

## Au-delà de l'algèbre de Boole, avec l'arithmétique modulo 2 par Gérard Pinson

C'est en 1980 lors d'un colloque tenu à l'Université Paul Valéry de Montpellier que nous avons initié une réflexion sur les relations entre « la pensée » on parle aujourd'hui de « processus cognitifs » et les caractéristiques quantitatives et qualitatives de l'information. Nous cherchions à étendre la portée de ce concept, tel qu'on le connaît dans différents domaines techniques comme la Théorie de l'Information (ex. : compression d'information), le Traitement des Signaux (holographie notamment) ou l'Intelligence Artificielle (en liaison avec les neurosciences), au contenu significatif d'information que comporte tout message traité par un processus supposé « intelligent ». Il est apparu que cette extension du concept technique numérique d'information au champ sémantique formel nécessitait une théorisation des états informationnels transmis ou non-transmis à la lumière de différents formalismes. Avec, pour conséquence, la classification de certains processus cognitifs en fonction des structures de l'information, et réciproquement la caractérisation de méthodes de traitement de l'information adaptées à chaque classe. Dans les formalismes employés pour construire cette théorie cognitive de l'information, il apparut notamment que l'opérateur « OU Exclusif » jouait un rôle central (voir figure 2 dans l'article). Aussi fut-ce avec un grand intérêt que je pris connaissance d'un article de Gérard Langlet (« La propagation asymétrique de la parité, enjeu de l'abandon du continu ? », Bio-Math, n°115, p 5-38, 1991) énonçant des idées similaires sur cet opérateur et sur la compression d'information en général. Je pris contact avec l'auteur, qui me fit découvrir ses travaux, puis ceux de Michael Zaus, ainsi que le langage APL que je ne connaissais pas et... Les Nouvelles d'APL. Comme je l'explique dans l'article ci-joint, si la physique des processus cognitifs passe par une première description en terme de lois physico-mathématiques analogiques (par ex. : diffraction, interférences, transformation de Fourier, etc.), sa modélisation sur un ordinateur implique l'adaptation de ces lois au monde binaire. C'est précisément en ce point qu'interviennent les travaux remarquables de Gérard Langlet, qui a poursuivi le raisonnement jusqu'à son terme et débouché sur l'arithmétique modulo 2 matricielle. Comme il le disait lui-même, les caractéristiques tout-à-fait particulières du langage APL ont joué un rôle décisif. C'est avec un vif regret que j'ai appris la disparition de Gérard Langlet. On comprendra, au vu de cette brève relation historique, combien cette disparition apparaît prématurée, et combien son enthousiasme et ses connaissances encyclopédiques font aujourd'hui défaut. Cela va de soi, cet article lui est dédié.

Gérard PINSON, ENSERG-INPG Professeur Agrégé de Sc. Physiques,

Lycée La Martinière-Terreaux,

18 place Gabriel Rambaud,

69283 Lyon Cedex 01.

### Résumé

- 1) Les processus informationnels associent implicitement à l'information digitalisée qu'ils traitent une information formelle continue. Il existe ainsi plusieurs classes de calculabilité : automates finis, machines de Turing (infini dénombrable), opérateurs continus (infini non dénombrable).
- 2) La classe des opérateurs continus est irréductible à la classe des opérateurs numériques. Elle met en œuvre des procédures (Transformation de Fourier, convolution,...) qui agissent sur des formes (signaux, vecteurs, ...) et non sur des nombres (scalaires). Cela ouvre des perspectives de calcul nouvelles : non commutativité, modalité, existence de points fixes, logique multivaluée...
- 3) La simulation sur un ordinateur du calcul formel continu consiste à implémenter ces opérateurs de transformation des signaux sous une forme matricielle. La précision requise pour exécuter de longues

chaînes de calcul impose l'usage de transformations algébriques en arithmétique modulo  $p$ . Cela se traduit au niveau des dispositifs booléens par le choix de  $p = 2$ . On rejoint ainsi les conclusions que Langlet et Zaus ont largement développées dans cette revue.

## 1 Introduction

Les ordinateurs sont des automates finis. Ils travaillent sur des données booléennes, possèdent une mémoire finie, et sont capables à tout instant de calculer leur état futur à l'aide de fonctions combinatoires booléennes, à partir de leur état présent et des entrées.

Les machines de Turing sont des machines abstraites travaillant sur des nombres entiers. L'ensemble de leurs états internes est lui aussi fini, mais comme ces machines travaillent sur des entiers aussi grands que l'on veut, leur mémoire (un "ruban" d'écriture-lecture) n'est pas limitée, et est donc infinie. L'intérêt du concept de machine de Turing réside dans le fait que les fonctions calculables par une procédure effective sont les fonctions calculables par une machine de Turing (thèse de Turing-Church). Mais ces machines abstraites ne sont pas réalisables, les ordinateurs les "simulant" en imposant une limite inévitable à la quantité de mémoire disponible.

Des considérations théoriques issues de la théorie de l'information et de la théorie de la calculabilité tendent à mettre en évidence l'existence d'une autre limitation fondamentale du calcul abstrait (Pinson 1995, 1996, 1997). Cette limitation réside à la frontière entre les entiers et les réels, c'est-à-dire entre les ensembles dénombrables et ceux qui ont la puissance du continu. Elle conduit à imaginer l'existence d'opérateurs continus, qui généralisent les opérateurs scalaires booléens ou arithmétiques en étendant considérablement leurs possibilités de calcul. Leur simulation dans un automate fini suggère une architecture matérielle et logicielle essentiellement matricielle et arithmétique (modulo 2).

Le but de cet article est d'exposer ces principes de calcul, issus d'une classification détaillée des différents types de calculabilité.

## 2 Entropie et complexité algorithmique

Une source  $\mathcal{S}$ , sans mémoire, générateur d'ÉvÈnements ou états aléatoires, est un ensemble dénombrable de messages notés  $s_i$ . Soit  $p$  la distribution de probabilité de ces messages. On appelle  $H(s) = -\sum_i p(s_i) \log p(s_i)$  la *quantité d'information* moyenne par objet de la source ou *entropie statistique*<sup>[1]</sup>.

Soit  $\mathbf{B} = \{0,1\}$  l'alphabet binaire. On note  $\mathbf{B}^*$  l'ensemble des mots  $w$  sur cet alphabet. Soit  $\varepsilon$  le mot vide. Il existe une correspondance bijective entre  $\mathbf{B}^*$  et  $\mathbf{N}$ , qui est l'ordre lexicographique :

$(\varepsilon, 0), (0,1), (1,2), (00,3), (01,4), (10,5), (11,6), (000,7), (001,8), \dots$

L'encodage d'une source  $\mathcal{S}$  est une application de  $\mathcal{S}$  dans un sous-ensemble  $\mathcal{W}$  de  $\mathbf{B}^*$ . Le code est dit régulier si cette application est injective et déchiffrable si deux messages distincts sont codés par deux mots différents. Si on impose au canal de communication de ne pouvoir être lu que *séquentiellement* (ce qui est la traduction formelle de l'*irréversibilité du temps* : au fur et à mesure que le récepteur reçoit les données, il lui est impossible de "remonter le temps" pour concaténer les données reçues dans un ordre autre que celui du texte), alors le code est *instantané*, appelé encore "code auto-délimité" ou "code irréductible".

Soit  $p(s_i)$  la probabilité du message  $s_i$  auquel correspond le mot code  $w_i$ . Soit  $\langle l \rangle = \sum_i p(s_i) l(w_i)$  la longueur moyenne des mots. Le théorème de la voie sans bruit (1<sup>er</sup> théorème de Shannon) montre qu'il existe un code irréductible pour lequel  $\langle l \rangle$  est aussi voisine qu'on veut de sa borne inférieure, égale à l'entropie de la source dans le cas d'un alphabet binaire :  $\langle l \rangle \geq H(s)$ .

La théorie algorithmique de l'information précise la notion d'information à partir de celle de calcul effectif : la complexité d'un objet, c'est-à-dire l'information qu'il contient, est la taille minimale de sa description algorithmique (développée initialement par Kolmogorov, Solomonoff et Chaitin, cette théorie

est présentée par exemple dans Li et Vitányi 1997). Il s'agit d'une théorie de l'information liée à un objet particulier, et complète ainsi la théorie statistique de la communication, élaborée par Shannon, qui est une théorie probabiliste liée à un ensemble d'objets. Ces deux théories sont complémentaires. Soit  $\mathcal{S}$  un ensemble énumérable d'objets  $s$ , et  $\mathcal{W}$  un ensemble énumérable d'objets  $w$ . Soit  $l$  une fonction de volume définie de  $\mathcal{W}$  dans  $\mathbf{N}$  (par exemple le nombre de caractères). Un mode de description ou code est un ensemble récursivement énumérable  $E \subseteq \mathcal{W} \times \mathcal{S}$ , et  $\langle w, s \rangle \in E$  est un encodage de  $s$ . La complexité de  $s$  selon  $E$  est :

$$C_E(s) = \min\{l(w) : \langle w, s \rangle \in E\}$$

Par convention,  $C_E(s) = \infty$  si une telle description n'existe pas.

Comment réaliser cet encodage ? L'énumération effective des machines de Turing  $T_1, T_2, \dots$  détermine l'énumération effective des fonctions récursives partielles  $\Phi_1, \Phi_2, \dots$  telles que  $\Phi_i$  est la fonction calculée par  $T_i$  pour tout  $i$ . Il existe une fonction récursive partielle universelle  $\Phi_0$  calculée par une machine de Turing universelle  $U$  telle que :

$$\Phi_0(E(np)) = \Phi_n(p)$$

où  $E(np)$  désigne un encodage auto-délimité de la  $n^{\text{ième}}$  machine de Turing  $T_n$  exécutant le programme  $p$ . La complexité d'un objet  $s \in \mathcal{S}$  est alors la longueur minimale du programme  $p$  appliqué au calculateur  $\Phi_n$  qui produit  $s$  :

$$C_{\Phi_n}(s) = \min\{l(p) : \Phi_n(p) = s\}$$

On montre (théorème fondamental de cette théorie, dit théorème d'invariance) que la fonction  $\Phi_0$  est telle que :

$$C_{\Phi_0}(s) \leq C_{\Phi_n}(s) + c_{\Phi_n}$$

La complexité d'une chaîne ne dépend donc pas de la machine utilisée, à un facteur constant près. Une machine universelle étant donnée, chaque machine de Turing particulière est assimilable à un programme exécutable. Alors la complexité d'un objet est la longueur du plus petit programme qui peut produire cet objet. Dans le cas d'une chaîne aléatoire, la longueur du programme qui la code est de l'ordre de grandeur de la chaîne elle-même (c'est le programme PRINT  $s$ , dont le nombre de bits est au moins égal à  $l(s)+1$ ).

### 3 Quatre entropies algorithmiques

Il y a quatre possibilités pour décrire un objet dans l'ensemble  $\mathcal{W} \times \mathcal{S}$  [Uspensky, 1992] :

1-Complexité simple, ou "naïve", qui correspond historiquement à la première théorie algorithmique de l'information :  $C_C(s) = C_\phi(s) = \min\{l(p) : \phi(p) = s\}$ . On n'impose ni à  $p$  ni à  $s$  d'être auto-délimités.

Cela revient à décrire un objet quelconque pris dans un ensemble similaire à  $\mathbf{N}$  par une chaîne appartenant à un langage de même structure : la description d'un nombre est un nombre.

2-Complexité uniforme :  $C_K(s) = C(s; l(s))$  qui décrit un objet auto-délimité (donc de longueur connue) par un entier : la description d'un mot est un nombre. Cela correspond à ce que l'on entend habituellement par "décodage" ou "décision".

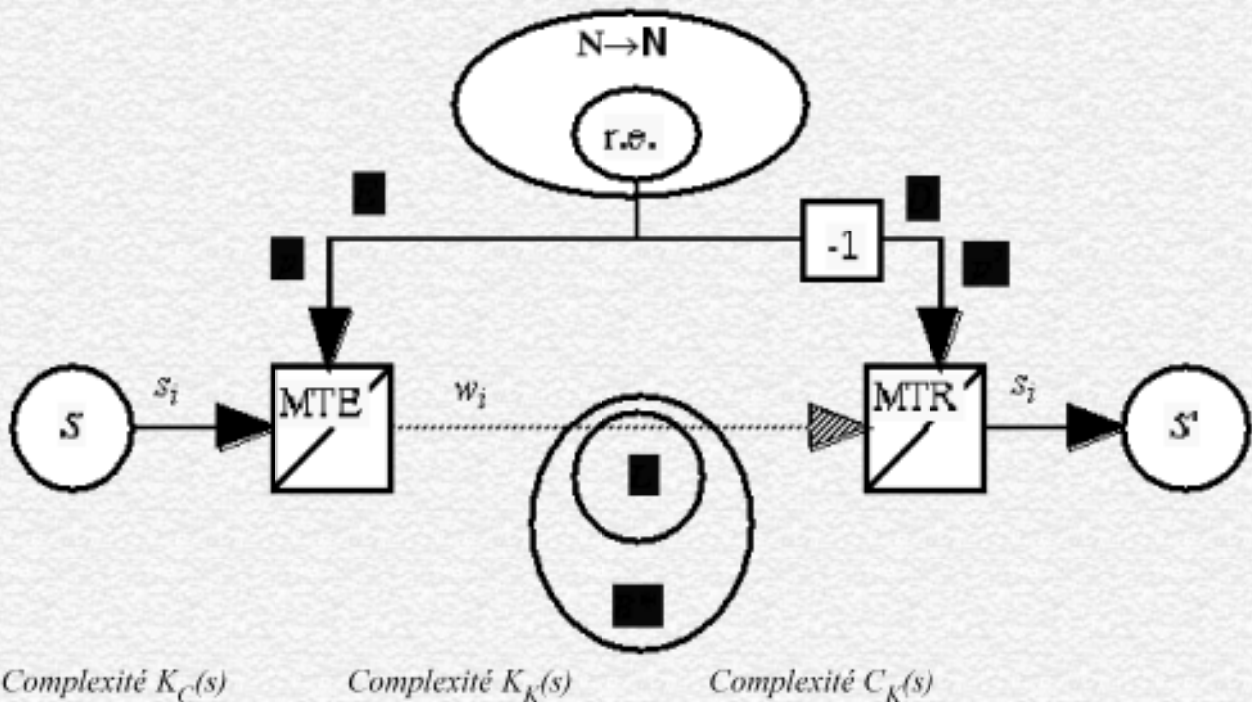
3-Complexité préfixée :  $K_C(s) = C(s)$  avec  $p$  auto-délimité. On évalue un objet  $s$  quelconque pris dans un ensemble similaire à  $\mathbf{N}$  à l'aide d'une chaîne auto-délimitée. La description d'un nombre est un mot. Cela correspond à ce que l'on entend habituellement par "codage".

4-Complexité aléatoire :  $K_K(s) = \min\{l(p) : \phi(p) = s\} = -\log m(s)$  où on évalue un objet  $s$  autodélimité par une chaîne  $p$  autodélimitée ( $m(s)$  est une mesure de probabilité "algorithmique"). La description d'un mot est un mot. Cela correspond à ce que l'on entend habituellement par "traduction" ou "interprétation".

#### 4 Schéma canonique de communication

Nous avons proposé par ailleurs [Pinson 1994, 1995] une classification de différents types informationnels selon le schéma suivant : la communication d'une information suppose l'existence :

- d'une information transmise sous forme :
  - d'un *contenu d'information* proprement dit
  - d'un *contenu d'identification explicite*, ou "référence", qui spécifie ce contenu par rapport à
  - un *contenu d'identification implicite*, ou "contexte", qui est de l'information non-transmise.



**Figure 1 : schéma canonique de communication**

Au cours d'un processus de transfert informationnel, l'information transmise est constituée des mots code  $w$  émis d'une machine de Turing Émettrice (MTE) vers une machine de Turing réceptrice (MTR). Ces mots, ainsi que les symboles contenant ces mots, sont répertoriés (énumérés) par rapport à une référence commune, qui peut être par exemple le premier mot émis. Par contre, cette transaction s'accompagne d'une information non-transmise, car présumée, qui consiste notamment dans l'information contenue dans le canal, c'est-à-dire dans l'organisation temporelle et spatiale de la mémoire partagée, et dans l'information contenue dans les codeur/décodeur. Au total, cette information non-transmise se trouve dans les programmes d'encodage/décodage,  $E$  et  $D = E^{-1}$ , dont la définition implique celle des processus de calcul et de communication, et réciproquement. Plus formellement, des ensembles de messages  $s_j$ , de mots code  $w_{ij}$  et de programmes  $p_j$  sont des

ensembles dénombrables. Soit  $P(\mathbf{B}^*)$  l'ensemble des langages définis sur l'alphabet  $\{0,1\}$ . On note  $d(A)$  la cardinalité d'un ensemble  $A$ . L'ensemble des parties d'un ensemble dénombrable n'étant pas dénombrable,  $d(P(\mathbf{B}^*)) = \aleph_1$ . Il existe donc une infinité non dénombrable de façons  $R_{ij}$  d'associer les programmes  $p_j$  aux données  $s_j$  par une fonction de  $\mathbf{N}$  dans  $\mathbf{N}$ . Or c'est précisément la relation  $R$  qui à  $s$  associe la description  $p$ , mais ne fait l'objet d'aucun transfert informationnel au cours d'une transaction : il s'agit d'un "standard" sur lequel émetteur et récepteur sont préalablement accordés.

Un transfert informationnel suppose une information  $K_K(s)$  préalablement connue de l'émetteur et du récepteur, contenue dans le standard  $R$  à l'aide duquel s'effectue la traduction par un récepteur de la description que fait un émetteur des états internes d'une source : un transfert informationnel est une description de description, ce qui, formellement, correspond à un ensemble de parties. Il faut donc étendre au continu la notion de quantité d'information, pour tenir compte de l'information contextuelle inhérente à tout processus de traitement.

## 5 Extensions au cas continu

Concernant  $H(S)$ , l'extension de la formule de l'entropie au cas continu pose problème. L'évaluation de la quantité d'information d'une source repose sur un calcul pondéré des *états discernables* d'un système. Aussi le passage à la limite dans la définition de l'entropie d'une source continue :

$$H(S) = -\sum_{i=1}^n p_i \log p_i \quad \rightarrow \quad H(S) = -\int_a^b p(s) \log p(s) ds$$

soulève un certain nombre de difficultés, comme le remarque Shannon lui-même. L'entropie d'une variable aléatoire à distribution continue :

- peut être négative.
- peut être infiniment grande
- au contraire du cas discret, n'est pas nécessairement invariante dans un changement de coordonnées.

Concernant  $C(s)$ , l'extension au cas continu pose également problème pour l'évaluation de  $K_K(s)$  : la probabilité a priori dans un ensemble dénombrable n'est évidemment pas la même que la probabilité définie comme une mesure de densité dans un ensemble continu. Il vient :

$$K_{\aleph_0}(s) = \min\{l(p) : \phi(p) = s\} = -\log(m(s)) \quad \wedge \quad K_{\aleph_1}(s) = -\log(\mu(s))$$

où  $\mu(s)$  est une mesure définie sur un ensemble continu. On montre [Gács, 1983] que  $K_{\aleph_0}(s)$  et  $K_{\aleph_1}(s)$  diffèrent d'un ordre de grandeur égal à l'inverse de la fonction d'Ackermann, fonction calculable mais qui n'est pas primitive récursive.

## 6 Trois classes de calculabilité

Ces considérations théoriques conduisent donc à faire la distinction entre trois classes de calculabilité (Anceau, 1997) :

- celle des automates finis, qui manipulent de l'information booléenne (les ordinateurs...)
- celle des machines de Turing, qui manipulent des entiers.
- celle des processus communicants, qui supposent la manipulation implicite du continu.

L'approximation que réalise un ordinateur du fonctionnement d'une machine de Turing s'apparente à un "fenêtrage" (*windowing*, au sens du traitement des signaux), c'est-à-dire une coupure dans un ensemble potentiellement infini de données. Elle se traduit par le codage, dans un alphabet de valence finie, des entiers de l'ensemble  $\mathbf{N}$  par des mots de longueur finie, qui sont des suites de bits dans le cas d'un

alphabet de valence 2.

Maintenant, on peut ajouter que l'approximation que réalise ce même ordinateur des processus de calcul vus sous l'angle général de processus de communication s'apparente à un échantillonnage (*sampling*, selon la terminologie du traitement des signaux), qui quantifie le continu en unités élémentaires de volume non nul.

Un calcul effectué par un processeur physique n'est donc qu'une traduction approchée d'un processus fondamentalement continu. Si l'on symbolise ce processus comme une "boîte noire" à qui l'on fournit des données et qui produit un résultat, ce n'est plus le concept de fonction récursive, prise dans un ensemble dénombrable, qui s'applique, mais celui de fonction continue qui associe à l'intrant un "noyau" propre au système pour produire une sortie. Une des bases mathématiques de la théorie des systèmes continus est un théorème de L. Schwarz qui montre qu'il est possible de décrire toute "boîte noire" comme un noyau  $x \nabla$  agissant sur l'intrant  $y$  par voltérration :

$$(x \nabla y)(u, v) = \int_u^v x(u, t) g(t, v) dt$$

Cette composition de fonctions, définie pour la première fois par Volterra, conduit dans le cas le plus simple à la convolution. Celle-ci doit donc jouer un rôle central dans la théorie du traitement continu de l'information.

Il est alors possible de considérer les structures algébriques du traitement de l'information comme dessinant un schéma hiérarchique à trois niveaux : fini (ensembles discrets finis, logique de Boole, arithmétique modulo), infini dénombrable (entiers naturels, arithmétique), infini non-dénombrable (formes et signaux, analyse harmonique)(Figure 2). Un même opérateur présent dans deux niveaux hiérarchiques successifs possède une double signification, selon le niveau dans lequel il se trouve. Pour chaque niveau algébrique, un opérateur de distinction (noté Op) lie deux opérateurs de composition duaux qui relèvent des niveaux algébriques inférieur et supérieur respectivement :

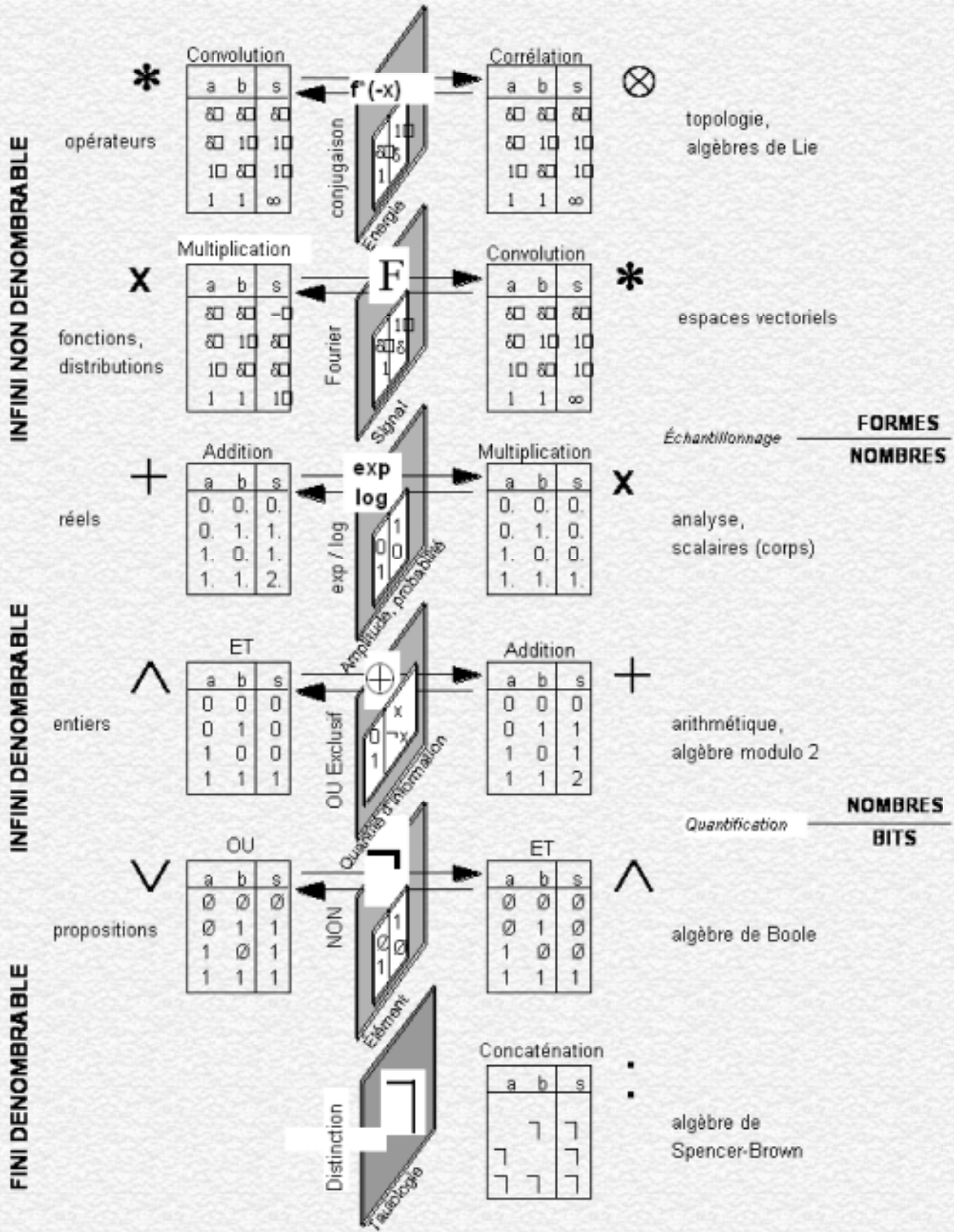
$$\text{Op}(a * b) = \text{Op}(a) \bullet \text{Op}(b) \quad \text{et} \quad \text{Op}^{-1}(a \bullet b) = \text{Op}^{-1}(a) * \text{Op}^{-1}(b)$$

"Op" est un homomorphisme transformant un opérateur noté \* à un niveau donné en un opérateur noté • dans le niveau suivant (ce processus est réversible). En changeant de niveau, les objets changent de contexte, donc de contenu d'identification.

Fondamentalement, l'opérateur Op est une involution (exemple : opérateur NON). Cette règle est en fait plus complexe, car la transformation de Fourier appliquée deux fois produit un effet de "miroir", alors que les opérateurs duaux EXP et LOG ne peuvent s'appliquer qu'alternativement :

$$\text{Exp}(\text{Log}(\text{Exp}(\text{Log}...)))$$

Figure 2 : Le traitement de l'information fait appel à : a) des opérateurs de distinction, b) des couples d'opérateurs d'association duaux. Ces opérateurs agissent sur des ensembles informationnels finis dénombrables (bits), infinis dénombrables (nombres entiers), infinis non dénombrables (formes).



## 7 Traitement formel de l'information

Les analogies formelles entre calcul booléen et analyse harmonique sont bien connues :  $\{\neg, \vee, \wedge, \emptyset, 1\}$  «  $\{F, *, \cdot, \delta, 1\}$  » où  $F$  et  $*$  sont respectivement la transformation de Fourier et la convolution,  $\delta$  et  $1$  les fonctions de Dirac et unité. Ce parallèle n'est pas exact à cause de "l'impossibilité" de la multiplication des distributions. En réalité il ne s'agit pas d'une impossibilité, mais seulement d'une définition insuffisamment précise de la distribution de Dirac, dont on peut spécifier le "carré" en se donnant une méthode précise de passage à la limite à partir d'une fonction [Colombeau, 1992].

**Tableau 1 : Boole vs Fourier**

Distinction	$\neg \neg a = a$	$F[F[x(t)]] = x(-t)$
Association	$c = a \wedge b$	$z(t) = x(t) \cdot y(t)$
	$a = a \wedge 1$	$x(t) = x(t) \cdot 1(t)$
	$c = a \vee b$	$z(t) = x(t) * y(t)$
	$a = a \vee \emptyset$	$x(t) = x(t) * \delta(t)$
Variables	$1 = \neg \emptyset$	$1(t) = F[\delta(t)]$
	$\emptyset = \neg 1$	$\delta(t) = F[1(t)]$
Dualité	$\neg a \vee \neg b = \neg(a \wedge b)$	$F[x] * F[y] = F[x \cdot y]$
	$\neg a \wedge \neg b = \neg(a \vee b)$	$F[x] \cdot F[y] = F[x * y]$

Comme les opérateurs NON et exp/log, la transformation de Fourier peut être considérée comme un processus de distinction qui peut opérer entre deux limites opposées :

$$F\left[A \cdot e^{-b \cdot t^2}\right] = A \cdot e^{-B \cdot \nu^2} \quad \text{(Gauss)}$$

$$F[\text{peigne de Dirac}] = \text{peigne de Dirac}$$

Un contenu significatif d'information se situe ainsi entre deux invariants extrêmes, le désordre d'une fonction gaussienne d'une part, et l'ordre absolu et infini d'un peigne de Dirac (noté *sha*  $\text{peigne}$ ) idéal d'autre part.

## 8 Propriétés nouvelles du calcul "superbooléen"

Quelles propriétés nouvelles est-on en droit d'attendre de processus de calcul fondés sur les opérateurs continus de l'analyse harmonique par rapport au calcul booléen de base ?

### 8.1 Algèbre non commutative

Pour des fonctions quelconques (c'est-à-dire pas seulement des fonctions paires), la double négation n'est pas équivalente à un simple opérateur NON-NON, puisque  $F[F[x(t)]] = x(-t)$ . Plus généralement, il faut tenir compte de la symétrie hermitienne qui transforme la convolution en corrélation :

$$\begin{aligned} x(t) \longrightarrow x^*(-t) &\Rightarrow x(t) \otimes y(t) = x(t) * y^*(-t) \\ &x(t) \otimes y^*(-t) = x(t) * y(t) \end{aligned}$$

Convolution et corrélation ont deux structures algébriques distinctes [Borsellino et Poggio, 1973]. La



convolution est un opérateur commutatif et associatif, la corrélation ne l'est pas. Pour étudier les deux algèbres notées  $A^\bullet$  (avec  $\bullet = * \text{ ou } \otimes$ ) on pose :

$$[x, y]^\bullet = x \bullet y - y \bullet x \quad (\text{commutateur})$$

$$[f, g, h]^\bullet = f \bullet (g \bullet h) - (f \bullet g) \bullet h \quad (\text{associateur})$$

On montre facilement que :

$$[f, g]^* = 0 \quad [f, g, h]^* = 0$$

$$[f, g]^\otimes \neq 0 \quad [f, g, h]^\otimes \neq 0$$

Ces relations caractérisent  $A^*$  comme une algèbre abélienne (comme l'algèbre de Boole), tandis que les relations :

$$[f, f]^\otimes = 0$$

$$[[f, g]^\otimes, h]^\otimes + [[g, h]^\otimes, f]^\otimes + [[h, f]^\otimes, g]^\otimes = 0$$

montrent que  $A^\otimes$  est une algèbre de Lie. On rejoint ici un formalisme de base de la mécanique quantique. Sur un domaine limité d'intégration (cas des signaux physiques et des filtres), le *début* de la convolution est calculé en sommant le produit du *début* de la forme  $x$  par le *début* de la forme  $y$ . Ce calcul peut être effectué en place entre les données d'entrée et la réponse impulsionnelle du système, comme un codage instantané. Tandis que le *début* de la corrélation est calculé en sommant le produit du *début* de la forme  $x$  par la *fin* de la forme  $y$  ou, calcul symétrique mais pas commutatif, en sommant le *début* de la forme  $y$  par la *fin* de la forme  $x$ . Aussi, il n'est plus équivalent de calculer  $a \vee b$  ou  $b \vee a$ . L'orientation de l'information devient significative.

## 8.2 Logique modale

Le calcul logique sur des vecteurs au lieu de scalaires introduit la phase comme deuxième paramètre opérationnel après l'amplitude. Le diagramme suivant souligne l'existence d'une structure commune entre l'analyse harmonique et une logique modale :

$$F[x(t-\tau)] = e^{-2i\pi\nu\tau} \cdot F[x(t)] \quad \neg z p = \alpha \neg p$$

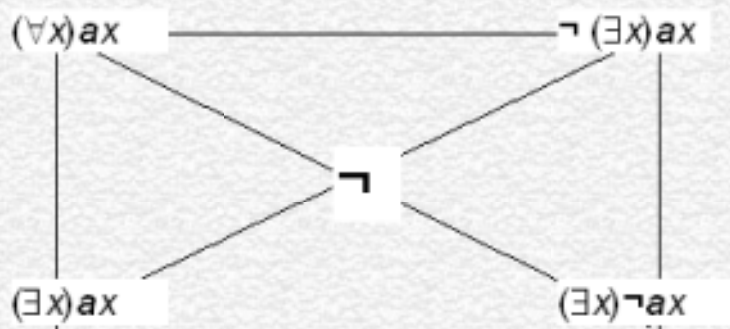
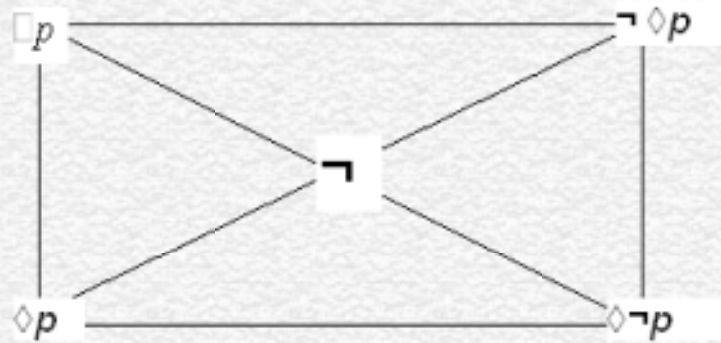
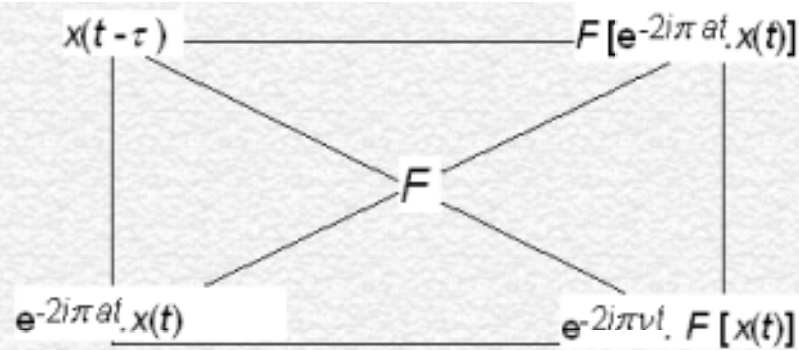
$$F[e^{-2i\pi a t} \cdot x(t)] = X(\nu-a) \quad \neg \alpha p = \Box \cup p$$

$$x(t-\tau) = F[e^{-2i\pi\nu\tau} \cdot X(\nu)] \quad \Box p = \cup \alpha \neg p$$

$$e^{-2i\pi a t} \cdot x(t) = F[X(\nu-a)] \quad \alpha p = \cup \Box \cup p$$

Par exemple, on peut interpréter la modalité à l'aide des quantificateurs " et \$ :

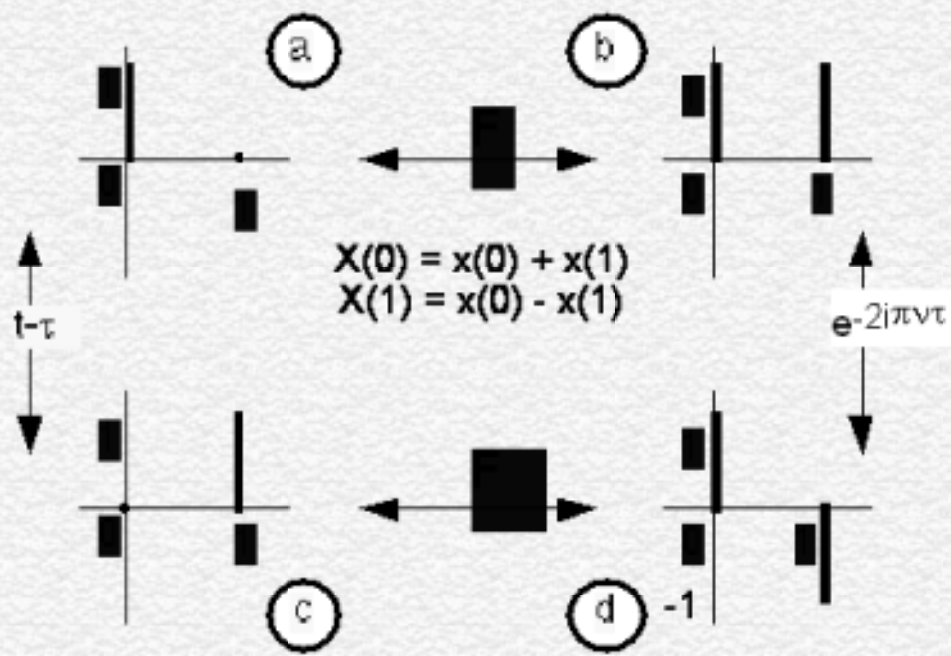
### Tableau 2 : modulation & logique modale



Une des conséquences de ce schéma de calcul est que l'on doit tenir compte de la position des échantillons par rapport à l'origine. Considérons par exemple une FFT sur deux points dont les signaux sont  $a = \delta_1(k)$  et  $b = 1(k)$ . Dans ces conditions, la FFT est établie comme indiqué tableau 3.

Considérons le signal  $c$ , qui est égal au signal  $a$  translaté (d'un seul et unique pas d'échantillonnage, puisque le calcul est effectué sur deux points seulement). Si l'on calcule les opérations superbooléennes de base, on obtient les résultats suivants :

Tableau 3 : FFT sur 2 points avec des signaux déphasés.



$x y$	$x \text{ AND } y$	$x \text{ OR } y$	$x \rightarrow y$
a a	a	a	b
a b	a	b	b
b a	a	b	a
b b	b	b	b
c c	c	a	-d
c b	c	b	0
b c	c	b	c
b b	b	b	b

Les résultats produits par les superportes ET et OU ne diffèrent pas fondamentalement des calculs logiques ordinaires. Comme noté précédemment, si l'on mesure l'amplitude résultante en valeur absolue, ces opérations sont strictement identiques au cas booléen. Par contre il n'en est pas de même pour l'implication logique, si l'on considère que le zéro arithmétique (noté 0) signifie une opération "nulle", qui n'est pas signifiante (en quelque sorte une condition "NAM" — *Not-A-Meaning* — comme il existe une condition "NAN" — *Not-A-Number* — en langage de haut niveau). Alors, quelque chose qui est faux ne peut impliquer une conséquence vraie, contrairement à l'implication logique usuelle, mais conduit seulement à un non-sens (dans un tel cas, on peut considérer que le calcul s'arrête, faute de données).

### 8.3 Points fixes

Comme cela a été noté plus haut, une fonction gaussienne (à l'échelle près) ou un peigne de Dirac de pas 1 sont des points fixes pour l'opérateur TF. Dans le cas discret, la TFD d'un scalaire, c'est-à-dire une TFD calculée sur un seul point, est égale à ce scalaire lui-même :

$$F[k] = k$$

D'une part, cela signifie que les "superbits" sont nécessairement des vecteurs pour pouvoir être distingués les uns des autres par un opérateur NON, ce qui implique, comme on l'a noté : 1) que la position de l'information par rapport à l'origine devient significative (contenu d'identification explicite), d'où l'extension de la logique de Boole à une logique modale ; 2) que la direction de l'information par rapport à l'orientation du référentiel devient aussi significative (contenu d'identification implicite), d'où l'extension de l'algèbre de Boole à une algèbre de Lie non commutative.

En revanche, en choisissant convenablement la valeur du nombre d'échantillons  $N$  (il faut que  $\sqrt{N}$  soit entier, soit  $m$  impair), il apparaît que l'on peut construire une logique intégrant la notion de point fixe, puisque la TFD d'un peigne de pas  $\sqrt{N}$  est un peigne de pas  $N/\sqrt{N} = \sqrt{N}$ .

Il est alors aisé d'étendre la notation de Spencer-Brown au cas auto-référentiel, comme l'a fait Varela (Varela, 1974), grâce aux règles de calcul suivantes :

$$I1 : \text{dominance} \quad \neg v = \neq$$

$$I2 : \text{ordre} \quad \neg \neg =$$

$$I3 : \text{consistance} \quad \neg Z = Z$$

$$I4 : \text{nombre} \quad Z Z = Z$$

Soit " $\neg$ " =  $\delta_1(k)$ ,  $\neq$  =  $1(k) = \llcorner \llcorner \llcorner_N$ ,  $Z = \llcorner \llcorner_{N/2}$ . Il vient :

$$I1 : 1 \vee v = F_1[F_1[1] \cdot F_1[v]] = F_1[\delta_1 \cdot F_1[v]] = F_1[\delta_1] = 1$$

$$I2 : F_1[F_1[\delta_1]] = \delta_1$$

$$I3 : F_1[\llcorner \llcorner_{N/2}] = \llcorner \llcorner_{N/2}$$

$$I4 : \llcorner \llcorner_{N/2} \vee \llcorner \llcorner_{N/2} = F_1[F_1[\llcorner \llcorner_{N/2}] \cdot F_1[\llcorner \llcorner_{N/2}]] = F_1[\llcorner \llcorner_{N/2} \cdot \llcorner \llcorner_{N/2}] = \llcorner \llcorner_{N/2}$$

Cette propriété de point fixe (pour les ensembles d'échantillons de cardinal pair) est essentielle. D'une part on montre que les paradoxes de la logique classique (Russell...) et les théorèmes de limitation (Gödel, Tarski,...) dérivent d'une structure plus fondamentale exprimée comme une propriété autoréférentielle de point fixe (Lawvere, 1969). En introduisant un ensemble autosimilaire (i.e. fractal) satisfaisant cette propriété, un point fixe ne conduit pas nécessairement à un paradoxe. D'autre part, elle autorise une extension de la logique booléenne à une logique tri-valuée dont les tables de vérité sont les suivantes (on note :  $\emptyset = \delta_1 = \llcorner \llcorner_1$  ;  $1 = 1 = \llcorner \llcorner_N$  ;  $Z = \llcorner \llcorner_{N/2}$ ) :

**Tableau 4 : tables de vérité superbooléennes avec point fixe :**

x	$\neg x$	$x \wedge y$	$\emptyset$	$\square$	1	$x \wedge y$	$\emptyset$	$\square$	1	$x \wedge y$	$\emptyset$	$\square$	1	$x \wedge y$	$\emptyset$	$\square$	1
$\emptyset$	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\square$	1	$\emptyset$	1	1	1	$\emptyset$	$\emptyset$	$\square$	1
$\square$	$\square$	$\square$	$\emptyset$	$\square$	$\square$	$\square$	$\square$	$\square$	1	$\square$	$\square$	$\square$	1	$\square$	$\square$	$\square$	$\square$
1	$\emptyset$	1	$\emptyset$	$\square$	1	1	1	1	1	1	$\emptyset$	$\square$	1	1	1	$\square$	$\emptyset$
	$\neg x$			$x \wedge y$			$x \vee y$			$x \rightarrow y$	$x \oplus y$						

En faisant varier  $N$ , on peut aisément étendre ces notions à une logique multivaluée, rejoignant ainsi les caractéristiques bien connues de la logique floue.

## 9 Réalisation physique par FFT

Il reste alors à voir comment un automate fini pourrait simuler au mieux ces principes fondamentaux de calcul.

### 9.1 FFT

Un premier pas consiste à remplacer les opérateurs continus de l'analyse harmonique par leurs équivalents digitaux.

L'analyse harmonique physique (et non théorique) des formes subit un certain nombre de limitations à travers les processus de traitement digital de l'information : quantification, fenêtrage, échantillonnage, blocage. Ces limitations physiques sont inévitables à partir du moment où l'on veut transcrire des capacités de traitement de l'information non-transmise dans des dispositifs réels, c'est-à-dire des processus communicants. Aussi l'algèbre  $\{F, *, \cdot, \delta, 1\}$  est-elle "digitalisée", la transformation de Fourier (TF, notée  $F$ ) devient une transformation de Fourier digitale (TFD, notée  $F$ ) dont le calcul est réalisé par exemple à l'aide de l'algorithme "papillon" bien connu (TF rapide, abrégée TFR, ou *FFT*, *Fast Fourier Transform*). Dans ce cas approché la multiplication des "distributions" devient possible.

### 9.2 Normalisation et opérateur NON

TF et TFI (TF inverse) sont deux opérations symétriques. Comme  $F[F[x(t)]] = x(-t)$ , nous obtenons, pour une distribution paire comme  $\delta$ , une relation similaire à la relation logique  $\neg\neg a = a$ .

Malheureusement, cela n'est pas vrai dans le cas digital, à cause du facteur en  $1/N$  qui est présent dans l'une des deux opérations, par exemple la TFI :

$$X(j) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi \frac{jk}{N}} \quad ; \quad x(k) = \frac{1}{N} \sum_{j=0}^{N-1} X(j) \cdot e^{+j2\pi \frac{jk}{N}}$$

Aussi, si l'on veut que  $\neg\neg 1(t) = 1(t)$  et  $\neg\neg \delta(t) = \delta(t)$ , nous devons traduire ces opérateurs en appliquant le principe de relativité d'échelle. Aussi le principal opérateur (TF) devient une transformation de Fourier digitalisée et normalisée à 1 (TFDN, notée  $F_1$ ) :

$$X_1(j) = F_1[x(k)] = \frac{F[x(k)]}{\sup |F[x(k)]|}$$

Avec cette procédure, toute amplitude résultante maximale est conservée égale à un. Mais ce principe de calcul ne modifie pas la forme des signaux. Aussi la TFDN  $F_1$  est vue comme un opérateur NON : un "dirac" normalisé à un a pour inverse  $1(t)$  et réciproquement. Avec ce principe de calcul, *l'information est donc codée par la forme du signal, et non par son amplitude.*

Cette mise à l'échelle statique ne doit pas être confondue avec la mise à l'échelle dynamique utilisée dans l'algorithme TFR pour éviter les dépassements numériques. Pour calculer  $X_1(j) = F_1[x(t)]$ , on peut utiliser en APL un algorithme comme :  $X_1 \leftarrow X \div \text{Sup } |X|$ .

### 9.3 Mise à l'échelle et opérateur ET

Pour exécuter une opération NON par TFR, et une opération ET par multiplication, il est nécessaire de suivre les règles suivantes :

NON :  $F_1$ (pleine échelle positive)  $\rightarrow$  pleine échelle positive

ET : pleine échelle positive  $\times$  pleine échelle positive  $\rightarrow$  pleine échelle positive

Ainsi, si tout échantillon possède une amplitude égale à 0 ou 1, le produit résultant (ET) est toujours égal à 0 ou 1 (puisque  $1 \times 1 = 1$ ).

#### 9.4 Opérateur OU

En ce qui concerne l'opérateur OU, nous pouvons : soit calculer directement le produit de convolution (par convolution circulaire) ; soit appliquer le théorème de convolution, comme cela peut être fait en logique booléenne en faisant appel aux relations de De Morgan. Pour être exact, il serait nécessaire d'étendre les vecteurs à  $2N$  points, en remplissant de zéros les segments compris entre  $N+1$  et  $2N$ , pour tenir compte du fenêtrage et du repliement du spectre. Mais si on se limite aux vecteurs tels que  $\delta_1(t)$ ,  $1(t)$  et aux peignes normalisés, il n'est pas nécessaire d'en tenir compte ici.

#### 9.5 "Superbits" : peignes de Dirac

Échantillonner les signaux rend leur spectre périodique, et réciproquement. Les calculs ci-dessus sont donc en réalité exécutés avec des distributions "peignes" digitalisées, dont les périodes valent 1 pour  $1(t)$  et  $N$  pour  $\delta_1(t)$ . Comme la TF est un opérateur qui possède la propriété d'homothétie :

$$F[x(at)] = \frac{1}{|a|} X\left(\frac{\nu}{a}\right)$$

nous pouvons considérer des peignes de période  $a$  et  $1/a$ . Aussi une valeur suffisamment grande de  $N$  autorise différents choix pour traduire les éléments binaire 0 et 1 en "superbits" représentés par des peignes de pas  $a$  et  $1/a$ . Il faut seulement que  $a$  soit une puissance de 2 comprise entre 1 et  $N/2$ .

#### 9.6 "Superoctets" : signaux à $M$ dimensions

La TF d'une fonction séparable à  $M$  dimensions est elle-même séparable. Par exemple en dimension 2, le calcul est effectué par :

$$X(j_1, j_2) = \sum_{k_2=0}^{N_2-1} \sum_{k_1=0}^{N_1-1} x(k_1, k_2) e^{-j2\pi \frac{k_1 j_1}{N_1}} e^{-j2\pi \frac{k_2 j_2}{N_2}}$$

Soit  $N_1 = N_2 = \dots = N$  (matrice carrée). Considérons les signaux :  $\delta_1(k_1) \cdot \delta_1(k_2)$  ;  $\delta_1(k_1) \cdot 1(k_2)$  ;  $1(k_1) \cdot \delta_1(k_2)$  ;  $1(k_1) \cdot 1(k_2)$ . Ces signaux correspondent aux mots de 2 bits : 00, 01, 10, 11. Nous avons (la concaténation étant notée ";" :

$$\neg \{a;b\} = \neg a; \neg b \quad \ll \quad F_1[x(k_1), y(k_2)] = F_1[x(k_1)] \cdot F_1[y(k_2)]$$

Ces relations s'étendent à un nombre quelconque de superbits, et les calculs sur des mots de  $M$  bits sont possibles, comme en logique classique. Soit :

$$z = x + y \quad \text{avec } x, y \in \{ \delta_1(k_1); 1(k_1) \}$$

$$z \in \{ \delta_1(k_1) \cdot \delta_1(k_2), \delta_1(k_1) \cdot 1(k_2), 1(k_1) \cdot \delta_1(k_2), 1(k_1) \cdot 1(k_2) \}$$

Alors :

$$s(k_1) = x(k_1) \oplus y(k_1) \quad s \in \{ \delta_1(k_1); 1(k_1) \}$$

$$c(k_2) = x(k_1) \wedge y(k_1) \quad c \in \{ \delta_1(k_2); 1(k_2) \}$$

Cet algorithme permet l'énumération des entiers, un entier de valeur  $N$  étant représenté par un volume dans un espace à  $M$  dimensions, avec  $M^3 \lceil \log_2 N \rceil$ .

## 10 Précision numérique

L'implémentation de l'ensemble de ces processus de calculs est réalisable soit directement en langage machine sur des processeurs de traitement numérique du signal spécialisés (processeurs DSP), comme nous l'avons proposé (Pinson 1996) à l'aide de processeurs *Texas Instrument TMS320C50*.

L'inconvénient d'une telle solution est sa lourdeur, puisqu'il faut remplacer chaque "porte" booléenne par un processeur DSP !

Une autre solution consiste à simuler ces fonctions sur un ordinateur classique dans un langage de haut niveau. Il est clair alors que des propriétés de consistance, de cohérence et de clarté du langage découleront directement les capacités du programme à simuler correctement ces processus. Une solution élégante parce que concise est l'utilisation du langage APL. Ce faisant, on est immédiatement conduit à poursuivre ce raisonnement, puisque l'APL met clairement en évidence le lien étroit qui existe entre FFT et transformation de Hadamard (Nishikawa 1994). Si l'on poursuit le raisonnement jusqu'à son terme, c'est à cette dernière opération qu'il convient de donner le rôle fondamental d'opérateur superbooléen.

Étant donné  $N$  échantillons quantifiés avec  $B$  bits dans une gamme  $L$ , la précision numérique ou résolution vaut  $q = 1/2^B$ . Pour conduire les calculs, deux solutions techniques sont à considérer : calculs en virgule flottante, ou calcul en virgule fixe.

-Calculs en virgule flottante : avec  $B = 32$ , on obtient une dynamique de calcul égale à  $20\log(2^{32}) = 192\text{dB}$ . Cette dynamique paraît importante, et suffit lorsqu'on applique l'analyse harmonique au calcul d'un spectre. Mais le bruit de calcul augmente rapidement si une succession d'opérations sont opérées en chaîne. Il s'agit d'une limitation fondamentale : quelle que soit la valeur de  $B$ , le décalage et la troncation ont des effets rapidement non négligeables. S'agissant d'opérations booléennes se succédant en grand nombre, cette solution devient impraticable.

-Calculs en virgule fixe : la première condition à remplir pour que les calculs soient exacts dans un ensemble  $E$  est que le produit ou la somme de deux éléments de l'ensemble  $E$  appartiennent à cet ensemble. Cette condition est vérifiée dans l'ensemble des entiers  $0, 1, \dots, L-1$  si les calculs sont faits modulo  $L$ . Un choix convenable de  $L$  permet de définir des transformations algébriques (TFA) ayant des propriétés comparables à la TFD [Bellanger, 1981].

Sur le plan pratique, ce choix est guidé par des considérations de simplicité des mécanismes de calcul. L'arithmétique modulo suppose une division entière par  $L$ , division réalisée très facilement par décalage pour  $L = 2^b$ , ou par décalage avec retenue pour  $L = 2^b \pm 1$  (arithmétique en complément à un).

Sur le plan théorique, on montre que l'existence d'une transformation algébrique et de son inverse se ramène à la condition suivante : pour tout facteur premier  $P$  de  $L$ , il faut que  $N$  divise  $P-1$ . Donc, si l'on choisit  $L$  premier,  $N$  doit diviser  $L-1$ . Finalement, on montre qu'un choix du module  $L$  très intéressant est un nombre de Fermat :

$$L = 2^{2^b} + 1$$

L'expression de la transformation algébrique correspondante se déduit de ces considérations théoriques :

Ordre de la transformée :  $N = 2^{b+1}$

Transformée directe :

$$X(j) = \sum_{k=0}^{N-1} x(k) 2^{jk}$$

Transformée inverse :

$$x(k) = 2^{-(2^{b+1} - b - 1)} \sum_{j=0}^{N-1} X(j) 2^{-jk}$$

Ces calculs impliquent toutefois que le nombre d'échantillons  $N$  soit lié à la résolution :  $B \approx N/2$ . Par exemple, si  $b = 5$ ,  $B = 2^b = 32$ ,  $N = 64$ ,  $L = 4.294.967.297$ .

Enfin, la normalisation à 1 précédemment décrite nécessaire pour introduire une symétrie entre TF et TFI suppose une division de tous les résultats de la TFA par sa valeur (absolue) maximale, ce qui est une autre forme de division modulo qui entre dans le cadre de ces règles de calcul.

## 11 Opérateurs superbooléens en arithmétique modulo 2

Arrivant au terme de cet article, notre démarche peut se résumer ainsi :

- l'analyse théorique du concept d'information conduit à supposer l'existence de trois classes de calculabilité : fini (booléen), infini dénombrable (arithmétique) et continu (harmonique).
- l'extension au continu des propriétés du calcul booléen conduit à généraliser les opérateurs booléens à l'aide des opérateurs de l'analyse harmonique, conservant ainsi les structures mathématiques fondamentales : involution, dualité, commutativité, associativité, etc. Cette généralisation, mettant en jeu des vecteurs au lieu de scalaires, permet de conjecturer l'existence de propriétés logiques nouvelles.
- réciproquement, tout processus opérationnel de calcul et de communication étant nécessairement fini et digitalisé, la simulation sur ordinateur de cette généralisation passe par une utilisation adaptée des algorithmes du traitement numérique des signaux (DSP).
- mais la précision requise pour effectuer des calculs en chaîne conduit à abandonner l'arithmétique en virgule flottante ou fixe au profit de l'arithmétique modulo  $p$ , avec  $p = 2$  dans le cas binaire.

Les transformations unitaires rapides sont une conséquence du théorème de Good (Kunt, 1981), qui énonce qu'une matrice  $A$  générée à partir d'une base de matrices hermitiennes orthogonales est décomposable en un produit de matrices élémentaires. Les transformations de Walsh-Hadamard et de Fourier sont alors des cas particuliers de ce théorème, et tous ces calculs se ramènent à des calculs matriciels. Mais, suivant en cela la démarche de Langlet 1994, l'application de l'arithmétique modulo 2 aux matrices conduit à chercher une implémentation des opérateurs harmoniques digitalisés (que nous avons appelé "superbooléens") sous forme binaire. On aboutit ainsi à une application de la classification des opérateurs de calcul (finis, dénombrables, continus) comme indiquée figure 2 au cas strictement binaire de l'arithmétique modulo 2, et cela donne le schéma exposé par Zaus 1995 page 67 de ces opérateurs vus sous l'angle binaire.

Cette démarche semble naturelle, si l'on constate que le point central de notre propre schématisation, fig. 2, est l'opérateur OU Exclusif, qui constitue le lien (encore bien mystérieux, au demeurant), entre les entiers naturels et les réels. Or les produits matriciels des transformations FFT et Hadamard fournissent les composantes des signaux ordonnées selon le code Gray, dont l'une des définitions possibles est précisément l'intégrale binaire de parité (Zaus 1994) :

$$G \rightarrow B \cup 0, 1 \setminus B$$

$$B \rightarrow \setminus G \quad (\text{intégrale binaire})$$

La simplification aux matrices purement binaires, en passant de :

$$X +, \cdot Y \quad (\text{produit matriciel})$$

à :

$$X \setminus, \wedge Y \quad (\text{produit matriciel binaire})$$



conduit ainsi à prendre l'opérateur XOR comme opérateur de calcul principal. D'où une représentation des signaux sur une base de fonctions binaires. Il reste à choisir le type exact de transformation binaire qui conserverait les propriétés de l'analyse harmonique. Zaus 1996 propose page 45 la transformée de Langlet :

$$C \rightarrow C, \quad \neg 1'B \rightarrow \neg 1 \setminus B$$

Les propriétés des matrices générées par cette transformation (pariton) sont en effet conformes à celles des matrices employées en FFT. Notamment, leur structure en "triangle de Pascal" dérive du développement des coefficients du binôme  $(a+b)^n$  dont l'enveloppe est une gaussienne (Zaus 1994 page 58). Ce point est essentiel si l'on veut pouvoir mettre en évidence une propriété de point fixe ( $\neg A \equiv A$ ), propriété centrale comme on l'a vu plus haut à partir de laquelle il est permis d'envisager une extension du calcul booléen. Il reste – et ceci constitue un programme de recherches qui est à effectuer – à analyser la logique de la parité pour s'assurer qu'elle possède bien toutes les propriétés requises qui ont été énoncées le long de cet article pour matérialiser convenablement les principes du calcul superbooléen.

En tout état de cause, il est fondamental de remarquer que cela n'est envisageable que dans un cadre matriciel, que Stern 1994 développe à l'aide de matrices binaires de rang deux, en systématisant les calculs booléens sur ces matrices. Mais au-delà d'une simple extension de l'algèbre de Boole aux matrices binaires de rang deux, comme le fait cet auteur, c'est bien une exploitation de telles matrices (comme le "géniton" de Langlet) au sens de l'analyse harmonique, fondée sur la notion de transformation unitaire rapide, qui permet d'envisager l'application opérationnelle de réelles propriétés logiques nouvelles. On aboutit alors, comme Langlet 1994 et Zaus 1995 le proposent, et comme nous-mêmes l'avons proposé à plusieurs reprises (voir par exemple Pinson 1994, 1995) à une possible application de ces procédures de calcul à l'étude des réseaux neuronaux et des caractéristiques de la cognition.

## Bibliographie

- ANCEAU F. : Le cerveau humain est-il un super-ordinateur ?, *XVIème Colloque International de Bio-Mathématique*, Paris, 11-13 septembre 1997.
- BORSELLINO A., POGGIO T. : Convolution and Corrélation Algebras, *Kybernetik*, vol. 13, 1973, p 113-122.
- COLOMBEAU, J.-F. : *Multiplication of distributions - A tool in mathematics, numerical engineering and theoretical physics*, LNM 1532, Springer-Verlag, Berlin, 1992.
- GÁCS P. : On the relation between descriptonal complexity and algorithmic probability, *Theoretical Computer Science*, n° 22, p 71-93, 1983.
- KUNT, M. : *Traitement numérique des signaux*, Dunod, Paris, 1981.
- LANGLET, G. : De l'algèbre de Hadamard à la Transformation Cognitive, *Les Nouvelles d'APL*, AFAPL, Paris, n°11, p 65-92, 1994a.
- LANGLET, G. : Vers le codage optimal de l'information, *Les Nouvelles d'APL*, AFAPL, Paris, n°11, p 93-101, 1994b.
- LAWVERE F. W. : *Category Theory, Homology Theory and their applications II*, Hilton ed., Springer-Verlag, Berlin, 1969.
- LI M., VITÁNYI P. : *An Introduction to Kolmogorov Complexity and Its Applications*, Springer-Verlag, Berlin, 1993, 1997.
- NISHIKAWA, T. : Algorithme généralisé de transformation rapide de Fourier et de Hadamard implanté en APL2, *Les Nouvelles d'APL*, AFAPL, Paris, n°12-13, p 67-70, 1994.
- PINSON G. : Cognition et Multicrucialité, *Rev. Int. Systémique*, vol 8, n°2, p 183-210, 1994.
- PINSON G. : Une théorie cognitive de l'information, *Rev. Int. Systémique*, vol 9, n°1, p 27-66, 1995a.
- PINSON G. : Cognitive Information Theory, *Proc. 14th Intern. Congress on Cybernetics*, Ass. Int. de

Cybernétique, Namur, Belgique, 1995b.

PINSON G. : Beyond Boole Algebra with TMS320 DSP Multiprocessing, *Proc. of The First European DSP Education and Research Conference*, Texas Instrument, ESIEE, Paris, 25-26/09/1996.

PINSON G. : Complexité algorithmique des processus de transfert informationnel, *coll. Les modèles de représentation : quelles alternatives*, Neuchâtel, 3-5 septembre 1997.

SPENCER-BROWN G. : *Laws of form*, Dutton, New-York, 1972.

STERN, A. : *Matrix Logic and Mind*, North-Holland/Elsevier, Amsterdam, 1992.

USPENSKY V. A. : Complexity and Entropy : An Introduction to the Theory of Kolmogorov

Complexity, in WATANABE O. (Ed.), *Kolmogorov Complexity and Computational Complexity*, Springer, EATCS, p 85-102, 1992.

VARELA F. J. : A calculus for Self-reference, *Int. J. General Systems*, vol. 2, 1975, p 5-24.

ZAUS, M. : La Logique de la Parité, théorique et appliquée, *Les Nouvelles d'APL*, AFAPL, Paris, n°12-13, p 42-66, 1994.

ZAUS, M. : Intégration de Parité, Milieux Excitables et Neuro-Calcul, *Les Nouvelles d'APL*, AFAPL, Paris, n°15, p 46-74, 1995.

ZAUS, M. : Fondements Mathématiques de la Logique de la Parité, *Les Nouvelles d'APL*, AFAPL, Paris, n°18, p 27-48, 1996.

ZAUS, M. : Analyse du signal binaire en Logique de la Parité, *Les Nouvelles d'APL*, AFAPL, Paris, n°19, p 17-52, 1996.

ZAUS, M. : Perspectives Transdisciplinaires de la Logique de la Parité, *Les Nouvelles d'APL*, AFAPL, Paris, n°21, p 11-32, 1996.

[1] Tous les logarithmes sont comptés en base 2