Ces notes sont inspirées du support officiel du cours : «Méthodes mathématiques pour l'informatique », de Jacques Vélu, chez Dunod.

Rappels et notations

On suppose connue les notions d'ensembles, de fonctions, de relations, d'entiers naturels. On note $\emptyset$ l'ensemble vide. Les éléments d'un ensemble X sont souvent appelés les *points de* X.

Ensembles d'entiers. On note $\mathbb{N}$ l'ensemble des entiers naturels. Soient p et q deux entiers tels que $q \geq p$. On note $[p,q]_{\mathbb{N}}$ ou plus simplement $[p,q]$ l'ensemble des entiers compris au sens large entre p et q :

$$[p,q] = \{n \in \mathbb{N}, p \leq n \leq q\}.$$

On convient que si $q < p$ alors $[p,q]$ est vide. Cette convention est compatible avec le fait que l'ensemble $[1,n]$ contient exactement n entiers pour tout entier n, en particulier pour $n = 0$.

Fonction vide. On convient que pour tout ensemble X il y a exactement une fonction de l'ensemble vide dans X, appelée la *fonction vide*.

Tuples, couples, produit cartésien. Soit X un ensemble et n un entier. Un *n-uple* d'éléments de X est une suite de n points de X, c'est à dire une fonction de l'ensemble $[1,n]$ dans X. Conformément à l'usage en matière de suites, si x est un n-uple, on note x_i (au lieu de $x(i)$) l'image de l'entier i et on écrit $x = (x_1, \ldots, x_n)$. Un n-uple est donc comme un ensemble à n éléments sauf que les éléments sont *numérotés* et peuvent apparaître plusieurs fois. On note X^n l'ensemble des n-uples d'éléments de X. Un *couple* est un 2-uple.
Dans le cas où $n = 0$ il y a exactement un 0-uple, le *uple vide*, qui est la fonction vide de $[1,0]$ dans X. Par conséquent X^0 est un singleton.
Soient X et Y deux ensembles. Le *produit cartésien* de X et Y noté $X \times Y$ est l'ensemble des couples d'éléments de $X \cup Y$ dont le premier élément appartient à X et le second à Y :

$$X \times Y = \{(x,y) \in (X \cup Y)^2 \text{ tel que } x \in X, y \in Y\}.$$

Dans le cas où $X = Y$ on a $X \times X = X^2$. Plus généralement $X^n = X \times \cdots \times X$.

Ensemble des parties. Soit X un ensemble. Une *partie* P de X est un sous-ensemble de X, c'est à dire un ensemble dont tous les éléments sont éléments de X ; on écrit alors $P \subset X$. On note $\mathcal{P}(X)$ l'ensemble des parties de X :

$$\mathcal{P}(X) = \{P, P \subset X\}.$$

Ensemble des fonctions. Dans ces notes, les termes *fonction* et *application* sont synonymes. Soient X et Y deux ensembles. On note Y^X l'ensemble des fonctions de X dans Y.

Remarque

> On a vu que un n-uple de points de X est une fonction de $[1,n]$ dans X, donc que X^n est l'ensemble des fonctions de $[1,n]$ dans X. Avec la notation que l'on vient d'introduire on a donc $X^n = X^{[1,n]}$.

Cardinal d'un ensemble fini. Soit X est un ensemble fini. On note $|X|$ le *cardinal* de X, c'est à dire le nombre d'éléments de X. Voici un certain nombre de résultats connus sur les cardinaux : soient X et Y deux ensembles finis et n un entier ;

- $\left|Y^X\right| = |Y|^{|X|}$;
- $|X \times Y| = |X| \,.\, |Y|$;

– $|X^n| = |X|^n$;
– $|\mathcal{P}(X)| = 2^{|X|}$.

Ces équations sont vraies, y compris quand X et/ou Y sont vides, grâce à notre convention sur la fonction vide (en adoptant toutefois la convention arithmétique que pour tout entier n on a $n^0 = 1$).

Remarque

La première équation justifie la notation Y^X pour l'ensemble des fonctions de X dans Y.
À cause de la dernière équation on trouve souvent la notation 2^X pour l'ensemble des parties de X. On verra plus loin que cette équation se justifie par le fait qu'une partie de X est complètement déterminée par sa *fonction caractéristique* qui est une fonction de X dans $\{0, 1\}$, un ensemble à deux éléments.

Opérations n-aire. Étant donnée une opération binaire associative, par exemple l'addition, on définit en général une notation pour la version n-aire de l'opération ; par exemple la version n-aire de l'addition est dénotée par le signe $\sum$. Dans le cas où $n = 0$ la convention est toujours de prendre l'élément neutre de l'opération : par exemple si $n = 0$ alors

$$\sum_{i=1}^{n} x_i = 0, \qquad \prod_{i=1}^{n} x_i = 1,$$
$$\bigcup_{i=1}^{n} x_i = \emptyset, \qquad \bigcap_{i=1}^{n} x_i = X.$$

Dans le cas de l'intersection on suppose que celle-ci est définie sur $\mathcal{P}(X)$ où X est un ensemble fixé.

Chapitre 1

Calcul booléen

1.1 Treillis

Ensembles ordonnés, borne supérieure et inférieure, treillis. Soit X un ensemble ordonné et x, y deux éléments de X. On dit que z est *la borne supérieure* de x et y si :

- z majore x et y, c'est à dire $z \geq x$ et $z \geq y$;
- z est le plus petit majorant de x et y, c'est à dire : pour tout u, si $u \geq x$ et $u \geq y$ alors $u \geq z$.

On a de même la définition de la borne inférieure de x et y en remplaçant $\geq$ par $\leq$.

1.1 Définition (Treillis, treillis distributif)
Si toute paire d'éléments de X admet une borne inférieure et une borne supérieure on dit que X est un treillis.
Si de plus la borne supérieure distribue sur la borne inférieure et réciproquement, c'est à dire si l'on a :

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$
$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

alors le treillis est dit distributif.

Exemple.

> L'ensemble $\mathbb{N}$ des entiers naturels ordonné par la relation « divise » (p divise n si il existe q tel que $n = pq$) est un treillis : la borne inférieure est le pgcd et la borne supérieure est le ppcm ; le plus grand élément est 0 et le plus petit est 1. Est-il distributif ?
>
> Si X est une ensemble, l'ensemble des parties de X ordonné par inclusion est un treillis distributif : la borne supérieure est la réunion et la borne inférieure est l'intersection. Le plus grand élément est X lui-même, et le plus petit est l'ensemble vide.
>
> L'ensemble des parties *finies* de X est également un treillis distributif, mais il n'a pas de plus grand élément.
>
> L'ensemble (d'ensembles d'entiers) $\{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2, 3\}\}$ est un treillis pour l'inclusion mais il n'est pas distributif.

1.2 Proposition
Soit X un treillis. On a alors :

- *la borne supérieure (ou inférieure) de x et y est unique ; on la notera désormais $x \vee y$ (ou $x \wedge y$) ;*
- *la borne supérieure et la borne inférieure sont des opérations* commutatives *et* associatives *: $x \vee y = y \vee x$ et $(x \vee y) \vee z = x \vee (y \vee z)$;*

– *si X a un plus grand élément, alors celui-ci est unique ; on le notera $\top$ ou 1 selon le contexte. S'il existe, le plus grand élément est neutre pour la borne inférieure ($x \wedge \top = x$ pour tout x) et absorbant pour la borne supérieure ($x \vee \top = \top$ pour tout x).*

De même si X a un plus petit élément (noté $\bot$ ou 0), celui-ci est neutre pour la borne supérieure et absorbant pour la borne inférieure.

Groupes, anneaux. Un *groupe commutatif* ou *abélien* est un ensemble G muni d'une opération $+$ commutative, associative, admettant un élément neutre dans G noté 0 et telle que tout élément x de G a un opposé $-x$. L'élément neutre s'il existe est unique, de même pour l'opposé.
Un *anneau commutatif* est un ensemble A muni d'une opération d'addition $+$ telle que A est un groupe commutatif pour $+$ et d'une opération de multiplication associative, commutative et distributive sur $+$:

$$x(y+z) = xy + xz.$$

Si la multiplication admet un élément neutre, noté 1, l'anneau est dit *unitaire*.

1.2 Algèbre de Boole

1.3 Définition (algèbre de Boole)
Soit B un treillis distributif avec plus grand et plus petit élément. Supposons que tout élément x de B a un complémentaire *y vérifiant :*

$$x \vee y = \top$$
$$x \wedge y = \bot$$

On dit alors que B est une algèbre de Boole *ou un* treillis de Boole.

Exemple.

L'ensemble $\mathbb{N}$ ordonné par « divise » n'est pas une algèbre de Boole.

L'ensemble ordonné $\{0, 1\}$ est une algèbre de Boole.

Soit X un ensemble. On a déjà vu que l'ensemble des parties de X est un treillis distributif, c'est également une algèbre de Boole en définissant le complémentaire comme on l'imagine.

Par contre l'ensemble des parties finies de X n'est pas une algèbre de Boole. En revanche l'ensemble des parties finies ou cofinies de X est une algèbre de Boole (une partie *cofinie* de X est une partie de X dont le complémentaire est fini).

Si B est une algèbre de Boole alors le complémentaire de tout élément x de B est unique. On le notera x^c.

1.4 Proposition (Loi de de Morgan)
Pour tout x et y dans l'algèbre de Boole B on a :

$$(x^c)^c = x$$
$$(x \vee y)^c = x^c \wedge y^c$$
$$(x \wedge y)^c = x^c \vee y^c$$

1.5 Définition (Anneau de Boole)
Soit A un anneau commutatif unitaire. On dit que A est un anneau de Boole *si tous ses éléments sont* idempotents *pour la multiplication, c'est à dire si :*

$$xx = x$$

pour tout x dans A.

EXEMPLE.

L'ensemble $\{0,1\}$ en définissant l'addition par : $0+0=1+1=0$ et $1+0=0+1=1$, et la multiplication de la manière naturelle est un anneau de Boole. Cet anneau est aussi connu sous le nom de $\mathbb{Z}/2\mathbb{Z}$ et c'est le seul anneau de Boole qui soit également un corps.

L'ensemble des parties de X en prenant la différence symétrique pour l'addition et l'intersection pour la multiplication.

Par contre l'ensemble des entiers naturels avec l'addition et la multiplication usuelles n'est pas un anneau de Boole : sauf 0 et 1, aucun entier n'est idempotent.

Soit A un anneau de Boole.

1.6 Proposition
Pour tout x dans A, l'opposé de x est x, c'est à dire $x+x=0$. Par conséquent on a $x(1+x)=0$ pour tout x.

Preuve. On montre que $x+x=0$ en calculant $(1+x)^2$. Par idempotence on a $(1+x)^2=1+x$ mais en développant (puisque la multiplication distribue sur l'addition) on a $(1+x)^2=1+x+x+x^2=1+x+x+x$, donc $1+x=1+x+x+x$ et en ajoutant l'opposé de $1+x$ de chaque côté on obtient $0=x+x$.

On définit une relation $x \leq y$ sur A par :

$$x \leq y \text{ ssi } x=xy.$$

1.7 Théorème
La relation $\leq$ ainsi définie est une relation d'ordre. Muni de cette relation, l'anneau de Boole A est une algèbre de Boole.

Preuve. On définit $x \vee y = x+y+xy$ et $x \wedge y = xy$. On peut alors vérifier que $x \vee y$ est bien la borne supérieure de x et y, que $x \wedge y$ est bien la borne inférieure et que A est distributif. Le complémentaire est donné par $x^c = 1+x$.

1.8 Proposition
Dans un anneau de Boole A, on a pour tous x et y

$$x \leq 1+y \text{ ssi } xy=0.$$

Soit maintenant B une algèbre de Boole. On définit deux opérations d'addition et de multiplication sur B :

$$\begin{aligned} x+y &= (x \vee y) \wedge (x \wedge y)^c \\ xy &= x \wedge y \end{aligned}$$

REMARQUE

En jouant avec les lois de de Morgan et la distributivité, on voit que l'on a :

$$\begin{aligned} x+y &= (x \vee y) \wedge (x^c \vee y^c) \\ &= (x \wedge y^c) \vee (y \wedge x^c) \end{aligned}$$

1.9 Théorème
Muni de ces opérations l'algèbre de Boole B est un anneau de Boole.

Une algèbre de Boole peut donc indifféremment être vue comme un treillis de Boole ou un anneau de Boole.

Isomorphismes. On va finir cette section avec un petit théorème très utile. Un *isomorphisme* entre deux ensembles ordonnés X et Y est une paire d'applications $\varphi : X \mapsto Y$ et $\psi : Y \mapsto X$, inverse l'une de l'autre et preservant l'ordre, c'est à dire telles que :

$$\varphi(x) \leq_Y \varphi(x') \text{ ssi } x \leq_X x'$$
$$\psi(y) \leq_X \psi(y') \text{ ssi } y \leq_Y y'$$

1.10 Théorème
Soit A une algèbre de Boole et X un ensemble ordonné. S'il existe un isomorphisme d'ordre *entre X et A, alors X est également une algèbre de Boole.*

1.3 Théorème de Stone

1.11 Théorème (Stone fini)
Soit B une algèbre de Boole finie. *Il existe un ensemble fini X tel que B est isomorphe à $\mathcal{P}(X)$, l'ensemble des parties de X.*

Avant de prouver le théorème on va établir quelques lemmes.

1.12 Lemme
Soit B une algèbre de Boole et $b_0 \in B$; on note B_0 l'ensemble des minorants de b_0 dans B. Alors B_0 est une algèbre de Boole dont le plus petit élément est $\bot$, le plus grand est b_0 et dans laquelle le complémentaire d'un élément b est $b^c \wedge b_0$.

Preuve. Par définition on a

$$B_0 = \{b \in B \text{ tel que } b \leq b_0\}.$$

Comme B_0 est un sous-ensemble de B, B_0 est ordonné ; son plus grand élément est b_0 et le plus petit est $\bot$.

Soient b et b' deux éléments de B_0. On vérifie très facilement que $b \wedge b'$ est la borne inférieure dans B_0 de b et b'. De plus comme $b \leq b_0$ et $b' \leq b_0$ on a $b \vee b' \leq b_0$; on en déduit que $b \vee b'$ est la borne supérieure dans B_0 de b et b'. L'ensemble B_0 est donc un treillis.

Comme B est distributif et comme les bornes inférieure et supérieure dans B_0 sont celles de B, on voit que B_0 est également distributif.

Finalement si $b \in B_0$ montrons que $b' = b^c \wedge b_0$ est son complémentaire dans B_0 :

- tout d'abord il est clair que $b' \leq b_0$ donc $b' \in B_0$;
- on a $b \wedge b' = b \wedge (b^c \wedge b_0) = \bot \wedge b_0 = \bot$;
- de même $b \vee b' = b \vee (b^c \wedge b_0) = (b \vee b^c) \wedge (b \vee b_0) = \top \wedge b_0$ car $b \leq b_0$. Comme $\top$ est le plus grand élément de B, on a $\top \wedge b_0 = b_0$, donc $b \vee b' = b_0$ qui est le plus grand élément de B_0.

B_0 est donc un treillis distributif, muni d'un plus petit et d'un plus grand élément et complémenté : c'est une algèbre de Boole.

1.13 Lemme
Soit B une algèbre de Boole finie *à deux éléments au moins. Alors B est* atomique, *c'est à dire que B contient au moins un élément a qui est un successeur de $\bot$; un tel élément est appelé un* atome.

Preuve. Par définition un atome de B est donc un élément $a \in B$ tel que :

- $a \neq \bot$;
- pour tout $x \in B$, si $\bot \leq x \leq a$ alors $x = \bot$ ou $x = a$.

On montre par récurrence sur le nombre d'éléments de B que B est atomique.

Si B n'a que deux éléments, alors un seul d'entre eux est différent de $\bot$: il s'agit de $\top$. Dans ce cas particulier il est clair que $\top$ est un atome (pourquoi ?).

Supposons maintenant que B a strictement plus que deux éléments et soit b_0 un élément différent de $\bot$ et différent de $\top$. Notons B_0 l'ensemble des minorants de b_0. D'après le théorème précédent, B_0 est une algèbre de Boole.

Comme $b_0 \neq \top$, $\top \notin B_0$ donc B_0 a strictement moins d'éléments que B et on peut lui appliquer l'hypothèse de récurrence : il y a donc un atome a_0 dans B_0. Reste à voir que c'est aussi un atome dans B. Soit donc $x \in B$ tel que $\bot \leq x \leq a_0$. Comme $a_0 \in B_0$ on a $a_0 \leq b_0$, donc $x \leq b_0$ et donc $x \in B_0$. Mais comme a_0 est un atome dans B_0 ceci entraine que $x = \bot$ ou $x = a_0$. Donc x est bien un atome dans B également.

REMARQUE

L'algèbre de Boole $\mathcal{P}(X)$ des parties d'un ensemble X est atomique (même si X est infini) ; ses atomes sont les singletons.

Si b est un élément d'une algèbre de Boole finie B, on note $\overline{b}$ l'ensemble des atomes de B qui minorent b :

$$\overline{b} = \{a \in B \text{ tel que } a \text{ est un atome et } a \leq b\}$$

1.14 Lemme
Soit B une algèbre de Boole finie et soient b et b' deux éléments B. On a $b \leq b'$ ssi $\overline{b} \subset \overline{b}'$.

Preuve. Si $b \leq b'$ alors tout minorant de b est aussi un minorant de b' donc en particulier $\overline{b} \subset \overline{b}'$.

Réciproquement supposons que $\overline{b} \subset \overline{b}'$. Soit $b_0 = b \wedge b'^c$; on va montrer que $b_0 = \bot$. Si $b_0 \neq \bot$ alors considérons B_0 l'algèbre de Boole des minorants de b_0 ; celle-ci est finie et contient au moins deux éléments, donc elle contient un atome a_0. Comme $a_0 \leq b_0$, on a en particulier $a_0 \leq b$, c'est à dire $a_0 \in \overline{b}$. Mais $\overline{b} \subset \overline{b}'$ et comme a_0 est un atome de B_0 donc de B (pourquoi ?), on a $a_0 \in \overline{b}'$, c'est à dire $a_0 \leq b'$. Mais $a_0 \leq b_0$ entraine également que $a_0 \leq b'^c$. Donc on a $a_0 \wedge b' = a_0 \wedge b'^c = a_0$ donc $a_0 = (a_0 \wedge b') \wedge (a_0 \wedge b'^c) = \bot$, ce qui est impossible car a_0 est un atome.

Donc $b_0 = \bot$, c'est à dire $b \wedge b'^c = \bot$ d'où l'on déduit $b \leq b'$ (comment ?).

Montrons enfin le théorème de Stone.

Preuve. Puisque B est finie on sait qu'elle a des atomes. Notons A l'ensemble des atomes de B ; A est donc fini. On considère la fonction :

$$\begin{array}{rcl} \varphi : B & \mapsto & \mathcal{P}(A) \\ b & \to & \overline{b} \end{array}$$

Soit b et b' tels que $\overline{b} = \overline{b}'$. Le lemme ci-dessous nous assure alors que $b = b'$; la fonction φ est donc injective.

Soit $\beta = \{a_1, \ldots, a_k\}$ une partie de A et considérons l'élément $b = a_1 \vee \ldots \vee a_k$ (on rappelle que dans le cas particulier où $k = 0$, c'est à dire si β est la partie vide, par définition $b = \bot$). On va montrer que $\overline{b} = \beta$ d'où l'on déduit que la fonction φ est surjective.

Soit a un atome de B et supposons que $a \leq b$; on a donc $a \wedge b = a$. Mais $a \wedge b = (a \wedge a_1) \vee \ldots \vee (a \wedge a_k)$. Si a est différent de chaque a_i alors on a $a \wedge a_i = \bot$ pour chaque i, dont $a \wedge b = \bot \vee \ldots \vee \bot = \bot$. D'autre part si a est égal à l'un des a_i alors on a $a \wedge b = \bot \vee \ldots \vee a_i \vee \ldots \vee \bot = a_i$, donc $a \leq b$. Autrement dit on vient de montrer que $a \leq b$ ssi a est égal à l'un des a_i ; ce qui revient exactement à dire que $\overline{b} = \beta$.

Enfin le lemme nous assure que la fonction est un morphisme d'ordre ; il s'agit donc bien d'un isomorphisme et le théorème est démontré.

1.15 Corollaire
Si B est une algèbre de Boole finie alors le nombre d'éléments de B est 2^n où n est le nombre de d'atomes de B.

1.4 Fonctions booléennes

À partir de maintenant on note $\mathcal{B}$ l'algèbre de Boole $\{0, 1\}$. Les éléments de $\mathcal{B}$ sont appelés des *bits*.
Soit n un entier positif ; on rappelle que $\mathcal{B}^n$ est l'ensemble des n-uplets de bits. Par exemple $\mathcal{B}^2$ est l'ensemble des couples de bits, soit :

$$\mathcal{B}^2 = \{(0,0), (0,1), (1,0), (1,1)\}.$$

L'ensemble $\mathcal{B}^n$. Comme on va beaucoup utiliser les n-uplets de bits, on allège la notation en supprimant les parenthèses et les virgules. Par exemple le couple $(0,1)$ s'écrit directement 01, le triplet $(1,0,1)$ s'écrit 101, etc. Un n-uplet de 0 et de 1 s'appelle aussi *un mot de longueur n sur l'alphabet* $\{0,1\}$. Avec cette notation $\mathcal{B}^2$ s'écrit :

$$\mathcal{B}^2 = \{00, 01, 10, 11\}.$$

On rappelle que $\mathcal{B}^n$ a exactement 2^n éléments[1]. On va convenir une fois pour toute d'une énumération de $\mathcal{B}^n$, c'est à dire d'une manière de numéroter chaque élément de $\mathcal{B}^n$ en commençant par 0 :

Numéro	n-uplet
0	$0\ldots000$
1	$0\ldots001$
2	$0\ldots010$
3	$0\ldots011$
$\vdots$	$\vdots$
2^n-2	$1\ldots110$
2^n-1	$1\ldots111$

Autrement dit l'élément numéro k de $\mathcal{B}^n$ est l'écriture en base 2 de k sur n chiffres. Ou réciproquement, le numéro du n-uplet $(b_1,\ldots,b_n)$ est :

$$\sum_{k=0}^{n-1} b_{n-k}2^k = \sum_{k=1}^{n} b_k 2^{n-k}.$$

Ainsi le numéro du n-uplet $(0,\ldots,0)$ est 0, celui de $(1,\ldots,1)$ est $\sum_{k=0}^{n-1} 2^k = 2^n - 1$, comme indiqué sur le tableau ci-dessus.

Remarque

Il existe bien d'autres manières d'énumérer $\mathcal{B}^n$. Ici on a choisit de lire les n-uplets comme des entiers en base 2 où le *bit de poids faible* est à droite. On pourrait choisir le bit de poids faible à gauche ; par exemple pour $n=3$ le triplet de numéro 0 serait 000, celui de numéro 1 serait 100 (au lieu de 001), celui de numéro 2 serait 010, celui de numéro 3 serait 110 (au lieu de 011), etc. Dans ce cas le n-uplet $(b_1,\ldots,b_n)$ aurait pour numéro :

$$\sum_{k=0}^{n-1} b_{k+1}2^k = \sum_{k=1}^{n} b_k 2^{k-1}.$$

On pourrait également choisir de grouper les n-uplets par nombre de 1 : par exemple pour $n=3$:

Nombre de 1 = 0	
0	000

Nombre de 1 = 1	
1	100
2	010
3	001

Nombre de 1 = 2	
4	110
5	101
6	011

Nombre de 1 = 3	
7	111

L'inconvénient de cette dernière méthode est de ne pas fournir de formule simple permettant de retrouver le numéro d'un n-uplet.

Il existe encore bien d'autres énumérations de $\mathcal{B}^n$; par exemple les *codes de Gray* énumèrent les n-uplets de manière à ce que un seul bit change entre deux n-uplets consécutif (exo : trouver un code de Gray pour $n=3$).

[1] En particulier $\mathcal{B}^0$ a un élément : le *mot vide* que l'on note souvent ϵ, c'est à dire le seul 0-uplet : ().

Fonctions booléennes. Une *fonction booléenne d'arité n* est une fonction de $\mathcal{B}^n$ dans $\mathcal{B}$. On note $\mathcal{F}_n$ l'ensemble des fonctions booléennes d'arité n.
Comme il y a 2^n n-uplets de bits, l'ensemble $\mathcal{F}_n$ contient 2^{2^n} fonctions booléennes d'arité n. Par exemple $\mathcal{F}_1$ contient $4 = 2^{2^1}$ éléments f_1, f_2, f_3, f_4 définies par :

b	$f_1(b)$	$f_2(b)$	$f_3(b)$	$f_4(b)$
0	0	0	1	1
1	0	1	0	1

De même $\mathcal{F}_2$ contient $16 = 2^{2^2}$ fonctions booléennes d'arité 2, etc.

1.16 Théorème
L'ensemble $\mathcal{F}_n$ des fonctions booléennes d'arité n ordonné par :

$$f \leq g \text{ ssi pour tout } (b_1, \ldots, b_n) \text{ dans } \mathcal{B}^n,\ f(b_1, \ldots, b_n) \leq g(b_1, \ldots, b_n)$$

forme une algèbre de Boole. Les opérations d'algèbre (et d'anneau) de Boole sont données par :

$$\begin{aligned}(f \wedge g)(b_1, \ldots, b_n) &= f(b_1, \ldots, b_n) \wedge g(b_1, \ldots, b_n)\\ (f \vee g)(b_1, \ldots, b_n) &= f(b_1, \ldots, b_n) \vee g(b_1, \ldots, b_n)\\ (f + g)(b_1, \ldots, b_n) &= f(b_1, \ldots, b_n) + g(b_1, \ldots, b_n)\\ f^c(b_1, \ldots, b_n) &= (f(b_1, \ldots, b_n))^c\end{aligned}$$

et le plus petit élément de $\mathcal{F}_n$ est la fonction identiquement nulle, le plus grand élément est la fonction constante 1.

La preuve est une simple vérification des différentes propriétés d'algèbre de Boole. En fait on a un théorème un peu plus général :

1.17 Théorème
Si X est un ensemble non vide alors l'ensemble $\mathcal{F}_X$ des fonctions de X dans $\mathcal{B}$ ordonné par

$$f \leq g \text{ ssi pour tout } x \in X,\ f(x) \leq g(x)$$

est une algèbre de Boole.

Pour voir cela on montre le lemme :

1.18 Lemme
$\mathcal{F}_X$ est isomorphe à $\mathcal{P}(X)$, l'ensemble des parties de X. Plus précisément les applications $\varphi : \mathcal{F}_X \mapsto \mathcal{P}(X)$ définie par :

$$x \in \varphi(f) \text{ ssi } f(x) = 1$$

et $\psi : \mathcal{P}(X) \mapsto \mathcal{F}_X$ définie par :

$$\psi(A)(x) = 1 \text{ ssi } x \in A$$

sont réciproques l'une de l'autre et préservent l'ordre :

$$\begin{aligned}f \leq g \text{ ssi } \varphi(f) \subset \varphi(g)\\ A \subset B \text{ ssi } \psi(A) \leq \psi(B)\end{aligned}$$

Étant donnée une partie A de X, l'application $\psi(A) : X \mapsto \mathcal{B}$ est appelée la *fonction caractéristique* de A.

Tables de vérité. Il existe plusieurs méthodes pour décrire complètement une fonction booléenne. La première est de donner sa *table de vérité*, c'est à dire la liste des valeurs de la fonction pour chaque n-uplet de bits. Par exemple la table de vérité de la fonction identiquement nulle sur $\mathcal{B}^2$ est :

00	0
01	0
10	0
11	0

et celle de la fonction : $(b_1, b_2) \mapsto b_1 + b_2$ est :

00	0
01	1
10	1
11	0

Remarque

Comme annoncé, on a listé les éléments de $\mathcal{B}^2$ selon le principe énoncé au début de la section.

Par définition la table de vérité d'une fonction booléenne d'arité n contient 2^n lignes ce qui n'est pas très économique ; par exemple ici on a beaucoup plus vite fait de décrire la fonction en disant qu'elle est *identiquement nulle* dans le premier cas, ou qu'il s'agit de la somme dans l'anneau de Boole $\mathcal{B}$ dans le second cas. L'un des problèmes principaux à propos des fonctions booléennes est de trouver des manières économiques de les décrire et de les calculer.

On va commencer par voir que toutes les fonctions booléennes peuvent être décrites en utilisant uniquement les opérations binaires de l'algèbre de Boole $\mathcal{B}$.

1.5 Formes normales

Variables. On appelle *variables* de $\mathcal{F}_n$ les n fonctions de $\mathcal{F}_n$ notées x_1, ..., x_n définies par :

$$x_i(b_1, \ldots, b_n) = b_i.$$

Remarque

Soit f une fonction booléenne d'arité p, f_1, ..., f_p des fonctions booléennes d'arité n et g la fonction booléenne d'arité n définie par :

$$g(b_1, \ldots, b_n) = f(f_1(b_1, \ldots, b_n), \ldots, f_p(b_1, \ldots, b_n))$$

qui est simplement la composée de f avec les f_i. Quand g est définie ainsi on écrit souvent

$$g = f(f_1, \ldots, f_p).$$

Supposons maintenant que f est d'arité n ; par définition des x_i on a :

$$f(b_1, \ldots, b_n) = f(x_1(b_1, \ldots, b_n), \ldots, x_n(b_1, \ldots, b_n))$$

ce qui avec la notation que l'on vient d'introduire s'écrit :

$$f = f(x_1, \ldots, x_n).$$

C'est pour cette raison que les fonctions booléennes x_i sont appelées des *variables*. Il faut bien garder à l'esprit que l'objet noté $f(x_1, \ldots, x_n)$ est un élément de $\mathcal{F}_n$ alors que $f(b_1, \ldots, b_n)$ (où les b_i sont des bits) est un élément de $\mathcal{B}$.

Les variables sont les fonctions de base à partir desquelles, et au moyen des opérations de l'algèbre de Boole $\mathcal{F}_n$, on va pouvoir exprimer toutes les fonctions de $\mathcal{F}_n$. Par exemple on peut maintenant écrire la fonction somme de $\mathcal{F}_2$: $(b_1, b_2) \mapsto b_1 + b_2$ comme $x_1 + x_2$. En effet, on a :

$$\begin{aligned}(x_1 + x_2)(b_1, b_2) &= x_1(b_1, b_2) + x_2(b_1, b_2) \\ &= b_1 + b_2\end{aligned}$$

Littéraux. Si x_i est une variable, on appelle *négation de* x_i la fonction booléenne x_i^c :

$$x_i^c(b_1, \ldots, b_n) = (x_i(b_1, \ldots, b_n))^c = b_i^c.$$

Les variables et leurs négations sont appelées les *littéraux* de $\mathcal{F}_n$. Il y a exactement $2n$ littéraux dans $\mathcal{F}_n$: x_1 et x_1^c, ..., x_n et x_n^c. Les littéraux x_i et x_i^c sont dits *d'indice* i et on les note l_i.
On va maintenant voir que toutes les fonctions de $\mathcal{F}_n$ peuvent s'écrire comme des combinaisons de littéraux au moyen des opérations $\vee$ et $\wedge$.

Atomes. On appelle *atome* de $\mathcal{F}_n$ une fonction qui prend la valeur 1 pour exactement un n-uplet de $\mathcal{B}^n$.

Remarque

Dans le livre de J. Vélu, les atomes de $\mathcal{F}_n$ sont appelés des *mintermes*.

Comme toutes les fonctions booléennes d'arité n, les atomes sont des fonctions caractéristiques de sous-ensembles de $\mathcal{B}^n$. Mais comme par définition ils ne prennent la valeur 1 que pour un seul n-uplet, ce sont les fonctions caractéristiques des singletons inclus dans $\mathcal{B}^n$.

Remarque

La table de vérité d'un atome de $\mathcal{F}_n$ contient exactement une ligne portant la valeur 1. Comme chaque table de vérité a 2^n lignes, il y a 2^n atomes dans $\mathcal{F}_n$.

1.19 Théorème
Soit m une fonction booléenne d'arité n. Les deux conditions suivantes sont équivalentes :

- *m est un atome ;*
- *il existe des littéraux l_1 d'indice 1, l_2 d'indice 2, ..., l_n d'indice n tels que :*

$$m = l_1 \wedge \ldots \wedge l_n.$$

En général on utilise plutôt la notation produit des anneaux de Boole :

$$m = l_1 \ldots l_n.$$

Pour cette raison les atomes de $\mathcal{F}_n$ sont aussi appelés des *monômes*.

Preuve. Le fait qu'un produit de littéraux est un atome est une simple vérification. Réciproquement supposons que m est un atome et soit $(b_1, \ldots, b_n)$ l'unique n-uplet tel que $m(b_1, \ldots, b_n) = 1$. Pour $i = 1, \ldots, n$ on définit un littéral l_i d'indice i par :

$$l_i = \begin{cases} x_i & \text{si } b_i = 1, \\ x_i^c & \text{sinon.} \end{cases}$$

Par définition des l_i on a :

$$l_i(b_1, \ldots, b_n) = 1 \text{ pour } i = 1, \ldots, n.$$

Soit f la fonction booléenne d'arité n définie par $f = l_1 \wedge \ldots \wedge l_n$. On a

$$f(b_1, \ldots, b_n) = l_1(b_1, \ldots, b_n) \wedge \ldots \wedge l_n(b_1, \ldots, b_n) = 1.$$

Si maintenant on prend un n-uplet $(c_1, \ldots, c_n)$ distinct de $(b_1, \ldots, b_n)$ alors il y a au moins un i tel que c_i est différent de b_i et par définition de l_i on a $l_i(c_i) = 0$; mais alors

$$f(c_1, \ldots, c_n) = l_1(c_1, \ldots, c_n) \wedge \ldots \wedge l_i(c_1, \ldots, c_n) \wedge \ldots \wedge l_n(c_1, \ldots, c_n) = 0.$$

Donc f prend une valeur non nulle sur le seul n-uplet $(b_1, \ldots, b_n)$ ce qui montre que $f = m$, c'est à dire :

$$m = l_1 \wedge \ldots \wedge l_n$$

comme annoncé.

Formes normales. Soit f une fonction booléenne d'arité n. On considère l'ensemble des n-uplets où f est non nulle. À chaque tel n-uplet on sait associer un atome ; notons ceux-ci $m_1, \ldots, m_p$. Alors on a :

$$f = m_1 \vee \ldots \vee m_p.$$

L'écriture de f sous cette forme est appelée *forme normale disjonctive canonique*.

Remarque

Le nombre p ci-dessus est égal au nombre de lignes de la table de vérité de f portant la valeur 1. Pour un f quelconque c'est en moyenne la moitié des lignes, soit 2^{n-1} atomes. Chaque atome m_i s'écrit comme une borne inférieure de n littéraux. Au total la taille de la forme normale canonique est $n2^{n-1}$, c'est un peu plus de la moitié de la taille du tableau de vérité entier.

On pourrait penser que l'on a fait beaucoup d'effort pour un résultat un peu faible. En fait on a obtenu une chose très importante : on a montré que l'on pouvait exprimer f au moyen des seuls opérations de sup, inf, complémentaire et des variables, ce qui n'était pas évident à priori. On va pouvoir maintenant employer toutes les propriétés des algèbres de Boole pour tenter de simplifier l'expression de f. De plus cette expression fournit en soit une méthode pour calculer la valeur de f sur les n-uplets.

Formes normales conjonctives. Soit f une fonction booléenne d'arité n. Considérons la fonction f^c que nous pouvons écrire en forme canonique disjonctive :

$$f^c = m_1 \vee \ldots \vee m_p$$

où les m_i sont des atomes. En passant au complémentaire, comme $(f^c)^c = f$ et en utilisant les lois de de Morgan on voit que :

$$f = m_1^c \wedge \ldots \wedge m_p^c.$$

Un atome est une fonction qui prend la valeur 1 pour exactement un n-uple. Le complémentaire d'un atome est donc une fonction qui prend la valeur 0 pour exactement un n-uple. On appellera de telles fonctions des *co-atomes* (J. Vélu les appelle des *maxtermes*). Toujours grâce aux lois de de Morgan on voit que tout co-atome h peut s'écrire :

$$h = l_1 \vee \ldots \vee l_n$$

où pour $i = 1, \ldots, n$, l_i est un littéral d'indice i. Cette expression de f comme inf de co-atomes, où chaque co-atome est un sup de littéraux, est appelée la *forme normale conjonctive canonique* de f.

1.6 Formules du calcul propositionnel

On va finir ce chapitre sur le calcul booléen avec une rapide présentation de l'algèbre de Boole des *propositions*. On commence par définir le *calcul propositionnel*. Il n'y a pas à proprement parler un seul calcul propositionnel car la définition dépend d'un certain nombre de données détaillées ci-dessous.
On se donne un ensemble $\mathcal{V}$ fini ou dénombrable[2]. Les éléments de $\mathcal{V}$ sont appelés les *variables propositionnelles* et on utilisera les lettres p et q pour les dénoter. On se donne également un ensemble fini $\mathcal{C}$ contenant au moins trois éléments distingués notés $\neg$, $\vee$ et $\wedge$. Les éléments de $\mathcal{C}$ sont appelés des *connecteurs*. On se donne enfin une *fonction d'arité* $a : \mathcal{C} \mapsto \mathbb{N}$ qui associe un entier (possiblement nul) à chaque connecteur (appelé l'*arité du connecteur*) et telle que l'arité du connecteur $\neg$ est 1, celles de $\wedge$ et $\vee$ sont toutes deux égales à 2.

Remarque

Les connecteurs $\neg$, $\vee$ et $\wedge$ sont appelés respectivement *négation*, *disjonction* et *conjonction*. En général on trouve d'autres connecteurs dans $\mathcal{C}$: l'*implication* noté $\rightarrow$ et d'arité 2, les constantes *vrai* et *faux* notées $\boldsymbol{tt}$ et $\boldsymbol{ff}$ d'arité 0 (c'est pourquoi on les appelle des *constantes*), l'*équivalence* notée $\leftrightarrow$, ...

Ces données seront complétées au début de la section 1.6 par la définition des *fonctions de vérité* associées à chaque connecteur.

[2]On rappelle qu'un ensemble D est *dénombrable* si il existe une bijection $\varphi : D \mapsto \mathbb{N}$ ou autrement dit si chaque élément de D peut se voir affecter un numéro unique.

Formules. Pour chaque entier n on définit par récurrence sur n un ensemble $\mathcal{T}_n$ dont les éléments sont appelés les *formules de hauteur au plus* n :

- $\mathcal{T}_0$ est l'ensemble $\mathcal{V}$ des variables propositionnelles auquel on ajoute tous les connecteurs d'arité 0 de $\mathcal{C}$ s'il y en a (les constantes propositionnelles) :

$$\mathcal{T}_0 = \mathcal{V} \cup \{c \in \mathcal{C} \text{ tel que } a(c) = 0\}.$$

- $\mathcal{T}_{n+1}$ est $\mathcal{T}_n$ auquel on ajoute pour chaque $k > 0$ l'ensemble des $k+1$-uplets $(c, A_1, \ldots, A_k)$ où c est un connecteur d'arité k et A_1, …, A_k sont des éléments de $\mathcal{T}_n$:

$$\mathcal{T}_{n+1} = \mathcal{T}_n \cup \bigcup_{k \geq 1} \{(c, A_1, \ldots, A_k) \in \mathcal{C} \times \mathcal{T}_n \times \cdots \times \mathcal{T}_n \text{ tel que } a(c) = k\}.$$

Remarquons que par définition on a toujours $\mathcal{T}_n \subset \mathcal{T}_{n+1}$ et donc $\mathcal{T}_n \subset \mathcal{T}_p$ dès que $n \leq p$. Enfin on note $\mathcal{T}$ la réunion de tous les $\mathcal{T}_n$:

$$\mathcal{T} = \bigcup_{n \geq 0} \mathcal{T}_n.$$

Les éléments de $\mathcal{T}$ sont appelées les *formules propositionnelles* et sont en général dénotés par les lettres A, B, C, …

Remarque

> Si on prend $\mathcal{V}$ vide et $\mathcal{C} = \{\neg, \vee, \wedge\}$ alors l'ensemble $\mathcal{T}$ est vide ! En effet $\mathcal{T}_0$ est vide puisque $\mathcal{V}$ est vide et il n'y a aucun connecteur d'arité 0 dans $\mathcal{C}$; mais alors $\mathcal{T}_1$ est vide aussi puisque il n'y a aucune formule dans $\mathcal{T}_0$; et $\mathcal{T}_2$ est vide aussi, ainsi que $\mathcal{T}_3$, etc. Finalement $\mathcal{T}$ est une réunion d'ensembles vides, donc est vide.
>
> Pour cette raison, à partir de maintenant on supposera que $\mathcal{V}$ est choisi non vide.

Hauteur d'une formule. Comme $\mathcal{T}$ est la réunion des $\mathcal{T}_n$, pour tout élément A de $\mathcal{T}$ il existe un entier n tel que $A \in \mathcal{T}_n$. On appelle *hauteur* de la formule A le plus petit entier h tel que $A \in \mathcal{T}_h$. La hauteur de A est notée $|A|$. Par définition de la hauteur, si A appartient à $\mathcal{T}_n$ alors sa hauteur est inférieure ou égale à n.

Remarque

> En utilisant le langage de la théorie des graphes, les formules propositionnelles sont des *arbres* finis dont chaque nœud est étiqueté par un connecteur de $\mathcal{C}$ et doit respecter l'arité de ce connecteur, et dont les feuilles sont étiquetés par des variables ou des constantes propositionnelles. C'est pour cela que l'on parle de *hauteur* d'une formule.

Soit A une formule de hauteur h. Si $h = 0$ alors A appartient à $\mathcal{T}_0$ et donc A est soit une variable propositionnelle, soit une constante propositionnelle.
Si $h = h' + 1$ alors, par définition de la hauteur, A appartient à $\mathcal{T}_{h'+1}$, mais pas à $\mathcal{T}_{h'}$. Donc on a $A = (c, A_1, \ldots, A_k)$ pour un connecteur c d'arité k et des formules A_1, …, A_k de $\mathcal{T}_n$. On a donc le lemme :

1.20 Lemme
Soit A une formule de hauteur $h+1$; il existe un (unique) connecteur c de $\mathcal{C}$, d'arité k, et des (uniques) formules A_1,…, A_k dont les hauteurs sont toutes inférieures ou égales à h tel que :

$$A = (c, A_1, \ldots, A_k).$$

L'unicité est immédiate puisque si $A = (c, A_1, \ldots, A_k)$ et $A = (c', B_1, \ldots, B_m)$ alors $(c, A_1, \ldots, A_k) = (c', B_1, \ldots, B_m)$. Remarquons que par définition de la hauteur, l'une au moins des formules A_i doit avoir la hauteur h.
On dit que c est le *connecteur principal* de la formule A et que les A_i sont les *sous-formules immédiates* de A.

Substitution et instances de formules. Soient n un entier, A, B_1, ..., B_n des formules, p_1, ..., p_n des variables propositionnelles. On définit par récurrence sur la hauteur de A une formule notée $A[B_1/p_1, \ldots, B_n/p_n]$ appelée « A dans laquelle les variables p_i sont substituées par les formules A_i » (ou encore « A dans laquelle les A_i sont substituées aux p_i ») :

- si $|A| = 0$ alors A est une formule ou une constante propositionnelle p et il y a deux cas :
  - p est égale à l'une des variables p_i auquel cas $A[B_1/p_1, \ldots, B_n/p_n] = B_i$;
  - sinon $A[B_1/p_1, \ldots, B_n/p_n] = p$.
- si $|A| = h+1$ alors $A = (c, A_1, \ldots, A_k)$ où c est un connecteur d'arité k et A_1, ..., A_k sont des formules de hauteurs au plus h ; dans ce cas

$$A[B_1/p_1, \ldots, B_n/p_n] = (c, A_1[B_1/p_1, \ldots, B_n/p_n], \ldots, A_k[B_1/p_1, \ldots, B_n/p_n]).$$

La formule $A[B_1/p_1, \ldots, B_n/p_n]$ est appelée une *instance* de la formule A.

Remarque

Soit A une formule et p_1, ..., p_n des variables propositionnelles. Définissons la fonction $\alpha : \mathcal{T}^n \mapsto \mathcal{T}$ par $\alpha(B_1, \ldots, B_n) = A[B_1/p_1, \ldots, B_n/p_n]$. Dans le cas où A est la variable propositionnelle p_i, la définition ci-dessus nous donne $\alpha(B_1, \ldots, B_n) = B_i$. Autrement dit, vue comme une fonction de $\mathcal{T}^n$ dans $\mathcal{T}$, la variable propositionnelle p_i est une fonction projection, exactement comme, dans le cas des fonctions booléennes, on appelait *variable* les fonctions x_i de projection de $\mathcal{B}^n$ dans $\mathcal{B}$: $x_i(b_1, \ldots, b_n) = b_i$.

Notations des formules. Lorsque l'on manipule des formules propositionnelles, il est souvent malcommode de les écrire comme des tuples. C'est pourquoi on introduit une notation plus praticable. Si c est un connecteur unaire (la négation) et A une formule alors on note $c(A)$ la formule (c, A) ; dans le cas où A est de hauteur 0, on fait même l'économie des parenthèses : par exemple la formule $(\neg, p)$ où p est une variable propositionnelle s'écrit $\neg p$.
Si c est un connecteur binaire et A et B sont deux formules alors on écrit $(A)\, c(B)$ pour la formule (c, A, B) et on fait même l'économie des parenthèses si il n'y a pas d'ambigüité : par exemple la formule $(\vee, p, (\neg, q))$ s'écrit $p \vee \neg q$.
Enfin, toujours dans l'idée d'économiser l'usage des parenthèses, on convient que le connecteur d'implication *associe à droite* : c'est à dire que si A, B et C sont des formules, la formule $(\rightarrow, A, (\rightarrow, B, C))$ qui devrait s'écrire $A \rightarrow (B \rightarrow C)$, s'écrit plus simplement $A \rightarrow B \rightarrow C$. Par contre la formule $(\rightarrow, (\rightarrow, A, B), C)$ s'écrit $(A \rightarrow B) \rightarrow C$ et on ne peut lui enlever de parenthèses sans la confondre avec la précédente.

1.7 Valeurs de vérité

Interprétation des connecteurs. Pour terminer la définition de l'algèbre de Boole des propositions, on va se donner une *interprétation booléeenne* des connecteurs. À chaque connecteur c d'arité k de $\mathcal{C}$ on associe une fonction booléenne $\bar{c}$ d'arité k appelée la *fonction de vérité* de c. Voici les interprétations des connecteurs usuels :

- si les constantes ***ff*** et ***tt*** appartiennent à $\mathcal{C}$, leurs fonctions de vérité sont les constantes booléennes 0 et 1 (une constante est une fonction d'arité 0) ;
- la fonction de vérité de $\neg$ est définie par $\bar{\neg}(b) = b^c$;
- les fonctions de vérité de $\vee$ et $\wedge$ sont définies par : $\bar{\vee}(b_1, b_2) = b_1 \vee b_2$ et $\bar{\wedge}(b_1, b_2) = b_1 \wedge b_2$;
- si $\rightarrow$ est dans $\mathcal{C}$, sa fonction de vérité est donnée par : $\bar{\rightarrow}(b_1, b_2) = b_1^c \vee b_2$;
- si $\leftrightarrow$ est dans $\mathcal{C}$, sa fonction de vérité est donnée par : $\bar{\leftrightarrow}(b_1, b_2) = (b_1 + b_2)^c$.

Remarque

La définition de la fonction de vérité de l'implication peut sembler contradictoire avec l'intuition que l'implication exprime une relation de cause à effet. Par exemple avec cette définition il est vrai le théorème des valeurs intermédiaires en analyse implique le théorème de décomposition en facteurs premiers en arithmétique, alors que ces deux théorèmes n'ont pas grand chose à voir. Il est également vrai, avec cette définition, qu'un énoncé faux implique un énoncé faux : par exemple $0 = 1$ implique « je suis le pape » (exemple dû à David Hilbert, grand mathématicien et logicien de la fin du XIXème - début du XXème siècle).

On peut néanmoins se convaincre que l'on a pas le choix sur les valeurs de vérité de l'implication en considérant par exemple l'énoncé : pour tout entier n, si n est un multiple de 4 alors n est pair. Il ne fait guère de doute que cet énoncé est vrai, et partant, il doit l'être quel que soit le choix de n. Si on prend $n = 2$, 3 ou 4 on verra apparaître tous les cas de vérité de l'implication : « faux implique vrai », « faux implique faux » et « vrai implique vrai » (cette discussion est largement inspirée de celle de Lascar et Cori dans le chapitre « Calcul propositionnel » de leur cours « Logique Mathématique » publié chez Masson ; je leur ai en particulier emprunté ce dernier exemple.)

Valuation, valeur de vérité d'une formule. On appelle *valuation* ou *distribution de valeurs de vérité* une fonction $\rho : \mathcal{V} \mapsto \mathcal{B}$ associant à chaque variable propositionnelle de $\mathcal{V}$ une *valeur de vérité* appartenant à $\mathcal{B}$ (l'algèbre de Boole $\{0, 1\}$).
Soit ρ une valuation et A une formule. La *valeur de vérité de A relativement à ρ* est l'élément de $\mathcal{B}$ noté $[\![A]\!]_\rho$ défini par récurrence sur la hauteur de A :

- si $|A| = 0$ alors il y a deux cas :
  - A est une variable propositionnelle p, dans ce cas $[\![A]\!]_\rho = \rho(p)$;
  - A est une constante propositionnelle c, dans ce cas $[\![A]\!]_\rho = \bar{c}$;
- sinon $|A| = h+1$ et il existe un connecteur c d'arité k et des formules $A_1, \ldots, A_k$, de hauteurs inférieures ou égales à h tels que $A = (c, A_1, \ldots, A_k)$. On définit alors :

$$[\![A]\!]_\rho = \bar{c}([\![A_1]\!]_\rho, \ldots, [\![A_k]\!]_\rho).$$

Substitution et valeur de vérité. Soit ρ une valuation et $p_1, \ldots, p_n$ des variables propositionnelles deux à deux distinctes, $b_1, \ldots, b_n$ des bits. On note $\rho[p_1 = b_1, \ldots, p_n = b_n]$ la valuation λ définie par :

$$\lambda(p) = \begin{cases} b_i & \text{si } p = p_i \text{ pour un } i = 1, \ldots, n\,, \\ \rho(p) & \text{si } p \text{ est une variable propositionnelle distincte de tous les } p_i\ . \end{cases}$$

1.21 Théorème (Substitution)
Soient A une formule, $p_1, \ldots, p_n$ des variables propositionnelles deux à deux distinctes, $B_1, \ldots, B_n$ des formules et ρ une valuation. Alors on a :

$$[\![A[B_1/p_1, \ldots, B_n/p_n]]\!]_\rho = [\![A]\!]_{\rho[p_1=[\![B_1]\!]_\rho, \ldots, p_n=[\![B_n]\!]_\rho]}.$$

Preuve. Notons λ la valuation $\rho[p_1 = [\![B_1]\!]_\rho, \ldots, p_n = [\![B_n]\!]_\rho]$ et A' la formule $A[B_1/p_1, \ldots, B_n/p_n]$. Avec ces notations, il faut montrer que $[\![A']\!]_\rho = [\![A]\!]_\lambda$.

La démonstration se fait par récurrence sur la hauteur de A. Si $|A| = 0$ alors il y a trois cas :

- A est une constante propositionnelle c; dans ce cas $[\![A]\!]_\sigma = \bar{c}$ pour toute valuation σ, donc l'égalité ci-dessus est bien réalisée.
- A est une variable propositionnelle p distincte de chaque p_i. Dans ce cas on a $A' = A$ par définition de la substitution, et $\lambda(p) = \rho(p)$ par définition de λ, donc, comme $[\![A']\!]_\rho = \rho(p)$ et $[\![A]\!]_\lambda = \lambda(p)$, encore une fois l'égalité est réalisée.
- A est l'une des variables propositionnelles p_i. Dans ce cas on a $A' = B_i$ par définition de la substitution, donc $[\![A']\!]_\rho = [\![B_i]\!]_\rho$. Mais on a $[\![A]\!]_\lambda = [\![p_i]\!]_\lambda = \lambda(p_i) = [\![B_i]\!]_\rho$ par définition de λ. Donc l'égalité est encore vérifiée.

Si maintenant $|A| = h+1$ alors il existe un connecteur c d'arité k et des formules $A_1, \ldots, A_k$ de hauteurs inférieures ou égales à h telles que $A = (c, A_1, \ldots, A_k)$. Pour $i = 1, \ldots, k$ notons A'_i la formule $A_i[B_1/p_1, \ldots, B_n/p_n]$. Comme les A_i sont toutes de hauteurs inférieures ou égales à h, on peut leur appliquer l'hypothèse de récurrence : on a donc $[\![A'_i]\!]_\rho = [\![A_i]\!]_\lambda$ pour $i = 1, \ldots, k$. Par définition de la substitution on a $A' = (c, A'_1, \ldots, A'_k)$, donc $[\![A']\!]_\rho = \bar{c}([\![A'_1]\!]_\rho, \ldots, [\![A'_k]\!]_\rho) = \bar{c}([\![A_1]\!]_\lambda, \ldots, [\![A_k]\!]_\lambda) = [\![A]\!]_\lambda$ et l'égalité est démontrée.

Tautologies, formules satisfaisables, insatisfaisable. Une *tautologie* est une formule dont la valeur de vérité est toujours égale à 1, quelle que soit la valuation choisie. Par exemple $p \vee \neg p$ est une tautologie. Une formule A est *satisfaisable* si il existe une valuation ρ telle que $[\![A]\!]_\rho = 1$; par exemple p est une formule satisfaisable. Si la formule A n'est pas satisfaisable, elle est dite *insatisfaisable* ou que c'est une *antilogie*.

1.22 Théorème
Soit A une tautologie et A' une instance de A. Alors A' est une tautologie.

Preuve. Par définition d'instance, il y a des variables propositionnelles $p_1, \ldots, p_n$ et des formules $B_1, \ldots, B_n$ telles que $A' = A[B_1/p_1, \ldots, B_n/p_n]$. Soit ρ une valuation quelconque et λ la valuation $\rho[p_1 = [\![B_1]\!]_\rho, \ldots, [\![B_n]\!]_\rho]$. D'après le théorème de substitution on a $[\![A']\!]_\rho = [\![A]\!]_\lambda$ mais comme A est une tautologie $[\![A]\!]_\lambda = 1$, donc $[\![A']\!]_\rho = 1$.

Conséquence logique et équivalence logique. Soit A et B deux formules. On dit que B est *conséquence logique* de A, ou que A *entraîne logiquement* B si pour toute valuation ρ on a :

$$[\![A]\!]_\rho \leq [\![B]\!]_\rho.$$

Autrement dit A entraîne logiquement B si $[\![B]\!]_\rho$ vaut 1 dès que $[\![A]\!]_\rho$ vaut 1.
Deux formules A et B sont dites *logiquement équivalentes* si pour toute valuation ρ on a :

$$[\![A]\!]_\rho = [\![B]\!]_\rho.$$

1.23 Théorème
Pour toutes formules A et B, A entraîne logiquement B ssi la formule $A \rightarrow B$ est une tautologie ; A et B sont logiquement équivalentes ssi la formule $A \leftrightarrow B$ est une tautologie.

Algèbre de Boole des propositions. L'algèbre de Boole des propositions est l'ensemble des formules du calcul propositionnel ordonné par la relation d'implication logique. Notons que cette relation n'est qu'une relation de préordre (elle ne satisfait pas l'antisymétrie) et qu'il faut donc *quotienter* $\mathcal{T}$ par la relation d'équivalence logique pour en faire un ensemble ordonné. C'est à dire que l'on considére les formules à équivalence logique près : deux formules équivalentes sont considérées comme égales dans l'algèbre de Boole des propositions et en particulier toutes les tautologies sont égales à la constante propositionnelle $\boldsymbol{tt}$.

Chapitre 2

Graphes

Dans ce chapitre on va introduire les définitions et propriétés de base des *graphes*. Les graphes sont des objets extrêmement courant, dans la vie en générale, et dans les mathématiques en particulier. Un plan de métro est (une représentation d') un graphe, mais également le réseau internet, ou la carte des départements français ; dès que l'on commence à dessiner un ensemble d'objets (des départements) et à les relier par des traits pour signifier des relations entre eux (ils sont limitrophes), on dessine un graphe. La notion intuitivement claire est donc celle de *dessin de graphe*. Cette notion n'est toutefois pas commode pour faire des mathématiques sur les graphes. Tout comme les objets de la géométrie ne sont pas des dessins mais des espaces affines ou vectoriels, des transformations linéaires, isométriques, etc. les graphes demandent une définition plus opératoire que juste « des objets dessinés et reliés par des traits ou des flèches ».

2.1 Définition

Graphe. Un *graphe* G est la donnée de deux ensembles S_G et A_G et d'une fonction $\delta_G : A_G \mapsto S_G^2$.

Terminologie. Les éléments de S_G sont appelés les *sommets* du graphe G (en anglais *vertex/vertices*) et on utilisera les lettres a, b, c, i, j, m, n, p, q, r, s, t pour les dénoter ; les éléments de A_G sont les *arêtes* ou les *flèches* de G (*edges* ou *arrows*) et on utilisera les lettres f, g, h ou α, β pour les dénoter. La fonction δ_G est parfois appelée *fonction d'incidence*. Elle indique pour chaque arête le couple de sommets reliés par cette arête.

Fonctions source et but. On définit deux fonctions s_G et b_G associant un sommet à chaque arête : si $\delta(f) = (s, b)$ est le couple de sommets associé à l'arête f alors $s_G(f) = s$ et $b_G(f) = b$, c'est à dire que pour toute arête f, $s_G(f)$ et $b_G(f)$ sont définies par :

$$(s_G(f), b_G(f)) = \delta_G(f).$$

Le sommet $s_G(f)$ est appelé *source* de l'arête f, et le sommet $b_G(f)$ est le *but* de l'arête f.
Soit a et b deux sommets (possiblement égaux) d'un graphe G. On note $G(a, b)$ l'ensemble des arêtes de source a et de but b :

$$G(a, b) = \{f \in A_G \text{ tel que } s_G(f) = a \text{ et } b_G(f) = b\}.$$

Remarque

> Les arêtes d'un graphe G sont *orientées* de leur source vers leur but et c'est pourquoi on les appelle aussi des *flèches*. Dans la littérature on trouve souvent la terminologie *graphe orienté* pour ce type de graphe.
> Il faut aussi bien voir que entre deux sommets il peut y avoir plusieurs arêtes et même qu'il peut y avoir plusieurs arêtes dont la source et le but sont le même sommet ; on les appelle des *boucles*.

Graphes simples et relations. Un graphe G est dit *simple* si entre chaque paire de sommets il y a au plus une arête, c'est à dire si pour tout s, t dans S_G, l'ensemble $G(s,t)$ est vide ou est un singleton. Notons qu'un graphe simple peut contenir des boucles.
Les graphes simples sont importants car ils servent à représenter des relations. Soit S un ensemble et R une relation binaire sur S. On peut construire un graphe simple G qui représente la relation R sur S de la manière suivante : les sommets de G sont les éléments de S, c'est à dire $S_G = S$. Les arêtes de G sont les couples (s,t) de sommets tels que $s\,R\,t$. La source d'une arête (s,t) est le sommet s et son but est le sommet t. Par définition il y a donc 0 ou une arête entre toute paire de sommet et donc G est simple.
Réciproquement si l'on a un graphe simple G, on peut définir une relation R sur les sommets de G par : $s\,R\,t$ ssi il existe une arête de source s et de but t dans A_G. Ainsi on a une correspondance bi-univoque entre les graphes simples et les relations binaires.

EXEMPLE.

Le diagramme de Hasse d'une relation d'ordre sur un ensemble fini S n'est pas autre chose que le graphe associé à la relation de *successeur* : t est un successeur de s si t est plus grand que s mais qu'il n'y a aucun point strictement compris entre s et t :

$$s < t \text{ et } \text{ pour tout } u \text{ dans } S, \text{ si } s \leq u \leq t \text{ alors } u = s \text{ ou } u = t.$$

Graphe fini. On dit que le graphe G est *fini* si les deux ensembles S_G de sommets et A_G d'arêtes sont finis (possiblement vides). L'*ordre* d'un graphe fini G est le nombre de ses sommets, c'est à dire le cardinal de S_G.
À partir de maintenant tous les graphes considérés seront supposés finis.

Arête émergente, incidente. Soit G un graphe (fini donc). Si a est une arête de source s ($s_G(a) = s$), on dit que a est *émergente en* s ; si a est une arête de but s ($b_G(a) = s$) on dit que a est *incidente en* s. Si deux arêtes f et g sont émergentes en un même sommet s on dit qu'elles sont *co-émergentes* en s et si f et g ont le même but t on dit que f et g sont *co-incidentes* en t.
Une même arête peut très bien être à la fois émergente et incidente en un même sommet, auquel cas on parle de *boucle*.

Degré d'un sommet, d'un graphe. Soit G un graphe. Le *degré émergent* ou *sortant* $d_G^+(s)$ d'un sommet s de G est le nombre d'arêtes émergentes en s :

$$d_G^+(s) = |\{f \in A_G, s_G(f) = s\}|\,.$$

Le *degré incident* ou *entrant* $d_G^-(s)$ du sommet s est le nombre d'arêtes incidentes en s :

$$d_G^-(s) = |\{f \in A_G, b_G(f) = s\}|\,.$$

Enfin le *degré* $d_G(s)$ de s est la somme de son degré sortant et de son degré entrant :

$$d_G(s) = d_G^+(s) + d_G^-(s).$$

Remarquons que les arêtes qui sont à la fois émergentes et incidentes en s sont comptées deux fois.
Le degré $d(G)$ du graphe G est la somme des degrés de chacun de ses sommets.

2.1 Théorème
Soit G un graphe. Le degré de G est égal à deux fois le nombre d'arêtes de G :

$$d(G) = \sum_{s \in S_G} d_G(s) = 2\,|A_G|$$

Ce théorème a pour corollaire le théorème suivant :

2.2 Théorème
Soit G un graphe. Le nombre de sommets de G de degré impair est pair.

2.2 Graphes non orientés

Paires. Soit S un ensemble. On note $S^{(2)}$ l'ensemble des paires d'éléments de S :

$$S^{(2)} = \{\{p, q\}, p \in S, q \in S\}.$$

Autrement dit $S^{(2)}$ est l'ensemble des couples d'éléments de S (usuellement noté S^2) mais dans lequel on identifie chaque couple (s_1, s_2) avec son opposé (s_2, s_1).

Graphes non orientés. Un *graphe non orienté* G est la donnée de deux ensembles S_G et A_G et d'une fonction $\delta_G : A_G \mapsto S_G^{(2)}$.
La terminologie sur les graphes non orientés est la même que dans le cas orienté, à ceci près que les deux sommets associés à une arêtes sont ses *extrémités*, c'est à dire que l'on ne distingue plus un sommet source et un sommet but, et que l'on ne parle plus d'arêtes *émergentes* : une arête d'extrémité s est *incidente* en s.

Degré dans le cas non orienté. Si G est un graphe non orienté, le degré d'un sommet s est le nombre d'arêtes incidentes en s, en comptant pour 2 les boucles en s :

$$d(s) = |\{f \in A_G, s \in \delta_G(f) \text{ et } |\delta_G(f)| = 2\}| + 2\,|\{f \in A_G, \delta_G(f) = \{s\}\}|\,.$$

Les boucles sont comptées double afin de retrouver dans le cas non orienté, le même théorème de parité que celui des graphes orientés.

Graphe non orienté sous-jacent. Soit G un graphe (orienté donc). On construit $\bar{G}$, le *graphe non orienté sous-jacent* de G de la manière suivante : les sommets et les arêtes de $\bar{G}$ sont les mêmes que ceux de G, c'est à dire $S_{\bar{G}} = S_G$ et $1_{\bar{G}} = A_G$, et la fonction d'incidence $\delta_{\bar{G}}$ est donnée par :

$$\delta_{\bar{G}}(f) = \{s_G(f), b_G(f)\}.$$

2.3 Chemins dans les graphes

Morphismes de graphes Soient G et H deux graphes. Un *morphisme de graphe* ϕ de G dans H est une paire d'application $\phi_S : S_G \mapsto S_H$ et $\phi_A : A_G \mapsto A_H$ respectant les applications source et but, c'est à dire vérifiant :

$$s_H(\phi_A(f)) = \phi_S(s_G(f)) \text{ et } b_H(\phi_A(f)) = \phi_S(b_G(f))$$

pour toute arête f de G.

Cas non orienté. Si G et H sont maintenant des graphes non orientés, un morphisme de G dans H est une paire d'applications ϕ_S et ϕ_A compatibles avec la les fonctions d'incidence :

$$\delta_H(\phi_A(f)) = \{\phi_S(s), s \in \delta_G(f)\}$$

Remarque

> En général lorsque s est un sommet de G on note $\phi(s)$ pour $\phi_S(s)$ et lorsque f est une arête de G on note $\phi(f)$ pour $\phi_A(f)$.

Isomorphisme de graphes. Un morphisme ϕ de graphes orientés ou non est un *isomorphisme* lorsque les deux applications ϕ_S et ϕ_A sont des bijections.

Chemins, connexité, cycles

Composabilité, cas orienté. Soit G un graphe et a, b deux sommets de G. On dit que a et b sont *adjacents* si il existe une arête f reliant a à b, c'est à dire telle que $s_G(f) = a$ et $b_G(f) = b$.
Soient deux arêtes f et g de G. On dit que f et g sont *adjacentes* ou *composables* en s si le but de f et la source de g sont tous deux le sommet s : $b_G(f) = s_G(g) = s$.

Composabilité, cas non orienté. Soient maintenant G un graphe non orienté. On dit que deux sommets a et b de G sont adjacents si il existe une arête f d'extrémités a et b : $\delta_G(f) = \{a, b\}$.
Soient f et g deux arêtes ; on dit que f et g sont adjacentes ou composables en s si elles ont une extrémité commune s : $s \in \delta_G(f) \cap \delta_G(g)$.

Chemins. Soit G un graphe orienté ou pas et n un entier. Un *chemin de longueur* n dans G est une suite *alternée* $(s_0, f_1, s_1 \ldots, f_n, s_n)$ où les s_i sont des sommets de G, les f_i sont des arêtes et pour chaque $i < n$ les arêtes f_i et f_{i+1} sont composables en s_i. On dit que le chemin *va de* s_0 *à* s_n ou qu'il *relie* s_0 à s_n. On dit également que s_0 et s_n sont *connectés* et que s_n est *accessible* à partir de s_0.

Remarque

Tout sommet s est accessible à partir de lui même grâce au chemin de longueur 0 : (s)

Remarque

Dans un graphe non orienté, dès qu'il y a un chemin de a vers b, il y a aussi un chemin inverse de b vers a ; ceci est faux dans un graphe orienté.

2.3 Théorème
Soient G et H deux graphes et ϕ un morphisme de G dans H. Si deux arêtes f et g de G sont composables dans G alors $\phi(f)$ et $\phi(g)$ sont composables dans H. Si $\gamma = (s_0, f_1, \ldots, f_n, s_n)$ est un chemin de G alors $\phi(\gamma) = (\phi(s_0), \phi(f_1), \ldots, \phi(f_n), \phi(s_n))$ est un chemin de H.

Connexité. Soit G un graphe orienté ou pas. On dit G est *étoilé* si il existe un un sommet s_0 de G tel que tout sommet s de G est accessible à partir de s_0. On dit que G est *connexe* si pour tout couple (a, b) de sommets de G, il existe un chemin de a à b.

2.4 Théorème
Un graphe non orienté *est connexe ssi il est étoilé.*

Composition de chemins. Soit $\gamma = (s_0, f_1, \ldots, f_n, s_n)$ et $\delta = (t_0, g_1, \ldots, g_m, t_m)$ deux chemins dans un graphe G orienté ou pas. On dit que γ et δ sont *composables* si $s_n = t_0$, et dans ce cas on définit le chemin composé ;

$$\gamma\delta = (s_0, f_1, \ldots, f_n, s_n, g_1 \ldots, g_m, t_m)$$

L'opération (partielle) de composition de chemins ainsi définie est clairement associative et on notera $\gamma_1 \ldots \gamma_n$ la composition de n chemins γ_1, …, γ_n.

Cycles Un *cycle* ou *circuit* dans un graphe G orienté ou non est un chemin de longueur non nulle $(s_0, f_0, s_1, \ldots, f_n, s_{n+1})$ tel que :

- $s_0 = s_{n+1}$;
- pour $i = 0, \ldots, n-1$, si $f_i = f_{i+1}$ alors f_i est une boucle.

Remarque

La seconde condition, qui peut sembler un peu étrange, est là pour assurer, dans le cas d'un graphe non orienté, qu'un chemin de la forme (a, f, b, f, a) où a et b sont les extrémités de l'arête f, n'est pas un cycle. Dans le cas d'un graphe orienté, cette seconde condition est toujours réalisée et est donc inutile.

Cycles élémentaires. On appelle *cycle élémentaire* dans un graphe G (orienté ou non) un chemin de longueur non nulle de la forme $(s_0, f_0, \ldots, f_n, s_{n+1})$ où :

- $s_0 = s_{n+1}$;
- pour tout $i, j = 0, \ldots, n$ on a $s_i \neq s_j$ et $f_i \neq f_j$.

Autrement dit un cycle élémentaire est un cycle qui ne passe pas deux fois par la même arête ou le même sommet (en dehors de son sommet de départ et d'arrivée).

2.5 Théorème (Décomposition des cycles)
Soit G un graphe orienté ou non et γ un cycle de G. Si γ est élémentaire alors γ est un cycle minimal, *c'est à dire que γ ne contient aucun cycle : pour tout chemins γ_0, γ_1 et tout cycle σ, si $\gamma = \gamma_0\sigma\gamma_1$ alors γ_0 et γ_1 sont des chemins de longueurs nulles et $\gamma = \sigma$.*

Si γ n'est pas élémentaire alors il existe des chemins γ_0, ..., γ_n et des cycles σ_0, ..., σ_n tels que :

- *pour $i = 0, \ldots, n-1$, le chemin γ_i est composable avec chacun des chemins γ_{i+1} et σ_i ;*
- *le chemin composé $\gamma_0 \ldots \gamma_n$ est un cycle élémentaire ;*
- $\gamma = \gamma_0\sigma_0\gamma_1\sigma_1 \ldots \gamma_n\sigma_n$.

Remarque

Comme σ_i est un cycle et que γ_i est supposé composable avec σ_i et γ_{i+1} on voit que σ_i est également composable avec γ_{i+1}, ce qui fait que la composition de chemins ci-dessus est bien définie.

Circuit eulérien. Soit G un graphe orienté ou non. Un *circuit eulérien* dans G est une circuit dans lequel toute arête de G apparaît exactement une fois. Le fameux problème des ponts de Königsberg se ramène à celui de trouver un circuit eulérien dans le plan de la ville.

2.6 Théorème
Un graphe orienté G admet un circuit eulérien ssi G est connexe et pour tout sommet s de G on a :

$$d^+(s) = d^-(s)$$

Un graphe non orienté G admet un circuit eulérien ssi G est connexe et pour tout sommet s de G, le degré de s est pair.

Remarque

C'est en remarquant que cette dernière propriété (parité du degré des sommets) est nécessaire qu'Euler démontra l'impossibilité de résoudre le problème des ponts de Königsberg, posant ainsi la première pierre de ce qui devait devenir la théorie des graphes.

Circuit hamiltonien. Soit G un graphe orienté ou non. Un *circuit hamiltonien* dans G est une circuit dans lequel chaque sommet de G apparaît exactement une fois. Contrairement aux circuits eulériens, on ne connait pas de condition nécessaire et suffisante assurant l'existence d'un circuit hamiltonien.

2.4 Matrices d'adjacence

Numérotation des sommets. Soit G un graphe d'ordre n. On se donne une *numérotation* des sommets de G, c'est à dire une bijection de S_G dans $[1, n]$. Désormais on notera i le sommet de numéro i.

Matrice d'adjacence. La *matrice d'adjacence* est la matrice carrée A d'ordre n donnée par :

$$A_{ij} = |\{f \in A_G, \delta_G(f) = (i,j)\}|.$$

Autrement dit le coefficient sur la $i$ème ligne, $j$ème colonne est le nombre d'arêtes de source le sommet (de numéro) i et de but le sommet (de numéro) j.
Si G est non orienté, sa matrice d'adjacence est donnée par une formule similaire :

$$A_{ij} = |\{f \in A_G, \delta_G(f) = \{i,j\}\}|.$$

Dans ce cas la matrice d'adjacence est *symétrique*.

Remarque

Lorsque le graphe est simple, les coefficients de la matrice d'adjacence sont tous égaux à 0 ou 1. En général les coefficients sont des entiers positifs.

2.7 Théorème
Soit A la matrice d'adjacence du graphe G et k un entier. Le coefficient de la $i$ème ligne, $j$ème colonne de la matrice A^k est le nombre de chemins de longueur k du sommet (de numéro) i au sommet (de numéro) j.

Graphes étiquetés. Soit G un graphe et R un ensemble. On appelle *étiquetage de G par R* une application $w : A_G \mapsto R$. On dit aussi que les arêtes sont *étiquetées* ou *pesées* par les éléments de R et l'élément $w(f)$ de R associé à l'arête f est appelé l'*étiquette* ou le *poids* de l'arête f.
Si R est muni d'un produit associatif (c'est à dire si R est un monoïde, voir chapitre suivant), alors on peut étendre la notion de poids aux chemins. Soit $\gamma = (s_0, f_1, \ldots, f_n, s_n)$ un chemin dans G. Le poids de γ est donné par :

$$w(\gamma) = w(f_1) \ldots w(f_n)$$

c'est à dire que le poids de γ est le produit des poids des arêtes de γ.
Si R est un semi-anneau, c'est à dire si R dispose d'une addition commutative et d'un produit distribuant sur l'addition, par exemple si R est un des ensembles de nombres $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ ou $\mathbb{C}$, ou encore si R est un anneau de Boole, alors on peut associer au graphe étiqueté (G, w) une matrice de poids W définie, comme pour la matrice d'adjacence par :

$$W_{ij} = \sum_{f:\delta_G(f)=(i,j)} w(f)$$

et l'on trouve alors une généralisation du théorème précédent :

2.8 Théorème
Pour tout entier k, le coefficient de la $i$ème ligne, $j$ème colonne de la matrice W^k est la somme des poids des chemins du sommet i au sommet j.

Remarque

Ce théorème généralise le précédent. En effet la matrice d'adjacence de Gs'obtient en donnant le poids 1 à toutes les arêtes, et dans ce cas, le poids d'un chemin de longueur non nulle est toujours égal à 1.

Graphes probabilistes. Voici un exemple particulièrement important de graphe étiqueté : un *graphe probabiliste* est un graphe pesé par des réels positifs tel que pour tout sommet s, la somme des poids des arêtes émergentes en s est égale à 1. Le poid d'une arête f de source s est interprété comme la probabilité, lorsqu'on se trouve en s, de quitter s par l'arête f. La matrice de poids de G est une matrice probabiliste, et sa puissance k donne, pour chaque sommets i et j, la probabilité pour, partant de i, arriver en j par un chemin de longueur k.

Chapitre 3

Langages réguliers

3.1 Monoïdes

Définition

Monoïde. Un *monoïde* est un ensemble muni d'une opération binaire associative admetant un élément neutre. On utilise en général la notation multiplicative pour dénoter l'opération et l'élément neutre. Un monoïde satisfait donc les équations suivantes :

$$u1 = 1u = u,$$
$$(uv)w = u(vw).$$

Exemple.

En fait il est difficile de trouver un exemple d'une structure qui *ne soit pas* un monoïde. En effet quasiment toutes les opérations usuelles en mathématiques (addition, multiplication, composition, produit de convolution, produit vectoriel, produit tensoriel, etc.) sont associatives. Il est vrai que certaines (le produit vectoriel) n'ont pas d'élément neutre.

Pour un exemple d'opération non associative (qui ne donne donc pas la structure de monoïde) on peut prendre l'ensemble des arbres binaires finis munis de l'opération de construction d'arbre : étant donnés deux arbres a et b, on note $a.b$ l'arbre dont le fils gauche est a et le fils droit est b. Cette opération n'est pas associative (mais elle a un élément neutre, à savoir l'arbre vide).

Produits d'arité quelconque. Soit M un monoïde et n un entier. On définit par récurrence sur n le *produit d'arité* n dans M qui est une fonction de M^n dans M,

$$(u_1, \ldots, u_n) \mapsto \prod_{i=1}^{n} u_i.$$

- Si $n = 0$ alors il n'y a aucun u_i et le produit vide est égal à 1 ;
- sinon $\prod_{i=1}^{n+1} u_i = \left(\prod_{i=1}^{n} u_i\right) u_{n+1}$.

On voit facilement que l'associativité s'étend aux produits généraux, c'est à dire que pour tous entiers n, p et tous $n+p$-uplets $(u_1, \ldots, u_n, u_{n+1}, \ldots, u_{n+p})$ d'éléments de M, on a :

$$\left(\prod_{i=1}^{n} u_i\right)\left(\prod_{i=n+1}^{n+p} u_i\right) = \prod_{i=1}^{n+p} u_i.$$

On notera $u_1 \ldots u_n$ le produit d'arité n des u_i. Si tous les u_i sont égaux à un même u, on note u^n leur produit.

Morphismes de monoïdes. Soient M et P deux monoïdes. Un *morphisme* de M dans P est une application $\varphi : M \mapsto P$ vérifiant :

$$\varphi(uv) = \varphi(u)\varphi(v)$$

pour tous éléments u et v de M. Si l'application est bijective on dit que φ est un *isomorphisme* et on vérifie facilement que l'application inverse est également un morphisme.

Congruences. Soit M un monoïde. Une *congruence* sur M est une relation d'équivalence $\equiv$ vérifiant

$$\text{si } u \equiv u' \text{ et } v \equiv v' \text{ alors } uv \equiv u'v'$$

3.1 Théorème
Soit M et P deux monoïdes et $\varphi : M \mapsto P$ un morphisme de monoïde. Alors la relation binaire définie sur M par

$$u \equiv v \text{ ssi } \varphi(u) = \varphi(v)$$

est une congruence.

3.2 Le monoïde des mots

Alphabet. On se donne un ensemble Σ non vide que l'on appelle *l'alphabet*. En général Σ est choisi fini mais ça n'est pas obligatoire. Les éléments de Σ sont appelés des *lettres* et on les notera a, b, c.

Mots. Un *mot* m sur l'alphabet Σ est une suite finie d'éléments de Σ, c'est à dire la donnée de :

- un entier $|m|$;
- une application $\sigma_m : \{1, \ldots, |m|\} \mapsto \Sigma$.

L'entier $|m|$ est appelé la *longueur* du mot m. Dans le cas où $|m|$ est nul, on dit que m est *le mot vide*. On note ε ou 1 le mot vide.

Les lettres sont des mots. Dans le cas où $|m| = 1$, le mot m détermine une unique lettre $a = \sigma_m(1)$. Réciproquement pour toute lettre a de Σ, il existe un unique mot m de longueur 1 tel que $\sigma_m(1) = a$. On a donc une correspondance biunivoque entre les lettres de Σ et les mots de longueur 1 de Σ^*. On notera a le mot de longueur 1 constitué de la seule lettre a.

Concaténation de mots. On note Σ^* l'ensemble des mots sur l'alphabet Σ. Soient u et v deux mots. On appelle *concaténation* de u et v le mot noté uv défini par :

- $|uv| = |u| + |v|$;
- $\sigma_{uv}(i) = \begin{cases} \sigma_u(i) & \text{si } 1 \leq i \leq |u|, \\ \sigma_v(i - |u|) & \text{si } |u| + 1 \leq i \leq |u| + |v|. \end{cases}$

3.2 Théorème
Muni de l'opération de concaténation, l'ensemble Σ^* des mots sur l'alphabet Σ est un monoïde d'élément neutre le mot vide ε.

Autrement dit la concaténation des mots est un produit associatif.

Les mots sont des produits de lettres. Si l'on combine la notation de produit n-aire des monoïdes et notre convention d'identifier une lettre a avec le mot de longueur 1 constitué de la seule lettre a, on s'autorise à écrire :

$$u = a_1 \dots a_n$$

dès que u est un mot de longueur n dont les lettres sont successivement a_1, ..., a_n, c'est à dire tel que $\sigma_u(i) = a_i$ pour $i = 1, \dots n$. En particulier si tous les a_i sont égaux à une même lettre a de Σ, on écrira $u = a^n$.

Préfixes, suffixes. Soit u un mot de longueur n. Un *préfixe* de u est un mot v de longueur $p \leq n$ qui est égal à un début de u, c'est à dire tel que pour tout $i = 1, \dots, p$ on a $\sigma_v(i) = \sigma_u(i)$. Un *suffixe* de u est un mot v de longueur $p \leq n$ qui est égal à une fin de u c'est à dire tel que $\sigma_v(i) = \sigma_u(n - p + i)$ pour $i = 1, \dots, p$. Alternativement on peut définir un préfixe de u comme un mot v tel qu'il existe un mot w vérifiant $u = vw$. De même un suffixe de u est un mot w tel qu'il existe un préfixe v de u vérifiant $u = vw$. Remarquons que la relation préfixe « u est un préfixe de v » est une relation d'ordre sur les mots : en effet elle est clairement réflexive et transitive et si u est un préfixe de v et v est un préfixe de u alors $u = v$.

3.3 Langages réguliers

Dans cette section on se donne un alphabet fini non vide Σ.

Langages. Un *langage* sur l'alphabet Σ est un sous-ensemble du monoïde Σ^*.
L'un des problèmes principaux de la théorie des langages est de répondre à la question : étant donné un langage L sur un alphabet Σ, existe-il une méthode pour déterminer si un mot u appartient à L ou non ? Une telle méthode s'appelle une *méthode de décision* de L. On peut faire une classification des langages en caractérisant leur méthode de décision : typiquement la classe des langages *indécidables* est celle des langages qui n'ont pas de méthode de décision algorithmique (implémentable sur un ordinateur).
À l'autre bout de l'échelle, les langages réguliers que nous allons maintenant définir forment la classe de langages la plus simple au sens où il existe un procédé algorithmique très simple permettant de résoudre le problème du mot pour tout langage régulier.

Composition. Soient L et L' deux langages. On note LL' le langage *composé* de L et L' défini par :

$$LL' = \{uv, u \in L, v \in L'\}$$

On notera L^k le langage L composé k fois avec lui-même :

$$L^k = \underbrace{L \dots L}_{k\times} = \{u_1 \dots u_k \text{ tel que } u_i \in L \text{ pour } i = 1, \dots, k\}$$

En particulier $L^0 = \{\varepsilon\}$ et $L^1 = L$.

Répétition. Soit L un langage. On note L^* le langage

$$L^* = \bigcup_{k \geq 0} L^k = \{u_1 \dots u_k, k \in \mathbb{N}, u_1, \dots, u_k \in L\}$$

Exemple.

Prenons $L = \{a\}$, où a est une lettre de Σ. Alors $L^* = \{\varepsilon, a, aa, aaa, \ldots\} = \{a^n, n \geq 0\}$.
Si on prend $L = \Sigma$ alors $\mathcal{L}^*$ est justement l'ensemble Σ^*.

Langages réguliers. La classe des *langages réguliers* sur Σ est le plus petit ensemble R de langages sur Σ tel que :

- R contient le langage vide $\emptyset$, et les langages $\{a\}$ pour chaque lettre $a \in \Sigma$;
- si L et L' sont deux langages appartenant à R alors le langage composé LL' appartient également à R ;
- si L et L' sont deux langages de R alors $L \cup L'$
- si L est un langage de R alors L^* est un langage de R.

3.3 Théorème
Si L est un langage fini alors L est régulier.

Exemple.

Si a et b sont deux lettres distinctes de Σ alors $\{ab\}$ est régulier puisque c'est la composition des langages $\{a\}$ et $\{b\}$ qui sont tous deux réguliers. De même $\{ab, bab\}$ est régulier puisque c'est la réunion du précédent et de la composition des langages $\{b\}$, $\{a\}$ et $\{b\}$.
L'ensemble Σ, considéré comme ensemble de mots de longueur 1 est régulier puisqu'il est la réunion (finie car Σ est supposé fini) des ensembles $\{a\}$ pour chaque $a \in \Sigma$. Par conséquent le monoïde des mots Σ^* est également régulier.
L'ensemble $\Sigma^{(2)}$ des mots de longueur 2 est régulier car il est fini (puisque Σ est fini). Par conséquent l'ensemble des mots de longueur paire est régulier car c'est $\left(\Sigma^{(2)}\right)^*$.
Il y a bien entendu des langages qui ne sont pas réguliers. Par exemple, soient a et b deux lettres distinctes de Σ ; alors le langage $\{a^k b^k, k \geq 0\}$ n'est pas régulier ; la démonstration se trouve après le lemme de pompage ci-dessous.

Expressions régulières. Les *expressions régulières* sont un système de notation pour décrire les langages réguliers. On utilise les notations définies ci-dessus sauf que l'on supprime l'usage des accolades : un mot u dénote le langage singleton $\{u\}$ et un langage fini $\{u_1, \ldots, u_n\}$ se dénote $u_1 \cup \cdots \cup u_n$ (la réunion des langages singleton contenant chaque mot u_i). On utilise souvent le symbole $+$ ou le symbole $|$ au lieu de $\cup$: $u_1 + \cdots + u_n$.

Exemple.

Supposons que $\Sigma = \{a_1, \ldots, a_n\}$. On a vu que, en considérant chaque lettre comme un mot de longueur 1, Σ est un langage régulier. Une expression régulière dénotant Σ est $a_1 + \cdots + a_n$.
Le monoïde Σ^* peut se dénoter par l'expression régulière $(a_1 + \cdots + a_n)^*$. Le langage des mots de longueur paire peut se dénoter $((a_1 + \cdots + a_n)(a_1 + \cdots + a_n))^*$. Le langage des mots contenant un mot u donné peut s'écrire $(a_1 + \cdots + a_n)^* u (a_1 + \cdots + a_n)^*$.

3.4 Automates finis

Questions sur les langages réguliers. Le problème de la définition de langages réguliers que nous venons de donner est qu'elle n'est pas très opératoire, c'est à dire qu'elle ne permet pas de répondre facilement à des questions simples sur les langages réguliers. Voici un liste de questions relativement naturelles que l'on peut se poser à propos des langages réguliers et pour lesquelles la définition ci-dessus ne propose pas de réponse immédiate :

- soient L un langage régulier ; son complémentaire L^c dans Σ^* est-il régulier ?
- soient L et L' deux langages réguliers ; leur intersection $L \cap L'$ forme-t-elle un langage régulier ?

– existe-il une procédure simple résolvant le problème du mot pour un langage régulier donné ?

À toutes ces questions la réponse est oui, mais pour le voir il faut faire un petit détour par les automates.

Définition. Un *automate fini* A est un graphe orienté étiqueté par des mots avec un sommet distingué et un ensemble de sommets distingués. Plus précisément un automate A est la donnée d'un quadruplet (G_A, w_A, i_A, F_A) où :

– G_A est un graphe orienté fini ;

– w_A est une fonction d'étiquetage des arêtes de G_A par des mots de Σ^* ;

– i_A est un sommet de G_A ;

– F_A est un ensemble de sommets de G_A.

Les sommets de G_A sont appelés les *états* et les arêtes sont appelées les *transitions* de l'automate. Le sommet i_A est appelé l'état *initial* et les sommets appartenant à F_A sont les états *finaux*.
Lorsque l'on dessine les automates, on représente les états par des ronds, on dispose le mot étiquetant une arête à côté de l'arête, on met une petite flèche entrant vers l'état initial et une petite flèche sortante sur chaque état final.

Langage reconnu par un automate. Soit A un automate fini. Comme Σ^* est un monoïde et que G_A est étiqueté par des mots de Σ^*, on peut associer à tout chemin γ de G_A un mot $w_A(\gamma)$. On note L_A le langage des chemins de l'état initial de A à l'un de ses états finaux :

$$L_A = \{w_A(\gamma), \gamma \text{ est un chemin de } i_A \text{ à } s \in F_A\}.$$

Le langage L_A est appelé le *langage de* A ; on dit aussi que A *reconnait* le langage L_A. Déterminer si un mot u appartient à L_A revient donc à construire un chemin dans G_A de source i_A, de but l'un des sommets appartenant à F_A et dont le poids est u. Si un tel chemin existe, on dit que A reconnait u.

Langages automatiques. Un langage L sur l'alphabet Σ est *automatique* si il existe un automate fini A tel que $L = L_A$.

Remarque

En général il existe plusieurs automates qui reconnaissent le même langage.

3.4 Théorème
La classe des langages automatiques contient tous les langages finis, est close par réunion, composition et répétition.
Par conséquent tout langage régulier est automatique.

Preuve. Le langage vide est reconnu par n'importe quel automate dont l'ensemble d'états finaux est vide.

Soit $u = a_1 \dots a_n$ un mot de longueur n. On construit un automate U qui reconnait uniquement le mot u, c'est à dire le langage singleton $\{u\}$: G_U est une chaîne de $n+1$ sommets, c'est à dire le graphe à $n+1$ sommets $s_1, \dots, s_{n+1}$ et n arêtes $f_1, \dots, f_n$. Pour $i = 1, \dots, n$, la source et le but de f_i sont respectivement s_i et s_{i+1}. L'étiquette de f_i est la lettre a_i (mot de longueur 1). Le sommet initial est s_1 et il n'y a qu'un sommet final : s_{n+1}.

Soit A et A' deux automates. On construit un automate B pour $L_A \cup L_{A'}$: le graphe G_B est la réunion des deux graphes G_A et $G_{A'}$ à laquelle on ajoute un sommet i_B et deux arêtes f et f' : la source de f et de f' est le sommet i_B, le but de f est le sommet initial de A, le but de f' est le sommet initial de A', et les deux arêtes sont étiquetés par le mot vide. Le sommet initial de B est i_B et les sommets finaux sont ceux de A et ceux de A'.

Ainsi on peut obtenir un automate pour chaque langage fini puisque un langage fini est un ensemble fini de mots, c'est à dire une réunion finie de singletons.

On construit de même un automate C pour $L_A L_{A'}$ en prenant la réunion des deux graphes mais on n'ajoute aucun sommet ; par contre on ajoute une arête issue de chaque sommet final de A dont le but est le sommet

initial de A' et l'étiquette est le mot vide. Le sommet initial de B est celui de A et les sommets finaux de B sont ceux de A'.

Enfin on construit un automate D pour L_A^* en prenant le graphe G_A auquel on ajoute une arête issue de chaque sommet final de A de but le sommet initial de A et étiquetée par le mot vide. Le sommet initial de D est celui de A et les sommets finaux de D sont ceux de A.

La classe des langages automatiques est donc close par réunion, composition et répétition : elle contient donc la classe des langages réguliers puisque c'est la plus petite classe close par ces opérations.

Langages automatiques et langages réguliers. La réciproque du théorème précédent est également vraie.

3.5 Théorème
Si L est un langage automatique alors L est régulier.

Preuve. Dans ce qui suit on dira qu'un chemin γ passe par un sommet s si γ contient s ailleurs qu'en première ou dernière position.

Soit A un automate reconnaissant le langage L. Étant donné un ensemble S non vide de sommets de A et deux sommets (possiblement égaux) s et s' appartenant ou pas à S, on note $L(s, s', S)$ le langage des chemins de s à s' passant uniquement par des sommets de S :

$$L(s, s', S) = \{w_A(\gamma), \gamma = (s_0, f_1, s_1, \ldots, f_n, s_n), s_0 = s, s_n = s' \text{ et } s_i \in S \text{ pour } i = 1, \ldots, n-1\}$$

On va montrer le lemme

3.6 Lemme
Pour tout s, s' et S, le langage $L(s, s', S)$ est régulier.

On en déduit immédiatement le résultat car on a $L = \bigcup_{s \in F_A} L(i_A, s, S_A)$ où S_A est l'ensemble de tous les états de A ; L est donc un réunion finie de langages réguliers, c'est donc un langage régulier.

Pour montrer le lemme, on raisonne par récurrence sur $|S|$. Supposons $|S| = 0$. Un chemin de s à s' et ne passant que par les sommets de S, c'est à dire par aucun sommet, est forcément de la forme (s, f, s') ou f est une arête de s à s'. Donc $L(s, s', S) = \{w_A(f), f \text{ est une arête de } s \text{ à } s'\}$ est un ensemble fini et est donc régulier.

Supposons maintenant que $|S| \geq 1$. Alors il existe un sommet t dans S. Tout chemin γ de s à s' ne passant que par S peut se décomposer en $\gamma = \gamma_0\gamma_1 \ldots \gamma_p$ où γ_0 est un chemin de s à t ne passant pas par t, chaque γ_i pour $i = 1, \ldots, p-1$ est un chemin de t à t ne passant pas par t, et γ_p est un chemin de t à s' ne passant pas par t. Notons S^- l'ensemble S auquel on a enlevé le sommet t. De ce qui précède on déduit que :

$$L(s, s', S) = L(s, t, S^-)L(t, t, S^-)^* L(t, s', S^-)$$

Comme $|S^-| = |S| - 1$, on peut supposer par récurrence que chacun des trois langages du membre droit sont réguliers, d'où l'on déduit que $L(s, s', S)$ est régulier.

3.5 Automates déterministes.

Les automates que l'on a vus jusqu'à maintenant sont *non déterministes* au sens où, étant donné un mot u, il existe en général plusieurs sommets que l'on peut atteindre en partant du sommet initial par un chemin de poids u. Par conséquent, lorsque l'on essaye de reconnaître u en construisant un chemin de poids u issu du sommet initial, on peut se tromper et arriver dans un état non final, alors qu'une solution existe ; il faut donc recommencer jusqu'à ce que, soit on trouve une solution, soit on ait épuisé toutes les possibilités sans trouver de solution ; accessoirement il n'est pas évident (même si c'est vrai), que l'on ne va pas chercher indéfiniment.

Un automate *déterministe* est un automate A tel que :

- les arêtes sont étiquetées par des lettres, c'est à dire : pour toute arête f de G_A, $w_A(f)$ est une lettre (un mot de longueur 1) ;

– la fonction w_A d'étiquetage est injective sur les arêtes issues d'un même sommet, c'est à dire : pour toutes arêtes f et f', si $s_{G_A}(f) = s_{G_A}(f')$ et si $w_A(f) = w_A(f')$ alors $f = f'$.

L'automate est *complet* si on ne reste jamais bloqué en un sommet faute de lettre, c'est à dire : pour tout sommet s de G_A et toute lettre a de Σ, il existe une arête f de source s telle que $w_A(f) = a$.

3.7 Théorème
Soit A un automate déterministe. Pour tout mot u et tout état s de A il existe au plus un chemin dans G_A de source s et de poids u. Si de plus A est complet alors il existe exactement un chemin issu de s de poids u.

Déterminisation. Le problème du mot devient plus facile à résoudre avec un automate déterministe : contrairement au cas non déterministe, si une solution existe elle est unique donc on la trouve du premier coup et l'on peut donc répondre en une seule tentative. C'est ce qui explique, entre autres raisons, l'importance du théorème :

3.8 Théorème (Déterminisation)
Soit A un automate. Il existe un automate déterministe D qui reconnait le même langage que A, c'est à dire tel que $L_A = L_D$.
Soit D un automate déterministe ; il existe un automate complet C qui reconnait le même langage que D.

Preuve. La deuxième partie du théorème est assez facile : on construit C en ajoutant un nouveau sommet t et pour chaque lettre a de Σ, une boucle en t étiquetée par a. Une fois ceci fait, à chaque sommet s auquel il manque une arête étiquetée par une lettre a, on ajoute un arête de s vers t étiquetée par a.

La déterminisation de A est un petit peu plus compliquée. Tout d'abord on remplace A par un automate dont toutes les arêtes sont étiquetées par des mots de longueur au plus 1 : pour cela il suffit de remplacer toute arête f étiquetée par un mot u de longueur supérieure à 2 par une chaîne issue de la source de f, arrivant au but de f et où les arêtes sont étiquetées par les lettres successives de u.

On suppose maintenant que toutes les arêtes de A sont étiquetées par des mots de longueur au plus 1, c'est à dire par le mot vide ou par une lettre. Soit s un sommet de G_A et u un mot ; on note s^u l'ensemble des sommets t de G_A tels qu'il existe un chemin de s à t de poids u. Si S est un ensemble de sommets de G_A, on note S^u la réunion des s^u pour tous les $s \in S$: $S^u = \bigcup_{s \in S} s^u$.

Le lemme suivant nous sera utile :

3.9 Lemme
Soit s un sommet de A, u_1 et u_2 deux mots ; alors on a

$$s^{u_1 u_2} = \bigcup_{t \in s^{u_1}} t^{u_2} = (s^{u_1})^{u_2}$$

Par conséquent si $u = a_1 \dots a_n$ alors

$$s^u = (\dots (s^{a_1})^{a_2} \dots)^{a_n}$$

L'automate D est construit de la manière suivante : ses sommets sont les ensembles de sommets de A. Soit S un ensemble de sommets de G_A, c'est à dire un sommet de G_D. Pour chaque lettre a de Σ on ajoute à D une arête étiquetée par a de source S et de but S^a. L'état initial de D est le singleton contenant l'état initial de A et les états finaux sont tous les ensembles de sommets contenant au moins un état final de A.

L'automate D ainsi construit est clairement déterministe. Le fait qu'il reconnait le même langage que A est une conséquence immédiate du lemme.

Remarque

La procédure de déterminisation décrite produit un automate dont le nombre d'états est 2^n où n est le nombre d'états de A. On peut souvent simplifier l'automate résultant, en particulier parceque le graphe obtenu est en général non connexe (au sens orienté) et que l'on peut alors supprimer toutes les composantes connexes qui ne contiennent pas l'état initial. Toutefois, même après ces simplifications, l'automate déterministe est en général de grande taille par rapport à l'automate de départ.

3.10 Théorème

La classe des langages automatiques est close par complémentaire et intersection.

Preuve. Soit A un automate, D un automate déterministe complet reconnaissant L_A : l'automate reconnaissant le complémentaire de L_A est tout simplement l'automate D mais où les sommets finaux sont exactement ceux qui ne sont pas finaux dans D. Remarquons que l'hypothèse de complétude est cruciale pour obtenir le résultat.

Puisque le complémentaire d'un langage automatique est automatique, et que la réunion de deux langages automatiques est automatique, on en déduit que l'intersection de deux langages automatiques est automatique. On peut faire un peu mieux en donnant une construction explicite de l'automate I reconnaissant $L_A \cap L_{A'}$, en supposant que A et A' sont déterministes : les sommets du graphe G_I sont les couples d'un sommet de G_A et d'un sommet de $G_{A'}$. On met une arête étiquetée par a entre deux sommets (s, s') et (t, t') ssi il y a une arête étiquetée par a entre s et t dans G_A et une arête étiquetée par a entre s' et t' dans $G_{A'}$. Le sommet initial de I est le couple formé des sommets initiaux de A et A' et les sommets finaux de I sont les couples formés d'un sommet final de A et d'un sommet final de A'.

Lemme de pompage. On va terminer cette section avec un résultat simple permettant dans beaucoup de cas de déterminer qu'un langage n'est pas automatique, et donc pas régulier.

3.11 Lemme (Lemme de pompage)

Soit L un langage automatique infini. Il existe un entier N tel que pour tout mot u de L de longueur plus grande que N, on peut trouver des mots v, p et w tels que :

- $u = vpw$;
- $|p| > 0$;
- $|vp| \leq N$;
- *pour tout entier $k \geq 0$, le mot $vp^k w$ appartient à L.*

Preuve. Soit A déterministe qui reconnait L. On prend pour N le nombre d'états de A plus 1. Ainsi tout chemin γ de longueur supérieure à N repasse au moins deux fois dans un même état. Soit u un mot de L de longueur supérieure à N. Comme l'automate reconnait u, il existe un chemin γ issu du sommet initial et arrivant à un sommet final et tel que $w_A(\gamma) = u$. Mais comme A est déterministe on voit que la longueur de γ est égale à celle de u, donc est supérieur à N.

Soit s le premier état par lequel γ passe deux fois ; alors γ se décompose en $\gamma = \gamma_0\gamma_1\gamma_2$ où γ_0 va de i_A à s et ne passe pas par s, γ_1 est un cycle élémentaire en s. En particulier la longueur de $\gamma_0\gamma_1$ est inférieure à N. Si on prend pour u, p et w les poids respectifs de γ_0, γ_1 et γ_2, on a réalisé toutes les propriétés du lemme. En particulier le « pompage » vient du fait que γ_1 est un cycle.

Remarque

Le terme « pompage » est une traduction de l'anglais « pumping lemma » qui fait référence à l'idée que l'on peut « pomper » le sous-mot p.

Exemple.

On peut facilement voir que le langage $L = \{a^k b^k, k \geq 0\}$ n'est pas automatique, et donc pas régulier non plus. Supposons, pour la contradiction, que si. Soit N l'entier fournit par le lemme de pompage et considérons le mot $u = a^N b^N$ qui est dans L. D'après le lemme, comme $|u| \geq N$, on peut écrire $u = vpw$ où $|vp| \leq N$, $|p| > 0$ et $vp^k w$ est dans L pour tout k. Posons $n = |vp|$. Comme $n \leq N$ et $u = a^N b^N$, on a $vp = a^n$, $w = a^{N-n} b^N$; en particulier $p = a^m$ pour un m non nul. C'est à dire que $u = a^{n-m} a^m a^{N-n} b^N$ et que l'on devrait avoir $a^{n-m}(a^m)^k a^{N-n} b^N = a^{n-m} a^{mk} a^{N-n} b^N \in L$ pour tout k ce qui est faux dès que $k \neq 1$ puisqu'alors le nombre de a est $n-m+mk+N-n = N+m(k-1)$ qui est différent de N, le nombre de b.