Fondements Mathématiques de la Logique de la Parité

par Michael Zaus

(traduction : Gérard A. Langlet. Rapport original en anglais, daté du 14 février 1996)

Cet article a pour but de fournir une vue d'ensemble de l'espace binaire B^n et des concepts associés, en particulier de l'opération de OU_*Exclusif* (XOR) $x \oplus y$ ainsi que sa généralisation à l'intégrale scalaire binaire $\oplus^n_{i=1} x_i$ et à l'intégrale vectorielle binaire $\square^n_{i=1} x_i$ pour les vecteurs $x \in B^n$. Nous présentons ces fondements mathématiques selon une nomenclature standard en informatique, mais nous nous rapporterons aussi à la nomenclature d'APL, car l'intégrale vectorielle binaire, exposée dans la section 2.3 tire son origine d'APL ([IVE62], [IVE78], [LAN93], [LOC89], [ZA95]).

2.1 L'Espace $B^n = \{0,1\}^n$

Les objets à étudier essentiellement en logique de la parité sont des vecteurs de dimension n contenant des composantes binaires. Selon le contexte, ces vecteurs peuvent avoir une signification différente; ils comprennent des points, des motifs, des mots, des items de mémoire, des données, des paramètres, des chromosomes artificiels, des images, des événements, des ensembles flous, des signaux de télévision, des signaux corporels (EEG, ECG, EOG etc...), des vecteurs d'état de milieu excitable, et bien plus encore. Les représentations de ces entités sont basées sur l'espace B^n que nous obtenons par produit cartésien n-uple sur $B = \{0,1\}$:

$B^n = B \times B \ldots \times B$
n *fois*

En développant l'espace $B^n = \{0,1\}$ on obtient l'ensemble des vecteurs binaires à n dimensions :

(1) $\quad B^n = \{x : x = (x_1, x_2, ..., x_n); x_i \in \{0,1\}; i = 1, 2, ..., n\}$

Le nombre de points dans B^n est 2^n. Du fait que la représentation binaire des entiers sur n chiffres binaires dans l'intervalle $[0, 2^n]$ s'identifie avec les éléments de B^n, on *peut* les considérer comme identiques, à condition que ceci ait un sens pour une application spécifique. Par exemple, dans les algorithmes génétiques et l'optimisation de fonctions, l'ensemble $B = \{0,1\}$ est un alphabet de symboles et l'application e : $B^n \rightarrow D$ est une fonction de codage pour un certain domaine discret et fini D. Le codage des éléments de D est une application à partir des chaînes (chromosomes artificiels) de longueur n de B vers D, et B^n est appelé *l'espace de recherche*. La dimension n de l'espace dépend strictement de la précision requise pour la fonction soumise à optimisation (ZA95c]).

Dans d'autres domaines d'appplications, par exemple dans les réseaux de neurones, en particulier les modèles de mémoire distribuée de manière peu dense, les éléments de B^n peuvent toujours s'écrire comme des entiers à n bits, en représentation binaire, p. ex. 1001 ... 0111, mais ils représentent ici des items de mémoire en termes de motifs de zéros et de un à n composantes, sans que l'on assigne une signification spécifique aux composantes, du fait que ce sont des caractéristiques abstraites. Ici, B^n doit avoir une grande dimension de sorte que la mémoire soit répartie de façon peu dense dans le modèle à mémoire distribuée.

Un espace à relativement basse dimension B^n est utilisé dans la pratique courante des puces floues en technologie VLSI. Ici, 0 ou bien 0000 représente une non-appartenance, tandis que 15 ou bien 1111 représente une appartenance complète. Les autres nombres intermédiaires sont utilisés pour représenter des points équidistants dans l'intervalle unité [0,1] selon la progression 1/15, 2/15, ... 14/15. Chaque vecteur de quatre bits représente donc un degré de flou entre 0 et 1, et si l'univers du discours d'un ensemble flou est discrétisé en 31 éléments, il en résulte une quantification en 124 bits de la fonction d'appartenance ([WAT92]).

Un autre exemple concerne l'analyse de signaux binaires, où les éléments de B^n représentent des coefficients binomiaux modulo 2 lesquels à leur tour représentent des fonctions standard du temps (composantes spectrales). L'espace sous-jacent (B^n, N) est alors utilisé pour déterminer des fonctions discrètes du temps, c'est-à-dire des signaux binaires. Ceci sera développé dans les sections suivantes. Le triangle de Pascal modulo 2, sa complétion en un carré de Pascal, ainsi que son identification avec la matrice S d'Iverson et avec la matrice retournée de Langlet appelée pariton va jouer un rôle central ([BOP81], [IVE62], [LAN92a], [ZA95b]).

Le fait de considérer les éléments de B^n comme des entiers est certainement plus une exception qu'une règle, au vu de ces exemples et de ceux donnés au début de cette section. L'espace B^n a plein de structure, et cela dépend de l'introduction de relations et d'opérations dans B^n. Afin de fournir une base convenable pour effectuer de la modélisation à l'aide d'espaces binaires, il est avantageux de caractériser une

paire de structures intimement associées dans B^n avant de passer à un traitement plus détaillé de ses opérations et de ses opérateurs. D'abord, la paire $\langle B^n, \leq \rangle$ est un ensemble partiellement ordonné, où la relation $\leq$ est réflexive, antisymétrique et transitive. Dans B^n, l'ordre partiel de la relation $\leq$ est défini comme des points ou des coordonnées (bit à bit) selon :

(2) $$x \leq y = (x_1 \leq x_2, \ldots x_n \leq y_n) .$$

Ainsi, de manière à être partiellement ordonnés, les vecteurs x et y doivent satisfaire l'une des égalités $0 \leq 0$, $0 \leq 1$, ou $1 \leq 1$. La paire $\langle B^n, \leq \rangle$ est importante pour des ordres métriques de la représentation de B^n par des diagrammes de *Hasse* qui permettent des représentations spatiales de B^n.

Ensuite, si l'on introduit dans B^n les opérations d'intersection ($\wedge$) , d'union ($\vee$) , de complémentation ($^-$) ainsi que les définitions point à point telles que :

(3) $$x \wedge y = (x_1 \wedge y_1, \ldots, x_n \wedge y_n)$$

(4) $$x \vee y = (x_1 \vee y_1, \ldots, x_n \vee y_n)$$

(5) $$x = (x_1, x_2, \ldots, x_n),$$

nous obtenons une algèbre booléenne $\langle B^n, \wedge, \vee, ^- \rangle$. Cette structure est importante, mais moins intéressante pour les systèmes binaires dynamiques en général, et pour la représentation de signaux binaires et leur analyse en particulier. En ce qui concerne ces derniers, nous avons besoin de concepts métriques et topologiques dans B^n , spécialement d'applications *préservant les distances*, c'est-à-dire des isométries et des déplacements, en particulier des réflexions, des rotations et des translations. En outre, il faudra des opérations *préservant l'entropie* pour des propagations, résistant à l'erreur, de différences concernant les signaux et les transformées de ces signaux dans B^n.

Ces conditions nécessaires peuvent se trouver satisfaites par au moins trois approches, surtout par le passage naturel aux anneaux booléens, ou par le passage plus radical à des groupes finis binaires ou même à des corps de Galois en algèbre modulo 2 ([BOP81], [LAN92b, [AIG93]). Nous nous restreindrons à une brève caractérisation de la première option en faisant ressortir le passage de $\langle B^n, \wedge, \vee, ^- \rangle$ vers un anneau booléen.
Le concept central de ce passage est XOR, l'opération de OU-*eXclusif* (en anglais « *eXclusive*-OR »), définie selon la loi de de Morgan par

(6) $$(x \oplus y) = [(x \wedge y) \vee (x \wedge y)] = (x \neq y) .$$

Remarquer que nous avons défini XOR en utilisant la notation $\oplus$ aussi bien que $\neq$. Cela ne devrait pas constituer une surprise pour le lecteur que OU-*exclusif* (XOR) et *différent* soient des concepts synonymes. Nous mettons l'accent sur ce fait, car ce n'est pas seulement une affaire de notation mais aussi une affaire de sens. XOR représente la *parité*, le concept permettant de distinguer une grandeur paire d'une grandeur impaire, comme nous le montrerons plus loin. Si *inégal* a le sens de *différent*, alors *non inégal* a le sens d'*équivalent*, et c'est précisément le sens de la définition par la dualité de la loi de de Morgan :

(7) $$\overline{(x \oplus y)} = [(x \vee y) \wedge (x \vee y)] = (x = y) .$$

Ainsi, les définitions (6) et (7) utilisent la dualité des lois de de Morgan, et le fait de penser à XOR comme $\neq$ va apporter un fondement formel d'une manière élégante avec des concepts nouveaux et généralisés. En notation standard, la définition ponctuelle de XOR dans B^n est donnée par

(8) $$(x \oplus y) = (x_1 \oplus y_1 , \wedge \ldots , x_n \oplus y_n) ,$$

et en posant $x \cdot y = x \wedge y$ en incluant sa définition ponctuelle, on obtient finalement :

(9) $$B R (n) = \langle B^n , \oplus, \cdot \rangle ,$$

c'est-à-dire un anneau booléen avec identité, du fait que la propriété $x \cdot x = xx = x$ est satisfaite pour tous les éléments de $x \in B^n$. Théoriquement, on peut toujours repasser de $\langle B^n, \wedge, \vee, ^- \rangle$ à $\langle B^n , \oplus, \cdot \rangle$, et vice-versa, car nous pouvons toujours recapturer les opérations d'intersection ($\wedge$), d'union ($\vee$) et de complément ($^-$) par $x \wedge y = x \cdot y, x \vee y = x \oplus y \oplus (x \cdot y)$ et $x = 1 \oplus x$. L'intérêt des anneaux booléens provient du fait que l'opération $\oplus$ nous permet d'introduire des fonctions de distance, des propriétés de réversibilité, des applications congruentes ainsi que le groupe des déplacements dans B^n . Remarquer encore que l'algèbre $\langle B^n, \wedge, \vee, ^- \rangle$ nous permet aussi d'introduire l'opération $x \square y = x \wedge y$, appelée soustraction booléenne. Une « symétrisation » de la différence $x \square y$ s'obtient par $(x , y) \vee (y \square x)$, mais ceci est équivalent à $x \oplus y$, d'où à XOR. L'équivalence des expressions

(10) $(x \oplus y) = [(x \wedge y) \vee (x \wedge y)] = [(x \square y) \vee (y \square x)] = (x \square y)$

est aisément confirmée et est une caractéristique bien connue pour les anneaux booléens et l'algèbre binaire. Ainsi, les deux opérations $\oplus$ et $\square$ sont identiques dans les anneaux booléens $\langle B^n, \oplus, \cdot \rangle$, et XOR est bien l'opération principale des modèles basés sur l'espace B^n. Ainsi, nous avons montré comment l'algèbre $\langle B^n, \wedge, \vee, \neg \rangle$ détermine l'anneau booléen $\langle B^n, \oplus, \cdot \rangle$. Cette étape nous rend capables d'appliquer la théorie algébrique des anneaux à l'étude des modèles dans B^n.

2.2 Propriétés fondamentales de XOR

La caractérisation de l'espace B^n et de ses structures nécessitait de connaître plusieurs propriétés de XOR. Le but de cette section consiste à donner une vue plus systématique de cette opération. La section 2.3 présente sa généralisation à l'intégrale binaire scalaire et à l'intégrale binaire vectorielle. La plupart de ces propriétés sont exprimées dans des théorèmes, des lemmes et des corollaires dont les preuves sont données dans Zaus ([ZA95c]). Le lecteur peut utiliser cette section comme un guide pour explorer XOR, ou comme référence pour les sections suivantes. Rappelons d'abord la définition de XOR :

Définition 2.1 $(x \oplus y) = [(x \wedge y) \vee (x \wedge y)] = (x \neq y)$

Théorème 2.1 L'opération $\oplus$ a les propriétés suivantes pour tout $x, y, z, w \in B^n$:

(2.1.1)	$x \oplus y = y \oplus x$	commutativité
(2.1.2)	$x \oplus 0 = x$	
(2.1.3)	$x \oplus x = 1$	
(2.1.4)	$x \oplus (y \oplus z) = (x \oplus y) \oplus z$	associativité
(2.1.5)	$(x \oplus y) \oplus (z \oplus w) = (x \oplus z) \oplus (y \oplus w)$	bisymétrie
(2.1.6)	$x \wedge (y \oplus z) = (x \wedge y) \oplus (x \wedge z)$	distributivité
(2.1.7)	$x \oplus x = 0$	
(2.1.8)	$x \oplus y = x \oplus z \Rightarrow y = z$	annulation
(2.1.9)	$1 \oplus y = x$	complément

Le Théorème 2.1 recouvre des propriétés structurales importantes de XOR. La bisymétrie du théorème (2.1.5) résulte de (2.1.1) et de (2.1.4) et est la propriété la plus importante de XOR, à cause du fait que l'on peut échanger les termes à gauche ce qui produit une permutation équivalente à droite. La bisymétrie correspond à la *loi de la conservation de l'entropie*, et un examen plus approfondi de XOR montre qu'il s'agit effectivement d'une opération préservant l'entropie. Par exemple, si x, y, z et w sont des signaux ou des vecteurs d'états, alors le vecteur résultant $u = (x \oplus y) \oplus (z \oplus w)$ demeure *invariant* si on réordonne les termes deux à deux. Ceci est généralisable au cas *n-symétrique*. XOR est manifestement une opération *isentropique*.

Des mesures élémentaires pour définir le poids des vecteurs binaires ou des distances entre les vecteurs sont introduites comme suit ([BASS83], [BOP81], [LAN92a]) :

Définition 2.2 La masse totale de x est $dim\ (x)$, le nombre de composantes de x.

Définition 2.3 La masse pesante $mp\ (x)$ est le nombre de 1 dans x, c'est-à-dire $\|x\|_1 = \Sigma^n_{i=1} x_i$.

Définition 2.4 La masse cachée $mc\ (x)$ est le nombre de zéros dans x, c'est-à-dire $\|x\|_0 = \Sigma^n_{i=1}\ 1 \oplus x_i$.

Les deux dernières mesures sont utilisées pour déterminer la majorité $M(x)$; c'est une fonction caractéristique définie par :

(11) $$M(x) = \begin{cases} 1 \text{ si } \|x\|_1 > \|x\|_0 \\ 0 \text{ si } \|x\|_1 \leq \|x\|_0 \end{cases}$$

La notation indique que la masse pesante est la *norme* de x, tandis que la masse cachée est la *conorme* de x, car la première est aussi la distance à l'origine 0, tandis que la dernière est la distance de son complément au sommet 1. Les deux mesures admettent des relativisations, le taux de norme ($\|x\|_1 \div n$) et le taux de conorme ($\|x\|_0 \div n$). Ces grandeurs sont parfois plus informatives que la seule majorité $M(x)$. Ceci sera démontré sur des exemples numériques. Pour établir une relation entre ces mesures et la métrique de B^n, nous avons besoin de la fonction de distance suivante.

Définition 2.5 Pour deux vecteurs quelconques $x, y \in B^n$, leur distance est définie par

(12) $$d(x,y) = \sum_{i=1}^{n} (x_i \oplus y_i) = \sum_{i=1}^{n} (x_i \neq y_i)$$

La définition 2.5 définit la *distance de Hamming* bien connue entre deux vecteurs binaires, c'est-à-dire le nombre de dimensions par

lesquelles x et y diffèrent. C'est un produit interne généralisé, appelé le *produit interne « somme-Ou_exclusif »* $x +.\neq y$ en APL. Le théorème suivant montre que la fonction de distance $d(x,y)$ a toutes les propriétés de la fonction commune de distance dans l'espace euclidien avec l'exception de la propriété d'être un nombre réel non négatif.

Théorème 2.2 Dans B^n la fonction de distance $d(x,y)$ satisfait les propriétés suivantes :

(2.2.1)	$x = y \rightarrow$	$d(x,y) = 0$	non négativité
(2.2.2)	$x \neq y \rightarrow$	$d(x,y) \neq 0$	positivité
(2.2.3)	$d(x,y) =$	$d(y,x)$	commutativité
(2.2.4)	$d(x,y) \leq$	$d(x,z) + d(z,x)$	inégalité du triangle

Nous sommes maintenant prêts à présenter quelques exemples numériques qui expliquent les concepts précédents à titre comparatif.

Exemples La Table 1 concerne n = 6 avec B^6, x =100100 et y =111101

1	*dim* (x)	$d(x) = 6$	*dim* (y)	$d(y) = 6$
2	*norme* (x)	$\|\|x\|\|_1 = 2$	*norme* (y)	$\|\|y\|\|_1 = 5$
3	*conorme* (x)	$\|\|x\|\|_0 = 4$	*conorme* (y)	$\|\|y\|\|_0 = 1$
4	*distance* (x,y)	$\Sigma^n_{i=1}\, x_i \oplus y_i = 3$	*norme* $(x \oplus y)$	$\|\|x \oplus y\|\|_1 = 3$
5	*taux de norme (x)*	$\|\|x\|\|_1 \div 6 = 0{,}33$	*taux de norme (y)*	$\|\|y\|\|_1 \div 6 = 0{,}83$
6	*taux de conorme (x)*	$\|\|x\|\|_0 \div 6 = 0{,}67$	*taux de conorme (y)*	$\|\|y\|\|_0 \div 6 = 0{,}17$
7	*majorité* (x)	$M(x) = 0$	*majorité* (y)	$M(y) = 1$
8	*orthogon.*(x,y)	$x\ fy = 1 \leftrightarrow d_{x,y} = n/2$	$d(x,y) = 3$	$x\ fy = 1$
9	*complément* (x)	$1 \oplus x =$ 011011	*complément* (y)	$1 \oplus y =$ 000010

Table 1

Remarquer que la norme et la conorme dans les exemples 2 et 3 ont pour somme la dimension de x dans l'exemple 1. L'exemple 4 montre que les distances et les normes entre deux vecteurs binaires sont identiques. Les exemples 5 - 7 montrent que les mesures relatives peuvent être plus informatives que la fonction majorité *M*. L'exemple 8 montre la condition d'orthogonalité pour des vecteurs binaires et l'exemple 9 les compléments de x et de y.

Le théorème suivant établit la propriété de *réversibilité* de XOR. En patrticulier, une opération est logiquement réversible si on peut toujours déduire les données à partir des résultats. L'intersection ($\wedge$) et l'union ($\vee$), par exemple, sont des opérations *irréversibles*, tandis que le complément ($\bar{}$) est réversible, c'est une involution. La réversibilité de XOR va jouer un rôle important pour l'intégrale vectorielle binaire ci-dessous, mais discutons d'abord de l'importance du Théorème 2.3 et du Corollaire 2.1 en ce qui concerne les *différentielles binaires*.

Théorème 2.3 Si $x,y \in B^n$ avec $x \oplus y = z$, alors $y = x \oplus z$ et $x = y \oplus z$.

Corollaire 2.1 Si $x,z \in B^n$, alors il existe exactement un seul $y \in B^n$ de telle sorte que $x \oplus y = z$.

Le théorème et son corollaire montrent que B^n a trois entités distinctes : (a) des sommets $x \in B^n$, (b) des directions $dx \in B^n$ et (c) des côtés $(x, dx) \in B^n$. Considérez maintenant les tables 2a et 2b ci-après :

x_i	y_i	$z_i = x_i \oplus y_i$
0	0	
0	1	0
1	0	
1	1	1
		1
		0

x_i	dx_i	$x_i^* = x_i \oplus y_i$
0	0	
0	1	0
1	0	
1	1	1
		1
		0

Table 2a **Table 2b**

Le fait que les deux tables présentent la table de vérité de XOR n'est pas le point central. L'important c'est que la table 2a montre la *partie-si* du théorème 2.3 pour les composantes $x_i, y_i, z_i \in x, y, z \in B^n$ de sorte que x_i et y_i sont considérées comme des variables binaires indépendantes. La preuve de $y_i = x_i \oplus z_i$ est basée sur l'implication :

$((x_i \oplus y_i = z_i)$ et $(x_i \oplus y_i = z_i)) \rightarrow ((x_i \wedge y_i) = (x_i \wedge z_i)$ et $(x_i \wedge y_i) = (x_i \wedge z_i))$.
Ainsi, $y_i = (x_i \wedge z_i) \vee (x_i \wedge z_i) = (x_i \oplus z_i)$. De la même manière, on prouve que $x_i = y_i \oplus z_i$. De là, le fait d'échanger l'entrée et la sortie (les données et les résultats) montre que XOR est une opération réversible.

Le Corollaire 2.1, d'un autre côté, révèle l'existence de *différentielles binaires* uniques. Ceci est le message de la table 2b ci-dessus. Le vecteur $z = x \oplus y$ dans le corollaire 2.1 est appelé la différence de x et de y . Maintenant, soient $x, x^* \in B^n$ deux vecteurs distincts, alors $dx = x \oplus x^*$ est un nouveau vecteur $dx = (dx_1, dx_2, ..., dx_n)$. Il est engendré point à point par XOR et est appelé la différentielle dx. Puisque dx définit une *direction* dans B^n , il s'agit d'un vecteur d'orientation qui décrit le changement de x en x^* à chaque coordonnée $x_i \in x$. Avec x, il constitue un côté (x, dx) dans B^n qui pointe en x^*. Finalement, puisque $dx = x \oplus x^*$ implique $x^* = x \oplus dx$ par le théorème 2.3, nous reconnaissons les relations entre les tables 2a et 2b comme indiqué pour les composantes x_i , dx_i et x_i^* . Ceci établit l'existence de différentielles binaires uniques.

Pour donner un autre exemple soit x valant 0101 et soit x^* valant 1100. Alors dx = 1001. Remarquer que dx contient en chaque coordonnée dx_i la parité des coordonnées respectives x_i et x_i^*. Ceci montre que chaque différentielle binaire est aussi la *différentielle de parité* de x, et donc l'*inverse* de l'*intégrale* binaire vectorielle, définie plus loin.

Le corollaire 2.1 établit aussi que tout élément donné de B^n forme une base métrique pour B^n. Ceci signifie qu'il n'y a pas de triplets isocèles de vecteurs distincts deux à deux dans B^n , ce qui, à son tour, est important pour le groupe des déplacements dans B^n. Cela sera considéré dans la section 2.4 sur les déplacements, les produits internes, et le géniton.

2.3 Fondements des Opérateurs XOR (Ou_exclusif) Généralisés

Tournons-nous maintenant vers les opérateurs XOR généralisés, c'est-à-dire l'intégrale binaire scalaire et l'intégrale binaire vectorielle. Ces concepts ont besoin d'une notation mathématique rigoureuse en accord avec ce qui vient d'être développé formellement. Toutefois, nous éprouvons de la difficulté, parce que ni les mathématiques conventionnelles ni la terminologie de l'informatique n'offrent une notation consistante en ce qui concerne les opérateurs monadiques généralisés. C'est pourquoi nous prendrons un compromis en introduisant une notation partiellement nouvelle, mais logiquement saine. La table 3 ci-dessous prévoit ce à quoi le lecteur doit s'attendre en ce qui concerne le XOR généralisé.
La rangée 1 de la table 3 présente le compromis pour une notation standard en informatique et la rangée 2 présente la notation APL standard.

	1	2	3	4
	notation standard	forme duale de (1,1)	notation standard	forme duale de (1,3)
1	somme de XOR $z = \oplus^n_{i=1} x_i$ intégrale binaire scalaire par différent	somme de non-XOR $z = \square^n_{i=1} x_i$ intégrale binaire scalaire par égal	intégration de XOR $z = \square^n_{i=1} x_i$ intégrale binaire vectorielle par différent	intégration de non-XOR $z = \square^n_{i=1} x_i$ intégrale binaire vectorielle par égal
	notation APL	notation APL	notation APL	notation APL
2	opérateur monadique $z \leftarrow \neq/ x$ réduction par différent	opérateur monadique $z \leftarrow =/ x$ réduction par égal	opérateur monadique $z \leftarrow \neq\backslash x$ balayage par différent	opérateur monadique $z \leftarrow =\backslash x$ balayage par égal
	réduction par XOR	réduction par non-XOR	balayage par XOR	balayage par non-XOR

Table 3

Des détails sur la table 3 vont suivre pas à pas. Ce sur quoi les opérateurs agissent est formellement appelé des *opérandes*. Un opérateur *f* prend une fonction □ et la convertit en quelque chose de différent appelé fonction dérivée $z \leftarrow \square/ x$ pour un certain argument *x*. Dans la table 3, l'opération binaire XOR est la fonction logique ≠ , et le fait de la soumettre à l'opérateur de réduction / produit la fonction dérivée $z \leftarrow \neq/ x$ c'est-à-dire la réduction généralisée par XOR. De même, si ≠ est soumis à l'opérateur de balayage cumulatif \ , nous obtenons la fonction dérivée $z \leftarrow \neq\backslash x$, c'est-à-dire l'opérateur de balayage par XOR généralisé. Ceci s'applique pour n'importe quelle opération binaire □ , et cela explique la notation de la rangée 2 de la table 3 ci-dessus. Remarquer que chaque opérateur a son opérateur dual en vertu des lois de de Morgan [1]. Les propriétés formelles de ces opérateurs sont décrites dans ce qui suit.

Définition 2.6 L'intégrale binaire scalaire (IBS) est la somme par XOR.

$$z = \overset{n}{\underset{i=1}{\oplus}} x_i = (...(x_1 \oplus x_2) \oplus x_3) \oplus ...) \oplus x_n) = \tag{13}$$

$$= x_1 \oplus x_2 \oplus \ ... \ \oplus x_n \ .$$

Théorème 2.4 [2]

$$\overset{n}{\underset{i=1}{\oplus}} x_i = z = \begin{cases} 1 & \text{ssi} \| x \|_1 = \textit{pair} \\ 0 & \text{ssi} \| x \|_1 = \textit{impair} \end{cases}$$

Nous avons utilisé un moyen subtil de définir l'IBS pour nous assurer que le lecteur réalise la différence avec l'intégrale binaire vectorielle de la définition 2.7 ci-dessous. A partir de la définition 2.6 et du théorème 2.4, il s'ensuit que l'IBS détermine la parité pour tout $x \in B^n$, c'est-à-dire qu'un vecteur *x* a pour parité 1 ssi la norme de *x* est impaire, sinon *x* a pour parité 0. Par exemple, le vecteur $x = 100100$ des exemples de la table 1 a pour parité 0, tandis que $y = 111101$ a pour parité 1. Le corollaire 2.2 ci-dessous montre que l'IBS est équivalente à la somme *modulo 2* de $x \in B^n$.

Corollaire 2.2

$$\oplus^{n}_{i=1} x_i = z = 2|_0 \Sigma^{n}_{i=1} x_i$$

Ceci signifie que $\oplus^{6}_{i=1} 111101 = (2|_0 5) = 1$ et que $\oplus^{6}_{i=1} 100100 = (2|_0 2) = 0$. Remarquer toutefois que l'IBS est un opérateur booléen non-absorbant non- numérique, tandis que la somme *modulo 2* est basée sur l'opération numérique d'addition et est donc un opérateur numérique.

Le théorème 2.5 ci-dessous est un compagnon de la loi de de Morgan et montre que la forme duale de l'IBS, la somme par *non*-XOR dans l'entrée (1,2) de la

table 3, puisque $z \leftarrow (=/ x) = (\overline{\neq/ x})$. La somme par *non*-XOR est ainsi la somme binaire d'équivalence, comme le montre le théorème 2.5 .

Théorème 2.5

$$[1 \oplus \oplus^{n}_{i=1} 1 \oplus x_i] = \square^{n}_{i=1} x_i$$

Le corollaire 2.3 ci-dessous est la version duale du corollaire 2.3 ci-dessus et résulte directement du théorème 2.5 . Il montre aussi comment agit l'opérateur du théorème 2.5 sur ses arguments.

Corollaire 2.3

$$\square^{n}_{i=1} x_i = \overline{2|_0 \Sigma^{n}_{i=1} x_i}$$

En particulier, $\square^{6}_{i=1} 111101 = [(1=1=1=1=0=1)=0] = [(\overline{2|_0 1}) = 0]$, et de manière

analogue $\square^{6}_{i=1} 100100 = [(1=0=0=1=0=0)=1] = [(\overline{2|_0 4}) = 1]$. D'où le fait que la somme binaire par l'équivalence détermine si *x* et *y* sont chacun de parité *paire*, ou non.

Jusqu'ici nous avons fourni la base formelle des colonnes 1 et 2 de la table 3 pour l'intégrale binaire scalaire, la somme par XOR et son dual, la somme binaire par l'équivalence. Le prochain opérateur est l'intégrale binaire vectorielle, l'intégration par XOR de l'entrée (1,3) de la table 3 ci-dessus. On l'appelle balayage par l'inégalité (« unequal-scan » en anglais) en APL, le balayage par préfixe (« prefix-scan ») en informatique, et intégrale de parité en logique de la parité ([LAN92a], [ZA95b]).

Définition 2.7 L'intégrale binaire vectorielle (IBV) est la somme **cumulée** par XOR.

$$\overset{n}{}$$

(14) $\square x_i = (x_1, (x_1 \oplus x_2), \ldots, (x_1 \oplus x_2 \oplus \ldots \oplus x_n)) = z_1, z_2, \ldots, z_n$ $i=1$

de sorte que

$x_i = z_1$
$x_1 \oplus x_2 = z_2$
$x_1 \oplus x_2 \oplus x_3 = z_3$
.
.
.
$x_1 \oplus x_2 \oplus \ldots \oplus x_n = z_n$

L'IBV de la définition 2.7 est une généralisation directe de XOR, mais elle mérite quelques commentaires, quelques explications ainsi que des illustrations informatiques, avant d'en faire ressortir plusieurs théorèmes et corollaires.

La premier fait important au sujet de l'IBV est qu'elle détermine *toutes les sommes par XOR* pour un argument vectoriel x. Maintenant, pensez à des statistiques et à une distribution de probabilités telle que 0,02 0,04 0,06 0,11 0,20 0,31 0,16 0,07 0,02 0,01 , et calculez sa fonction de distribution cumulée discrète (*f.d.c.*) par l'expression (15) :

$p_1 = 0{,}02$

$0{,}02 + 0{,}04 = 0{,}06$

(15) $F(x) = \Sigma\, p_i$

$0{,}02 + 0{,}04 + 0{,}06 = 0{,}13$
$x_i < x$
$0{,}02 + 0{,}04 + 0{,}06 + 0{,}11 + \ldots + 0{,}01 = 1$

Dans les deux cas, le résultat n'est pas un nouveau scalaire. C'est un nouveau n-tuple dont les composantes sont les valeurs cumulées réduites, les sommes réduites par XOR concernant l'IBV, et les termes réduits par sommation concernant $F(x)$ = 0,02 0,06 0,12 0,23 0,43 0,74 0,90 0,97 0,99 1 .

Ensuite, soit $x \in B^n$. En prenant d'abord la première composante, puis les deux premières composantes, puis les trois premières composantes, et ainsi de suite, nous obtenons les vecteurs $1x = x_1$, $2x = x_1,x_2$, $3x = x_1,x_2,x_3$, etc... On les appelle *préfixes* de x en informatique. L'IBV de la définition 2.7 produit donc la *réduction* par XOR de tous les préfixes de l'argument vectoriel x, et c'est pourquoi l'opérateur est aussi appelé le *balayage des préfixes* (« *prefix-scan* »). Il s'agit d'un opérateur spécial dans la « connection machine » de Hillis pour le calcul parallèle ([HIL85]), mais il est beaucoup plus général que cela. Nous allons le démontrer maintenant pour de simples vecteurs binaires et pour des IBV itérées dans le but de faire ressortir le concept d'*intégration de parité*.

Pour voir comment l'opérateur $\square_{i=1}^{n} x_i$ agit sur les éléments de $x \in B^n$, nous

choisissons le simple vecteur $x = 10000000 \in B^8$. La table 4a ci-dessous montre comment l'opérateur engendre l'intégrale binaire vectorielle de x, de haut en bas par une série de réductions par XOR. Le résultat est l'intégrale vectorielle z = 11111111. Ce vecteur est l'*intégrale de parité* de x, puisque chaque *étape de calcul* détermine la parité par paires entre les termes. Chaque composante de z est une parité et la dernière indique la parité du vecteur x, l'argument original. Ceci vaut pour tout élément $x \in B^n$.

x = 1 0 0 0 0 0 0 0
1 = 1 1 ⊕ 0 = 1 1 ⊕ 0 ⊕ 0 = 1 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 = 1 1 ⊕ 0 ⊕ 0⊕ 0 ⊕ 0 ⊕ 0 = 1 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 = 1 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕

intégrales	1 0 0 0 0 0 0 0	$x^{(0)}$ & propagation
intégrale 1	1 1 1 1	$x^{(1)}$: onde 1
intégrale 2	1 1 1 1	$x^{(2)}$: onde 2
intégrale 3	1 0 1 0	$x^{(3)}$: onde 3
intégrale 4	1 0 1 0	$x^{(4)}$: onde 4
intégrale 5	1 1 0 0	$x^{(5)}$: onde 5
intégrale 6	1 1 0 0	$x^{(6)}$: onde 6
intégrale 7	1 0 0 0	$x^{(7)}$: onde 7
intégrale 8	1 0 0 0	$x^{(8)}$: onde 8
	1 1 1 1	
	0 0 0 0	
	1 0 1 0	
	0 0 0 0	

0 = 1			1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0	
		successives		d'ondes

Table 4a **Table 4b**

Si l'opérateur $\sum_{i=1}^{n} x_i$ est appliqué à des matrices $n \times m$ au lieu de vecteurs, alors on doit spécifier l'axe d'intégration.

(16) (16.1) $M^*_{ij} = \sum_{i=1}^{n} M_{ij}$ (16.2) $M^*_{ij} = \sum_{j=1}^{m} M_{ij}$

Dans (16.1) ci-dessus, l'opérateur agit sur les colonnes, tandis que dans (16.2), il agit sur les lignes et détermine leurs intégrales de parité respectives. La table 4b, d'autre part, démontre ce qui arrive quand nous itérons l'opérateur successivement selon :

(17) $$x^{(t+1)} = \sum_{i=1}^{n} x_i(t) \ ,$$

où t dénote la variable d'itération. Le résultat est une matrice de parité 8×8 avec une périodicité $t = n$, puisque le vecteur d'entrée original $x(0)$ réapparaît comme la *ne* intégrale de parité. La matrice de parité de la table 4b, appelée *pariton* ([LAN91a,b], [LAN92a], [ZA94b]) sera considérée à nouveau presque dans chacune des sections suivantes à cause de son importance.

Observer que l'opérateur dans l'équation (17) agit comme un générateur d'ondes asymétrique, en ce sens qu'il propage des différences élémentaires de gauche à droite, et de haut en bas par itérations successives. Ainsi, c'est, entre autres choses, le bloc de construction des machines à rétroaction de parité pour modéliser et analyser les milieux excitables ([ZA95b]). D'autres détails relatifs à l'IBV vont suivre dans les sections ultérieures, alors, retournons à ses propriétés formelles concernant les colonnes 3 et 4 de la table 3 .

Théorème 2.6

$$\sum_{i=1}^{n} x_i = (z_1, z_2, ..., z_n) \quad et \quad z_n = \begin{cases} 1 \text{ ssi } ||x||1 = impair \\ 0 \text{ ssi } ||x||_1 = pair. \end{cases}$$

Le théorème ci-dessus est un cousin du théorème 2.4 et établit que la parité de $x \in B^n$ est déterminée dans la dernière composante de l'intégrale de parité z. Ainsi, le vecteur x a pour parité 1 ssi z_n vaut 1, sinon x a pour parité 0. Donc l'IBV contient le contrôle de parité de son argument vectoriel dans le dernier item de z. La table 4a ci-dessus le montre explicitement pour chaque intégrale de parité de la matrice 8×8.

Définition 2.8 La contre-partie cumulative de $\Sigma^{n}_{i=1} x_i$ est la *somme de tous les partiels*

(18) $$\sum_{i=1}^{n} x_i = (x_1, (x_1 + x_2) \ldots (x_1 + \ldots + x_n)) \ .$$

Observer que la fonction de distribution cumulative dans l'équation (15) est bien définie pour des variables aléatoires discrètes. Il n'existe, toutefois, aucune notation pour la somme de tous les partiels, c'est pourquoi nous nous sommes décidés pour celle-ci [3].

Le corollaire 2.4 ci-dessous est le pendant du corollaire 2.2 concernant l'intégrale binaire scalaire. Elle montre l'équivalence entre l'IBV et la *somme cumulée modulo 2*.

Corollaire 2.4

$$\sum_{i=1}^{n} x_i = z = 2|_0 \sum_{i=1}^{n} x_i$$

Ainsi, $\sum_{i=1}^{n} 111101 = (2|_0 123445) = 101001$, et, de manière correspondante,

n

$\square_{i=1}$ 100100 = (2|$_0$ 111222) = 111000. L'IBV est, comme l'IBS, un opérateur booléen non numérique et ne nécessite pas de lourd calcul numérique (« bit crunching ») *modulo 2*.

Le théorème 2.7 ci-dessous est encore un parent de la loi de de Morgan. Il montre que la forme duale de l'IBV, l'intégration par non-XOR dans l'entrée (1,4) de la table 3, est l'équivalent de l'opérateur *balayage par égal* dans l'entrée (2,4) de la

table 3, puisque $z \leftarrow (=\backslash x) = \overline{(\neq\backslash x)}$.

Théorème 2.7

$$[1 \oplus \square_{i=1}^{n} 1 \oplus x_i] = \square_{i=1}^{n} x_i$$

L'intégration par non-XOR est donc la somme d'équivalence cumulée binaire.

Nous parvenons finalement au corollaire 2.5, la contre-partie du corollaire 2.3 . Cela peut frapper le lecteur que ce corollaire soit aussi étrangement abstrait, mais nous en avons besoin pour compléter la table 3, à cause de la dualité des opérateurs.

Corollaire 2.5

$$\square^{n}_{i=1} x_i = 2|_0 \overline{\square^{n}_{i=1} x_i}$$

Il montre comment l'opérateur du théorème 2.7 agit sur ses arguments. Il suffit d'un seul exemple :

$$\square^{n}_{i=1} \overline{111101} = [1,(1=1), \ldots,(1=1=1=1=0=1)] = \overline{[2|_0 000011]} = 111100.$$

Théorème 2.8

$$\exists x_i \in x : x_i \neq x_{ij} \rightarrow \square^{n}_{i=1} x_1, x_2, \ldots, x_n \neq \square^{n}_{i=1} x_n, x_{n-1}, \ldots, x_1 \; ; \; ij=1,2,\ldots n; \; (i \neq j)$$

Le théorème 2.8 ci-dessus établit que l'IBV est asymétrique pour tout $x_i \in B^n$ excepté l'origine (000...00) et le sommet (111...11). Cela signifie que s'il existe un simple x_i qui soit différent de tous les autres éléments possiblement équivalents dans x, alors une rotation de x autour de sa première coordonnée produit un résultat différent. Ceci se produit déjà pour le vecteur à 2 composantes $x = x_1, x_2, = 1\ 0$ et sa rotation 0 1 :

(19)
$$\boxed{\begin{array}{l}\square^{2}_{i=1} 1\ 0 \\ = 1\ 1\end{array}} \quad \neq \quad \boxed{\begin{array}{l}\square^{2}_{i=1} 0\ 1 \\ = 0\ 1\end{array}}$$

Ceci vaut pour toute dimension n de x. L'importance du théorème 2.8 est que l'opérateur est un propagateur asymétrique de différences symétriques. Le fait que l'opérateur soit également isentropique et réversible résulte de la table 4a et du mécanisme de rétroaction de l'équation (17). A nouveau, le vecteur à deux composantes x = 1 0 et son intégrale de parité intégrée clarifient la situation d'un simple coup d'œil.

(20)
$$\boxed{\begin{array}{l}\square^{2}_{i=1} 1\ 0 \\ = 1\ 1\end{array}} \quad \rightarrow \quad \boxed{\begin{array}{l}\square^{2}_{i=1} 1\ 1 \\ = 1\ 0\end{array}}$$

Le point important de l'implication contenue dans (20) est que cette itération double engendre la structure la plus importante de la logique de la parité, la matrice de parité :

(21)
$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

appelée *pariton génétique*, ou *géniton* en abrégé ([LAN92a]). Ses propriétés formelles sont traitées dans la section suivante, une fois introduits les concepts des déplacements et des produits internes binaires. Les deux opérateurs, l'IBS aussi bien que l'IBV, seront exploités de façon plus détaillée dans les sections concernant la logique de parité appliquée.

2.4 Mouvements, Produits Internes et Géniton

Le triangle de Pascal modulo 2 et son remplissage en carré de Pascal d'une part, ainsi que le groupe des déplacements concernant les matrices de parité binaires d'autre part, vont jouer un rôle central dans les sections suivantes. La même chose vaut pour les produits internes mettant en jeu XOR. Le fait que $\langle B^n, d \rangle$ soit un espace métrique de diamètre n a déjà été établi dans la définition 2.1 et le théorème 2.2 de la section 2.2 . Pour établir le concept des déplacements dans B^n nous avons besoin du concept d'isométrie, de la propriété involutive des déplacements préservant la distance, de l'identité et de la propriété des déplacements préservant les compléments. Les deux plus importantes

classes de déplacements seront les rotations de vecteurs et les réflexions de tableaux dans B^n pour des transformations effectives, ainsi que le groupe des opérateurs de transformation.

Définition 2.9 Un déplacement φ entre deux espaces métriques est une isométrie s'il préserve la distance, c'est-à-dire $\forall\, x,y \in X : (\varphi : X \to Y) \to d[\varphi(x), \varphi(y)] = d(x,y)$.

Noter que la définition 2.9 met en jeu deux espaces métriques. Une isométrie définie sur un seul espace *dans* ou *sur lui-même* est une application rigide. Pour B^n nous obtenons :

Définition 2.10 Une application préservant la distance sur $\varphi : B^n \to B^n$ est un *déplacement*.

Le fait qu'un déplacement de B^n soit biunivoque résulte du théorème 2.2.1 de la section 2.2, puisque $d(x,y) = 0$ ssi $x = y$. Beaucoup plus important est le lemme suivant :

Lemme 2.1 Une application φ préservant la distance dans $\varphi : B^n \to B^n$ est *involutive*.

Puisque les involutions jouent un rôle-clé en logique de la parité et dans les calculs réversibles, nous insérons la démonstration du lemme 2.1. Si $x \in B^n$, dénotons $\varphi(\varphi(x))$ par y. Puisque φ préserve la distance, $d(x, \varphi(x)) = d(\varphi(x),y)$. Ainsi, chaque vecteur binaire x,y a la même distance de $\varphi(x)$. Toutefois, puisque $\varphi(x)$ forme une base métrique pour B^n selon le théorème 2.3 et le corollaire 2.1, il s'ensuit que $y = x$. D'où pour tout $x \in B^n$: $\varphi(\varphi(x)) = x$. Bien que ceci ait l'air trivial, le lemme 2.1 a d'importantes implications pour les transformées dans B^n (par exemple les *transformées de Shegalkin* et les *transformées de Langlet* ([BOP81], [LAN92a], [ZA94b])).

Corollaire 2.6 $\langle B^n, d \rangle$ est un espace métrique monomorphe avec un diamètre n, de là toute application préservant la distance φ de B^n dans lui-même est un *déplacement*.

Corollaire 2.7 Si un déplacement φ de B^n laisse un élément fixe, c'est une *identité*.

Lemme 2.2 Si φ est un déplacement quelconque de B^n et que $x \in B^n$, alors $\varphi(x) = \overline{\varphi}(x)$, c'est-à-dire que tout déplacement φ de $x \in B^n$ *préserve son complément*.

Pour nous, les déplacements les plus importants sont les rotations et les réflexions. Une distinction significative entre une rotation autour d'un point dans le plan et une réflexion sur une droite dans le plan est qu'une rotation n'a qu'un seul point invariant, le centre de rotation, tandis qu'une réflexion a une droite de points invariants, l'axe de réflexion. Un autre type de déplacement, la translation, n'a aucun point invariant. La seule exception pour laquelle tous les points sont invariants est la transformation identité. Maintenant, l'ensemble de toutes les rotations autour d'un point ou d'une coordonnée p forme un groupe. La même chose vaut pour les réflexions. La réflexion sur une droite donnée l forme un groupe avec la transformation identité, puisque la réflexion sur l, suivie par une autre réflexion sur l, nous ramène au point de départ. Ces concepts couvrent l'algèbre binaire et l'espace B^n. Nous allons le démontrer pour le *géniton* et pour le groupe des réflexions, et ensuite pour des matrices de parité $n \times n$, c'est-à-dire les paritons.

Définition 2.11 Le **géniton** est la matrice de parité 2×2 symétrique $G = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

Théorème 2.9 Le géniton et ses rotations de 90° dans le sens des aiguilles d'une montre ou en sens inverse forment un groupe d'opérateurs matriciels c'est-à-dire de transformations.

Théorème 2.10 Le géniton et ses réflexions horizontale h, diagonale d et verticale v forment un groupe d'opérateurs matriciels c'est-à-dire de transformations G, Gh, Gd, Gv, GU **et** GU [NdT].

Pour pouvoir donner des détails sur le théorème 2.10, nous avons besoin du produit matriciel binaire et de la puissance d'une matrice [4].

Définition 2.12 Le produit matriciel binaire (PMB [NdT]) est le produit interne Ou_exclusif-ET (« XOR-AND ») :

$$(22) \qquad \boldsymbol{Z} = (\boldsymbol{X} \,\square\, \boldsymbol{Y}) = (\boldsymbol{X} \overset{\wedge}{\oplus} \boldsymbol{Y}) = \overline{(\boldsymbol{X} \overset{\vee}{=} \boldsymbol{Y})}$$

Le PMB est défini en termes des lois de de Morgan ([IVE62]), et la notation signifie que ce produit met en jeu les opérations $\oplus$ et $\wedge$ de sorte que les termes de X et de Y sont d'abord liés par conjonction, puis par XOR. Le produit dans (22) est la contre-partie *modulo 2* du produit interne de matrices habituel en algèbre. Les deux se distinguent très bien l'un de l'autre en APL par les notations (**X≠.∧Y**) et (**X+.×Y**). La

nomenclature standard concernant le produit interne XOR-ET d'une matrice binaire **X** mn et d'une matrice binaire **Y** np est donnée par :

$$i = 1, 2, ..., m$$

$$z_{ik} = (x_{i\,1} \wedge y_{1k}) \oplus (x_{i\,2} \wedge y_{2k}) \oplus \cdots \oplus (x_{in} \wedge y_{nk}) = \oplus^{n}_{r=1}\, x_{ir} \wedge y_{rk} :$$

$$k = 1,2, ..., p,$$

dont le résultat est une matrice binaire **Z** mp. La notation se simplifie quelque peu pour des matrices binaires carrées, mais la logique de la définition 2.12 devrait être transparente maintenant. Le produit binaire en puissance PBP est simplement un produit matriciel binaire n-uple.

La table 5a résume les propriétés essentielles du géniton et du groupe de réflexion (théorème 2.10), tandis que la table 5b présente quelques détails sur les PMB et sur les PBP.

$G = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$		*G* est un tableau symétrique, périodique, auto-organisé, isentropique, auto-similaire, en moyenne semi-corrélé, donc fractal avec un bruit en 1/*f* ou bruit rose. *G* est un opérateur de symétrie ternaire.
$Gh = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$		*Gh* est la réflexion horizontale (*h*) le long de (x_{11}, x_{12}) de *G* ci-dessus. *Gh* est auto-inverse en ce qui concerne ⊕ et ∧. Donc *Gh* est involutif. *Gh* est non-symétrique, et, donc, un opérateur de symétrie binaire.
$Gd = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$		*Gd* est la réflexion horizontale (*d*) le long de (x_{21}, x_{12}) de la matrice *G*. *Gd* est la matrice binaire, carré matriciel de *G*, et est donc symétrique. *Gd* est comme *G* un opérateur de symétrie ternaire, avec pour cube *GU*.
$Gv = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$		*Gv* est la réflexion verticale (*v*) le long de (x_{12}, x_{22}) de la matrice *G*. *Gv* est, comme *Gh*, auto-inverse pour ⊕ et ∧, et est donc involutif. *Gv* est, comme *Gh*, non-symétrique et un opérateur de symétrie binaire.
$GU = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$		La matrice-unité est le carré, ou le produit matriciel, de *Gh* ou de *Gv*. *GU* est à son tour le double produit matriciel binaire de *G* ou *Gd*, c-à-d. *GU* est le cube de *G* ou de *Gd*, et est le carré de *Gh* ou de *Gv*.
$\overline{GU} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$		$\overline{GU}$ est le complément de la matrice-unité *GU*. Noter le lemme 2.2 sur la préservation du complément. Il est valable pour tous les opérateurs ci-dessus, matriciels, de transformation ou de symétrie.

Table 5a

La table 5b ci-dessous est surtout centrée sur *G* et sur sa réflexion verticale *Gv*. Les détails concernant *G* sont valables pour *Gd*, et ceux concernant *Gv* sont valables pour *Gh*. Chaque rangée de la table 5b est une instance du théorème 2.10 sans que l'on montre la totalité de ces instances. Nous utilisons la notation simplifiée de la définition 2.12.

calcul	signification	propriétés
$G^{T} = G$	transposée de *G*	*G* est symétrique
$Gv^{T} = Gh$	transposée de *Gv*	*Gv* est non-symétrique
$G \square G = Gd$	PMB de *G*	*Gd* est le carré de *G*
$Gv \square Gv = GU$	PMB de *Gv*	*Gv* est auto-inverse
$G \square G \square G = GU$	PBP de *G*	*GU* est le cube de *G*
$Gv \square G = \overline{GU}$	PMB de *Gv*,*G*	complément de *GUv*
$G \square Gv =$	PMB de	transformée horizontale de

Gh	*G*,*Gv*	*G*
G □ *Gh* = *Gv*	PMB de *G*,*Gh*	transformée verticale de *G*

Table 5b

Les deux tables sont des *objets d'étude* pour les matrices de parité en général. Il vaut mieux s'intéresser en détail aux propriétés du géniton, car ces propriétés valent pour des matrices de plus grande dimension n × n, comme nous le montrerons dans les sections suivantes. Nous refermons maintenant ce survol, plutôt condensé, des fondements mathématiques et nous allons aborder un autre domaine des fondements de la logique de parité, l'analyse binaire du signal.

Bibliographie

AIG93 : Aigner, M. 1993 *Diskrete Mathematik,* Vieweg, Braunschweig.

BAS83 : Basilevsky, A. 1983 *Applied Matrix Algebra in the Statistical Sciences,* North-Holland, Amsterdam.

BOP81 : Bochmann, D. & Posthoff, Ch. 1981 Binäre dynamische Systeme, Oldenbourg, München.

HIL85 : Hillis D. *The Connection Machine*, MIT Press, Cambridge, USA.

IVE62 : Iverson, K.E. 1962. ***A P**rogramming **L**anguage,* John Wiley & Sons, New York.

IVE78 : Iverson, K.E. 1978-9 Notation as a Tool of Thought. Turing Award Lecture of the ACM. Reprinted in McDonnel,E., 1981. *A Source Book in **APL***, Palo Alto Press, Palo Alto.

LAN91a : Langlet G.A. 1991 Les Génitons : Variations sur Pascal, Sierpinski et Fibonacci, APL-CAM Journal (Belgique), Vol. 13, N° 2, 399-421.

LAN91b : Langlet G.A. 1991 Paritons and Cognitons : Towards a new theory of information. APL-CAM Journal, Vol 13, N° 3, 709-743.

LAN92a : Langlet G.A. 1992 Towards the Ultimate APL-T.O.E. APL Quote-Quad, ACM Press, USA, 23, 1, July 1992, 118-132.

LAN93 : Langlet G.A. 1993 Symétrie, Forces et Phénomènes. APL-CAM Journal, Vol. 15, N° 1, 57-80.

LOC89 : Lochner, H. 1989, *APL2 Handbuch*, Springer, Heidelberg.

WAT92 : Watenabe, H., Symon, J.R., Detloff, W.D. & Yount, K.E. 1992 VLSI fuzzy chip and inference accelerator board. in Zadeh, L. & Kacprzyk, K. (Eds.) 1992 *Fuzzy Logic for the Management of Uncertainty*. Wiley & Sons, Chichester, G.B.

ZA94b : Zaus M., 1994 Theoretische und Andgewandte Paritätslogik. APL-CAM Journal, Vol. 16, N° 3, 447-469. Disponible en français : La Logique de la Parité, Théorique et Appliquée, Les Nouvelles d'APL, N° 12-13, 42-66 1994, Paris.

ZA95 : Zaus, M. 1995 Artificial Evolution in Cognitive Science and Technology, Interim-Report, 160 p. Chapman & Hall, London.

ZA95b : Zaus, M., Parity Integration, Excitable Media and Neural Computing, APL-CAM Journal, Vol. 17, N° 3, 409-432. Disponible en français : Intégration de Parité, Milieux Excitables et Neuro-Calcul, Les Nouvelles d'APL, N° 15, 46-74, 1994, Paris (disponible sur Internet : http://www.ensmp.fr/~scherer/langlet).

ZA95c : Zaus, M., *Artificial Evolution in Cognitive Science and Technology*, en préparation pour : Chapman & Hall, London, ISBN 0-412486601.

Note : Le rapport original (77p). de M. Zaus « Studies in the Foundations of Parity Logic » est disponible, soit à l'adresse suivante :

Institut für Kognitionsforschung, FB 5 - A6, Universität Oldenburg, P.O. Box 2503, Allemagne (Fax : {49} 441-7985170),

soit, en France, auprès de : Mme Derost, Bibliothèque SCM, CEA-Saclay, 91191-Gif sur Yvette Cedex.

Ce travail a été financièrement soutenu par la Fondation Scientifique Allemande (DFG), Bourse : Sche 298/5-2. Il a été effectué au sein du Groupe de Recherche Interdisciplinaire sur les Sciences Cognitives, Université de Brême & Université d'Oldenburg.

La seconde partie du rapport paraîtra en français sous forme d'article dans le numéro suivant des Nouvelles d'APL sous le titre « Représentations Analytiques des Signaux Binaires ».

Pensée à méditer :

« *L'aventure, c'est la recherche de la* différence »

Paul-Emile Victor

---

[1] Un traitement extensif des opérateurs monadiques (ou unaires) est fourni par Iverson ([IVE62], [IVE78]) et par Lochner [LOC89]. Leurs applications scientifiques sont discutées par Langlet ([LAN92a], [LAN93], [LAN94]) et Zaus ([ZA94], [ZA95b], [ZA95c]).

[2] Nous utilisons l'abréviation *ssi* pour « *si et seulement si* ».

[3] Elle correspond à la fonction dérivée +\x, l'opérateur de balayage par plus (« *plus-scan* ») en APL. L'« *œil de bœuf* » dans notre nomenclature standard signifie l'opérateur de balayage en APL, la rétrobarre \ .

NdT Le groupe à 6 éléments est formé des 4 rotations possibles des génitons, de la matrice-unité et de la rotation de cette dernière de 90°.

[4] Ces produits représentent un autre cas pour lequel la notation dans la littérature n'est pas adéquate, parce que les opérations mises en jeu sont supprimées dans la notation, par exemple XoY pour les produits internes généralisés. Ceci est évité dans la définition 2.12 ci-dessus.

NdT En anglais : BMP pour « *binary matrix product* »; on aura alors BPP pour « *binary power product* ».