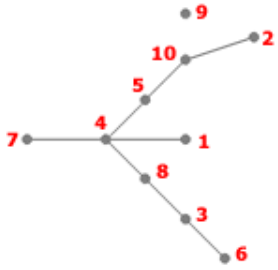
I={4,5,9,10}
S={5,10}



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

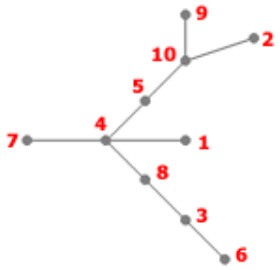
I={5,9,10}
S={10}



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={}

4 Théorème de Cayley

PROPRIÉTÉ IV (CAYLEY, 1857) : Le nombre d'arbres que l'on peut construire sur n ($n > 1$) sommets numérotés est égal à n^{n-2}.

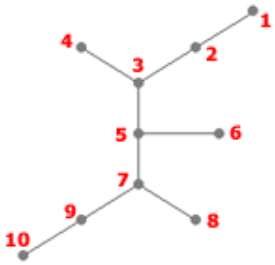
PREUVE 5 :
La preuve est immédiate en utilisant le codage de Prüfer.

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. ■

5 Exercices

Exercice 5. *Décrivez l'arbre ci-dessous à l'aide du codage de Prüfer.*



Exercice 6. *Dessinez l'arbre correspondant à la suite S=1,1,1,1,1,1,1,1,1.*

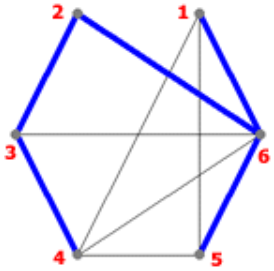
III. Arbres couvrants

1 Définition

DÉFINITION 3 (ARBRE COUVRANT). *Un* arbre couvrant *(ou* arbre maximal*) est un graphe partiel qui est aussi un arbre.* ♢

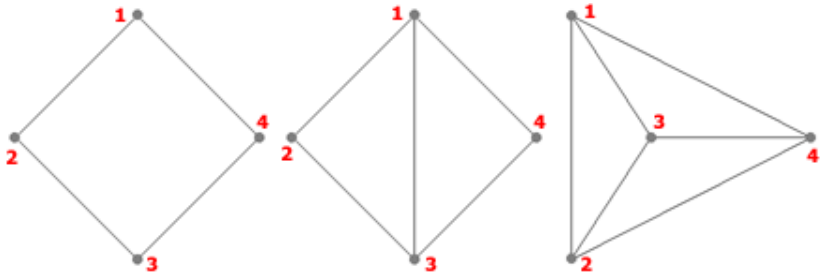
REMARQUE 1. On rappelle qu'un graphe partiel de G est obtenu en enlevant des arêtes (mais pas de sommets) à G.

EXEMPLE 3. Un des arbres couvrants (en bleu) d'un graphe donné.



1.1 Exercices

Exercice 7. *Combien d'arbres couvrants différents les graphes suivants possèdent-ils ?*



Exercice 8. *Trouvez un arbre couvrant minimal pour chaque graphe ci-dessus.*

---

2 Arbre maximal de poids minimum

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.

2.2 L'algorithme de Kruskal

Voici un algorithme célèbre permettant de résoudre ce problème :

PROPRIÉTÉ V (L'ALGORITHME DE KRUSKAL) : **Données :** Graphe $G = (V, E)$ ($|V| = n$, $|E| = m$) et pour chaque arête e de E, son poids $c(e)$.

Résultat : Arbre ou forêt maximale $A = (V, F)$ de poids minimum.

Algorithme : Trier et renuméroter les arêtes de G dans l'ordre croissant de leur poids : $c(e_1) < c(e_2) < ... < c(e_m)$.

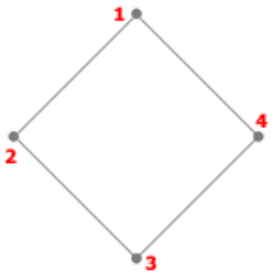
1. Poser $F := \varnothing$, $k := 0$
2. Tant que $k < m$ et $|F| < n - 1$ faire
   - Début
     - si e_{k+1} ne forme pas de cycle avec F alors $F := F \cup \{e_{k+1}\}$
     - $k := k + 1$
   - Fin

Exercice 9. *Le tableau suivant donne 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 10. *Trouvez un arbre maximal de poids minimum du graphe ci-dessous (les chiffres en bleu représentent le poids des arêtes) :*



IV. Arborescence

1 Définitions et exemples

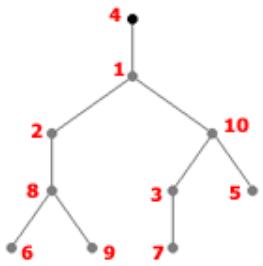
1.1 Définition d'une arborescence

DÉFINITION 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.

1.2 Exemple

EXEMPLE 4. Sur l'arborescence ci-dessous, la racine est le sommet 4. Les sommets 5, 6, 7 et 9 sont les feuilles.



1.3 Rang des sommets

On peut, dans une arborescence, assigner un rang aux sommets :

DÉFINITION 5 (RANG D'UN SOMMET). *Le* rang *d'un sommet d'une arborescence est la distance de ce sommet à la racine.* ♢

EXEMPLE 5. 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 6 (HAUTEUR D'UNE ARBORESCENCE). *On dira que la* hauteur *de l'arborescence est le rang maximum* ♢

EXEMPLE 6. L'exemple ci-dessus a une hauteur de 4.

---

2 Arborescences ordonnées

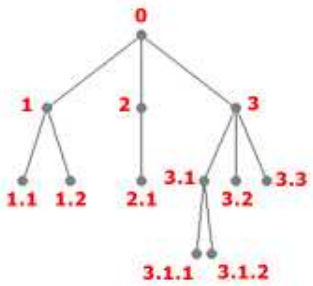
2.1 Définition

DÉFINITION 7 (ARBORESCENCE ORDONNÉE). *Une* arborescence ordonnée *est une arborescence dont les arêtes partant de chaque sommet sont ordonnées.* ♢

REMARQUE 2. On peut systématiquement étiqueter les sommets d'un tel arbre comme suit :

- on attribue 0 à la racine r,
- puis 1, 2, 3, ... aux sommets qui sont adjacents à r.
- 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 7. La figure ci-dessous illustre le procédé.



DÉFINITION 8. *Cet ordre (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.* ♢

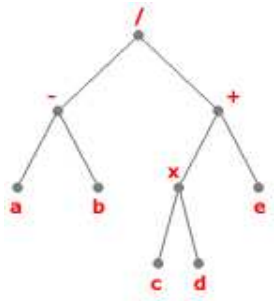
REMARQUE 3. Il est identique à l'ordre obtenu en parcourant de haut en bas la branche la plus à gauche de l'arbre, puis la branche immédiatement à droite, et ainsi de suite jusqu'à la branche la plus à droite.

REMARQUE 4. On parle aussi de *parcours en profondeur* de l'arbre, 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.

2.2 Lien avec les expressions algébriques

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 8. Par exemple, l'aborescence ci-dessous représente l'expression arithmétique $(a-b)/((c\times d)+e)$:



On observe que les variables de l'expression $(a, b, c, dete)$ sont les feuilles et que les opérations sont les autres sommets.

REMARQUE 5. L'arbre doit être ordonné car $a-b$ et $b-a$ conduisent au même arbre, mais pas au même arbre ordonné.

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 9 (NOTATION POLONAISE). *On appelle cette nouvelle notation la* notation polonaise *dans sa forme préfixée ou directe.* ♢

REMARQUE 6. Par analogie, en notation polonaise postfixée ou inverse, on place les symboles après les arguments ; certaines calculatrices - notamment les HP - utilisent la notation polonaise inverse).

EXEMPLE 9. L'expression $(a-b)/((c \times d)+e)$ devient ainsi $/-ab+\times cde$.

2.3 Exercices

Exercice 11. *Combien d'arborescences existe-t-il sur n sommets numérotés ?*

Exercice 12. *Étant donnée l'expression algébrique* $(2x+y)(5a-b)^3$,

1. *dessinez l'arborescence ordonnée correspondante ;*
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 en notation polonaise directe.*

3 Codage de Huffman

3.1 Présentation

Le *codage de Huffman* est une méthode de compression statistique de données qui permet de réduire la longueur du codage d'un alphabet.

Le code de Huffman (1952) est un code de longueur variable optimal, c'est-à-dire tel que la longueur moyenne d'un texte codé soit minimale. On observe ainsi des réductions de taille de l'ordre de 20 à 90%.

3.2 Principe

Le principe de l'algorithme de Huffman consiste à recoder les octets rencontrés dans un ensemble de données source avec des valeurs de longueur binaire variable. L'unité de traitement est ramenée au bit.

Huffman propose de recoder les données qui ont une occurrence très faible sur une longueur binaire supérieure à la moyenne, et recoder les données très fréquentes sur une longueur binaire très courte.

Ainsi, pour les données rares, nous perdons quelques bits regagnés pour les données répétitives.

EXEMPLE 10. Dans un fichier ASCII le ”w” apparaissant 10 fois aura un code très long : 0101000001000. Ici la perte est de 40 bits (10 x 4 bits), car sans compression, il serait codé sur 8 bits au lieu de 12.

Par contre, le caractère le plus fréquent comme le ”e” avec 200 apparitions sera codé par 1. Le gain sera de 1400 bits (7 x 200 bits). On comprend l'intérêt d'une telle méthode.

3.3 Propriété de préfixe :

Le codage de Huffman a 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 11. Si un caractère est représenté par la combinaison binaire 100 alors la combinaison 10001 ne peut être le code d'aucune autre information.

Dans ce cas, l'algorithme de décodage interpréterait les 5 bits comme une succession du caractère codé 100 puis du caractère codé 01

REMARQUE 7. Cette caractéristique du codage de Huffman permet une codification à l'aide d'une structure d'arbre binaire.

3.4 Méthode

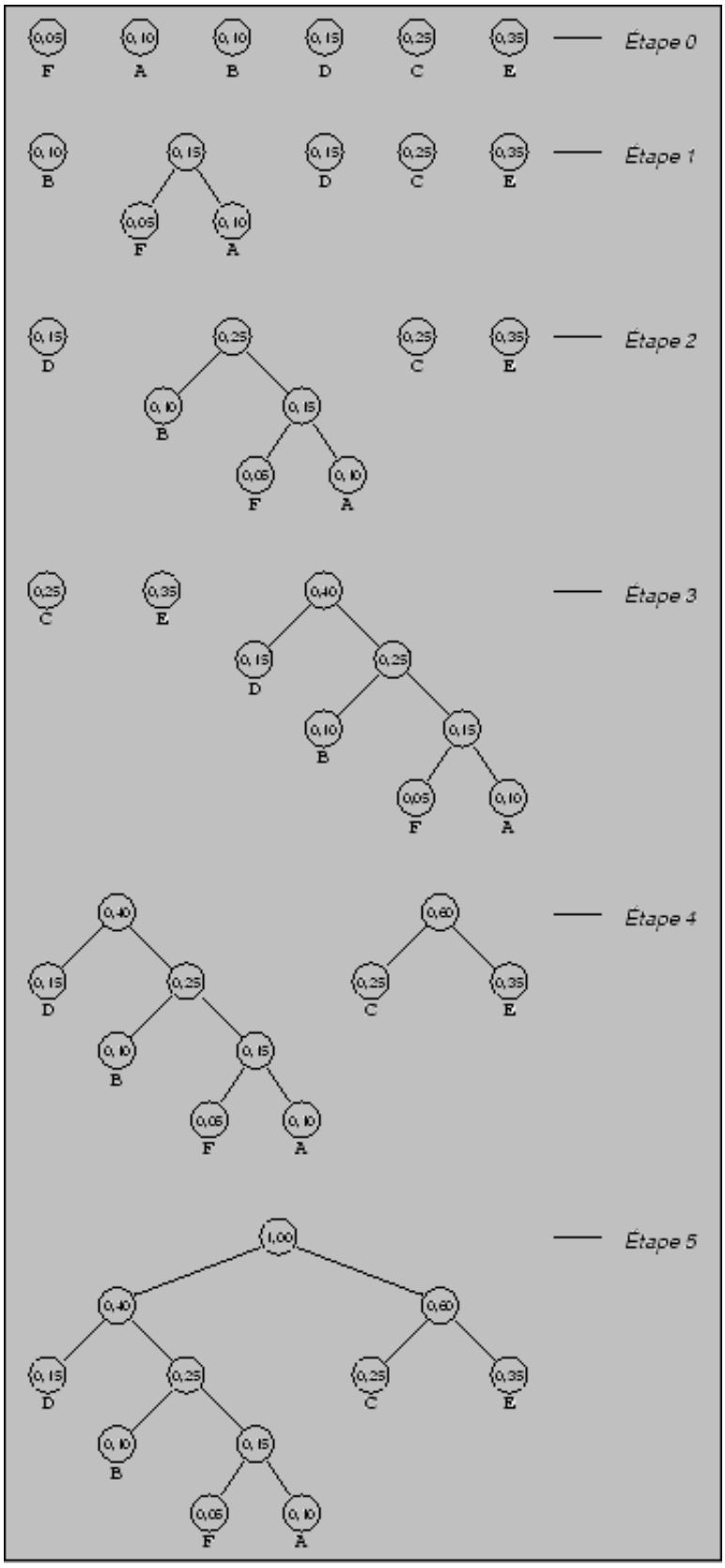
L'algorithme opère sur une forêt. Une forêt est ici un ensemble d'arbres étiquetés complets : tout noeud interne (c'est-à-dire qui n'est pas une feuille) a deux fils non-vides.

La forêt initiale est formée d'un arbre à un noeud pour chaque lettre du langage-source, dont l'étiquette est la probabilité de cette lettre.

La forêt finale est formée d'un unique arbre, qui est l'arbre de décodage du code.

L'algorithme est de type glouton : il choisit à chaque étape les deux arbres d'étiquettes minimales, soit x et y, et les remplace par l'arbre formé de x et y et ayant comme étiquette la somme de l'étiquette de x et de l'étiquette de y.

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.


0,05
F
0, 10
A
0, 10
B
0, 15
D
0,25
C
0,35
E
Étape 0
0, 10
B
0, 15
0,05
F
0, 10
A
0, 15
D
0,25
C
0,35
E
Étape 1
0, 15
D
0,25
0, 10
B
0, 15
0,05
F
0, 10
A
0,25
C
0,35
E
Étape 2
0,25
C
0,35
E
0,40
0, 15
D
0,25
0, 10
B
0, 15
0,05
F
0, 10
A
Étape 3
0,40
0, 15
D
0,25
0, 10
B
0, 15
0,05
F
0, 10
A
0,60
0,25
C
0,35
E
Étape 4
1,00
0,40
0, 15
D
0,25
0, 10
B
0, 15
0,05
F
0, 10
A
0,60
0,25
C
0,35
E
Étape 5

Le code d'une lettre est alors déterminé en suivant le chemin depuis la racine de l'arbre jusqu'à la feuille associée à cette lettre en concaténant successivement un 0 ou un 1 selon que la branche suivie est à gauche ou à droite.

EXEMPLE 12. Ainsi, sur la figure ci-contre, A=0111, B=010, C=10, D=00, E=11 et F=0110.

EXEMPLE 13. 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.

3.5 Conclusion

Ce principe de compression est aussi utilisé dans le codage d'image TIFF (Tagged Image Format File) spécifié par Microsoft Corporation et Aldus Corporation.

Par ailleurs, le codage d'image est fait en retranscrivant exactement le contenu d'un écran (image), en utilisant les méthodes traditionnelles de compression.

REMARQUE 8. Il existe des méthodes qui ne conservent pas exactement le contenu d'une image (méthodes non conservatives) mais dont la représentation visuelle reste correcte. Entre autres, il y a la méthode JPEG (Join Photographic Experts Group) qui utilise la compression de type Huffman pour coder les informations d'une image.

Malgré son ancienneté, cette méthode est toujours remise au goût du jour, et offre des performances appréciables. En effet, beaucoup de recherches en algorithmique ont permis d'améliorer les fonctionnalités de la méthode Huffman de base, par exemple les arbres binaires, les arbres équilibrés, etc.

3.6 Exercices

Exercice 13. *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 14. *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 %*

V. Parcours en largeur d'un graphe

1 Présentation

Le but est de visiter l'ensemble des sommets d'un graphe donné, sans en oublier aucun.

Le parcours en largeur est l'un des algorithmes de parcours les plus simples sur les graphes.

REMARQUE 9. Cet algorithme de parcours fonctionne encore dans le cas orienté.

2 Idée de l'algorithme

Soit s un sommet origine.

1. Parcourir tous les arcs du graphe accessibles depuis s.
2. Calculer la distance entre s et tous les sommets accessibles (cette distance correspond au plus petit nombre d'arêtes).
3. Construction d'une arborescence en largeur, de racine s, qui contient tous les sommets accessibles dans le graphe.

Fin du Chapitre

Chapitre 25

Problèmes de coloration

I. Coloration des sommets

1 Notion de stabilité

DÉFINITION 1 (STABLE). *Soit G = (V, E) un graphe. Un sous-ensemble S de V est un* stable *s'il ne comprend que des sommets non adjacents deux à deux.* ♢

DÉFINITION 2 (NOMBRE DE STABILITÉ). *Le cardinal du plus grand stable est le* nombre de stabilité *de* G ♢

NOTATION : On le note $a(G)$.

2 La coloration

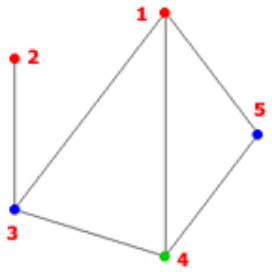
DÉFINITION 3 (COLORATION DES SOMMETS D'UN GRAPHE). *La* coloration des sommets *d'un graphe consiste à affecter à tous les sommets de ce graphe une couleur de telle sorte que deux sommets adjacents ne portent pas la même couleur.* ♢

REMARQUE 1. Une coloration avec k couleurs est donc une partition de l'ensemble des sommets en k stables.

DÉFINITION 4 (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.* ♢

NOTATION : Ce nombre chromatique est noté $g(G)$.

EXEMPLE 1. 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. On a donc trois stables : $\{1, 2\}$, $\{3, 5\}$ et $\{4\}$. On ne peut pas utiliser moins de couleurs, à cause des cliques 1-4-5 et 1-3-4.



REMARQUE 2. Remarquons enfin que le sommet 2 aurait pu aussi être vert. La coloration minimale n'est donc pas forcément unique.

3 Encadrement du nombre chromatique

3.1 Majoration

On a les deux majorations suivantes...

PROPRIÉTÉ 1 : $g(G) \leqslant r + 1$, où r est le plus grand degré de ses sommets.

PREUVE 6 :
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 II : $g(G) \leqslant n + 1 - a(G)$

PREUVE 7 :
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 $g(G) \leqslant 1 + (n - a(G))$. ■

3.2 Minoration

On a d'autres résultats, concernant la minoration...

PROPRIÉTÉ III : Le nombre chromatique d'un graphe est supérieur ou égal à celui de chacun de ses sous-graphes.

PREUVE 8 :
Ce résultat découle de la définition même du nombre chromatique. ■

PROPRIÉTÉ IV : $g(G) \geqslant w(G)$

PREUVE 9 :
Puisque, par définition, dans une clique d'ordre m, tous les sommets sont adjacents entre eux, il faudra m couleurs. Donc, forcément, le nombre chromatique du graphe sera supérieur ou égal à l'ordre de sa plus grande clique. ■

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, et attribuer à chacun des sommets son numéro d'ordre dans la liste obtenue.

Étape 2 En parcourant la liste dans l'ordre, attribuer une couleur non encore utilisée au premier sommet non encore coloré, et attribuer cette même couleur à chaque sommet non encore coloré et non adjacent à un sommet de cette couleur.

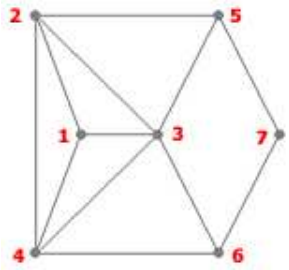
Étape 3 S'il reste des sommets non colorés dans le graphe, revenir à l'étape 2. Sinon, la coloration est terminée.

5 Exercices

Exercice 1. *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 ?*

Exercice 2. *Déterminez le nombre chromatique de ce graphe :*



Exercice 3. *Sept élèves, désignés par A, B, C, D, E, F et G, se sont rendus à la bibliothèque aujourd'hui. Le tableau suivant précise «qui a rencontré qui» (la bibliothèque étant petite, deux élèves présents au même moment se rencontrent nécessairement...).*

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*

De combien de places assises doit disposer la bibliothèque pour que chacun ait pu travailler correctement au cours de cette journée ?

Exercice 4. *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 5. *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 6. *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 ?*

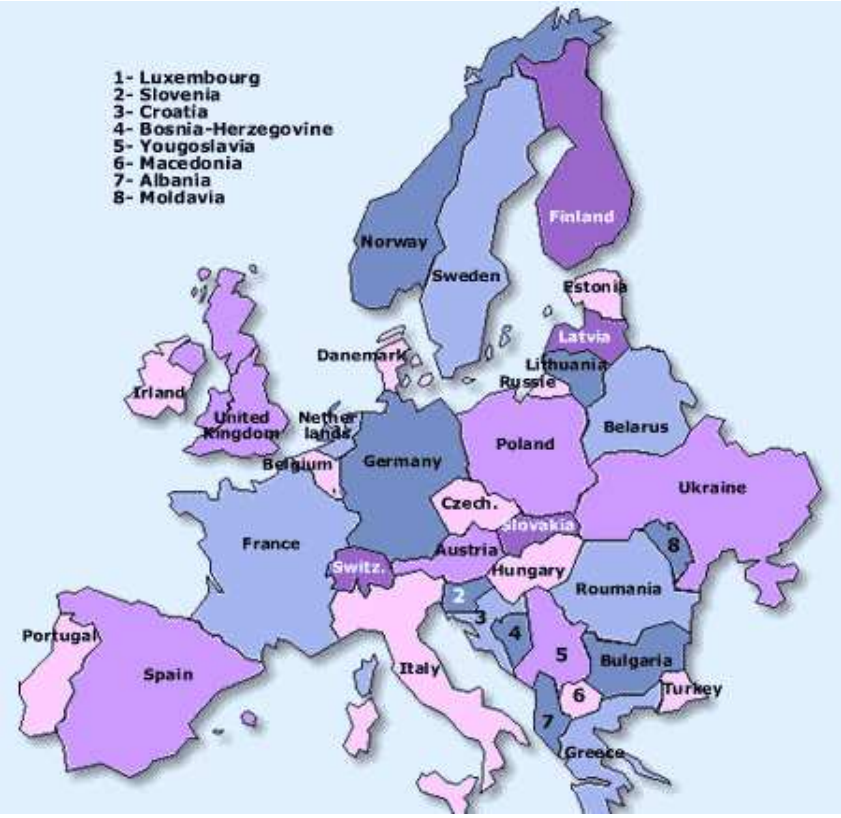
Exercice 7. *Utilisez l'algorithme de coloration de Welsh et Powell pour colorer les graphes des exercices précédents.*

Exercice 8. *Exprimez la résolution d'un Sudoku classique en termes de coloration de graphe. Décrivez le graphe (nombre de sommets, nombre d'arêtes, etc.). Combien faut-il de couleurs ?*

II. Coloration des sommets d'un graphe planaire

1 Présentation

Une question datant de la fin XIX^{e}, reliée aux graphes planaires (voir plus loin), est devenue célèbre sous le nom de problème des quatre couleurs : suffit-il de quatre couleurs pour dessiner n'importe quelle carte géographique ?



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 V (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.

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.

3 Exercice

Exercice 9. *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.*

III. Coloration des arêtes

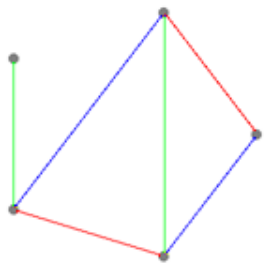
1 Présentation du problème

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 5 (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 2. 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.



---

2 Lien avec la coloration des sommets

2.1 Présentation

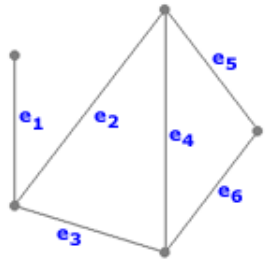
Pour colorer les arêtes d'un graphe, on peut se ramener au problème de la coloration des sommets. Il suffit pour cela de travailler non pas sur le graphe lui-même, mais sur le graphe adjoint, noté G', et que l'on définit ainsi :

- 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.

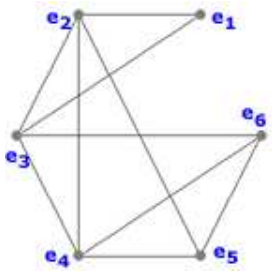
On peut ensuite appliquer par exemple l'algorithme de Welsh et Powell sur le graphe G' pour colorer ses sommets. Une fois cela fait, on colorera les arêtes de G de la même couleur que les sommets correspondants de G'.

2.2 Exemple de coloration d'arêtes

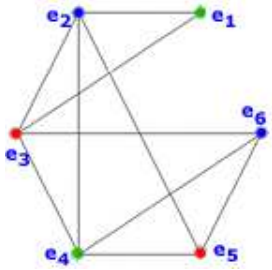
1. Un graphe G :



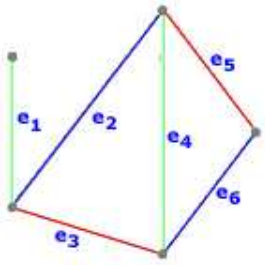
2. Son graphe adjoint G'



3. Coloration des sommets de G'



4. Coloration des arêtes de G



3 Exercice

Exercice 10. *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 26

Graphes orientés

I. Définitions

1 Digraphe (graphe orienté), sommet, arc

En donnant un sens aux arêtes d'un graphe, on obtient un digraphe, ou graphe orienté.

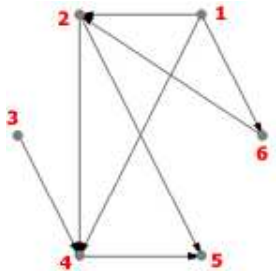
REMARQUE 1. De l'anglais directed graph.

DÉFINITION 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 1. Un exemple de digraphe :

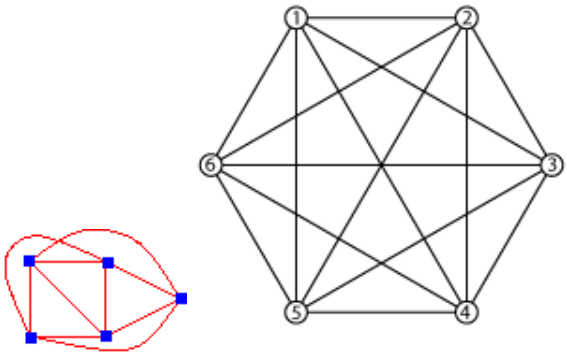


Exercice 1. *Construire un graphe orienté dont les sommets sont les entiers compris entre 1 et 12 et dont les arcs représentent la relation « être diviseur de ».*

Exercice 2. *Quel est le nombre maximal d'arêtes dans un graphe orienté d'ordre n qui ne possède pas d'arêtes parallèles ?*

2 Exemples

1. le graphe d'une relation binaire peut être assimilé à un graphe orienté,
2. les graphes orientés peuvent être représentés graphiquement, ce qui fournit toutes sortes d'exemples :



3 Degré d'un sommet d'un digraphe

Soit v un sommet d'un graphe orienté.

DÉFINITION 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 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 1 : On a :

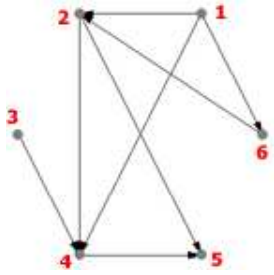
$$d(v) = d_+(v) + d_-(v)$$

.

Exercice 3. *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 4. *Trouvez les degrés extérieurs et intérieurs de chacun des sommets du graphe ci-dessous :*



Exercice 5. *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 ?*

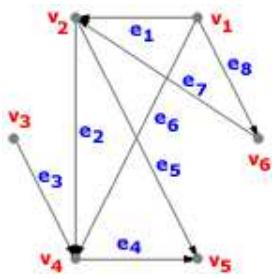
---

4 Chemins et circuits

4.1 Chemin

DÉFINITION 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 2. Sur le digraphe ci-dessous, on peut voir par exemple le chemin $(v_3, e_3, v_4, e_4, v_5)$.



REMARQUE 2. Par convention, tout chemin comporte au moins un arc.

4.2 Distance

DÉFINITION 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 3. 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 6. *Donnez un algorithme permettant de calculer la distance entre deux sommets x et y d'un digraphe connexe.*

4.3 Circuit

DÉFINITION 6 (CIRCUIT). *Un* circuit *est un chemin avec* $u_0 = u_k$. ♢

EXEMPLE 4. Le digraphe ci-dessus ne contient pas de circuit.

REMARQUE 3. Les notions de longueur, de chemins et de circuits sont analogues à celles des chaînes et des cycles pour le cas non orienté.

5 Circuits eulériens

PROPRIÉTÉ II : 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.

II. Digraphe fortement connexe

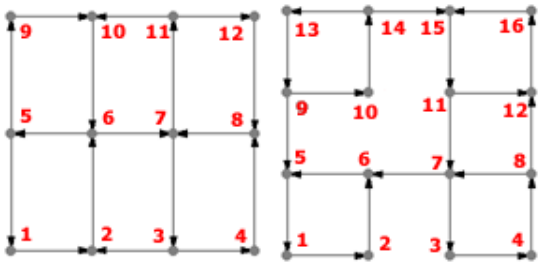
1 Définitions

1.1 Connexité forte

DÉFINITION 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 4. En d'autres termes, tout sommet est atteignable depuis tous les autres sommets par au moins un chemin.

Exercice 7. *Les graphes ci-dessous sont-il fortement connexes ? Si non, donnez leurs composantes fortement connexes.*



1.2 Composantes connexes

DÉFINITION 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).* ♢

2 Exercices

Exercice 8. *On souhaite prélever 4 litres de liquide dans un tonneau. Pour cela, nous avons à notre disposition deux récipients (non gradués !), l'un de 5 litres, l'autre de 3 litres.*

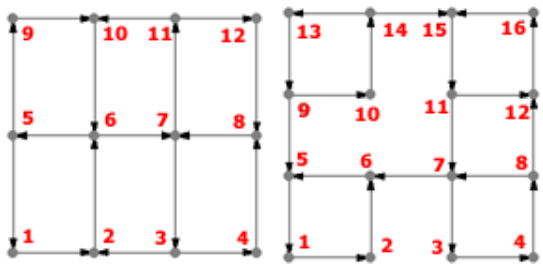
Comment doit-on procéder ?

Exercice 9. *Donnez un algorithme permettant de calculer la distance entre deux sommets x et y d'un digraphe connexe.*

Exercice 10. *Proposez un algorithme qui détermine si un graphe est fortement connexe ou non.*

Indication : utilisez un système de marquage des sommets.

Exercice 11. *Les graphes ci-dessous sont-il fortement connexes ? Si non, donnez leurs composantes fortement connexes.*



III. Matrice et listes d'adjacences

1 Matrice d'incidence

DÉFINITION 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* d *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 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.* ♢

1.1 Remarques

1. se donner un graphe orienté est équivalent à se donner les deux matrices J^+ et J^-.
2. on peut relier $d^+(s)$ (resp. $d^-(s)$) au nombre de 1 apparaissant dans J^+ (resp. J^-).

1.2 Résultats

PROPRIÉTÉ III : 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. $J^-J^{+t} = \begin{pmatrix} d^-(s_1) & & 0 \\ & \ddots & \\ 0 & & d^-(s_n) \end{pmatrix}$

2 Matrice d'adjacences

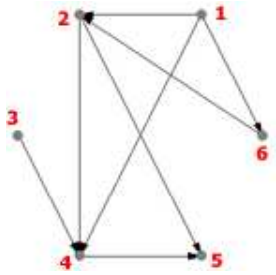
2.1 Définition

DÉFINITION 11 (MATRICE D'ADJACENCES). *On peut représenter un digraphe par une* matrice d'adjacences *:*

- *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.* ♢

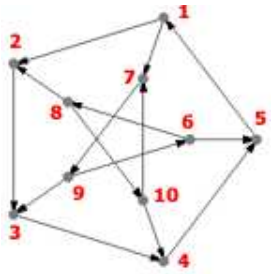
2.2 Exemple

Voici la matrice d'adjacences du digraphe G :



$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Exercice 12. *Décrivez le digraphe G ci-contre par une matrice d'adjacences.*



REMARQUE 5. Se donner un graphe orienté revient à se donner sa matrice d'adjacence.

2.3 Propriétés

PROPRIÉTÉ IV : 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 V (NOMBRE DE CHEMINS 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 de chemins de longueur k qui mènent de s à t.

PROPRIÉTÉ VI (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

$$\boxed{B = A + A^t}$$

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 13. *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 14. *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 VII : Soient $s_1, s_2, \ldots, s_n$ les sommets d'un graphe orienté. Alors

$$\begin{pmatrix} d(s_1) \\ \vdots \\ d(s_n) \end{pmatrix} = J \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

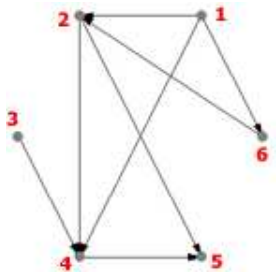
PROPRIÉTÉ VIII (RELATION ENTRE J, J^+ ET J^-) : On note J^+ et J^- les matrices d'incidences d'un graphe orienté, et J la matrice d'incidence du graphe non orienté qui lui est associé. Alors

$$\boxed{J = J^+ + J^-}$$

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 5. Voici les listes d'adjacences du digraphe G :



1 : 2, 4, 6
2 : 4, 5
3 : 4
4 : 5
5 : -
6 : 2

PROPRIÉTÉ IX : 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 15. *Décrivez le digraphe G de l'exercice précédent par des listes d'adjacences.*

IV. Digraphes sans circuits

1 Théorème

PROPRIÉTÉ X : Le digraphe $G = (V, E)$ est sans circuits 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 10 :
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. ■

2 Algorithme de calcul du rang

Donnée : digraphe $G = (V, E)$ sans circuit.

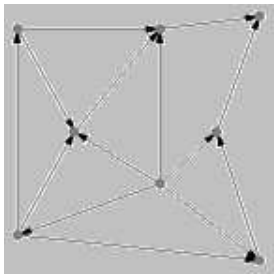
Résultat : rang $r(v)$ de chaque sommet $v \in V$ du digraphe G.

Début – $r := 0$, $X := V$
- R : l'ensemble des sommets de X sans prédécesseurs dans X
- Tant que X n'est pas vide faire
  - $r(v) := r$ pour tout sommet $v \in R$
  - $X := X - R$
  - R : l'ensemble des sommets de X sans prédécesseurs dans X
  - $r := r + 1$

Fin.

3 Exercice

Exercice 16. *Attribuez un rang aux sommets du digraphe ci-dessous en utilisant l'algorithme de calcul du rang :*



Fin du Chapitre

Chapitre 27

Problèmes de chemin

I. Algorithme de Dijkstra

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.

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.

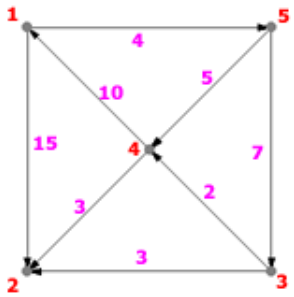
3 Description de l'algorithme de Dijkstra

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
  - $l(j) = l(i) + d(i, j)$
  - $p(j) = i$

4 Exemple



Initialisations – $S = \{1\}$;
- $T = \{2, 3, 4, 5\}$;
- $l = (0, 15, \infty, \infty, 4)$;
- $p = (NIL, 1, NIL, NIL, 1)$.

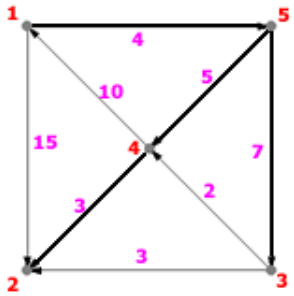
Première itération – $i = 5$ car $l(5) = min(15, \infty, \infty, 4) = 4$;
- $S = \{1, 5\}$; $T = \{2, 3, 4\}$;
- les successeurs de 5 dans T sont 3 et 4 ;
- $l(3)$ prend la nouvelle valeur $min(\infty; l(5) + d(5; 3)) = min(\infty; 4 + 7) = 11$; $p(3) = 5$;
- $l(4)$ prend la nouvelle valeur $min(\infty; l(5) + d(5; 4)) = 9$; $p(4) = 5$;
- 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 $p(5) = 1$.



5 Exercices

Exercice 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 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

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.

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.

3 Définitions

DÉFINITION 1 (SOMMET CRITIQUE). *Un sommet i est* critique *si $d_i = j_i$.* ♢

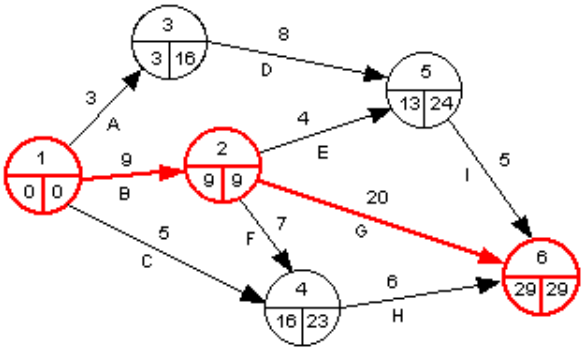
DÉFINITION 2 (ARC CRITIQUE). *Un arc (i, j) est* critique *si ses extrémités sont des sommets critiques et $d_{ij} = d_j - d_i$.* ♢

DÉFINITION 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 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.* ♢

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

5 Exercices

Exercice 3. *Refaites le graphe des précédences de l'exemple en utilisant l'algorithme du chemin critique.*

Exercice 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 28

Chaînes de Markov

I. Généralités

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.

2 Définitions

DÉFINITION 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 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.

3 Exemple

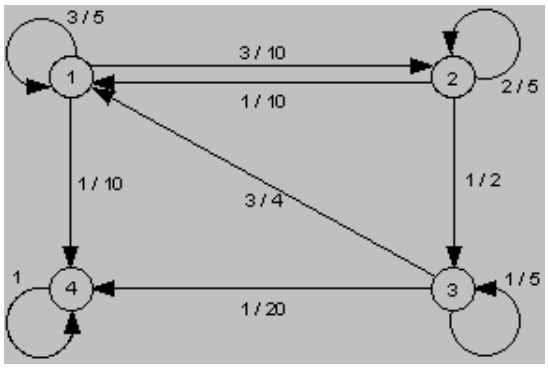
Pour représenter le passage d'une molécule de phosphore dans un écosystème, nous considérerons quatre états possibles :

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 :



4 Propriétés

PROPRIÉTÉ I : 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 II : $p(t) = p(0)P^t$

5 Exercice

Exercice 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

1 Présentation

On constate souvent que la distribution $p(t)$ converge vers une distribution limite p si $t \to \infty$.

DÉFINITION 3. *Si tel est le cas, on dit que cette dernière définit un* régime permanent *du processus stochastique.* ♢

REMARQUE 1. Le régime permanent n'est pas influencé par le choix de la distribution initiale.

2 Existence d'une distribution limite

PROPRIÉTÉ III : 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 2. La démonstration de cette condition d'existence dépasse le cadre de ce cours.

3 Exercices

Exercice 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 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 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 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

1 Généralités

1.1 Définitions

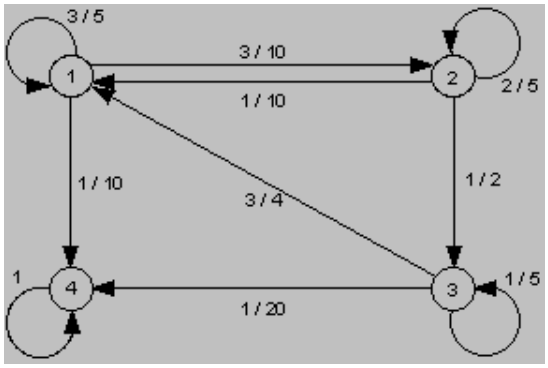
DÉFINITION 4 (ÉTAT ABSORBANT). *Un* état absorbant *est un état que l'on ne quitte plus lorsqu'on y pénètre.* ♢

REMARQUE 3. Autrement dit, l'état j est absorbant si $p_{jj} = 1$.

DÉFINITION 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 1. 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.

1.2 Propriété

PROPRIÉTÉ IV : 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.

2 Délais d'absorption et probabilité d'absorption

2.1 Présentation

Lorsque l'on a affaire à une chaîne de Markov absorbante, on est généralement intéressé par les deux questions suivantes :

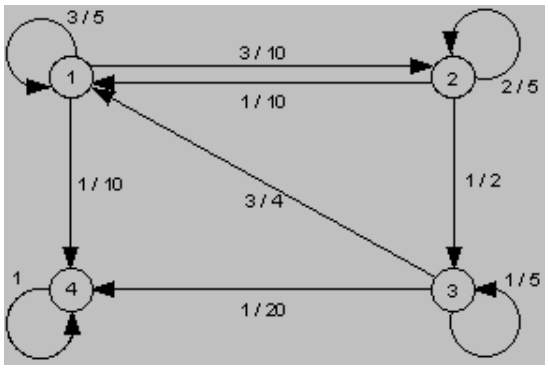
– Combien de temps faudra-t-il en moyenne pour arriver dans un état absorbant, étant donné son état initial ?
– S'il existe plusieurs états absorbants, quelle est la probabilité de tomber dans un état absorbant donné ?

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) :

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{10} & \frac{3}{5} & \frac{3}{10} & 0 \\ 0 & \frac{1}{10} & \frac{2}{5} & \frac{1}{2} \\ \frac{1}{20} & \frac{3}{4} & 0 & \frac{1}{5} \end{pmatrix}$$

2.3 Matrice fondamentale

DÉFINITION 6 (MATRICE FONDAMENTALE DE LA CHAÎNE ABSORBANTE). *La matrice* $N = (I - Q)^{-1}$ *est appelée la* matrice fondamentale de la chaîne absorbante. $\diamondsuit$

2.4 Résultats

PROPRIÉTÉ V : 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 VI : 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 2. 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 VII : 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 3. 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 !

---

3 Exercices

Exercice 6. *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 7. *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 8. *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 9 (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 29

Annales

I. Partiel du 22 octobre 2007 (S1)

Dire, pour chacune des assertions suivantes, si elle est vraie ou fausse (+1 pour une bonne réponse, -1 pour une mauvaise).

QUESTION 1 : Soit $A = \{\{a, b, \}, \{c\}, \{d, e, f\}\}$. Dire si les affirmations suivantes sont justes ou fausses.

1. $a \in A$
2. $\{c\} \subset A$
3. $\{d, e, f\} \in A$
4. $\{\{a, b\}\} \subset A$
5. $\emptyset \in A$
6. $\emptyset \subset A$

QUESTION 2 : On reconsidère l'ensemble A de la question précédente. Dire si les affirmations suivantes sont justes ou fausses.

1. $P(A)$ *contient 3 éléments*
2. $P(A)$ *contient 4 éléments*
3. $P(A)$ *contient 8 éléments*
4. $P(A)$ *contient 64 éléments*

QUESTION 3 : Soit E un ensemble. On note $P(E)$ l'ensemble de ses parties. Pour tout A et B, parties de E, on définit un opérateur δ par

$$\delta(A, B) = (A \cup B) \cap (E \setminus (A \cap B)).$$

On suppose que l'on sait montrer que pour C et D, parties de E, on a

$$C \cap (E \setminus D) = C \setminus D$$

On pose $A = \{a, b, c, d, e, f\}$, $B = \{a, c, e, g\}$, $C = \{b, d, f\}$ et

$$E = A \cup B \cup C \cup \{h, i\}$$

. Dire si les affirmations suivantes sont justes ou fausses.

1. *Il y a quatre éléments dans* $\delta(A, B)$
2. $\delta(a, c) \neq \emptyset$
3. $\delta(b, c)$ *contient sept éléments.*

QUESTION 4 : Dire si les affirmations suivantes sont justes ou fausses.

1. *Pour tout ensemble* A *et* B *inclus dans* E, $\delta(A, B) \in E$.
2. $\delta : E \times E \to E$.
3. *Si* $A \subset P(E)$ *et si* $B \subset P(E)$, $\delta(A, B)$ *a du sens.*

QUESTION 5 : Soit A, A' et B trois parties de E non vides telles que $A' \subset A$. Dire si les affirmations suivantes sont justes ou fausses.

1. $\delta(A', B)$ *peut être égale à l'ensemble vide même si* $\delta(A, B) \neq \emptyset$.
2. $\delta(A', B) \supset \delta(A, B)$.
3. $\delta(A', B) = \delta(A, B)$.

QUESTION 6 : Soit A, B et C trois parties de E non vides. Dire si les affirmations suivantes sont justes ou fausses.

1. $\delta(A, B) = \delta(A, C)$ *si* $B = C$.
2. $\delta(A, B) = \delta(A, C)$ *donc* $B = C$.

3. $\delta(A, B) = \delta(A, C)$ *ssi* $B = C$.

QUESTION 7 : Soit t_1 et t_2 deux termes exprimés dans une algèbre de Boole munie des opérateurs classiques $+$, ., et -.

Si $t_1.t_2 = 0$ alors...

1. $t_1 = 0$ *et* $t_2 = 0$.
2. $t_1 = 0$ *ou* $t_2 = 0$.
3. $t_1 = \overline{t_2}$.
4. *Il existe un terme* k *tel que* $t_1 = k.\overline{t_2}$.

QUESTION 8 : On reconsidère la question précédente, sauf qu'au lieu de $t_1.t_2 = 0$, on suppose que $t_1 + t_2 = 1$. Les affirmations suivantes sont-elles exactes ?

1. $t_1 = 1$ *et* t_2 *vaut autre chose et vice versa.*
2. *Il existe un terme* k *tel que* $t_1 = k + \overline{t_2}$
3. t_1 *ou* t_2 *vaut 1*
4. $t_1 = \overline{t_2}$

QUESTION 9 : Si a et b sont des constantes et x est une variable. L'équation booléenne $ab + abx = ax + abx$,

1. *a une unique solution* b.
2. *est vérifiée par* $b, ab, \overline{a} + b$.
3. *admet n'importe quelle valeur de* x *pour solution.*

QUESTION 10 : On donne une expression E dans une algèbre de Boole munie des opérateurs classiques $+$, ., et -.

La version la plus simplifiée de $E = \overline{a}.b + a.c + \overline{b}.c + \overline{c}$ est

1. *1*
2. a
3. *0*

4. *une autre expression*

QUESTION 11 : La version la plus simplifiée de $E = \overline{a}(a+b)(a+c)(a+d)(a+e)$ est

1. *1*
2. $\overline{a}bcde$
3. *0*
4. *une autre expression*

QUESTION 12 : La version la plus simplifiée de $E = (\overline{\overline{a} + \overline{b}} + c)(d + ab)$ est

1. *1*
2. *0*
3. $a + b + c + d$
4. *une autre expression*

QUESTION 13 : La version la plus simplifiée de $E = a.(\overline{b} + c).(\overline{a.b} + ac) + \overline{a.(\overline{b} + c)}.\overline{\overline{a.b} + a.c}$ est

1. *1*
2. *0*
3. $a + c$
4. *une autre expression*

QUESTION 14 : Soit n variables booléennes, k le nombre de mintermes et l le nombre de maxtermes. Quelle relation existe entre k et l ?

1. $k + l = 2^n - 1$
2. $k = l = 2^n$
3. $k = l = n$

QUESTION 15 : Soit $m_i^{(n)}$ le i^{eme} minterme à n variables. L'expression $m_0^{(2)} + m_1^{(2)} + m_2^{(2)} + m_3^{(2)}$ est égale à

1. *0*
2. *1*
3. $m_6^{(2)}$
4. $m_0^{(2)}$

QUESTION 16 : Quel est l'ensemble des (i, j) tels que $m_i^{(7)}.m_j^{(7)} \neq 0$

1. $\emptyset$
2. $\mathbb{N} \times \mathbb{N}$
3. $\{(i, i) \mid i \in \mathbb{N}$ *et* $i \leq 6\}$

QUESTION 17 : Soit $E(x, y, z)$ une fonction booléenne à trois variables x, y, z.
Si $E(x, y, z) = x\overline{y}.y$, la forme canonique disjonctive est

1. $xy + x\overline{z}$
2. $xyz + xy\overline{z} + x\overline{yz}$
3. *une somme de 4 mintermes*
4. *autre chose*

QUESTION 18 : Si $E(x, y, z) = z(\overline{x} + y) + \overline{y}$, la forme canonique disjonctive est

1. $\overline{x}z + yz + \overline{y}$
2. *une somme de 8 mintermes*
3. *une somme de 6 mintermes*
4. *autre chose*

QUESTION 19 : Si $E(x, y, z) = \overline{\overline{x} + y} + \overline{x}.y$, la forme canonique disjonctive est

1. *une somme de 2 mintermes*

2. *une somme de 4 mintermes*
3. *autre chose*

QUESTION 20 : Si $E(x, y, z) = x + yz$, la forme canonique **conjonctive** est

1. $xyz + xy\overline{z} + x\overline{y}z + x\overline{yz} + \overline{x}yz$
2. *elle même : c'est déjà en FCC*
3. *un produit de quatre maxtermes*
4. *un produit de trois maxtermes*

QUESTION 21 : Soit $E(x, y)$ telle que la FCD de sa négation est $xy + x\overline{y} + \overline{xy}$. La FCC de $E(x, y)$ est alors

1. *impossible à calculer*
2. $x + \overline{y}$
3. *un produit de quatre maxtermes*
4. *autre chose*

II. Partiel du 22 octobre 2007 (S3)

Dire, pour chacune des assertions suivantes, si elle est vraie ou fausse (+1 pour une bonne réponse, -1 pour une mauvaise).

QUESTION 22 : Soit $I = \{a, b\}$ un alphabet. Combien y a-t-il d'automates de Moore à 1 état ?

1. *0.*
2. *1.*
3. *2.*
4. *Une infinité.*

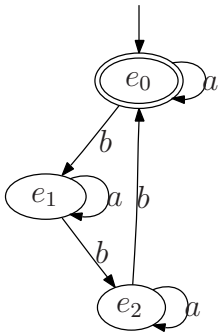


FIG. 29.1 – Automate de la question 25

QUESTION 23 : Avec le même alphabet $I = \{a, b\}$ qu'à la première question, combien y a-t-il d'automates de Moore à 2 états accessibles ?

1. *6.*
2. *9.*
3. *36.*
4. *Une autre quantité.*

QUESTION 24 : Avec le même alphabet $I = \{a, b\}$ qu'à la première question, la liste des mots de longueur 5 contient

1. *10 éléments.*
2. *25 éléments.*
3. *32 éléments.*
4. *Un autre nombre d'éléments.*

QUESTION 25 : Soit l'automate donné à la figure 29.1. Quelle affirmation est fausse ?

1. *L'automate n'accepte que les mots dont le nombre de b est un multiple de 3.*
2. *Le mot vide est reconnu par l'automate.*
3. *$bbbaaa$ est reconnu par l'automate.*
4. *L'automate n'accepte que les mots dont la taille est un multiple de 3.*

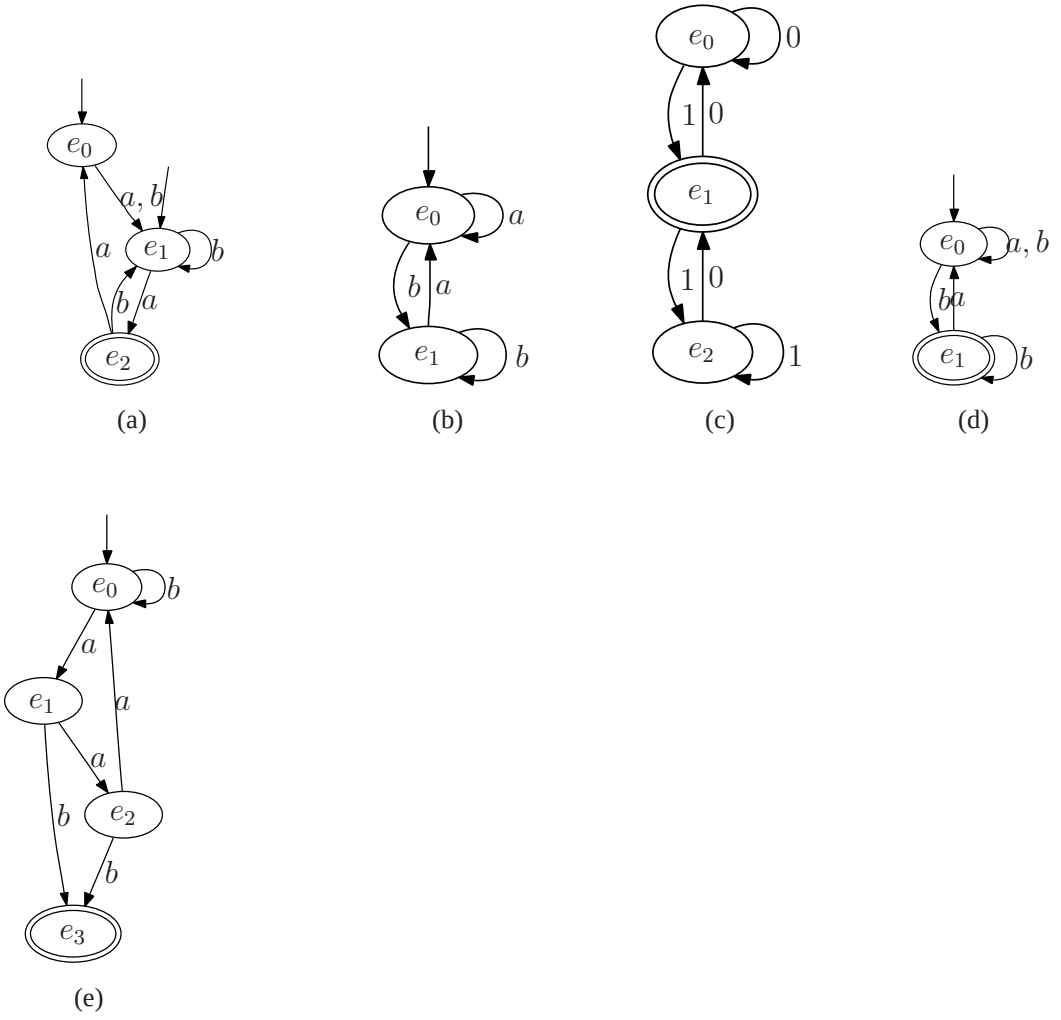


FIG. 29.2 – Automates de Moore ?

QUESTION 26 : Soit l'automate donné à la figure 29.1. Il reconnaît l'expression rationnelle...

1. $(a^*ba^*ba^*b)$.
2. $(a \mid b).(a \mid b).(a \mid b)$.
3. $(a^*ba^*ba^*b)^*$.

QUESTION 27 : La figure 29.2 contient de nombreux automates. Répondre par vrai ou faux à chacune des questions suivantes.

	0	1
$e0$	$\{e_1, e_2\}$	$\{e_2, e_3\}$
$e1$	$\{e_3\}$	$\{e_1\}$
$e2$	$\{e_2\}$	$\{e_3\}$

(a)

	0	1
A	B	B
B	D	C
C	F	E
D	F	G
E	F	E
F	H	H
G	G	F
H	H	H

(b)

	0	1
$\{e_0, e_1\}$	$\{e_1, e_2, e_3\}$	$\{e_1, e_2, e_3\}$
$\{e_1, e_2, e_3\}$	$\{e_2, e_3\}$	$\{e_1, e_3\}$
$\{e_1, e_3\}$	$\{e_3\}$	$\{e_1\}$
$\{e_2, e_3\}$	$\{e_3\}$	$\{e_2\}$
$\{e_1\}$	$\{e_3\}$	$\{e_1\}$
$\{e_3\}$	F	F
$\{e_2\}$	$\{e_2\}$	$\{e_3\}$
F	F	F

(c)

	0	1
$\{e_0, e_1\}$	$\{e_1, e_2, e_3\}$	$\{e_1, e_2, e_3\}$
$\{e_1, e_2, e_3\}$	$\{e_2, e_3\}$	$\{e_1, e_3\}$
$\{e_1, e_3\}$	$\{e_1, e_3\}$	$\{e_3\}$
$\{e_2, e_3\}$	$\{e_3\}$	$\{e_2, e_3\}$
$\{e_3\}$	$\{e_3\}$	$\{e_3\}$

(d)

	0	1
A	B	B
B	C	C
C	D	D
D	D	D

(e)

FIG. 29.3 – Tables de transitions

1. *La figure 29.2(a) est un automate de Moore.*
2. *La figure 29.2(b) est un automate déterministe.*
3. *La figure 29.2(c) est un automate de Moore.*
4. *La figure 29.2(d) est un automate de Moore.*
5. *La figure 29.2(e) est un automate de Moore.*

QUESTION 28 : Les figures 29.3 définissent des relations de transitions, et l'on considère les automates associés ; ils possèdent de plus deux états initiaux e_0 et e_1 et un état final e_2 (pour (a), (c) et (d)), un état initial A et un état final B pour (b) et (e).

1. (a) *correspond à un automate de Moore.*
2. (b) *correspond à un automate déterministe.*

3. (c) *et* (e) *reconnaissent le même langage.*
4. (d) *est un automate non déterministe.*
5. (e) *est un automate de Moore.*

QUESTION 29 : On considère un automate à 6 états : e_0, ..., e_5. Son état initial est e_0 et son état final est e_4. Sa table de transition 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 (réflexive, symétrique et transitive) engendrée par $e_0\mathcal{R}e_2$, $e_1\mathcal{R}e_3$ et $e_3\mathcal{R}e_5$. Quel est le nombre d'états de l'automate quotient ?

1. *2.*
2. *3.*
3. *4.*
4. *5.*

QUESTION 30 : Avec la même table de transition et la même relation qu'à la question précédente, répondre par vrai ou faux : $\mathcal{R}$ n'est pas une congruence d'automate.

QUESTION 31 : Lorsqu'un AFND A est défini par une relation de transition t non déterministe, la méthode du dual consiste, dans le pire des cas

1. *en trois déterminisations et deux inversions.*
2. *deux déterminisations et deux inversions.*
3. *un nombre de quotients qui dépend de* $\mathcal{R}$.
4. *une déterminisation et une inversion.*

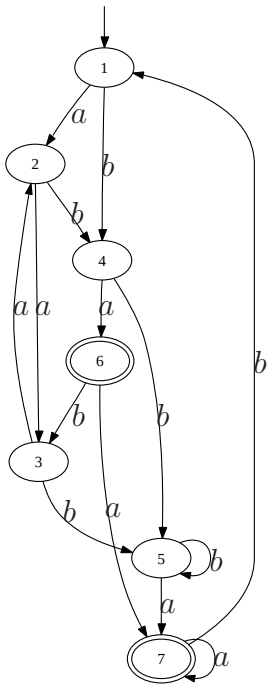


FIG. 29.4 – Automate à trop grand nombre d'états (Question 32)

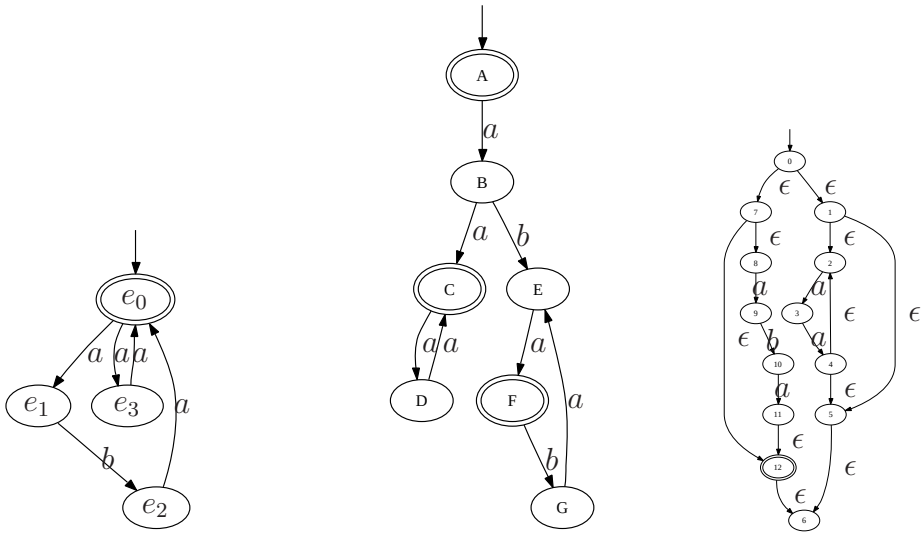


FIG. 29.5 – Automates reconnaissant une expression régulière

QUESTION 32 : On considère l'automate donné à la figure 29.4.

Combien l'automate minimal de Moore reconnaissant le même langage contient-il d'états ?

1. *2.*
2. *4.*
3. *7.*
4. *Un autre nombre.*

QUESTION 33 : Cette question est liée à la figure 29.5. Choisir la bonne réponse. L'expression régulière $((aba) \mid (aa))$

1. *est reconnue par l'automate 29.5(a).*
2. *est reconnue par l'automate 29.5(b).*
3. *est reconnue par l'automate 29.5(c).*
4. *est reconnue par un autre automate*

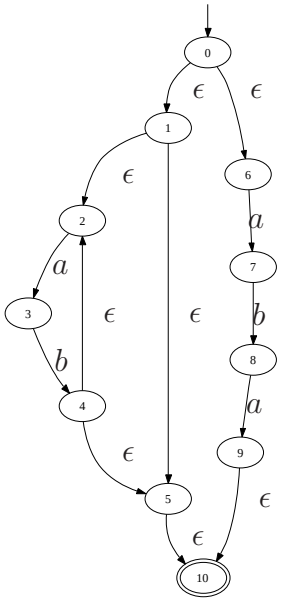


FIG. 29.6 – Automate de Thompson

QUESTION 34 : Cette question est liée à l'automate A donné à la figure 29.6. De combien d'états de A est constitué l'état initial de la version sans epsilon-transition.

1. *1.*
2. *3.*
3. *5.*
4. *6.*

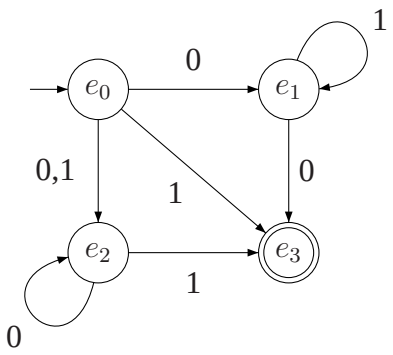
QUESTION 35 : Cette question est aussi liée à l'automate donné à la figure 29.6. Combien d'états a la forme sans epsilon-transition qui correspond à cet automate.

1. *6.*
2. *7.*
3. *10.*
4. *11.*

QUESTION 36 : Cette question est aussi liée à l'automate donné à la figure 29.6. Combien d'états d'acceptation a la forme sans epsilon-transition qui correspond à cet automate.

1. *1.*
2. *3.*
3. *4.*
4. *Un autre nombre.*

QUESTION 37 : On considère l'automate :

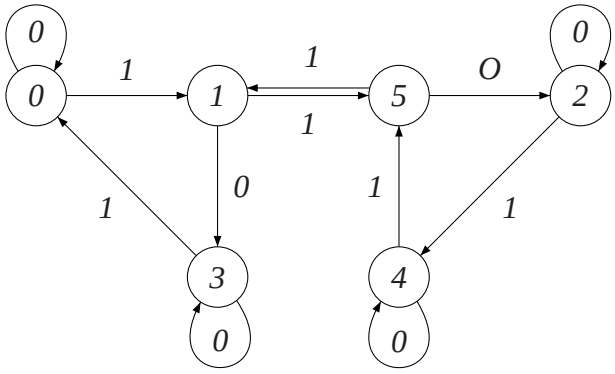


et la relation binaire $e_i \mathcal{R} e_j$ si et seulement si i et j ont la même parité.

1. *$\mathcal{R}$ est une relation d'équivalence sur E.*
2. *$\mathcal{R}$ est une congruence d'automate sur E.*
3. *La table de transition de l'automate quotient est*

E	a	b
$A = \{e_0, e_2\}$	$\{e_0, e_2\}$	$\{e_4\}$
$B = \{e_1, e_3, e_5\}$	$\{e_1, e_3, e_5\}$	$\{e_0, e_2\}$
$C = \{e_4\}$	$\{e_4\}$	$\{e_1, e_3, e_5\}$

4. *Le graphe de l'automate quotient est*


0
1
1
O
0
0
1
5
2
1
0
1
1
1
3
4
0
0

Chapitre 30

Bibliographie

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.

Chapitre 31

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.

Index