UNIVERSITÉ DU QUÉBEC À MONTRÉAL

LES DÉCOMPOSITIONS DE FONCTIONS EN PITS

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

PAR

PATRICK SIMARD

DÉCEMBRE 2006

REMERCIEMENTS

En premier lieu, j'aimerais glisser un petit mot pour remercier du fond du coeur mes directeurs de maîtrise, Louise Laforest et Gilbert Labelle, pour leur dévouement exceptionnel. Merci pour votre gentillesse et pour votre attention à mon égard. J'ai passé d'agréables rencontres avec vous. Vous ne réalisez sûrement pas à quel point je suis ravi de vous avoir eu comme superviseurs. Mille mercis pour votre générosité et pour votre patience. Merci pour vos explications et pour vos gestes amicaux. Tout fut apprécié ! Vous êtes des personnes remarquables et je suis content que vous m'ayez légué une partie de votre savoir. Merci pour tout !

En second lieu, j'aimerais aussi prendre le temps de remercier les membres de ma famille pour leur soutien. Il n'est pas évident de travailler et d'étudier en même temps mais, ma famille a tout le temps été présente pour m'aider. De simples gestes sont parfois très appréciés. J'aimerais remercier mes parents pour avoir cru en moi durant toutes ces années d'études. J'aimerais remercier mes deux frères qui sont respectivement actuaire et informaticien, pour m'avoir donné des conseils judicieux tout au long de mes études et pour m'avoir incité à étudier en mathématiques option informatique. Merci à ma famille !

AVANT-PROPOS

Dans le cadre de mes tâches d'auxiliaire d'enseignement pour le cours d'organisation des ordinateurs et assembleur, «INF2170», je me suis intéressé à divers sujets. J'ai été captivé par les tables de vérité, les tables de Karnaugh, les diagrammes de Venn, les hyper-cubes, les simplifications de formules booléennes avec l'algèbre de Boole et les portes logiques. Je trouve intéressant les correspondances qui existent entre les divers objets mathématiques entourant la logique. Le mémoire présenté est connexe à ces sujets.

TABLE DES MATIÈRES

AVANT-PROPOS iii

LISTE DES TABLEAUX vi

RÉSUMÉ vii

INTRODUCTION 1

CHAPITRE I
LES CHAPEAUX 3

1.1 Introduction 3

1.2 L'univers des booléens 3

1.3 Les formes de fonctions 9

1.4 Les encodages 10

1.4.1 Les encodages de la forme descriptive 10

1.4.2 Les encodages de la forme polynomiale 13

1.5 Le chapeau de Gilbert Labelle 14

1.5.1 Définition du chapeau 14

1.5.2 Théorème du chapeau 15

1.5.3 L'involution du chapeau 20

1.6 Autres résultats 23

CHAPITRE II
PITS ET DÉCOMPOSITIONS DE FONCTIONS 27

2.1 Introduction 27

2.2 Les pits 27

2.3 Première décomposition de fonctions en pits 29

2.3.1 Théorème de Wilson 29

2.3.2 Théorème de Lucas 32

2.3.3 Théorème de décomposition 1 41
2.4 Deuxième décomposition de fonctions en pits 44
2.4.1 Le petit Théorème de Fermat 44
2.4.2 Le delta de Kronecker 45
2.4.3 Théorème de décomposition 2 47
2.5 Troisième décomposition de fonctions en pits 50
2.5.1 Matrice associée aux deltas de Kronecker 50
2.5.2 Théorème de décomposition 3 55
CHAPITRE III
RÉSULTATS DES DÉCOMPOSITIONS DE FONCTIONS 58
3.1 Introduction 58
3.2 Programmes de décompositions 59
3.2.1 Décomposition 1 60
3.2.2 Décomposition 2 68
3.2.3 Décomposition 3 73
3.3 Généralisations 76
3.3.1 Incrémentation 77
3.3.2 Addition 78
3.3.3 Produit 82
3.3.4 Monôme 84
3.4 Applications 84
CONCLUSION 86
ANNEXE A
PROGRAMMES MAPLE SUR LA FONCTION CHAPEAU 88
ANNEXE B
ÉNUMÉRATION DES CHAPEAUX 95
BIBLIOGRAPHIE 125

LISTE DES TABLEAUX

1.1 Conjonction · ($\wedge$) et disjonction exclusive + ($\oplus$). 4

1.2 La forme générale d'une table de vérité 6

1.3 Exemple de table de vérité 8

1.4 Tableau d'inclusions 16

RÉSUMÉ

En 1971, Gilbert Labelle a introduit la fonction chapeau qui est une traduction entre deux représentations de fonctions booléennes. Cette fonction intimement liée au calcul propositionnel possède de remarquables propriétés et permet de trouver le polynôme associé à une table de vérité et réciproquement. La fonction chapeau est involutive et nous en fournissons une démonstration car l'article original de Gilbert Labelle n'en présentait pas.

Pour une base de numération fixée p où p est premier, un nombre entier est identifié par une suite de chiffres appelés «pits» par analogie aux bien connus bits. Toute fonction définie sur $\mathbb{N}$ est exprimable par une fonction définie sur les pits. Une telle fonction est décomposable en une suite de sous-fonctions qui expriment individuellement chaque chiffre de sortie de la fonction originelle à partir des chiffres en entrée. Différentes décompositions de fonctions en pits sont présentées.

Les calculs liés à ces décompositions sont difficiles et des algorithmes astucieux sont développés en Maple pour obtenir quelques résultats qui suggèrent des formules générales que nous prouvons par la suite. Un bit est un cas particulier des pits et il y a une bijection entre les opérateurs d'addition/produit et les portes logiques. Il est alors possible pour un concepteur en électronique de réaliser une implémentation parallèle de fonctions logiques/arithmétiques à partir des décompositions.

Mots clés :
- Représentations de fonctions
- Calcul propositionnel
- Décompositions de fonctions
- Programmation Maple
- Calcul parallèle

INTRODUCTION

Le premier chapitre introduit le calcul propositionnel et la fonction chapeau. La matière concernant la logique et l'univers des booléens/bits est abordée en début de chapitre. Une table de vérité peut être représentée à partir d'une formule mathématique appelée «forme descriptive». En développant la forme descriptive, il est possible de trouver le polynôme (modulo deux) associé à cette table de vérité. Il existe cependant une façon alternative permettant de trouver le polynôme d'une table de vérité. La fonction chapeau permet de traduire une représentation de la forme descriptive en une représentation polynomiale. La fonction chapeau utilise comme algorithme un dénombrement d'objets au lieu d'effectuer des calculs arithmétiques. La forme descriptive et la forme polynomiale sont représentées de manière ensembliste via des opérateurs d'encodage. La fonction chapeau est bijective et involutive, deux remarquables propriétés dont une démonstration est fournie. La fonction permet donc de trouver indirectement le polynôme associé à une table de vérité et réciproquement.

Le chapitre deux concerne les fonctions définies sur les entiers représentés en base p, où p est premier. Les chiffres de la représentation sont appelés «pits» par analogie aux bits. Toute fonction définie sur $\mathbb{N}$ est représentable par une famille de fonctions définies sur des pits. La valeur de chacune des fonctions de cette fammile est calculable par une série modulo p. Il existe différents algorithmes permettant la traduction de fonctions définies sur les entiers en fonctions définies sur les pits. Cette traduction se fait à partir de différentes décompositions de fonctions qui utilisent le Théorème de Lucas, la série du

binôme de Newton et le delta de Kronecker exploité sous diverses formes. Les résultats que l'on obtient sont tous égaux lorsque développés peu importe l'algorithme de décomposition utilisé. Notons que le Théorème de Wilson, le petit Théorème de Fermat et le Théorème de Kummer sont aussi utilisés dans ce chapitre pour démontrer ces décompositions de fonctions.

Les décompositions de fonctions utilisent des sommations infinies pour exprimer les pits d'une fonction à partir des pits en entrée. Il est pourtant possible d'implémenter de façon pratique ces formules en utilisant un nombre fini d'étapes via quelques astuces. C'est ce qui est exposé au chapitre trois. Des programmes Maple permettent d'éviter de longs calculs pour chaque décomposition de fonctions en pits. Cependant, ces fonctions doivent avoir des caractéristiques particulières. Les différents exemples d'utilisations des programmes Maple suggèrent des formules mathématiques dont certaines sont démontrées. Comme les pits peuvent être calculés de manière indépendante, des techniques de calcul parallèles peuvent être utilisées dans le but de calculer des fonctions définies sur des entiers. Ces résultats obtenus pourraient avoir des applications en micro-électronique dans la réalisation de diagrammes de portes logiques et pourraient peut-être un jour révolutionner les ordinateurs de demain. De plus, des programmes Maple relatifs à la fonction chapeau sont présentés en annexe. Ces programmes ont permis de faire une énumération de la fonction chapeau appliquée sur toutes représentations de tables de vérité possibles à une, deux et trois variables. Notons que le nombre de table de vérité à n variables est 2^{2^n}.

CHAPITRE I

LES CHAPEAUX

1.1 Introduction

Dans ce chapitre, le calcul propositionnel et la fonction chapeau sont présentés.

1.2 L'univers des booléens

Les valeurs numériques un et zéro sont respectivement utilisées pour qualifier les valeurs booléennes *vrai* et *faux*. Un chiffre binaire[1], aussi appelé «bit», est soit le chiffre «0» ou bien le chiffre «1». Lorsque le mot chiffre est utilisé, il faut immédiatement penser au symbole et à la valeur qui lui est associé. Notons qu'en présence de valeurs numériques, la conjonction (l'opérateur binaire «et») correspond au produit, tandis que la disjonction exclusive (l'opérateur binaire «ou exclusif») est égale à la somme modulo deux. Cela permet d'utiliser respectivement les symboles arithmétiques «·» et «+», pour présenter les opérateurs logiques binaires de conjonction ($\wedge$) et disjonction exclusive ($\oplus$).

---

[1]Chiffre en base deux.

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

c	d	$c + d$
0	0	0
0	1	1
1	0	1
1	1	0

Tab. 1.1 Conjonction $\cdot$ ($\wedge$) et disjonction exclusive $+$ ($\oplus$).

Ces deux correspondances sont connues et souvent utilisées dans la littérature mathématique pour appréhender les formules logiques d'un point de vue arithmétique.

Exemple 1.2.1 *Soit a, b, c, d, e, f, g, h et i, des variables booléennes et T, une fonction 9-aire qui a pour entrée $(a, b, c, d, e, f, g, h, i)$. La formule logique*

$$T = (a \wedge b \wedge c) \oplus (d \wedge e \wedge f) \oplus (g \wedge h \wedge i), \tag{1.2.1}$$

est écrite avec les symboles arithmétiques de la manière suivante

$$T = (abc) + (def) + (ghi), \quad (modulo\ 2). \tag{1.2.2}$$

Dans la notation arithmétique, le produit (la conjonction) est un opérateur prioritaire sur l'addition (la disjonction exclusive) de manière similaire à leur sens usuel. Il est donc facultatif d'utiliser des parenthèses dans l'équation (1.2.2) pour marquer la priorité des opérations. Il existe cependant des cas où les parenthèses sont nécessaires afin de marquer la priorité des opérations.

Exemple 1.2.2

$$T = ((a \wedge (b \oplus c)) \oplus ((d \wedge e) \oplus f)) \wedge g. \tag{1.2.3}$$

En ce qui concerne l'opérateur unaire de négation ($\neg$), introduisons une fonction qui permettra d'appliquer la négation à x selon certaines circonstances.

Définition 1.1 *L'opération de «négation optionnelle», notée par $x^{[a]}$, appliquée au bit, x, est définie par l'équation*

$$x^{[a]} = \begin{cases} x & si \quad a = 1, \\ \neg x & si \quad a = 0. \end{cases} \tag{1.2.4}$$

L'idée sous-jacente à cette notation est d'introduire des formules qui peuvent avoir optionnellement une négation pour chaque variable. C'est ainsi que les formules satisfaisant des tables de vérité sont définies. Soulignons aussi que, puisque l'addition désigne la somme modulo deux, alors $\neg x = 1 + x$. Mais, comme la négation est involutive, il est vrai que $x = 1 + 1 + x$. L'opération de négation optionnelle vérifie donc l'égalité

$$x^{[a]} = 1 + a + x \quad \text{(modulo 2)}. \tag{1.2.5}$$

L'équation présentée ci-dessous permet d'évaluer rapidement $x^{[a]}$

$$x^{[a]} = \begin{cases} 1 & si \quad x = a, \\ 0 & si \quad x \neq a. \end{cases} \tag{1.2.6}$$

Cette formule est très pratique lors d'évaluations consécutives de nombreuses négations optionnelles. Les sorties d'une fonction booléenne sont souvent décrites par une table de vérité.

Définition 1.2 *Une table de vérité T d'ordre n, est un tableau qui décrit une fonction n-aire, $F : \{0,1\}^n \to \{0,1\}$*

Indices	x_1	x_2	...	x_{n-1}	x_n	$w(x_1, x_2, \ldots, x_n) \in \{0, 1\}$
0	0	0	...	0	0	$w(0, 0, \ldots, 0, 0)$
1	0	0	...	0	1	$w(0, 0, \ldots, 0, 1)$
2	0	0	...	1	0	$w(0, 0, \ldots, 1, 0)$
⋮	⋮	⋮		⋮	⋮	⋮
i	ϵ_1	ϵ_2	...	ϵ_{n-1}	ϵ_n	$w(\epsilon_1, \epsilon_2, \ldots, \epsilon_{n-1}, \epsilon_n)$
⋮	⋮	⋮		⋮	⋮	⋮
$2^n - 1$	1	1	...	1	1	$w(1, 1, \ldots, 1, 1)$

Tab. 1.2 La forme générale d'une table de vérité

Notons que l'indice de la ligne i est égal à la représentation en base 10 des entrées (chiffres binaires) de la table,

$$i = \epsilon_n 2^0 + \epsilon_{n-1} 2^1 + \epsilon_{n-2} 2^2 + \cdots + \epsilon_1 2^{n-1}. \tag{1.2.7}$$

Cependant, il est plus commode de manipuler des formules que des tables de vérité. Trouvons une formule mathématique qui est logiquement équivalente à une table de vérité.

Théorème 1.2.1 *Une table de vérité, définie par le tableau 1.2, peut être décrite par la fonction*

$$F(x_1, x_2, \ldots, x_n) = \sum_{(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) \in \{0,1\}^n} w(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \ldots x_n^{[\epsilon_n]}. \tag{1.2.8}$$

Pour toute table de vérité, il existe donc une formule satisfaisant cette table de vérité.

Démonstration du Théorème :

À chaque ligne de la table de vérité est associée un terme t_i qui est fonction de

toutes les variables de F et qui est strictement exclusif à cette ligne de telle sorte que

$$F(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{2^n-1} t_i. \tag{1.2.9}$$

Examinons ce qui se passe à la ligne $i = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ de la table de vérité.

1. Si $w(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) = 1$,

   (a) Si $\forall j : (x_j = \epsilon_j)$, alors $F(x_1, x_2, \ldots, x_n) = 1$.

   $$\begin{aligned} \forall j : (x_j = \epsilon_j) \quad & ssi \quad \forall j : (x_j^{[\epsilon_j]} = 1) \\ & ssi \quad (x_1^{[\epsilon_1]} = 1) \wedge (x_2^{[\epsilon_2]} = 1) \wedge \cdots \wedge (x_n^{[\epsilon_n]} = 1) \\ & ssi \quad x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \ldots x_n^{[\epsilon_n]} = 1. \end{aligned}$$

   (b) Si $\exists j : (x_j \neq \epsilon_j)$, alors $F(x_1, x_2, \ldots, x_n) = 0$.

   $$\begin{aligned} \exists j : (x_j \neq \epsilon_j) \quad & ssi \quad \exists j : (x_j^{[\epsilon_j]} = 0) \\ & ssi \quad x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \ldots x_n^{[\epsilon_n]} = 0. \end{aligned}$$

   Dans ce cas, la contribution du terme associé à cette ligne est

   $$t_i = x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \ldots x_n^{[\epsilon_n]}. \tag{1.2.10}$$

2. Si $w(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) = 0$, la contribution du terme associé à cette ligne est

   $$t_i = 0. \tag{1.2.11}$$

D'après les équations (1.2.10) et (1.2.11), la contribution du terme associé à la ligne i est

$$t_i = w(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \ldots x_n^{[\epsilon_n]}. \tag{1.2.12}$$

Il est permis de sommer les termes des entrées puisque ceux-ci sont considérés de façons exclusives. D'où,

$$F(x_1, x_2, \ldots, x_n) = \sum_{(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) \in \{0,1\}^n} w(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \ldots x_n^{[\epsilon_n]}. \tag{1.2.13}$$

■

Exemple 1.2.3 *Considérons la table* 1.3.

	x_1	x_2	x_3	$w(x_1, x_2, x_3)$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Tab. 1.3 Exemple de table de vérité

La fonction satisfaisant cette table de vérité est

$$F(x_1, x_2, x_3) = x_1^{[0]} x_2^{[0]} x_3^{[0]} + x_1^{[0]} x_2^{[1]} x_3^{[1]} + x_1^{[1]} x_2^{[0]} x_3^{[1]} + x_1^{[1]} x_2^{[1]} x_3^{[0]}. \tag{1.2.14}$$

Nous pouvons exhaustivement vérifier le tout :

$$\begin{aligned}
F(0,0,0) &= 0^{[0]}0^{[0]}0^{[0]} + 0^{[0]}0^{[1]}0^{[1]} + 0^{[1]}0^{[0]}0^{[1]} + 0^{[1]}0^{[1]}0^{[0]} \\
&= 1 \cdot 1 \cdot 1 + 1 \cdot 0 \cdot 0 + 0 \cdot 1 \cdot 0 + 0 \cdot 0 \cdot 1 \\
&= 1, \\
F(0,0,1) &= 0^{[0]}0^{[0]}1^{[0]} + 0^{[0]}0^{[1]}1^{[1]} + 0^{[1]}0^{[0]}1^{[1]} + 0^{[1]}0^{[1]}1^{[0]} \\
&= 1 \cdot 1 \cdot 0 + 1 \cdot 0 \cdot 1 + 0 \cdot 1 \cdot 1 + 0 \cdot 0 \cdot 0 \\
&= 0, \\
&\vdots
\end{aligned}$$

Remarque 1.1 *Il est possible de décrire une table de vérité avec une formule qui n'utilise qu'un seul opérateur binaire, la barre de Sheffer (Sheffer, 1913)*

$$p \,|\, q = \neg p \wedge \neg q. \tag{1.2.15}$$

Le tout est vérifiable avec les trois affirmations suivantes :

$$\begin{aligned} &(a) \quad \neg p = p \mid p, \\ &(b) \quad p \wedge q = \neg p \mid \neg q, \\ &(c) \quad p \vee q = \neg(\neg p \wedge \neg q) \quad \textit{(loi de DeMorgan).} \end{aligned} \tag{1.2.16}$$

Il est donc possible d'effectuer les opérations $\neg$, $\wedge$ *et* $\vee$ *en n'utilisant que la barre de Sheffer. Une table de vérité peut donc être décrite en utilisant que la barre de Sheffer.*

1.3 Les formes de fonctions

Une même fonction peut être représentée sous plusieurs formes. La forme change, mais le contenu reste le même. Dans cette section, la «forme descriptive» et la «forme polynomiale» d'une table de vérité sont présentées.

Définition 1.3 *Selon le Théorème 1.2.1, une table de vérité peut être décrite par la formule*

$$F(x_1, x_2, \ldots, x_n) = \sum_{(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) \in [0,1]^n} w(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) \prod_{i=1}^{n} x_i^{[\epsilon_i]}. \tag{1.3.1}$$

Appelons cette formule, la «forme descriptive» de $F(x_1, x_2, \ldots, x_n)$.

Définition 1.4 *En substituant* $x_j^{[\epsilon_j]}$ *par* $1 + \epsilon_j + x_j$ *dans la formule précédente, on obtient la «forme polynomiale» de* $F(x_1, x_2, \ldots, x_n)$

$$F(x_1, x_2, \ldots, x_n) = \sum_{(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) \in [0,1]^n} w(\epsilon_1, \epsilon_2, \ldots, \epsilon_n) \prod_{i=1}^{n} (1 + \epsilon_i + x_i). \tag{1.3.2}$$

De cette manière, l'opération de négation optionnelle est éliminée. Le tout peut être développé en distribuant la multiplication sur l'addition afin d'obtenir son polynôme lui est associé. D'où, son appellation «forme polynomiale» de $F(x_1, x_2, \ldots, x_n)$. Noter que les deux formules (1.3.1) et (1.3.2) sont équivalentes modulo deux, il n'y a que la présentation (la symbolique) qui est différente. Pour

trouver la table de vérité associée à la forme polynomiale, il suffit d'énumérer pour toutes les entrées $x_1, x_2, \ldots, x_n$ la sortie du polynôme.

Exemple 1.3.1 *La forme descriptive de la table de vérité 1.3 est*

$$F(x_1, x_2, x_3) = x_1^{[0]} x_2^{[0]} x_3^{[0]} + x_1^{[0]} x_2^{[1]} x_3^{[1]} + x_1^{[1]} x_2^{[0]} x_3^{[1]} + x_1^{[1]} x_2^{[1]} x_3^{[0]}. \tag{1.3.3}$$

En substituant $x_j^{[\epsilon_j]}$ *par* $1 + \epsilon_j + x_j$ $(\forall j)$ *dans la formule précédente, on obtient la forme polynomiale de* $F(x_1, x_2, x_3)$

$$(1 + x_1)(1 + x_2)(1 + x_3) + (1 + x_1) x_2 x_3 + x_1 (1 + x_2) x_3 + x_1 x_2 (1 + x_3). \tag{1.3.4}$$

En distribuant les multiplications sur les additions, on obtient le polynôme

$$F(x_1, x_2, x_3) = 1 + x_3 + x_2 + 2\,x_2 x_3 + x_1 + 2\,x_1 x_3 + 2\,x_1 x_2 + 4\,x_1 x_2 x_3. \tag{1.3.5}$$

Puisqu'il s'agit d'une congruence modulo 2, il en résulte que

$$F(x_1, x_2, x_3) = 1 + x_1 + x_2 + x_3 \quad \textit{(modulo 2)}. \tag{1.3.6}$$

Cependant, la forme polynomiale n'est pas toujours plus « courte » que la forme descriptive.

1.4 Les encodages

Dans cette section, on montre comment la forme descriptive et la forme polynomiale sont encodées à l'aide d'ensembles.

1.4.1 Les encodages de la forme descriptive

Définition 1.5 *Soit* ϕ*, l'opérateur permettant de transformer un terme*

$$x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} x_3^{[\epsilon_3]} \ldots x_i^{[\epsilon_i]} \ldots x_n^{[\epsilon_n]} \tag{1.4.1}$$

de la forme descriptive en ensemble s tel que

$$s = \phi(x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \dots x_n^{[\epsilon_n]}) = \{i : \epsilon_i = 1\} \tag{1.4.2}$$

qui contient les indices inférieurs des x où il y a [1] *comme indice supérieur de x.*

Exemple 1.4.1 $\phi(x_1^{[1]} x_2^{[0]} x_3^{[1]}) = \{1, 3\}$.

Exemple 1.4.2 $\phi(x_1^{[0]} x_2^{[0]} x_3^{[0]}) = \emptyset$.

Cependant, le décodage ϕ^{-1} appliqué à $\phi(x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \dots x_n^{[\epsilon_n]})$ ne se fait pas aussi bien que l'encodage. Pour tranformer un ensemble en un terme, le nombre de variables de la fonction F doit être connu. Illustrons ces propos avec un exemple.

Exemple 1.4.3 *Soit* $s = \{1, 3\}$. *Puisque le nombre de variables de la fonction n'est pas connu, l'une des équations suivantes est vraie*

$$\begin{aligned}
\phi^{-1}(\{1, 3\}) &= x_1^{[1]} x_2^{[0]} x_3^{[1]}, \\
\phi^{-1}(\{1, 3\}) &= x_1^{[1]} x_2^{[0]} x_3^{[1]} x_4^{[0]}, \\
\phi^{-1}(\{1, 3\}) &= x_1^{[1]} x_2^{[0]} x_3^{[1]} x_4^{[0]} x_5^{[0]}, \\
&\vdots \\
\phi^{-1}(\{1, 3\}) &= x_1^{[1]} x_2^{[0]} x_3^{[1]} x_4^{[0]} x_5^{[0]} \dots x_i^{[0]} \dots x_n^{[0]}, \\
&\vdots
\end{aligned}$$

Si $n = 3$, *c'est-à-dire si la fonction n'admet que les variables* x_1, x_2 *et* x_3, *alors il est clair que*

$$\phi^{-1}(\{1, 3\}) = x_1^{[1]} x_2^{[0]} x_3^{[1]}. \tag{1.4.3}$$

Définition 1.6 *Soit T, la forme descriptive de* $F(x_1, x_2, \dots, x_n)$,

$$T = \sum_{i=1}^{r} \tau_i = \sum_{(\epsilon_1, \epsilon_2, \dots, \epsilon_n) \in \{0,1\}^n} w(\epsilon_1, \epsilon_2, \dots, \epsilon_n) x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \dots x_n^{[\epsilon_n]}. \tag{1.4.4}$$

où les τ_i sont les termes non nuls de la forme descriptive,

$$\tau_i = x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \dots x_n^{[\epsilon_n]}. \tag{1.4.5}$$

La fonction $\Phi(T)$ donne un ensemble qui contient l'encodage individuel de chaque terme de la forme descriptive

$$S = \Phi(T) = \{\phi(\tau_1), \phi(\tau_2), \dots, \phi(\tau_r)\}. \tag{1.4.6}$$

Exemple 1.4.4 $\phi(x_1^{[1]} x_2^{[1]} x_3^{[0]}) = \{1, 2\}$.

Exemple 1.4.5 $\Phi(x_1^{[1]} x_2^{[1]} x_3^{[0]}) = \{\{1, 2\}\}$.

Exemple 1.4.6 *Soit T, la forme descriptive*

$$T = x_1^{[0]} x_2^{[0]} x_3^{[0]} + x_1^{[1]} x_2^{[1]} x_3^{[0]} + x_1^{[1]} x_2^{[0]} x_3^{[1]} + x_1^{[0]} x_2^{[1]} x_3^{[1]}. \tag{1.4.7}$$

Alors,

$$S = \Phi(T) = \{\emptyset, \{1, 2\}, \{1, 3\}, \{2, 3\}\}. \tag{1.4.8}$$

Notation 1 *À chaque petit*[2] *ensemble, est associé une variable s_i qui lui est égale. Un petit ensemble peut décrire un nombre binaire lorsqu'on considère les éléments de cet ensemble comme étant les indices des bits qui valent un. Le nombre i du s_i est la représentation décimale de ce nombre binaire représentant un petit ensemble. Soit $\alpha \in s$, un petit ensemble, alors $i = \sum_{j \in \alpha} 2^{j-1}$.*

Dans l'exemple précédent, S s'écrit de façon abrégée par

$$S = \{s_0, s_3, s_5, s_6\}. \tag{1.4.9}$$

[2]Un petit ensemble, s, se trouvant dans le gros ensemble S.

Exemple 1.4.7 *Les premiers s_i peuvent être énumérés comme ceci :*

$$\begin{aligned} s_0 &= \emptyset, \\ s_1 &= \{1\}, \\ s_2 &= \{2\}, \\ s_3 &= \{1,2\}, \\ s_4 &= \{3\}, \\ s_5 &= \{1,3\}, \\ &\vdots \end{aligned}$$

Notons que l'ensemble $\Phi(T)$ possède les propriétés suivantes :

1. $\Phi(T)$ est un ensemble d'ensembles si T a au moins un terme non nul.
2. $\Phi(0) = \emptyset$.
3. $\Phi(\prod_{j=1}^{n} x_j^{[0]}) = \{\emptyset\} = \{s_0\}$.

1.4.2 Les encodages de la forme polynomiale

Définition 1.7 *Soit ψ, l'opérateur prenant en entrée le monôme*

$$p = x_{a_1} x_{a_2} \dots x_{a_k} \tag{1.4.10}$$

où $\{a_1, a_2, \dots, a_k\} \subseteq \{1, 2, \dots, n\}$ et le transformant en l'ensemble

$$\psi(p) = \{a_1, a_2, \dots, a_k\}. \tag{1.4.11}$$

Un polynôme P,

$$P = \sum_{i=1}^{r} p_i, \tag{1.4.12}$$

où p_i est un monôme non nul, peut être représenté par un ensemble de représentations de monômes

$$\Psi(P) = \{\psi(p_1), \psi(p_2), \dots, \psi(p_r)\}. \tag{1.4.13}$$

Exemple 1.4.8 $\psi(x_1x_3x_5) = \{1,3,5\} = s_{21}$.

Exemple 1.4.9 $\Psi(x_1x_3x_5) = \{\{1,3,5\}\} = \{s_{21}\}$.

Exemple 1.4.10 $\Psi(1 + x_1 + x_2 + x_2x_3) = \{\emptyset, \{1\}, \{2\}, \{2, 3\}\} = \{s_0, s_1, s_2, s_6\}$.

1.5 Le chapeau de Gilbert Labelle

Il y a quelques années, Gilbert Labelle (Labelle, 1971) s'est intéressé à une fonction mathématique qui possède des propriétés très intéressantes, la fonction chapeau. Celle-ci permet de passer d'une représentation de table de vérité à une représentation de polynôme.

1.5.1 Définition du chapeau

Notation 2 *Posons* $[n] = \{1, 2, \ldots, n\}$.

Définition 1.8 *La fonction «chapeau», notée par «* $\widehat{\ }$ *», est une fonction qui a pour entrée une forme descriptive T encodée par* Φ *et qui retourne sa forme polynomiale P encodée par* Ψ

$$\widehat{\ } : \Phi(T) \mapsto \Psi(P). \tag{1.5.1}$$

L'ensemble de départ et d'arrivée de la fonction est $\mathcal{P}(\mathcal{P}([n]))$ où n est le nombre de variables des fonctions de T et P, tandis que $\mathcal{P}(E)$ est l'ensemble des sous-ensembles d'un ensemble E.

Notation 3 *La lettre S est utilisée pour désigner l'entrée de la fonction chapeau et* $\widehat{S}$ *pour désigner la sortie de cette fonction.*

Avec la fonction chapeau, il est possible de passer indirectement de la forme descriptive à la forme polynomiale.

Exemple 1.5.1 *Soit la forme descriptive suivante*

$$F(x_1, x_2, x_3) = x_1^{[0]}x_2^{[0]}x_3^{[0]} + x_1^{[1]}x_2^{[1]}x_3^{[0]} + x_1^{[1]}x_2^{[0]}x_3^{[1]} + x_1^{[0]}x_2^{[1]}x_3^{[1]}. \tag{1.5.2}$$

Il a été vu (voir l'exemple 1.3.1) que la forme polynomiale associée à ce F était

$$F(x_1, x_2, x_3) = 1 + x_1 + x_2 + x_3. \tag{1.5.3}$$

Si la fonction chapeau a pour entrée l'ensemble

$$S = \{\emptyset, \{1,2\}, \{1,3\}, \{2,3\}\}, \tag{1.5.4}$$

alors, la fonction chapeau retourne l'ensemble

$$\widehat{S} = \{\emptyset, \{1\}, \{2\}, \{3\}\}. \tag{1.5.5}$$

Cependant, il existe un algorithme plus intéressant qui permet d'arriver au même résultat.

1.5.2 Théorème du chapeau

Théorème 1.5.1 (Théorème du chapeau) *Soit S, une forme descriptive encodée par* Φ *et* $\widehat{S}$*, la forme polynomiale qui lui est associée et qui est encodée par* Ψ*. Alors* $\widehat{S}$ *satisfait*

$$\widehat{S} = \{t \subseteq [n] : t, \text{ un ensemble qui contient un nombre impair de } s \in S\}. \tag{1.5.6}$$

La preuve de ce Théorème vient un peu plus loin.

Exemple 1.5.2 *Soit* $S = \{\{2\}, \{1,2\}, \{1,2,3\}\}$ *et* $n = 3$*. Pour tout* $t \subseteq [n]$*, comptons le nombre de* $s \in S$ *qui sont des sous-ensembles de t.*

t	$\{2\}$	$\{1,2\}$	$\{1,2,3\}$	$\#\sqrt{}$
$\emptyset$				0
$\{1\}$				0
$\{2\}$	$\sqrt{}$			1
$\{3\}$				0
$\{1,2\}$	$\sqrt{}$	$\sqrt{}$		2
$\{1,3\}$				0
$\{2,3\}$	$\sqrt{}$			1
$\{1,2,3\}$	$\sqrt{}$	$\sqrt{}$	$\sqrt{}$	3

Tab. 1.4 Tableau d'inclusions

Il n'y a que les ensembles $\{2\}$, $\{2,3\}$ *et* $\{1,2,3\}$ *incluent dans* $\{1,2,3\}$ *qui contiennent un nombre impair d'éléments de S. Par le Théorème 1.5.1,*

$$\widehat{S} = \{\{2\},\{2,3\},\{1,2,3\}\}. \tag{1.5.7}$$

En d'autres mots,

$$x_1^{[0]}x_2^{[1]}x_3^{[0]} + x_1^{[1]}x_2^{[1]}x_3^{[0]} + x_1^{[1]}x_2^{[1]}x_3^{[1]} = x_2 + x_2x_3 + x_1x_2x_3. \tag{1.5.8}$$

Notation 4 *Soit* $u \subseteq [n]$*, alors, posons*

$$x_u = \prod_{i \in u} x_i. \tag{1.5.9}$$

Exemple 1.5.3 $x_{|1,2,4|} = x_1x_2x_4$.

Exemple 1.5.4 $x_\emptyset = 1$.

Lemme 1.5.2 $\forall I \subseteq [n]$,

$$\prod_{i \in I}(1 + x_i) = \sum_{u \subseteq I} x_u. \tag{1.5.10}$$

La preuve de ce Lemme se trouve un peu plus loin.

Examinons tout d'abord quelques exemples.

Exemple 1.5.5 *Pour* $I = \{1, 4, 5\}$,

$$\begin{aligned}\prod_{i\in I}(1+x_i) &= (1+x_1)(1+x_4)(1+x_5)\\ &= 1 + x_1 + x_4 + x_5 + x_1x_4 + x_1x_5 + x_4x_5 + x_1x_4x_5\\ &= 1 + x_{\{1\}} + x_{\{4\}} + x_{\{5\}}\\ &\quad + x_{\{1,4\}} + x_{\{1,5\}} + x_{\{4,5\}} + x_{\{1,4,5\}}\\ &= \sum_{u\subseteq I} x_u.\end{aligned}$$

Remarque 1.2 *Le Lemme peut aussi être défini avec I, un ensemble fini quelconque. Dans l'exemple précédent, l'ensemble* $I = \{\spadesuit, \nabla, 7\}$ *aurait pu être considéré à la place de l'ensemble* $\{1, 4, 5\}$. *Cependant, nous nous contenterons de* $I \subseteq [n]$ *puisqu'il est coutume d'indicer des objets avec des entiers.*

Exemple 1.5.6 *Pour* $I = \{1, 2, 3, 4, 5\}$,

$$\begin{aligned}\prod_{i=1}^{5}(1+x_i) &= 1 + x_1 + x_2 + x_3 + x_4 + x_5\\ &\quad + x_1x_2 + x_1x_3 + x_1x_4 + x_1x_5 + x_2x_3\\ &\quad + x_2x_4 + x_2x_5 + x_3x_4 + x_3x_5 + x_4x_5\\ &\quad + x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_4 + x_1x_3x_5\\ &\quad + x_1x_4x_5 + x_2x_3x_4 + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5\\ &\quad + x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5\\ &\quad + x_1x_3x_4x_5 + x_2x_3x_4x_5 + x_1x_2x_3x_4x_5.\\ &= \sum_{u\subseteq I} x_u.\end{aligned}$$

Démonstration du Lemme 1.5.2

Sans perte de généralité, posons l'ensemble $I = \{1, 2, \ldots, n\}$ car celui-ci peut

être mis en bijection avec n'importe quel ensemble à n éléments. Démontrons par récurrence sur I la formule (1.5.10).

Le cas de base pour $I = \emptyset$ est vérifié.

$$\prod_{i\in\emptyset}(1+x_i) = 1 = x_\emptyset = \sum_{u\subseteq\emptyset} x_u \tag{1.5.11}$$

Supposons que la formule soit vraie pour $I = \{1, 2, \ldots, n\}$ et montrons que la formule reste vraie pour $I \cup \{n+1\}$.

$$\begin{aligned}
\prod_{i\in I\cup\{n+1\}}(1+x_i) &= \left(\prod_{i\in I}(1+x_i)\right)(1+x_{n+1}) \\
&= \left(\sum_{u\subseteq I} x_u\right)(1+x_{n+1}) \qquad \text{(par l'hypothèse d'induction)} \\
&= \sum_{u\subseteq I} x_u + x_{n+1}\sum_{u\subseteq I} x_u \\
&= \sum_{u\subseteq I} x_u + \sum_{u\subseteq I\cup\{n+1\}\text{ t.q. }(n+1)\in u} x_u \\
&= \sum_{u\subseteq I\cup\{n+1\}\text{ t.q. }(n+1)\notin u} x_u + \sum_{u\subseteq I\cup\{n+1\}\text{ t.q. }(n+1)\in u} x_u.
\end{aligned}$$

D'où,

$$\prod_{i\in I\cup\{n+1\}}(1+x_i) = \sum_{u\subseteq I\cup\{n+1\}} x_u. \tag{1.5.12}$$

Le pas de la récurrence est donc vérifié et il en est de même pour la formule. ■

Démonstration du Théorème du chapeau.

Soit $F(x_1, x_2, \ldots, x_n)$ définie par

$$F(x_1, x_2, \ldots, x_n) = \sum_{(\epsilon_1,\epsilon_2,\ldots,\epsilon_n)\in\{0,1\}^n}\left(w(\epsilon_1, \epsilon_2, \ldots, \epsilon_n)\prod_{i=1}^{n}(1+\epsilon_i+x_i)\right). \tag{1.5.13}$$

Les termes $(1 + \epsilon_i + x_i)$ de la formule précédente peuvent être classifiés en deux familles : les termes avec $\epsilon_i = 0$ et ceux avec $\epsilon_i = 1$. Si ces deux familles sont

isolées et que le produit des éléments de chaque famille est effectué, il en résulte

$$F(x_1, x_2, \ldots, x_n) = \sum_{s \in S} \left(\prod_{i \in s} x_i \prod_{i \notin s} (1 + x_i) \right). \tag{1.5.14}$$

Par la Notation 4 et le Lemme 1.5.2,

$$F(x_1, x_2, \ldots, x_n) = \sum_{s \in S} \left(x_s \sum_{u \subseteq [n] \setminus s} x_u \right). \tag{1.5.15}$$

En distribuant la multiplication sur les additions, cela mène à

$$F(x_1, x_2, \ldots, x_n) = \sum_{s \in S} \sum_{u \subseteq [n] \setminus s} x_s x_u. \tag{1.5.16}$$

La Notation 4 permet de poser $t = s \cup u$ (puisque $s \cap u = \emptyset$) et d'obtenir $x_t = x_s x_u$. D'où,

$$F(x_1, x_2, \ldots, x_n) = \sum_{t \subseteq [n]} \sum_{s \in S,\, s \subseteq t} x_t. \tag{1.5.17}$$

On note ici que l'indice de sommation, $u \subseteq [n] \setminus s$, à été substitué par $t \subseteq [n]$. Il est permis de mettre en évidence la somme intérieure comme ceci

$$F(x_1, x_2, \ldots, x_n) = \sum_{t \subseteq [n]} \left(\sum_{s \in S,\, s \subseteq t} 1 \right) x_t. \tag{1.5.18}$$

Le coefficient de x_t détermine si le monôme x_t est ajouté ou non au polynôme. Il n'y a que deux valeurs possible pour les coefficients : zéro et un. Le coefficient de x_t est donc

$$\sum_{s \in S,\, s \subseteq t} 1 \quad \text{(modulo 2)}, \tag{1.5.19}$$

qui est égal à 1 si et seulement si t contient un nombre impair de $s \in S$. Il en résulte que

$$F(x_1, x_2, \ldots, x_n) = \sum_{t \subseteq [n]} x_t, \quad \text{tel que } t \text{ contient un nombre impair de } s \in S. \tag{1.5.20}$$

Autrement dit,

$$\widehat{S} = \{ t \subseteq [n] : t, \text{un ensemble qui contient un nombre impair de } s \in S \}. \tag{1.5.21}$$

■

Cette preuve du Théorème du chapeau n'avait pas été aussi détaillée dans l'article de Gilbert (Labelle, 1971). Une autre démonstration du Théorème chapeau est donnée plus tard dans le chapitre.

1.5.3 L'involution du chapeau

Théorème 1.5.3 (L'INVOLUTION DU CHAPEAU) *La fonction chapeau est involutive*[3], *c'est-à-dire que*

$$\widehat{\circ}\widehat{\circ} = Id. \tag{1.5.22}$$

Ce résultat surprenant avait été énoncé sans démonstration par G. Labelle (Labelle, 1971).

Définition 1.9 *Soit A un ensemble. La «fonction caractéristique», notée χ_A, est définie par :*

$$\chi_A(a) = \begin{cases} 1 & si \quad a \in A, \\ 0 & si \quad a \notin A. \end{cases} \tag{1.5.23}$$

Exemple 1.5.7 $\chi_{\{0,4,5\}}(4) = 1.$

Exemple 1.5.8 $\chi_{\{0,4,5\}}(3) = 0.$

---

[3]Dans des ouvrages mathématiques comme des mémoires, nous évitons souvent de faire sensation. Par un heureux hasard, on obtient involontairement la face souriante, $\widehat{\circ}\widehat{\circ}$, qui pourrait contagieusement faire sourire le lecteur. Méfiez-vous ! Car, derrière le chapeau de Gilbert Labelle se cache l'involution, $\widehat{\circ}\widehat{\circ}$.

Notation 5 *Si A est omis et que a est une proposition, alors $\chi(a)$ vaudra 1 si a est vraie et 0 si a est fausse. Autrement dit, si a est une proposition et A est omis, alors nous considérons que $\chi(a)$ vaut $\chi_{\{vrai\}}(a)$.*

Démonstration du Théorème 1.5.3.

Par le Théorème 1.5.1,

$$t \in \widehat{S} \Leftrightarrow \sum_{s \in S,\, s \subseteq t} 1 = 1 \quad \text{(modulo 2).} \tag{1.5.24}$$

Par la définition 1.9, cette dernière formule peut être écrite autrement

$$\chi_{\widehat{S}}(t) = 1 \Leftrightarrow \sum_{s} \chi(s \subseteq t)\chi_S(s) = 1, \tag{1.5.25}$$

qui est équivalente à

$$\chi_{\widehat{S}}(t) = \sum_{s} \chi(s \subseteq t)\chi_S(s). \tag{1.5.26}$$

Rappelons ici qu'une fonction caractéristique qui n'a pas l'argument A s'applique à une proposition et que nous avons $\chi(s \subseteq t) = 1$ si et seulement si $s \subseteq t$ est vrai.

Montrons que $\widehat{\widehat{S}} = S$, c'est-à-dire que $(u \in \widehat{\widehat{S}}) \Leftrightarrow (u \in S)$ ou bien que $\chi_{\widehat{\widehat{S}}}(u) = \chi_S(u)$. En substituant S par $\widehat{S}$ dans (1.5.26), on obtient

$$\chi_{\widehat{\widehat{S}}}(u) = \sum_{t} \chi_{\widehat{S}}(t)\chi(t \subseteq u). \tag{1.5.27}$$

En substituant (1.5.26) dans (1.5.27), on obtient

$$\chi_{\widehat{\widehat{S}}}(u) = \sum_{t} \left(\sum_{s} \chi(s \subseteq t)\chi_S(s) \right) \chi(t \subseteq u), \tag{1.5.28}$$

qui est égal à

$$\chi_{\widehat{\widehat{S}}}(u) = \sum_{s} \left(\sum_{t} \chi(s \subseteq t \subseteq u)\chi_S(s) \right), \tag{1.5.29}$$

en changeant l'ordre des sommations. Examinons ce qui se passe à l'intérieur de la sommation de s, lorsque s varie

$$\sum_t \chi(s \subseteq t \subseteq u)\chi_S(s). \tag{1.5.30}$$

1. Si $s = u$, la contribution de ce s dans l'expression (1.5.30) est

$$\sum_t \chi(u \subseteq t \subseteq u)\chi_S(u) = \chi(u \subseteq u \subseteq u)\chi_S(u) = \chi_S(u). \tag{1.5.31}$$

2. Si $s \neq u$, il y a deux cas à considérer :

   (a) Si $s \not\subseteq u$, alors l'expression (1.5.30) est nulle car

$$\forall t, \chi(s \subseteq t \subseteq u) = 0. \tag{1.5.32}$$

   (b) Si $s \subset u$ ($s \neq u$), l'expression (1.5.30) est nulle car

$$|\{t : s \subseteq t \subseteq u\}| = 2^{|u|-|s|} \equiv 0 \pmod 2 \tag{1.5.33}$$

   puisque $|u| > |s|$.

Conclusion,

$$\chi_{\widehat{S}}(u) = \sum_s \left(\sum_t \chi(s \subseteq t \subseteq u)\chi_S(s) \right) = \chi_S(u). \tag{1.5.34}$$

D'où, l'involution

$$\widehat{\ } \circ \widehat{\ } = Id. \tag{1.5.35}$$

■

Corollaire 1.5.4 *Lorsque la fonction chapeau a pour entrée une forme polynomiale encodée par Ψ, alors celle-ci retourne sa forme descriptive encodée par Φ.*

Démonstration du Corollaire 1.5.4

Notez tout d'abord que l'ensemble de départ de la fonction est le même que

l'ensemble d'arrivée de la fonction. Il est donc permis de prendre un élément de l'ensemble de sortie et de le mettre en entrée dans la fonction. Puisque l'opération chapeau est involutive et bijective, si on applique l'opération chapeau à $\widehat{S}$ alors, nous allons obtenir S. ■

Nous pouvons donc trouver la table de vérité d'un polynôme sans faire une énumération des sorties du polynôme pour toutes les entrées.

Exemple 1.5.9 *Soit :* $S = \{\{2\}, \{1,2\}, \{1,2,3\}\}$ *et* $\widehat{S} = \{\{2\}, \{2,3\}, \{1,2,3\}\}$. *Il y a deux interprétations possibles pour S et* $\widehat{S}$. *Par la définition 1.8,*

$$x_1^{[0]}x_2^{[1]}x_3^{[0]} + x_1^{[1]}x_2^{[1]}x_3^{[0]} + x_1^{[1]}x_2^{[1]}x_3^{[1]} = x_2 + x_2x_3 + x_1x_2x_3. \tag{1.5.36}$$

Par le Corollaire 1.5.4,

$$x_2 + x_1x_2 + x_1x_2x_3 = x_1^{[0]}x_2^{[1]}x_3^{[0]} + x_1^{[0]}x_2^{[1]}x_3^{[1]} + x_1^{[1]}x_2^{[1]}x_3^{[1]}. \tag{1.5.37}$$

Ces deux équations sont pourtant différentes.

1.6 Autres résultats

Quelques petits résultats importants sur la fonction chapeau sont énumérés ici.

Notation 6 *Dans cette section,* $S^{\frown}$ *désigne* $\widehat{S}$.

1. Si une forme descriptive est nulle, alors la forme polynomiale qui lui est associée est aussi nulle. Puisque $\Phi(0) = \emptyset$ et $\Psi(0) = \emptyset$, il en résulte que

$$\emptyset^{\frown} = \emptyset. \tag{1.6.1}$$

2. La table de vérité où il n'y a que des 1 à la sortie se traduit par la fonction

$$\sum_{(\epsilon_1,\epsilon_2,\ldots,\epsilon_n)\in\{0,1\}^n} x_1^{[\epsilon_1]}x_2^{[\epsilon_2]}\cdots x_n^{[\epsilon_n]} = 1. \tag{1.6.2}$$

Puisque $\Phi(\sum_{(\epsilon_1,\epsilon_2,\ldots,\epsilon_n)\in\{0,1\}^n} x_1^{[\epsilon_1]}x_2^{[\epsilon_2]}\cdots x_n^{[\epsilon_n]}) = \mathcal{P}([n])$ où $\mathcal{P}([n])$ désigne l'ensemble des parties de $[n]$ et $\Psi(1) = \{\emptyset\}$, il en résulte que

$$\mathcal{P}([n])^\frown = \{\emptyset\} \tag{1.6.3}$$

et par le Corollaire 1.5.4, notons que

$$\{\emptyset\}^\frown = \mathcal{P}([n]). \tag{1.6.4}$$

Cette deux dernières formules peuvent aussi être montrées en posant

$$\sum_{u\subseteq[n]} x_u = x_1^{[0]}x_2^{[0]}\cdots x_n^{[0]}. \tag{1.6.5}$$

3. Puisque $x_i^{[1]} = x_i$, alors $x_1^{[1]}x_2^{[1]}\cdots x_n^{[1]} = x_1x_2\cdots x_n$, alors $\Phi(x_1^{[1]}x_2^{[1]}\cdots x_n^{[1]}) = [n]$ et $\Psi(x_1x_2\cdots x_n) = [n]$, il en résulte que

$$[n]^\frown = [n]. \tag{1.6.6}$$

4. Soit $1 \leq i \leq n$, l'égalité suivante est satisfaite

$$x_i = x_i^{[1]} \cdot \sum_{(\epsilon_1,\epsilon_2,\ldots,\epsilon_{i-1},\epsilon_{i+1},\ldots,\epsilon_n)\in\{0,1\}^{n-1}} x_1^{[\epsilon_1]}x_2^{[\epsilon_2]}\cdots x_{i-1}^{[\epsilon_{i-1}]}x_{i+1}^{[\epsilon_{i+1}]}\cdots x_n^{[\epsilon_n]} \tag{1.6.7}$$

car,

$$\sum_{(\epsilon_1,\epsilon_2,\ldots,\epsilon_{i-1},\epsilon_{i+1},\ldots,\epsilon_n)\in\{0,1\}^{n-1}} x_1^{[\epsilon_1]}x_2^{[\epsilon_2]}\cdots x_{i-1}^{[\epsilon_{i-1}]}x_{i+1}^{[\epsilon_{i+1}]}\cdots x_n^{[\epsilon_n]} = 1. \tag{1.6.8}$$

Puisque $\Psi(x_i) = \{\{i\}\}$ et que

$$\Phi(x_i^{[1]} \cdot \sum_{(\epsilon_1,\epsilon_2,\ldots,\epsilon_{i-1},\epsilon_{i+1},\ldots,\epsilon_n)\in\{0,1\}^{n-1}} x_1^{[\epsilon_1]}x_2^{[\epsilon_2]}\cdots x_{i-1}^{[\epsilon_{i-1}]}x_{i+1}^{[\epsilon_{i+1}]}\cdots x_n^{[\epsilon_n]})$$
$$= \{t : t \subseteq [n], i \in t\}. \tag{1.6.9}$$

alors, il en résulte que,

$$\{\{i\}\}^\frown = \{t : t \subseteq [n], i \in t\}. \tag{1.6.10}$$

5. Si $s \subseteq [n]$ où $s = \{i, j, ...\}$ avec $1 \leq i, j, \ldots \leq n$ alors,

$$x_i x_j \cdots = x_i^{[1]} x_j^{[1]} \cdots \sum x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \cdots x_{i-1}^{[\epsilon_{i-1}]} x_{i+1}^{[\epsilon_{i+1}]} \cdots x_{j-1}^{[\epsilon_{i-1}]} x_{j+1}^{[\epsilon_{i+1}]} \cdots x_n^{[\epsilon_n]} \tag{1.6.11}$$

donne

$$\{s\}^\frown = \{t : t \subseteq [n], s \subseteq t\}, \tag{1.6.12}$$

6. Si $1 \leq i, j \leq n$ et[4] $i \neq j$, alors

$$\begin{aligned} x_i + x_j = \\ x_i^{[1]} \sum_{(\epsilon_1, \epsilon_2, \ldots, \epsilon_{i-1}, \epsilon_{i+1}, \ldots, \epsilon_n) \in \{0,1\}^{n-1}} x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \cdots x_{i-1}^{[\epsilon_{i-1}]} x_{i+1}^{[\epsilon_{i+1}]} \cdots x_n^{[\epsilon_n]} \\ + \; x_j^{[1]} \sum_{(\epsilon_1, \epsilon_2, \ldots, \epsilon_{j-1}, \epsilon_{j+1}, \ldots, \epsilon_n) \in \{0,1\}^{n-1}} x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \cdots x_{j-1}^{[\epsilon_{j-1}]} x_{j+1}^{[\epsilon_{j+1}]} \cdots x_n^{[\epsilon_n]} \end{aligned} \tag{1.6.13}$$

donne

$$\{\{i\}, \{j\}\}^\frown = \{t : t \subseteq [n], (i \in t \text{ ou } j \in t) \text{ et } \neg(\{i, j\} \subseteq t)\}. \tag{1.6.14}$$

7. Si $s_1, s_2 \subseteq [n]$ où $s_1 = \{i, j, ...\}$ et $s_2 = \{a, b, ...\}$ avec $1 \leq i, j, \ldots, a, b, \ldots \leq n$ et $s_1 \neq s_2$ alors,

$$\begin{aligned} x_i x_j \cdots + x_a x_b \cdots = \\ x_i^{[1]} x_j^{[1]} \cdots \sum x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \cdots x_{i-1}^{[\epsilon_{i-1}]} x_{i+1}^{[\epsilon_{i+1}]} \cdots x_{j-1}^{[c_{i-1}]} x_{j+1}^{[\epsilon_{i+1}]} \cdots x_n^{[\epsilon_n]} \\ + \; x_a^{[1]} x_b^{[1]} \cdots \sum x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \cdots x_{a-1}^{[\epsilon_{a-1}]} x_{a+1}^{[\epsilon_{a+1}]} \cdots x_{b-1}^{[\epsilon_{b-1}]} x_{b+1}^{[\epsilon_{b+1}]} \cdots x_n^{[\epsilon_n]} \end{aligned} \tag{1.6.15}$$

donne

$$\{s_1, s_2\}^\frown = \{t : t \subseteq [n], (s_1 \subseteq t \text{ ou } s_1 \subseteq t) \text{ et } \neg((s_1 \cup s_2) \subseteq t)\}. \tag{1.6.16}$$

Autrement dit,

$$\{s_1, s_2\}^\frown = \{t : t \subseteq [n], ((s_1 \setminus s_2) \subseteq t \text{ ou } (s_2 \setminus s_1) \subseteq t)\}. \tag{1.6.17}$$

---

[4] C'est-à-dire, si $1 \leq i \leq n$ et $1 \leq j \leq n$.

8. Si $S = \{s_1, s_2, s_3\}$ avec $s_1, s_2, s_3 \subseteq [n]$ où $s_1 = \{i, j, ...\}$, $s_2 = \{a, b, ...\}$ et $s_3 = \{u, v, ...\}$ avec $1 \leq i, j, \ldots, a, b, \ldots, u, v, \ldots \leq n$ et s_1, s_2, s_3 distincts, alors,

$$\begin{aligned} x_i x_j \cdots + x_a x_b \cdots + x_u x_v \cdots = \\ x_i^{[1]} x_j^{[1]} \cdots \sum x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \cdots x_{i-1}^{[\epsilon_{i-1}]} x_{i+1}^{[\epsilon_{i+1}]} \cdots x_{j-1}^{[\epsilon_{i-1}]} x_{j+1}^{[\epsilon_{i+1}]} \cdots x_n^{[\epsilon_n]} \\ + \; x_a^{[1]} x_b^{[1]} \cdots \sum x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \cdots x_{a-1}^{[\epsilon_{a-1}]} x_{a+1}^{[\epsilon_{a+1}]} \cdots x_{b-1}^{[\epsilon_{b-1}]} x_{b+1}^{[\epsilon_{b+1}]} \cdots \cdots x_n^{[\epsilon_n]} \\ + \; x_u^{[1]} x_v^{[1]} \cdots \sum x_1^{[\epsilon_1]} x_2^{[\epsilon_2]} \cdots x_{u-1}^{[\epsilon_{u-1}]} x_{u+1}^{[\epsilon_{u+1}]} \cdots x_{v-1}^{[\epsilon_{v-1}]} x_{v+1}^{[\epsilon_{v+1}]} \cdots \cdots x_n^{[\epsilon_n]} \end{aligned} \tag{1.6.18}$$

donne

$$\begin{aligned} \{s_1, s_2, s_3\}\widehat{\ } = \{ t : t \subseteq [n], ((s_1 \setminus (s_2 \cup s_3)) \subseteq t \\ \text{ou } (s_2 \setminus (s_1 \cup s_3)) \subseteq t \\ \text{ou } (s_3 \setminus (s_1 \cup s_2)) \subseteq t \\ \text{ou } (s_1 \cup s_2 \cup s_3) \subseteq t))\}, \end{aligned}$$

$$\{s_1, s_2, s_3\}\widehat{\ } = \{ t \subseteq [n] : t \text{ contient 1 ou bien 3 éléments } \in S\}. \tag{1.6.19}$$

9. De manière similaire au résultat précédent, si $S = \{s_1, s_2, \ldots\}$, alors

$$\widehat{S} = \{ t \subseteq [n] : t \text{ contient un nombre impair de } s \in S\}. \tag{1.6.20}$$

Ces résultats peuvent être démontrés en appréhendant l'entrée de la fonction chapeau soit comme une représentation de table de vérité ou bien comme une représentation de polynôme.

CHAPITRE II

PITS ET DÉCOMPOSITIONS DE FONCTIONS

2.1 Introduction

Diverses décompositions de fonctions permettant le passage de fonctions définies sur des entiers non négatifs en fonctions définies sur des chiffres sont étudiées.

2.2 Les pits

Dans ce chapitre, l'écriture $\equiv_p$ désigne l'équivalence modulo p où p est un nombre premier au lieu d'écrire «(modulo p)» à droite de l'équation. Dans de telles circonstances, prenons pour acquis que le résultat de la somme et du produit sont modulo p.

Définition 2.1 *Un nombre entier non négatif peut être écrit dans n'importe quelle base entière supérieure à un. Un nombre est écrit dans une «base première» si la base de numération utilisée est un nombre premier. Les chiffres d'un nombre écrit dans une base première sont appelés «pits».*

De manière similaire au mot «bit» qui vient de l'expression anglophone «binary digit» et qui signifie chiffre binaire, le mot «pit» défini par Gilbert

Labelle vient de l'expression «prime digit» et signifie, chiffre en base première. Puisque le nombre deux est un nombre premier, il est à noter qu'un bit est un pit.

Exemple 2.2.1 *Examinons le nombre* 4031 *en base cinq. Notons que le nombre cinq est premier et que les chiffres des nombres en base cinq peuvent être* 0, 1, 2, 3 *et* 4 *seulement. Le nombre* 4031 *possède quatre pits :* 4, 0, 3 *et* 1. *Ce nombre correspond à 516 dans la base décimale*

$$4031_{(5)} = 4 \times 5^3 + 3 \times 5 + 1 = 516_{(10)}. \tag{2.2.1}$$

Notation 7 *L'indice inférieur, mis entre parenthèses, d'un nombre désigne la base dans laquelle il est exprimé.*

Ce chapitre porte sur l'étude des décompositions de fonctions en pits. Mais, qu'est-ce qu'une décomposition de fonctions en pits ? Au chapitre précédent, des fonctions n-aires définies sur des bits ont été vues. Lorsque la base d'un nombre premier est p, la fonction n-aire portant sur les chiffres se décrit par

$$f : [p]^n \longrightarrow [p], \tag{2.2.2}$$

$$(a_0, a_1, \ldots, a_{n-1}) \longmapsto b_0. \tag{2.2.3}$$

L'entrée de la fonction peut aussi être interprété comme étant un nombre avec n pits $a = a_{n-1}a_{n-2}\ldots a_1a_0$ au lieu d'une suite de pits (d'un n-tuple) :

$$a = \ldots a_2a_1a_0 \longmapsto b_0. \tag{2.2.4}$$

Cependant, pour manipuler des fonctions plus générales

$$f : \mathbb{N} \longrightarrow \mathbb{N}, \tag{2.2.5}$$

$$a = \ldots a_2a_1a_0 \longmapsto b = \ldots b_2b_1b_0, \tag{2.2.6}$$

il suffit de décomposer la fonction f en plusieurs petites fonctions qui n'ont qu'un pit à la sortie.

$$\begin{aligned} f_0(a_0, a_1, a_2, \dots) &\equiv_p b_0, \\ f_1(a_0, a_1, a_2, \dots) &\equiv_p b_1, \\ &\vdots \\ f_m(a_0, a_1, a_2, \dots) &\equiv_p b_m, \\ &\vdots \end{aligned}$$

Ceci est une décomposition de fonctions en pits.

Notation 8 *Utilisons la notation a_i pour représenter le $i^{\text{ème}}$pit en base p d'un nombre entier non négatif a. En d'autres termes,*

$$a = a_0 + a_1 p + a_2 p^2 + \cdots + a_{n-1} p^{n-1}, \qquad (0 \le a_i < p). \tag{2.2.7}$$

Ainsi, si $z = f(x)$ alors, $z_i = (f(x))_i$ dénote le $i^{\text{ème}}$chiffre de la sortie de la fonction f. Dans ce chapitre, plusieurs formes de décomposions de fonctions z en pits z_i sont étudiées.

2.3 Première décomposition de fonctions en pits

Il est possible de décomposer la fonction $f(x)$ en fonctions z_i à l'aide de la série de Newton et du Théorème de Lucas. Mais, avant de se lancer dans des formules avec des coefficients binomiaux, il est important d'étudier le comportement des factoriels modulo p.

2.3.1 Théorème de Wilson

Lemme 2.3.1 *Supposons que x est son propre inverse multiplicatif, alors $x \equiv_p 1$ ou $x \equiv_p p - 1$.*

Démonstration du Lemme :

Par hypothèse,

$$x \cdot x = x^2 \equiv_p 1. \tag{2.3.1}$$

Cette dernière congruence équivaut à :

$$x^2 - 1 \equiv_p 0, \tag{2.3.2}$$

$$(x+1)(x-1) \equiv_p 0. \tag{2.3.3}$$

Donc,

$$x + 1 \equiv_p 0 \quad ou \quad x - 1 \equiv_p 0. \tag{2.3.4}$$

Autrement dit,

$$x \equiv_p -1 \equiv_p p - 1 \quad ou \quad x \equiv_p 1. \tag{2.3.5}$$

Alors, les seules solutions de la congruence $x^2 \equiv_p 1$ sont $x \equiv_p 1$ et $x \equiv_p p - 1$. ■

Théorème 2.3.2 *(*Théorème de Wilson*, (Dickson, 1971), (Hardy, 1960)) Si p est un nombre premier alors,*

$$(p-1)! \equiv_p -1. \tag{2.3.6}$$

Notation 9 *Un nombre modulo p est compris entre* 0 *et* $p-1$ *inclusivement. Lorsqu'il est écrit* -1*, il est sous-entendu qu'il s'agit de* $p-1$*.*

Démonstration du Théorème :

D'après le Lemme précédent, seuls les éléments 1 et $(p-1)$ sont leur propre inverse multiplicatif

$$1^2 = 1 \quad et \quad (p-1)^2 = p^2 - 2p + 1 \equiv_p 1. \tag{2.3.7}$$

Tous les autres éléments non nuls restants $n \in \{2, 3, \ldots, p-2\}$ ont un inverse multiplicatif qui n'est pas n.

$$n^{-1} \neq n, \quad \text{(modulo p)}. \tag{2.3.8}$$

Donc, si toutes les paires d'éléments, n et n^{-1}, appartenant à cet ensemble sont multipliées deux à deux, ils s'annulent et on obtient

$$(p-2)! \equiv_p 1. \tag{2.3.9}$$

Si cette dernière équation est multipliée par $p-1$, il en résulte que

$$(p-1)! \equiv_p -1. \tag{2.3.10}$$

■

Notons que

$$p! \equiv_p 0 \tag{2.3.11}$$

puisque

$$p! = p(p-1)!, \tag{2.3.12}$$

qui est trivialement divisible par p.
Jusqu'à présent, il a été montré que

$$(p-2)! \equiv_p 1, \qquad (p-1)! \equiv_p -1, \qquad p! \equiv_p 0. \tag{2.3.13}$$

Mais, que vaut $(p-k)!$ de manière générale ? Le théorème qui suit est un petit résultat fort intéressant que j'ai découvert.

Théorème 2.3.3 *Si p est un nombre premier et k un entier tel que $0 < k \leq p$, alors*

$$(p-k)! \equiv_p \frac{(-1)^k}{(k-1)!}. \tag{2.3.14}$$

Notons que le membre de droite de la congruence a un sens puisque $(k-1)!$ est inversible modulo p.

Démonstration du Théorème :
Par le Théorème de Wilson,

$$-1 \equiv_p (p-1)! \tag{2.3.15}$$

Mais,

$$(p-1)! = (p-1)(p-2)\cdots(p-(k-1))(p-k)! \quad (2.3.16)$$

$$-1 \equiv_p (-1)(-2)\cdots(-(k-1))(p-k)! \quad (2.3.17)$$

$$-1 \equiv_p (-1)^{k-1}(k-1)!(p-k)! \quad (2.3.18)$$

Donc,

$$(p-k)! \equiv_p \frac{(-1)^k}{(k-1)!} \quad (2.3.19)$$

■

2.3.2 Théorème de Lucas

Il est possible de décomposer un coefficient binomial en un produit de plus petits coefficients binomiaux dans une arithmétique modulo p. Le Théorème de Lucas montre une méthode efficace permettant de calculer un coefficient binomial.

Théorème 2.3.4 (Théorème de Lucas, *(Fine, 1947), (Lucas, 1878)) Soient A et B, deux nombres en base première p tels que :*

$$A = A_0 + A_1 p + A_2 p^2 + \ldots, \qquad (0 \leq A_i < p), \quad (2.3.20)$$

$$B = B_0 + B_1 p + B_2 p^2 + \ldots, \qquad (0 \leq B_i < p). \quad (2.3.21)$$

Alors,

$$\binom{A}{B} \equiv_p \binom{A_0}{B_0}\binom{A_1}{B_1}\binom{A_2}{B_2}\cdots. \quad (2.3.22)$$

Notons que le produit «infini» de droite a toujours un sens puisqu'il existe un m tel que pour tout $n > m$,

$$\binom{A_n}{B_n} = \binom{0}{0} = 1. \quad (2.3.23)$$

Illustrons le Théorème de Lucas avec un exemple.

Exemple 2.3.1 *Soit* $234_{(5)}$ *et* $104_{(5)}$, *deux nombres en base cinq, alors*

$$\binom{234_{(5)}}{104_{(5)}} \equiv_p \binom{2}{1}\binom{3}{0}\binom{4}{4}. \tag{2.3.24}$$

Cette dernière équation peut être vérifiée.

1. Le coefficient binomial calculé de manière usuelle donne

$$\binom{234_{(5)}}{104_{(5)}} = 23720460024918645912_{(10)} \equiv_5 2. \tag{2.3.25}$$

2. Le coefficient binomial calculé avec le Théorème de Lucas donne

$$\binom{2}{1}\binom{3}{0}\binom{4}{4} \equiv_5 2 \cdot 1 \cdot 1 = 2. \tag{2.3.26}$$

Il existe plusieurs démonstrations du Théorème de Lucas. Une preuve utilisant des séries formelles est présentée un peu plus tard. Mais tout d'abord, voici quelques Lemmes qui vont aider à la démonstration de ce Théorème.

Lemme 2.3.5 *Le nombre de facteurs de p dans n! est*

$$\left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \left\lfloor \frac{n}{p^3} \right\rfloor + \dots \tag{2.3.27}$$

Il en résulte que si M est un entier relativement permier avec p, alors

$$n! = Mp^{\left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \left\lfloor \frac{n}{p^3} \right\rfloor + \dots}. \tag{2.3.28}$$

Exemple 2.3.2 *Si* $p = 3$, *quel est le nombre de facteurs de p dans n! ?*

$$\begin{aligned} n! \;=\; & 1 \cdot 2 \cdot (3) \cdot 4 \cdot 5 \cdot (3 \cdot 2) \cdot 7 \cdot 8 \cdot (3^2) \\ & \cdot 10 \cdot 11 \cdot (3 \cdot 4) \cdot 13 \cdot 14 \cdot (3 \cdot 5) \cdot 16 \cdot 17 \cdot (3^2 \cdot 2) \\ & \cdot 19 \cdot 20 \cdot (3 \cdot 7) \cdot 22 \cdot 23 \cdot (3 \cdot 8) \cdot 25 \cdot 26 \cdot (3^3) \cdots n. \end{aligned}$$

Un facteur de p s'ajoute à tous les bonds de 3. Même chose en ce qui concerne les bonds de $3^2, 3^3, \dots$ le nombre de facteurs de p dans $n!$ est donc

$$\left\lfloor \frac{n}{3} \right\rfloor + \left\lfloor \frac{n}{3^2} \right\rfloor + \left\lfloor \frac{n}{3^3} \right\rfloor + \dots \tag{2.3.29}$$

Démonstration du Lemme :

Le tout peut être démontré de manière similaire à l'exemple précédent

$$n! = 1 \cdot 2 \cdots p \cdots (2p) \cdots (3p) \cdots (p^2) \cdots (2p^2) \cdots n. \tag{2.3.30}$$

Le nombre $n!$ comprend :

- $\left\lfloor \frac{n}{p} \right\rfloor$ occurrences de p pour les sauts de longueur p.
- $\left\lfloor \frac{n}{p^2} \right\rfloor$ occurrences supplémentaires de p pour les sauts de longueur p^2.
- $\left\lfloor \frac{n}{p^3} \right\rfloor$ occurrences supplémentaires de p pour les sauts de longueur p^3.
- ...

le nombre de facteurs de p dans $n!$ est donc

$$\left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \left\lfloor \frac{n}{p^3} \right\rfloor + \ldots \tag{2.3.31}$$

■

Remarque 2.1 *Pour faciliter les calculs, il est important de noter que*

$$\left\lfloor \frac{n}{p^{i+1}} \right\rfloor = \left\lfloor \frac{\left\lfloor \frac{n}{p^i} \right\rfloor}{p} \right\rfloor. \tag{2.3.32}$$

La nombre $\left\lfloor \frac{n}{p^{i+1}} \right\rfloor$ *peut donc être trouvé de manière itérative.*

Exemple 2.3.3 *Si* $p = 3$ *et* $n = 100_{(10)}$, *alors*

$$100! = 3^{33+11+3+1}M = 3^{48}M \tag{2.3.33}$$

où M *est un nombre relativement premier avec* p.

Lemme 2.3.6 *(Legendre, 1808). Si n est un nombre écrit en base première p,*

$$n = n_0 + n_1 p + n_2 p^2 + \ldots \tag{2.3.34}$$

alors, le nombre de facteurs de p dans n! est égal à

$$\frac{n - \sum_{i \geq 0} n_i}{p - 1}. \tag{2.3.35}$$

Exemple 2.3.4 *Si $p = 3$ et $n = 100_{(10)}$, quel est l'exposant de p dans «$n!$» ?*

$$100_{(10)} = 3^4 + 2 \cdot 3^2 + 1 = 10201_{(3)}. \tag{2.3.36}$$

Par le Lemme précédent, le nombre de facteurs de p dans $n!$ est

$$\frac{100 - 4}{3 - 1} = \frac{96}{2} = 48. \tag{2.3.37}$$

Démonstration du Lemme :

Le nombre de facteurs de p dans $n!$ est

$$\left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \left\lfloor \frac{n}{p^3} \right\rfloor + \dots \tag{2.3.38}$$

Or,

$$\begin{aligned} n &= n_0 + n_1 p + n_2 p^2 + n_3 p^3 + \dots, \\ \left\lfloor \frac{n}{p} \right\rfloor &= n_1 + n_2 p + n_3 p^2 + n_4 p^3 + \dots, \\ \left\lfloor \frac{n}{p^2} \right\rfloor &= n_2 + n_3 p + n_4 p^2 + n_5 p^3 + \dots, \\ &\vdots \end{aligned}$$

L'expression (2.3.38) est donc égale à

$$(n_1 + n_2 + \dots) + (n_2 + n_3 + \dots)p + (n_3 + n_4 + \dots)p^2 + \dots \tag{2.3.39}$$

Cette expression peut être réorganisé autrement,

$$0 + n_1 + n_2(1 + p) + n_3(1 + p + p^2) + n_4(1 + p + p^2 + p^3) + \dots \tag{2.3.40}$$

Mais,

$$\sum_{i=0}^{k-1} p^i = \frac{p^k - 1}{p - 1}. \tag{2.3.41}$$

En substituant cette égalité dans l'expression (2.3.40), on obtient

$$n_0\left(\frac{1-1}{p-1}\right) + n_1\left(\frac{p-1}{p-1}\right) + n_2\left(\frac{p^2-1}{p-1}\right) + n_3\left(\frac{p^3-1}{p-1}\right) + n_4\left(\frac{p^4-1}{p-1}\right) + \dots \tag{2.3.42}$$

En mettant en évidence $\frac{1}{p-1}$, nous obtenons

$$\frac{n_0(1-1)+n_1(p-1)+n_2(p^2-1)+n_3(p^3-1)+n_4(p^4-1)+\ldots}{p-1}. \tag{2.3.43}$$

Cette dernière expression est égale à

$$\frac{\sum_{i\geq 0} n_i p^i - \sum_{i\geq 0} n_i}{p-1}. \tag{2.3.44}$$

D'où, le nombre de facteurs de p dans $n!$ est

$$\frac{n-\sum_{i\geq 0} n_i}{p-1}. \tag{2.3.45}$$

■

Notation 10 *Posons $s(k)$ comme étant la somme usuelle des chiffres de k en base p.*

$$s(k) = k_0 + k_1 + k_2 + \ldots \tag{2.3.46}$$

Lemme 2.3.7 *Le nombre de facteurs de p dans $\binom{m}{n}$ est*

$$\frac{s(n)+s(m-n)-s(m)}{p-1}. \tag{2.3.47}$$

Démonstration du Lemme :

Puisque,

$$\binom{m}{n} = \frac{m!}{(m-n)!n!} \tag{2.3.48}$$

et que par le Lemme 2.3.6 et la Notation 10, le nombre de facteurs de p dans $n!$, $m!$ et $(m-n)!$ est respectivement égal à

$$\frac{n-s(n)}{p-1}, \quad \frac{m-s(m)}{p-1} \quad et \quad \frac{(m-n)-s(m-n)}{p-1} \tag{2.3.49}$$

alors, le nombre de facteurs de p dans $\binom{m}{n}$ est donc

$$\frac{m-s(m)}{p-1} - \frac{n-s(n)}{p-1} - \frac{(m-n)-s(m-n)}{p-1} \tag{2.3.50}$$

si on additionne les facteurs au numérateur et que l'on soustrait les facteurs du dénominateur de $\binom{m}{n}$. En simplifiant le tout, on obtient

$$\frac{s(n)+s(m-n)-s(m)}{p-1}. \tag{2.3.51}$$

comme nombre de facteurs de p dans $\binom{m}{n}$ ∎

Lemme 2.3.8 (KUMMER, *(Kummer, 1852), (Fine, 1947)*) *Si p est un nombre premier et $j > 0$, alors*

$$\binom{p^j}{i} \equiv_p 0, \qquad (0 < i < p^j). \tag{2.3.52}$$

Des outils sont définis dans le but de prouver le Théorème de Lucas. Le Théorème de Lucas ne peut être utilisé dans la démonstration de ce Lemme afin d'éviter une preuve circulaire.

Démonstration du Lemme :

Le nombre de facteurs de p dans $\binom{p^j}{i}$ est

$$\frac{s(i)+s(p^j-i)-s(p^j)}{p-1}. \tag{2.3.53}$$

Mais, puisque $p^j = 0 + 0p + 0p^2 + \cdots + p^j$, alors

$$s(p^j) = 1. \tag{2.3.54}$$

Comme $0 < i < p^j$ et $0 < (p^j - i) < p^j$, alors

$$s(i) \geq 1, \qquad s(p^j - i) \geq 1. \tag{2.3.55}$$

Il en résultera que

$$\frac{s(i)+s(p^j-i)-s(p^j)}{p-1} = \frac{s(i)+s(p^j-i)-1}{p-1} \geq \frac{1+1-1}{p-1} > 0 \tag{2.3.56}$$

Le nombre de facteurs de p dans $\binom{p^j}{i}$ est donc supérieur à zéro. Alors, $\binom{p^j}{i}$ admet au moins un facteur p. D'où,

$$\binom{p^j}{i} \equiv_p 0, \qquad (0 < i < p^j). \tag{2.3.57}$$

■

Démonstration du Théorème de Lucas :

Puisque $\sum_{i=0}^{n} \binom{n}{i} a^i b^{n-i} = (a+b)^n$, alors

$$\sum_{B=0}^{A} \binom{A}{B} x^B = (1+x)^A. \tag{2.3.58}$$

Par définition, $A = \sum_{i=0}^{n} A_i p^i$ et par substitution,

$$\sum_{B=0}^{A} \binom{A}{B} x^B = (1+x)^{A_0 + A_1 p^1 + \cdots + A_n p^n}. \tag{2.3.59}$$

Le tout peut être écrit autrement.

$$\sum_{B=0}^{A} \binom{A}{B} x^B = \prod_{j=0}^{n} \left((1+x)^{p^j}\right)^{A_j}. \tag{2.3.60}$$

Cependant,

$$(1+x)^{p^j} = \sum_{i=0}^{p^j} \binom{p^j}{i} x^i = 1 + x^{p^j} \tag{2.3.61}$$

car par le Lemme précédent,

$$\binom{p^j}{i} \equiv_p 0, \qquad (0 < i < p^j). \tag{2.3.62}$$

Donc,

$$\sum_{B=0}^{A} \binom{A}{B} x^B = \prod_{j=0}^{n} \left(1 + x^{p^j}\right)^{A_j}. \tag{2.3.63}$$

Mais, il est vrai que

$$(1 + x^{p^j})^{A_j} = \sum_{B_j=0}^{A_j} \binom{A_j}{B_j} (x^{p^j})^{B_j}. \tag{2.3.64}$$

D'où,

$$\sum_{B=0}^{A} \binom{A}{B} x^B = \prod_{j=0}^{n} \left(\sum_{B_j=0}^{A_j} \binom{A_j}{B_j} x^{B_j p^j} \right) = \sum_{B=0}^{A} \left(\prod_{j \geq 0} \binom{A_j}{B_j} \right) x^B. \tag{2.3.65}$$

Comme les intérieurs des sommations sont égaux, il en résulte que

$$\binom{A}{B} = \binom{A_0}{B_0}\binom{A_1}{B_1}\binom{A_2}{B_2}\cdots . \tag{2.3.66}$$

■

Démonstration (2) du Théorème de Lucas :

Par définition,

$$A = A_0 + p\left\lfloor \frac{A}{p} \right\rfloor \tag{2.3.67}$$

Dans cette preuve, il suffit de prouver que

$$\binom{A}{B} = \binom{A_0}{B_0}\binom{\lfloor \frac{A}{p} \rfloor}{\lfloor \frac{B}{p} \rfloor} \tag{2.3.68}$$

Par le binôme de Newton,

$$(1+x)^A = \sum_{B=0}^{A} \binom{A}{B} x^B \tag{2.3.69}$$

Néanmoins,

$$(1+x)^A = (1+x)^{A_0 + p\lfloor \frac{A}{p} \rfloor} = (1+x)^{A_0}((1+x)^p)^{\lfloor \frac{A}{p} \rfloor} \tag{2.3.70}$$

Mais,

$$(1+x)^p = \sum_{i=0} \binom{p}{i} x^i, \tag{2.3.71}$$

où,

$$\binom{p}{i} \equiv_p 0, \qquad (0 < i < p). \tag{2.3.72}$$

Donc,

$$(1+x)^A \equiv_p (1+x)^{A_0}(1+x^p)^{\lfloor \frac{A}{p} \rfloor}. \tag{2.3.73}$$

Par le binôme de Newton, on obtient

$$\sum_{B=0}^{A} \binom{A}{B} x^B \equiv_p \left(\sum_{k=0}^{A_0} \binom{A_0}{k} x^k \right) \left(\sum_{l=0}^{\lfloor \frac{A}{p} \rfloor} \binom{\lfloor \frac{A}{p} \rfloor}{l} x^{pl} \right), \tag{2.3.74}$$

qui après simplification donne

$$\sum_{B=0}^{A}\binom{A}{B}x^B \equiv_p \sum_{\substack{0\le k\le A_0\le p\\ 0\le l\le\left\lfloor\frac{A}{p}\right\rfloor}}\binom{A_0}{k}\binom{\left\lfloor\frac{A}{p}\right\rfloor}{l}x^{k+pl}. \tag{2.3.75}$$

Or, si nous regardons les exposants,

$$B = k + pl. \tag{2.3.76}$$

Ceci est vrai si

$$k = B_0, \qquad l = \left\lfloor\frac{B}{p}\right\rfloor. \tag{2.3.77}$$

Alors,

$$\sum_{B=0}^{A}\binom{A}{B}x^B \equiv_p \sum_{B=0}^{A}\binom{A_0}{B_0}\binom{\left\lfloor\frac{A}{p}\right\rfloor}{\left\lfloor\frac{B}{p}\right\rfloor}x^B. \tag{2.3.78}$$

Il en résulte que

$$\binom{A}{B} \equiv_p \binom{A_0}{B_0}\binom{\left\lfloor\frac{A}{p}\right\rfloor}{\left\lfloor\frac{B}{p}\right\rfloor}. \tag{2.3.79}$$

D'où, la congruence de Lucas. ■

Corollaire 2.3.9 *Si A et B sont deux nombres en base première p, alors :*

$$(\exists\, i)\ (A_i < B_i) \quad \Rightarrow \quad \binom{A_i}{B_i} = 0 \quad \Rightarrow \quad \binom{A}{B} \equiv_p 0. \tag{2.3.80}$$

Par le Théorème de Lucas et le Corollaire 2.3.9, il est facile de voir que

$$\binom{p^j}{i} \equiv_p 0, \qquad (0 < i < p^j). \tag{2.3.81}$$

Corollaire 2.3.10 *Soit z_i, le $i^{ème}$chiffre d'un entier z, alors*

$$z_i \equiv_p \binom{z}{p^i}. \tag{2.3.82}$$

Démonstration du Corollaire 2.3.10 :

Par le Théorème de Lucas, $\binom{z}{p^i} \equiv_p \binom{z_0}{0}\binom{z_1}{0}\dots\binom{z_{i-1}}{0}\binom{z_i}{1}\binom{z_{i+1}}{0}\cdots \equiv_p z_i$. ■

2.3.3 Théorème de décomposition 1

À partir du calcul des différences finies (Guelfond, 1963), il est possible de trouver une première décomposition de fonction en pits.

Théorème 2.3.11 (Théorème de décomposition 1) *Si p est un nombre premier et $z = f(x)$, alors*

$$z_i \equiv_p \sum_{m\geq 0} \alpha_m(i)\binom{x_0}{m_0}\binom{x_1}{m_1}\binom{x_2}{m_2}\cdots \tag{2.3.83}$$

où $\alpha_m(i)$ satisfait

$$\alpha_m(i) \equiv_p \sum_{\mu=0}^{m}(-1)^{m-\mu}\binom{m}{\mu}f(\mu)_i. \tag{2.3.84}$$

Démonstration du Théorème 2.3.11 :

La série de Newton est définie par

$$F(x) = \sum_{m\geq 0}\Delta^m F(0)\binom{x}{m}. \tag{2.3.85}$$

Mais, que vaut $\Delta^m F(0)$ dans cette équation ?

$$\begin{aligned}
\Delta F(0) &= F(1) - F(0),\\
\Delta^2 F(0) &= \Delta(\Delta F(0)),\\
&= \Delta(F(1) - F(0)),\\
&= (F(2) - F(1)) - (F(1) - F(0)),\\
&= F(0) - 2F(1) + F(2),\\
&= \binom{2}{0}F(0) - \binom{2}{1}F(1) + \binom{2}{2}F(2),\\
&\vdots
\end{aligned}$$

De manière générale, la formule suivante peut être démontrée par récurrence

$$\Delta^m F(0) = \sum_{0\leq\mu\leq m}(-1)^{m-\mu}\binom{m}{\mu}F(\mu). \tag{2.3.86}$$

Par le Théorème de Lucas,

$$\binom{x}{m} \equiv_p \binom{x_0}{m_0}\binom{x_1}{m_1}\binom{x_2}{m_2}\cdots \tag{2.3.87}$$

En substituant les deux dernières équations dans l'équation (2.3.85), on obtient

$$F(x) \equiv_p \sum_{m\geq 0} \alpha'_m(i)\binom{x_0}{m_0}\binom{x_1}{m_1}\binom{x_2}{m_2}\cdots \tag{2.3.88}$$

où $\alpha'_m(i)$ satisfait l'équation

$$\alpha'_m(i) = \sum_{\mu=0}^{m}(-1)^{m-\mu}\binom{m}{\mu}F(\mu). \tag{2.3.89}$$

Posons que

$$F(x) = \binom{f(x)}{p^i}. \tag{2.3.90}$$

Par le Corollaire 2.3.10,

$$z_i = f(x)_i = \binom{f(x)}{p^i}. \tag{2.3.91}$$

En substituant les deux dernières équations dans les équations (2.3.88) et (2.3.89), on obtient

$$f(x)_i \equiv_p \sum_{m\geq 0} \alpha_m(i)\binom{x_0}{m_0}\binom{x_1}{m_1}\binom{x_2}{m_2}\cdots \tag{2.3.92}$$

où $\alpha_m(i)$ satisfait l'équation

$$\alpha_m(i) \equiv_p \sum_{\mu=0}^{m}(-1)^{m-\mu}\binom{m}{\mu}f(\mu)_i. \tag{2.3.93}$$

■

Dans le cas de fonctions à plusieurs variables, la décomposition de la fonction $z = f(x, y, \dots)$ en pits est

$$z_i \equiv_p \sum_{m\geq 0, n\geq 0, \dots} \alpha_{m,n,\dots}(i)\binom{x_0}{m_0}\binom{x_1}{m_1}\cdots\binom{y_0}{n_0}\binom{y_1}{n_1}\cdots \tag{2.3.94}$$

où $\alpha_{m,n,\dots}(i)$ satisfait

$$\alpha_{m,n,\dots}(i) \equiv_p \sum_{0\leq\mu\leq m, 0\leq\nu\leq n} (-1)^{m-\mu}(-1)^{n-\nu}\binom{m}{\mu}\binom{n}{\nu}\cdots f(\mu,\nu,\dots)_i. \tag{2.3.95}$$

Exemple 2.3.5 *Soit $p = 2$ et $z = f(x) = x + 1$. Quel est la décomposition de fonction en pits de $f(x)$?*

$$z_0 \equiv_2 1 + \binom{x_0}{1}, \tag{2.3.96}$$

$$z_1 \equiv_2 \binom{x_0}{1} + \binom{x_1}{1}, \tag{2.3.97}$$

$$z_2 \equiv_2 \binom{x_0}{1}\binom{x_1}{1} + \binom{x_2}{1}, \tag{2.3.98}$$

$$z_3 \equiv_2 \binom{x_0}{1}\binom{x_1}{1}\binom{x_2}{1} + \binom{x_3}{1}, \tag{2.3.99}$$

$$z_4 \equiv_2 \binom{x_0}{1}\binom{x_1}{1}\binom{x_2}{1}\binom{x_3}{1} + \binom{x_4}{1}. \tag{2.3.100}$$

Mais, puisque

$$\binom{x_i}{1} = x_i, \tag{2.3.101}$$

il est clair que

$$z_0 \equiv_2 1 + x_0, \tag{2.3.102}$$

$$z_1 \equiv_2 x_0 + x_1, \tag{2.3.103}$$

$$z_2 \equiv_2 x_0x_1 + x_2, \tag{2.3.104}$$

$$z_3 \equiv_2 x_0x_1x_2 + x_3, \tag{2.3.105}$$

$$z_4 \equiv_2 x_0x_1x_2x_3 + x_4. \tag{2.3.106}$$

Les résultats précédents suggèrent la formule générale suivante.

$$z_n \equiv_2 \prod_{i=0}^{n-1} x_i + x_n. \tag{2.3.107}$$

Cette formule est prouvée au chapitre trois.

2.4 Deuxième décomposition de fonctions en pits

Une autre décomposition peut être construite à partir du delta de Kronecker. Cependant, examinons tout d'abord le petit[1] Théorème de Fermat.

2.4.1 Le petit Théorème de Fermat

Théorème 2.4.1 *(Le petit Théorème de Fermat, (Fermat, 1640)) Soit p, un nombre premier. Alors, pour tout entier a, l'équation suivante est satisfaite*

$$a^p \equiv_p a. \tag{2.4.1}$$

Démonstration du Théorème :

Une preuve par récurrence de cette congruence est effectuée sur a :

1. Si $a = 0$, alors

$$a^p \equiv_p 0. \tag{2.4.2}$$

2. Supposons que $a^p \equiv_p a$, montrons que $(a+1)^p \equiv_p (a+1)$.

$$(1+a)^p = \binom{p}{0} + \binom{p}{1}a + \binom{p}{2}a^2 + \cdots + \binom{p}{p}a^p. \tag{2.4.3}$$

Par le Théorème de Kummer,

$$\binom{p^j}{i} \equiv_p 0, \qquad (0 < i < p^j). \tag{2.4.4}$$

Donc,

$$(1+a)^p \equiv_p 1 + a^p. \tag{2.4.5}$$

Puisque $a^p \equiv_p a$, il en résulte que

$$(1+a)^p \equiv_p 1 + a. \tag{2.4.6}$$

---

[1]Petit, mais important ! Le mot petit est utilisé par opposition au grand Théorème de Fermat qui mentionne que $a^n + b^n \neq c^n$ si $abc \neq 0$ et $n \geq 3$, (a, b, c, n étant des entiers).

Ce qui complète le pas de la récurrence et termine la démonstration du petit Théorème de Fermat. ■

Exemple 2.4.1 *Le petit Théorème de Fermat affirme que* $3^5 \equiv_5 3$.

$$3^5 = 243_{(10)} \equiv_5 3. \tag{2.4.7}$$

Corollaire 2.4.2 *(*Corollaire du petit Théorème de Fermat*) Soit p, un nombre premier. Alors, pour tout entier $a \not\equiv_p 0$, l'équation suivante est satisfaite*

$$a^{p-1} \equiv_p 1. \tag{2.4.8}$$

Démonstration du Corollaire :
Par le petit Théorème de Fermat,

$$a^p - a \equiv_p 0. \tag{2.4.9}$$

Le facteur a peut être mis en évidence

$$a(a^{p-1} - 1) \equiv_p 0. \tag{2.4.10}$$

Si $a \not\equiv_p 0$, soit p divise a ou bien p divise $a^{p-1} - 1$.
Mais, p ne divise pas a. Donc,

$$a^{p-1} - 1 \equiv_p 0, \qquad (a \not\equiv_p 0). \tag{2.4.11}$$

■

2.4.2 Le delta de Kronecker

Il a été vu dans le chapitre précédent que $x_j^{[\epsilon_j]} \equiv_2 1 + \epsilon_j + x_j$. Cependant, il est possible de généraliser cette formule pour une base quelconque en utilisant le delta de Kronecker .

Définition 2.2 *Soit m et x, deux entiers. Le « delta de Kronecker » δ_m^x est défini par*

$$\delta_m^x = \begin{cases} 0 & si \quad x \neq m, \\ 1 & si \quad x = m. \end{cases} \tag{2.4.12}$$

Théorème 2.4.3 *Le delta de Kronecker satisfait les propriétés suivantes*

$$\begin{aligned} &(a) \quad \delta_m^x \equiv_2 1 + x + m, && x, m : bits, \\ &(b) \quad \delta_m^x \equiv_p 1 - (x-m)^{p-1}, && x, m : pits, \\ &(c) \quad \delta_m^x \equiv_p \tfrac{x-x^p}{x-m}, && x, m : pits. \end{aligned} \tag{2.4.13}$$

Démonstration du Théorème :

Vérifions que $\delta_m^x \equiv_p 1 - (x-m)^{p-1}$.

1. Si $x \equiv_p m$ alors,

$$\delta_m^m \equiv_p 1 - (m-m)^{p-1} \equiv_p 1. \tag{2.4.14}$$

2. Si $x \not\equiv_p m$ alors, par le Théorème de Fermat,

$$\delta_m^{x(x\neq m)} \equiv_p 1 - (x-m)^{p-1} \equiv_p 1 - 1 \equiv_p 0. \tag{2.4.15}$$

L'équation (b) satisfait donc la définition du delta de Kronecker.

Lorsque $p = 2$ dans l'équation (b), on obtient

$$\delta_m^x \equiv_2 1 - (x-m). \tag{2.4.16}$$

Mais, comme $a + b \equiv_2 a - b$, il en résulte que

$$\delta_m^x \equiv_2 1 + x + m. \tag{2.4.17}$$

L'équation (a) satisfait donc la définition du delta de Kronecker.

Si le numérateur et le dénominateur de (b) sont multipliés par $(x-m)$, on obtient

$$\delta_m^x \equiv_p \frac{(x-m) - (x-m)^p}{x-m}. \tag{2.4.18}$$

Mais,

$$(x-m)^p = \sum_{i=0}^{p} \binom{p}{i} x^i (-m)^{p-i}. \tag{2.4.19}$$

Par le Théorème de Kummer,

$$\binom{p^j}{i} \equiv_p 0, \qquad (0 < i < p). \tag{2.4.20}$$

Donc,

$$(x-m)^p = x^p - (-m)^p. \tag{2.4.21}$$

En faisant cette substitution dans l'équation du delta de Kronecker, on a

$$\delta_m^x \equiv_p \frac{(x-m) - (x^p - (-m)^p)}{x-m}. \tag{2.4.22}$$

Par le petit Théorème de Fermat,

$$(-m)^p \equiv_p -m. \tag{2.4.23}$$

Ce qui mène à

$$\delta_m^x \equiv_p \frac{x - x^p}{x-m}. \tag{2.4.24}$$

L'équation (c) satisfait donc la définition du delta de Kronecker. ∎

Il est à noter que dans l'équation (c), si $x = m$ alors, il y a une division par 0. Cependant, x n'est jamais évalué.

2.4.3 Théorème de décomposition 2

Théorème 2.4.4 (Théorème de décomposition 2) *Si p est un nombre premier et $z = f(x)$, alors*

$$z_i \equiv_p \sum_{m \geq 0} f(m)_i \prod_{j \geq 0} \delta_{m_j}^{x_j}. \tag{2.4.25}$$

Démonstration du Théorème 2.4.4 :

L'égalité suivante est vérifiée

$$z_i = f(x)_i \equiv_p \sum_{m\geq 0} f(m)_i \delta_m^x. \tag{2.4.26}$$

Ceci est trivial, car seul le terme $f(m)_i$ tel que $m = x$ est conservé.

Mais,

$$\delta_m^x \equiv_p \delta_{m_0}^{x_0} \delta_{m_1}^{x_1} \delta_{m_2}^{x_2} \cdots . \tag{2.4.27}$$

En substituant cette dernière formule dans la congruence (2.4.26), il en résulte que

$$z_i \equiv_p \sum_{m\geq 0} f(m)_i \delta_{m_0}^{x_0} \delta_{m_1}^{x_1} \delta_{m_2}^{x_2} \cdots . \tag{2.4.28}$$

■

Le delta de Kronecker peut prendre différentes formes. En substituant

$$\delta_m^x \equiv_p \frac{x - x^p}{x - m}, \tag{2.4.29}$$

dans la décomposition 2, on obtient le Corollaire suivant.

Corollaire 2.4.5 *Si p est un nombre premier et $z = f(x)$, alors*

$$z_i \equiv_p \sum_{m\geq 0} f(m)_i \prod_{j\geq 0} \frac{x_j - x_j^p}{x_j - m_j}. \tag{2.4.30}$$

Exemple 2.4.2 *Si $p = 2$ et $f(x) = x + 1$, alors*

$$\begin{aligned}
z_0 &\equiv_2 \frac{(x_0 - x_0^2)}{x_0}, \\
z_1 &\equiv_2 \frac{(x_0 - x_0^2)(x_1 - x_1^2)}{(x_0 - 1)x_1} + \frac{(x_0 - x_0^2)(x_1 - x_1^2)}{x_0(x_1 - 1)}, \\
z_2 &\equiv_2 \frac{(x_0 - x_0^2)(x_1 - x_1^2)(x_2 - x_2^2)}{(x_0 - 1)(x_1 - 1)x_2} + \frac{(x_0 - x_0^2)(x_1 - x_1^2)(x_2 - x_2^2)}{x_0 x_1 (x_2 - 1)} \\
&\quad + \frac{(x_0 - x_0^2)(x_1 - x_1^2)(x_2 - x_2^2)}{(x_0 - 1)x_1(x_2 - 1)} + \frac{(x_0 - x_0^2)(x_1 - x_1^2)(x_2 - x_2^2)}{x_0(x_1 - 1)(x_2 - 1)}.
\end{aligned}$$

Remarquons que par une des propriétés du delta de Kronecker

$$\delta_0^{x_0} = \frac{(x_0 - x_0^2)}{x_0}. \tag{2.4.31}$$

Mais, par une autre propriété du delta de Kronecker, on obtient

$$\delta_0^{x_0} = 1 + x_0. \tag{2.4.32}$$

En simplifiant mécaniquement z_0, z_1 et z_2 avec Maple, il en résulte :

$$z_0 \equiv_2 1 + x_0, \tag{2.4.33}$$

$$z_1 \equiv_2 x_0 + x_1, \tag{2.4.34}$$

$$z_2 \equiv_2 x_0 x_1 + x_2. \tag{2.4.35}$$

C'est une manière alternative à la méthode de la décomposition 1 qui utilise les coefficients du binôme de Newton et le Théorème de Lucas.

Cependant, il existe une autre forme relative à la décomposition 2. Au lieu de substituer

$$\delta_m^x \equiv_p \frac{x - x^p}{x - m}, \tag{2.4.36}$$

dans la décomposition 2, on peut substituer

$$\delta_m^x \equiv_p 1 - (x - m)^{p-1}. \tag{2.4.37}$$

Corollaire 2.4.6 *Si p est un nombre premier et $z = f(x)$, alors*

$$z_i \equiv_p \sum_{m \geq 0} f(m)_i \prod_{j \geq 0} \left(1 - (x_j - m_j)^{p-1}\right). \tag{2.4.38}$$

Puisque $a + b \equiv_2 a - b$, cette dernière formule est une généralisation de la forme polynomiale binaire vue au premier chapitre

$$z_i \equiv_2 \sum_{m \geq 0} f(m)_i \prod_{j \geq 0} \left(1 + x_j + m_j\right). \tag{2.4.39}$$

Exemple 2.4.3 *Si $p = 2$ et $f(x) = x + 1$, alors :*

$$\begin{aligned} z_0 &\equiv_2 1 + x_0, \\ z_1 &\equiv_2 (2 + x_0)(1 + x_1) + (1 + x_0)(2 + x_1), \\ z_2 &\equiv_2 (2 + x_0)(2 + x_1)(1 + x_2) + (1 + x_0)(1 + x_1)(2 + x_2), \\ &\quad + (2 + x_0)(1 + x_1)(2 + x_2) + (1 + x_0)(2 + x_1)(2 + x_2). \end{aligned}$$

Après simplification, en tenant compte du fait que $2 \equiv_2 0$, on obtient :

$$z_0 \equiv_2 1 + x_0, \tag{2.4.40}$$

$$z_1 \equiv_2 x_0 + x_1, \tag{2.4.41}$$

$$z_2 \equiv_2 x_0 x_1 + x_2. \tag{2.4.42}$$

En ce qui concerne les fonctions à plusieurs variables, si la fonction f admet les variables : $x, y, \ldots$ alors, nous pouvons montrer de manière similaire à ce que nous avons fait pour les fonctions à une seule variable que

$$z_i = f(x, y, \ldots)_i = \sum_{m,n,\ldots \geq 0} f(m, n, \ldots)_i \prod_{j \geq 0} \delta_{m_j}^{x_j} \delta_{n_j}^{y_j} \ldots. \tag{2.4.43}$$

2.5 Troisième décomposition de fonctions en pits

Il y a un polynôme modulo p qui décrit le delta de Kronecker. Il est possible d'élaborer une troisième décomposition de fonctions en pits avec les coefficients de ce polynôme.

2.5.1 Matrice associée aux deltas de Kronecker

Définition 2.3 *Le delta de Kronecker peut être défini comme étant un polynôme*

$$\delta_m^x \equiv_p 1 - (x - m)^{p-1} \equiv_p \sum_{i=0}^{p-1} \left\{ {i \atop m} \right\} x^i, \qquad 0 \leq \left\{ {i \atop m} \right\} \leq p - 1. \tag{2.5.1}$$

qui a comme coefficients $\left\{ {i \atop m} \right\}$.

Définition 2.4 *Définissons la «matrice $B_{p\times p}$ associée aux deltas de Kronecker»*

$$B_{p\times p} = \left[\left\{ {i \atop m} \right\}_{0\leq i,m<p} \right] \tag{2.5.2}$$

où i est l'indice des lignes et m est l'indice des colonnes, ces deux indices étant relatifs à 0. Cette matrice contient les coefficients de tous les polynômes des deltas de Kronecker pour un p donné. Un Théorème permettant de trouver rapidement ces valeurs de la matrice $B_{p\times p}$ est présenté un peu plus tard. Voici tout d'abord deux Lemmes qui aideront à la démonstration du Théorème.

Lemme 2.5.1 *Si p est premier et si $0 \leq i < p$, alors*

$$\binom{p-1}{i} \equiv_p (-1)^i. \tag{2.5.3}$$

Démonstration du lemme :

Par la définition du coefficient binomial,

$$\binom{p-1}{i} = \frac{(p-1)(p-2)\dots(p-i)}{i!}. \tag{2.5.4}$$

Puisque pour tout j,

$$p - j \equiv_p -j, \tag{2.5.5}$$

alors,

$$\binom{p-1}{i} \equiv_p \frac{(-1)(-2)\dots(-i)}{i!}. \tag{2.5.6}$$

En simplifiant le tout, il en résulte

$$\binom{p-1}{i} \equiv_p (-1)^i \frac{i!}{i!} = (-1)^i. \tag{2.5.7}$$

■

Lemme 2.5.2 *Si p est un nombre premier, alors δ_m^x satisfait*

$$\delta_m^x \equiv_p 1 - \sum_{i=0}^{p-1} m^{p-1-i} x^i. \tag{2.5.8}$$

Démonstration du Lemme :

Puisque $(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^i b^{n-i}$, alors

$$\delta_m^x \equiv_p 1 - (x-m)^{p-1} = 1 - \sum_{i=0}^{p-1} \binom{p-1}{i} (-m)^{p-1-i} x^i. \tag{2.5.9}$$

Par le Lemme 2.5.1,

$$\delta_m^x \equiv_p 1 - \sum_{i=0}^{p-1} (-1)^i (-m)^{p-1-i} x^i. \tag{2.5.10}$$

Dégroupons $(-m)^{p-1-i}$,

$$\delta_m^x \equiv_p 1 - \sum_{i=0}^{p-1} (-1)^i (-1)^{p-1-i} (m)^{p-1-i} x^i. \tag{2.5.11}$$

Par le Corollaire du petit Théorème de Fermat, $(-1)^{p-1} \equiv_p 1$, d'où,

$$\delta_m^x \equiv_p 1 - \sum_{i=0}^{p-1} (-1)^{i-i} m^{p-1-i} x^i. \tag{2.5.12}$$

Après simplification, il en résulte que

$$\delta_m^x \equiv_p 1 - \sum_{i=0}^{p-1} m^{p-1-i} x^i. \tag{2.5.13}$$

■

Théorème 2.5.3 *Soit p, un nombre premier. La matrice $B_{p\times p} = [\left\{ {i \atop m} \right\}_{0 \le i,m < p}]$ satisfait*

la congruence

$$B_{p\times p} \equiv_p \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1^{-1} & -2^{-1} & \cdots & -(p-1)^{-1} \\ 0 & -1^{-2} & -2^{-2} & \cdots & -(p-1)^{-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -1^{-(p-2)} & -2^{-(p-2)} & \cdots & -(p-1)^{-(p-2)} \\ -1 & -1 & -1 & \cdots & -1 \end{pmatrix}. \tag{2.5.14}$$

Notation 11 *Lorsque $\equiv_p$ est écrit pour des matrices, il est sous-entendu que tous les éléments homologues des matrices résultantes de chaque membre de l'égalité sont congruents modulo p.*

Démonstration du Théorème 2.5.3 :

Par le Lemme 2.5.2,

$$\delta_m^x \equiv_p 1 - \sum_{i=0}^{p-1} m^{p-1-i}x^i \equiv_p \sum_{i=0}^{p-1} \left\{ {i \atop m} \right\} x^i. \tag{2.5.15}$$

Trouvons ce que vaut $\left\{ {i \atop m} \right\}$ pour tout i et m

1. Si $i = 0$ (première ligne de B), alors

$$\left\{ {0 \atop m} \right\} \equiv_p 1 - (-m)^{p-1} \equiv_p \delta_m^0 \equiv_p \begin{cases} 1 & si \quad m = 0, \\ 0 & si \quad m \neq 0. \end{cases} \tag{2.5.16}$$

2. Si $m = 0$ (première colonne de B) et $i \neq 0$, alors :

   (a) Si $0 < i < p-1$, alors : $\left\{ {i \atop 0} \right\} \equiv_p -0^{p-1-i} \equiv_p 0$.

   (b) Si $i = p-1$, alors : $\left\{ {p-1 \atop 0} \right\} \equiv_p -0^{p-1-(p-1)} \equiv_p -0^0 \equiv_p -1$.

3. Si $m \neq 0$ et $i \neq 0$, alors par le Corollaire du petit Théorème de Fermat,

$$\left\{ {i \atop m} \right\} \equiv_p -m^{p-1-i} \equiv_p \begin{cases} -m^{-i} & si \quad 0 < i < p-1, \\ -1 & si \quad i = p-1. \end{cases} \tag{2.5.17}$$

Avec ces informations, tous les éléments de la matrice sont trouvés. ■

Exemple 2.5.1 *Pour $p = 3$,*

$$B_{p\times p} \equiv_p \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1^{-1} & -2^{-1} \\ -1 & -1 & -1 \end{pmatrix} \equiv_p \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix}. \tag{2.5.18}$$

Par définition, la propriété suivante est satisfaite

$$\begin{pmatrix} \delta_0^x & \delta_1^x & \delta_2^x \end{pmatrix} \equiv_p \begin{pmatrix} 1 & x & x^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix}. \tag{2.5.19}$$

Après calculs, on obtient

$$\delta_0^x \equiv_3 1 + 2x^2, \quad \delta_1^x \equiv_3 1 + 2x + 2x^2, \quad \delta_2^x \equiv_3 1 + x + 2x^2. \tag{2.5.20}$$

Une vérification exhaustive assure la validité du delta de Kronecker

$$\begin{aligned} &\delta_0^0 \equiv_3 1 + 2(0)^2 \equiv_3 1, \quad \delta_1^0 \equiv_3 1 + 2(0) + 2(0)^2 \equiv_3 0, \quad \delta_2^0 \equiv_3 1 + (0) + 2(0)^2 \equiv_3 0, \\ &\delta_0^1 \equiv_3 1 + 2(1)^2 \equiv_3 0, \quad \delta_1^1 \equiv_3 1 + 2(1) + 2(1)^2 \equiv_3 1, \quad \delta_2^1 \equiv_3 1 + (1) + 2(1)^2 \equiv_3 0, \\ &\delta_0^2 \equiv_3 1 + 2(2)^2 \equiv_3 0, \quad \delta_1^2 \equiv_3 1 + 2(2) + 2(2)^2 \equiv_3 0, \quad \delta_2^2 \equiv_3 1 + (2) + 2(2)^2 \equiv_3 1. \end{aligned} \tag{2.5.21}$$

Exemple 2.5.2 *Pour $p = 5$ et $p = 7$, on a*

$$B_{5\times 5} \equiv_5 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 2 & 3 & 1 \\ 0 & 4 & 1 & 1 & 4 \\ 0 & 4 & 3 & 2 & 1 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}, \qquad B_{7\times 7} \equiv_7 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 3 & 2 & 5 & 4 & 1 \\ 0 & 6 & 5 & 3 & 3 & 5 & 6 \\ 0 & 6 & 6 & 1 & 6 & 1 & 1 \\ 0 & 6 & 3 & 5 & 5 & 3 & 6 \\ 0 & 6 & 5 & 4 & 3 & 2 & 1 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}. \tag{2.5.22}$$

2.5.2 Théorème de décomposition 3

Notation 12 *Afin d'alléger la formule du prochain Théorème, définissons les nombres* $\left\{ {a \atop b} \right\}$, *modulo p, de façon similaire au Théorème de Lucas par*

$$\left\{ {a \atop b} \right\} = \left\{ {a_0 \atop b_0} \right\} \left\{ {a_1 \atop b_1} \right\} \left\{ {a_2 \atop b_2} \right\} \cdots \tag{2.5.23}$$

si $a = a_0 + a_1 p + a_2 p^2 + \ldots$ et $b = b_0 + b_1 p + b_2 p^2 + \ldots$

Théorème 2.5.4 (Théorème de décomposition 3) *Si p est un nombre premier et* $z = f(x)$, *alors*

$$z_i \equiv_p \sum_{m \geq 0} \beta_m(i) \prod_{j \geq 0} x_j^{m_j}, \qquad \beta_m(i) \equiv_p \sum_{\mu=0}^{m} \left\{ {m \atop \mu} \right\} f(\mu)_i. \tag{2.5.24}$$

Démonstration du Théorème 2.5.4 :

Par le Théorème de décomposition 2,

$$z_i \equiv_p \sum_{\mu \geq 0} f(\mu)_i \prod_{j \geq 0} \delta_{\mu_j}^{x_j}. \tag{2.5.25}$$

En substituant le delta de Kronecker par le polynôme qui lui est égal, on obtient

$$z_i \equiv_p \sum_{m \geq 0} f(\mu)_i \prod_{j \geq 0} \left(\sum_{0 \leq m_j \leq p-1} \left\{ {m_j \atop \mu_j} \right\} x_j^{m_j} \right). \tag{2.5.26}$$

Il est permis de déplacer la sommation intérieure le plus possible à l'extérieur comme ceci

$$z_i \equiv_p \sum_{\substack{m \geq 0 \\ 0 \leq m_0, m_1, m_2, \ldots \leq p-1}} \left(f(\mu)_i \left(\prod_{j \geq 0} \left\{ {m_j \atop \mu_j} \right\} x_j^{m_j} \right) \right). \tag{2.5.27}$$

En dégroupant les éléments du produit de la dernière équation et en posant

$$m = m_0 + m_1 p + m_2 p^2 + \ldots \tag{2.5.28}$$

on obtient

$$z_i \equiv_p \sum_{m\geq 0}\sum_{\mu\geq 0}\left(f(\mu)_i\left(\prod_{j\geq 0}\left\{\begin{matrix} m_j \\ \mu_j \end{matrix}\right\}\right)\left(\prod_{j\geq 0} x_j^{m_j}\right)\right). \tag{2.5.29}$$

Il est permis de changer les parenthèses comme ceci

$$z_i \equiv_p \sum_{m\geq 0}\left(\sum_{\mu\geq 0} f(\mu)_i \left\{\begin{matrix} \mu \\ m \end{matrix}\right\}\right)\prod_{j\geq 0} x_j^{m_j}. \tag{2.5.30}$$

■

Exemple 2.5.3 *Si $p = 2$ et $f(x) = x + 1$, notons tout d'abord que*

$$B_{2\times 2} \equiv_2 \begin{bmatrix} \left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\} & \left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\} \\ \left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} & \left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\} \end{bmatrix} \equiv_2 \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{2.5.31}$$

Par la décomposition 3, on peut trouver[2]

$$\begin{aligned}
z_0 &\equiv_2 1 + \left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} x_0 + \left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} + \left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}, \\
z_1 &\equiv_2 \left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\} x_0 + \left(\left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} + \left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}\right) x_1, \\
z_2 &\equiv_2 \left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\} x_1 + \left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}^2 x_0 x_1 + \left(\left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\}^2\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} + \left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}^2\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}\right) x_2 \\
&\quad + \left(\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}\left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} + \left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}^2\left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\right) x_0 x_2 \\
&\quad + \left(\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}\left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} + \left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}^2\left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\right) x_1 x_2 \\
&\quad + \left(\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}^2\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} + \left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\}^2\right) x_0 x_1 x_2 + \left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\}^2\left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} \\
&\quad + \left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\}\left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}^2\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\} + \left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\}^2\left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\} + \left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\}\left\{\begin{matrix} 0 \\ 0 \end{matrix}\right\}^2\left\{\begin{matrix} 1 \\ 1 \end{matrix}\right\}.
\end{aligned}$$

En simplifiant ces dernières équations, il en résulte que

$$z_0 \equiv_2 1 + x_0, \tag{2.5.32}$$

---

[2]Toujours en utilisant Maple.

$$z_1 \equiv_2 x_0 + x_1, \tag{2.5.33}$$

$$z_2 \equiv_2 x_0 x_1 + x_2. \tag{2.5.34}$$

Donc, quatre décompositions distinctes permettent d'arriver au même résultat pour $p = 2$ et $f(x) = x + 1$.

Une fonction à plusieurs variables $f(x, y, \dots)$ se décompose en pits de manière similaire au Théorème de décomposition 3

$$z_i = \sum_{m,n,\dots \geq 0} \beta_{m,n,\dots}(i) \prod_{j \geq 0} x_j^{m_j} y_j^{n_j} \cdots \tag{2.5.35}$$

où,

$$\beta_{m,n,\dots}(i) = \sum_{0<\mu<m, 0<\nu<n, \dots} \left\{ \begin{matrix} m \\ \mu \end{matrix} \right\} \left\{ \begin{matrix} n \\ \nu \end{matrix} \right\} \cdots f(\mu, \nu, \dots)_i. \tag{2.5.36}$$

CHAPITRE III

RÉSULTATS DES DÉCOMPOSITIONS DE FONCTIONS

3.1 Introduction

L'utilisation de Maple (Goupil, 2003) permet de calculer rapidement les pits de sortie

$$z_0, z_1, z_2, \ldots, z_k, \tag{3.1.1}$$

d'une fonction $z = f(x, y, \ldots)$ pour un p donné, selon les différentes décompositions de fonctions en pits. À partir de ces résultats, quelques formules générales de décompositions de fonctions sont prouvées. Des algorithmes de décompositions de fonctions en pits qui n'avaient jamais été programmés auparavant sont présentés dans ce chapitre. Cependant, puisque les programmes informatiques ont un caractère fini, il faut se restreindre dans le présent chapitre aux fonctions $z = f(x, y, \ldots)$ telles que chaque pit de sortie z_i est fonction d'un nombre fini de pits de $x, y, \ldots$

Exemple 3.1.1 *Voici une fonction qui ne satisfait pas cette condition.*

$$z = f(x) = x_0 + x_1 + x_2 + \ldots \qquad \textit{(somme ordinaire des pits de x).} \tag{3.1.2}$$

En effet, les nombres $x = p, p^2, p^3, \ldots$ ont tous la valeur $z = 1$ et la valeur du pit $z_0 = 1$. Néanmoins, certaines techniques développées au chapitre précédent

permettent d'énoncer le résultat suivant.

Théorème 3.1.1 *Si $p = 2$ et $z = x_0 + x_1 + x_2 + \ldots$, la somme ordinaire des bits de x alors,*

$$\begin{aligned}
z_0 &\equiv_2 x_0 + x_1 + x_2 + \ldots \\
z_1 &\equiv_2 x_0x_1 + \cdots + x_ix_j + \ldots \\
z_2 &\equiv_2 x_0x_1x_2x_3 + \cdots + x_ix_jx_kx_l + \ldots \\
&\vdots \\
z_k &\equiv_2 \textit{somme des produits des } x_i \textit{ pris } 2^k \textit{ à la fois.} \\
&\vdots
\end{aligned}$$

Dans les articles de G. Labelle (Labelle, 1978) et de P. Camion (Camion, 1960) sont présentés les détails de ce Théorème.

Dorénavant, supposons que les fonctions $z = f(x)$ sont telles que pour $i = 1, \ldots, i_{max}$

$$z_i \equiv_p \text{est fonction de } (x_0, x_1, \ldots, x_{n_{max}}), \tag{3.1.3}$$

c'est-à-dire, z_i admet un nombre fini de pits en entrée.

3.2 Programmes de décompositions

Dans cette section, nous décrivons des algorithmes de décompositions de fonctions en pits en Maple. J'aimerais souligner que ces algorithmes sont de ma propre contribution. Ces algorithmes sont importants car ils permettent d'ouvrir la voie à des applications.

3.2.1 Décomposition 1

Par le Théorème de décomposition 1, si p est un nombre premier et $z = f(x)$, alors

$$z_i \equiv_p \sum_{m\geq 0} \alpha_m(i)\binom{x_0}{m_0}\binom{x_1}{m_1}\binom{x_2}{m_2}\cdots \tag{3.2.1}$$

où $\alpha_m(i)$ satisfait

$$\alpha_m(i) \equiv_p \sum_{\mu=0}^{m}(-1)^{m-\mu}\binom{m}{\mu}f(\mu)_i. \tag{3.2.2}$$

Déclarons une fonction qui permet de retourner le $i^{\text{ème}}$chiffre d'un nombre.

Programme 3.2.1 (PIT) *La fonction Pit retourne le $i^{\text{ème}}$pit du nombre x en base p. Le nombre x est écrit en base décimale. Une valeur doit être préalablement attribuée à p. Le nombre i est une valeur positive ou égale à 0.*

Si $i = 0$, le programme retourne le premier chiffre (l'unité) de x en base p.

Si $i = 1$, le programme retourne le deuxième chiffre de x en base p.

⋮

Antécédent : le nombre i est un entier non-négatif.

```
> pit := proc(x,i)
>   local L;
>   if (x = 0) then
>      RETURN(0);
>   end if;
>   L := convert(x,base,p);
>   if (i > nops(L)-1) then
>      RETURN(0);
>   else
>      RETURN(L[i+1]);
>   end if;
> end proc:
```

Exemple 3.2.1 *Énumérons les pits de $39_{(10)}$ en base 5.*

```
> p := 5:
> for i from 0 to 3 do
>    pit(39,i);
> end do;
                                4
                                2
                                1
                                0
```

Programme 3.2.2 (BIN_P) *La fonction Bin_p permet de trouver le coefficient binomial de* $\binom{x}{m}$ *en utilisant le Théorème de Lucas. Le symbole x est une variable*[1] *et m est un entier non-négatif. Par Lucas, nous savons que*

$$\binom{x}{m} \equiv_p \binom{x_0}{m_0}\binom{x_1}{m_1}\cdots\binom{x_n}{m_n}. \tag{3.2.3}$$

Mais, il est à noter que $\binom{x_j}{0} = 1$. *Maple ne permet pas d'afficher un coefficient binomial de la forme* $\binom{a}{b}$. *La notation suivante en Maple est alors utilisée pour décrire le coefficient du binôme*

$$a^{\underline{b}} = \binom{a}{b}. \tag{3.2.4}$$

Une valeur doit être préalablement attribuée à p.

```
Bin_p := proc(x,m)
>   local i, aux, L;
>   if (m = 0) then
>      RETURN(1);
>   end if;
>   aux := 1;
>   L   := convert(m,base,p);
>   for i from 1 to nops(L) do
>      if (L[i] > 0) then
>         aux := aux*(x[i-1]^{L[i]});
>      end if;
>   end do;
```

[1] *x* n'est pas un nombre.

```
>   RETURN(aux);
> end proc:
```

Exemple 3.2.2 *Si $p = 2$, que vaut $Bin\_p(x, 100)$?*

$$100_{(10)} = 64_{(10)} + 32_{(10)} + 4_{(10)} = 1100100_{(2)} \tag{3.2.5}$$

```
> p := 2:
> Bin_p(x,100);
                          {1}       {1}       {1}
                      x[2]      x[5]      x[6]
```

Notation 13 *Il est à noter que x[i] veut dire x_i en Maple7.*

En d'autres mots,

$$\binom{x}{100_{(10)}} \equiv_2 \binom{x_2}{1}\binom{x_5}{1}\binom{x_6}{1}. \tag{3.2.6}$$

Il y a des simplifications qui ont été faites, car :

$$\binom{x_0}{0} = 1, \quad \binom{x_1}{0} = 1, \quad \binom{x_3}{0} = 1, \quad \binom{x_4}{0} = 1, \quad \binom{x_7}{0} = 1, \quad \binom{x_8}{0} = 1, \quad \dots \tag{3.2.7}$$

De façon pratique, dans le Théorème de décomposition 1, l'indice de sommation m peut être borné par un nombre «très élevé». Appelons ce nombre m_{max}.

$$z_i \equiv_p \sum_{m=0}^{m_{max}} \alpha_m(i)\binom{x_0}{m_0}\binom{x_1}{m_1}\binom{x_2}{m_2}\cdots \tag{3.2.8}$$

où $\alpha_m(i)$ satisfait

$$\alpha_m(i) \equiv_p \sum_{\mu=0}^{m} (-1)^{m-\mu}\binom{m}{\mu} f(\mu)_i. \tag{3.2.9}$$

Il est à noter que lorsque $m = 0$, alors

$$\alpha_m(i)\binom{x_0}{m_0}\binom{x_1}{m_1}\binom{x_2}{m_2}\cdots \equiv_p f(0)_i \tag{3.2.10}$$

Programme 3.2.3 (DÉCOMPOSITION1) *La fonction «décomposition1» retourne le $i^{\text{ème}}$pit d'une fonction (à une seule variable x) passée en entrée en utilisant le Théorème de décomposition1. Le $i^{\text{ème}}$pit est fonction des $(m_{max}+1)$ premiers chiffres de x*

$$z_i = f(x_0, x_1, \ldots, x_{m_{max}}) \tag{3.2.11}$$

Une valeur doit être préalablement attribuée à m_{max} et p.

```
> decomposition1 := proc(f, i)
>   local j, m, n, mu, aux;
>   aux := pit(f(0),i);
>   for m from 1 to mmax do
>      aux := aux +
>             modp(
>                    add(pit(f(mu),i)*(-1)^(m-mu)*
>                        modp(binomial(m,mu),p),
>                    mu = 0..mmax),p
>                 )*Bin_p(x,m);
>   end do;
>   RETURN(sort(aux));
> end proc:
```

Dans les exemples portant sur la décomposition1, l'indice de la sommation m est borné par le nombre 64 ($m_{max} = 64$).

Exemple 3.2.3 *Si $p = 2$ et $f(x) = x + 1$, la décomposition1 de la fonction f pour les premiers pits de sortie de f donne*

```
> p := 2:
> f := x -> x+1:
> for i from 0 to 4 do
>   z[i] = decomposition1(f,i);
> end do;
                                          {1}
                          z[0] = 1 + x[0]
```

```
                          {1}       {1}
        z[1] = x[0]     + x[1]

                    {1}       {1}       {1}
      z[2] = x[0]     x[1]    + x[2]

                {1}       {1}       {1}       {1}
    z[3] = x[0]     x[1]    x[2]    + x[3]

              {1}       {1}       {1}       {1}       {1}
  z[4] = x[0]     x[1]    x[2]    x[3]    + x[4]
```

Autrement dit, si $z = f(x) = x + 1$, alors

$$z_0 \equiv_2 1 + \binom{x_0}{1}, \tag{3.2.12}$$

$$z_1 \equiv_2 \binom{x_0}{1} + \binom{x_1}{1}, \tag{3.2.13}$$

$$z_2 \equiv_2 \binom{x_0}{1}\binom{x_1}{1} + \binom{x_2}{1}, \tag{3.2.14}$$

$$z_3 \equiv_2 \binom{x_0}{1}\binom{x_1}{1}\binom{x_2}{1} + \binom{x_3}{1}, \tag{3.2.15}$$

$$z_4 \equiv_2 \binom{x_0}{1}\binom{x_1}{1}\binom{x_2}{1}\binom{x_3}{1} + \binom{x_4}{1}. \tag{3.2.16}$$

Mais, ne pas oublier que

$$\binom{x_i}{1} \equiv_2 x_i. \tag{3.2.17}$$

Exemple 3.2.4 *Si $p = 2$ et $f(x) = 3x$, la décomposition1 de la fonction f pour les premiers pits de sortie de f donne*

```
> p := 2:
> f := x -> 3*x;
> for i from 0 to 4 do
```

```
>    z[i] = decomposition1(f,i);
> end do;
                              f := x -> 3 x
                                        {1}
                              z[0] = x[0]

                                     {1}       {1}
                          z[1] = x[0]    + x[1]

                               {1}       {1}     {1}       {1}
                    z[2] = x[1]    + x[0]    x[1]    + x[2]

                     {1}     {1}       {1}       {1}     {1}
        z[3] = x[0]    x[1]    + x[2]    + x[1]    x[2]
                       {1}     {1}     {1}       {1}
               + x[0]    x[1]    x[2]    + x[3]

                     {1}     {1}       {1}       {1}     {1}     {1}
        z[4] = x[1]    x[2]    + x[3]    + x[0]    x[1]    x[3]
                       {1}     {1}       {1}     {1}     {1}
               + x[2]    x[3]    + x[1]    x[2]    x[3]
                       {1}     {1}     {1}     {1}       {1}
               + x[0]    x[1]    x[2]    x[3]    + x[4]
```

Décrit d'une autre manière, si $z = f(x) = 3x$, alors

$$z_0 \equiv_2 \begin{pmatrix} x_0 \\ 1 \end{pmatrix}, \tag{3.2.18}$$

$$z_1 \equiv_2 \begin{pmatrix} x_0 \\ 1 \end{pmatrix} + \begin{pmatrix} x_1 \\ 1 \end{pmatrix}, \tag{3.2.19}$$

$$z_2 \equiv_2 \begin{pmatrix} x_0 \\ 1 \end{pmatrix} + \begin{pmatrix} x_2 \\ 1 \end{pmatrix} + \begin{pmatrix} x_0 \\ 1 \end{pmatrix}\begin{pmatrix} x_1 \\ 1 \end{pmatrix}, \tag{3.2.20}$$

$$z_3 \equiv_2 \begin{pmatrix} x_0 \\ 1 \end{pmatrix}\begin{pmatrix} x_1 \\ 1 \end{pmatrix} + \begin{pmatrix} x_2 \\ 1 \end{pmatrix} + \begin{pmatrix} x_1 \\ 1 \end{pmatrix}\begin{pmatrix} x_2 \\ 1 \end{pmatrix} + \begin{pmatrix} x_0 \\ 1 \end{pmatrix}\begin{pmatrix} x_1 \\ 1 \end{pmatrix}\begin{pmatrix} x_2 \\ 1 \end{pmatrix} + \begin{pmatrix} x_3 \\ 1 \end{pmatrix}, \tag{3.2.21}$$

$$z_4 \equiv_2 \begin{pmatrix} x_1 \\ 1 \end{pmatrix}\begin{pmatrix} x_2 \\ 1 \end{pmatrix} + \begin{pmatrix} x_3 \\ 1 \end{pmatrix} + \begin{pmatrix} x_0 \\ 1 \end{pmatrix}\begin{pmatrix} x_1 \\ 1 \end{pmatrix}\begin{pmatrix} x_3 \\ 1 \end{pmatrix} + \begin{pmatrix} x_2 \\ 1 \end{pmatrix}\begin{pmatrix} x_3 \\ 1 \end{pmatrix} + \begin{pmatrix} x_1 \\ 1 \end{pmatrix}\begin{pmatrix} x_2 \\ 1 \end{pmatrix}\begin{pmatrix} x_3 \\ 1 \end{pmatrix} + \begin{pmatrix} x_0 \\ 1 \end{pmatrix}\begin{pmatrix} x_1 \\ 1 \end{pmatrix}\begin{pmatrix} x_2 \\ 1 \end{pmatrix}\begin{pmatrix} x_3 \\ 1 \end{pmatrix} + \begin{pmatrix} x_4 \\ 1 \end{pmatrix}. \tag{3.2.22}$$

Exemple 3.2.5 *Si $p = 2$ et $f(x) = x^2$, la décomposition*1 *de la fonction f pour les premiers pits de sortie de f donne*

```
> p := 2:
> f := x -> x^2:
> for i from 0 to 5 do
>   z[i] = decomposition1(f,i);
> end do;
```

```
                                        {1}
                              z[0] = x[0]

                                 z[1] = 0

                                {1}       {1}     {1}
                     z[2] = x[1]    + x[0]    x[1]

                            {1}     {1}       {1}     {1}
                z[3] = x[0]    x[1]    + x[0]    x[2]

              {1}       {1}     {1}       {1}     {1}     {1}
  z[4] = x[2]    + x[1]    x[2]    + x[0]    x[1]    x[2]
                {1}     {1}
         + x[0]    x[3]

              {1}     {1}       {1}     {1}       {1}     {1}     {1}
  z[5] = x[1]    x[2]    + x[1]    x[3]    + x[0]    x[2]    x[3]
                {1}     {1}
         + x[0]    x[4]
```

Autrement dit, si $z = f(x) = x^2$, alors :

$$z_0 \equiv_2 \begin{pmatrix} x_0 \\ 1 \end{pmatrix}, \tag{3.2.23}$$

$$z_1 \equiv_2 0, \tag{3.2.24}$$

$$z_2 \equiv_2 \begin{pmatrix} x_1 \\ 1 \end{pmatrix} + \begin{pmatrix} x_0 \\ 1 \end{pmatrix}\begin{pmatrix} x_1 \\ 1 \end{pmatrix}, \tag{3.2.25}$$

$$z_3 \equiv_2 \binom{x_0}{1}\binom{x_1}{1} + \binom{x_0}{1}\binom{x_2}{1}, \tag{3.2.26}$$

$$z_4 \equiv_2 \binom{x_2}{1} + \binom{x_1}{1}\binom{x_2}{1} + \binom{x_0}{1}\binom{x_1}{1}\binom{x_2}{1} + \binom{x_0}{1}\binom{x_3}{1}. \tag{3.2.27}$$

Exemple 3.2.6 *Si $p = 3$ et $f(x) = x + 1$, la décomposition*1 *de la fonction f pour les premiers pits de sortie de f donne*

```
> p := 3:
> f := x -> x+1:
> for i from 0 to 3 do
>   z[i] = decomposition1(f,i);
> end do;
```

$$z[0] = x[0]^{\{1\}} + 1$$

$$z[1] = x[1]^{\{1\}} + x[0]^{\{2\}}$$

$$z[2] = x[0]^{\{2\}}\, x[1]^{\{2\}} + x[2]^{\{1\}}$$

$$z[3] = x[2]^{\{2\}}\, x[0]^{\{2\}}\, x[1]^{\{2\}} + x[3]^{\{1\}}$$

D'où,

$$z_0 \equiv_3 1 + \binom{x_0}{1}, \tag{3.2.28}$$

$$z_1 \equiv_3 \binom{x_0}{2} + \binom{x_1}{1}, \tag{3.2.29}$$

$$z_2 \equiv_3 \binom{x_0}{2}\binom{x_1}{2} + \binom{x_2}{1}, \tag{3.2.30}$$

$$z_3 \equiv_3 \binom{x_0}{2}\binom{x_1}{2}\binom{x_2}{2} + \binom{x_3}{1}. \tag{3.2.31}$$

Exemple 3.2.7 *Si $p = 5$ et $f(x) = x + 1$, la décomposition*1 *de la fonction f pour les premiers pits de sortie de f donne*

```
> p := 5:
> f := x -> x+1:
> for i from 0 to 3 do
>   z[i] = decomposition1(f,i);
> end do;
```

$$z[0] = x[0]^{\{1\}} + 1$$

$$z[1] = x[0]^{\{4\}} + x[1]^{\{1\}}$$

$$z[2] = x[1]^{\{4\}} x[0]^{\{4\}} + x[2]^{\{1\}}$$

$$z[3] = x[1]^{\{4\}} x[0]^{\{4\}} x[2]^{\{4\}} + x[3]^{\{1\}}$$

D'où,

$$z_0 \equiv_5 1 + \binom{x_0}{1}, \tag{3.2.32}$$

$$z_1 \equiv_5 \binom{x_0}{4} + \binom{x_1}{1}, \tag{3.2.33}$$

$$z_2 \equiv_5 \binom{x_0}{4}\binom{x_1}{4} + \binom{x_2}{1}, \tag{3.2.34}$$

$$z_3 \equiv_5 \binom{x_0}{4}\binom{x_1}{4}\binom{x_2}{4} + \binom{x_3}{1}. \tag{3.2.35}$$

La formule ci-présente est prouvée plus tard dans ce chapitre.

$$z_i \equiv_p \binom{x_0}{p-1}\binom{x_1}{p-1}\cdots\binom{x_{i-1}}{p-1} + \binom{x_i}{1}. \tag{3.2.36}$$

3.2.2 Décomposition 2

Le Théorème de décomposition 2 dit que,

$$z_i \equiv_p \sum_{m\geq 0} f(m)_i \prod_{j\geq 0} \delta_{m_j}^{x_j}. \tag{3.2.37}$$

Décomposition 2 a

En premier lieu, un Corollaire sous-jacent à la décomposition 2 affirme que

$$z_i \equiv_p \sum_{m\geq 0} f(m)_i \prod_{j\geq 0} \frac{x_j - x_j^p}{x_j - m_j}. \tag{3.2.38}$$

De façon pratique, pour rendre cette dernière formule calculable par un nombre fini d'étapes, les indices doivent être bornés par m_{max}

$$z_i \equiv_p \sum_{m=0}^{m_{max}} f(m)_i \prod_{j=0}^{|m|-1} \frac{x_j - x_j^p}{x_j - m_j}. \tag{3.2.39}$$

où $|m|$ représente le nombre de pits de m.

Programme 3.2.4 (DÉCOMPOSITION2A) *La fonction «décomposition2a» retourne le $i^{ème}$pit d'une fonction (à une seule variable x) passée en entrée en utilisant le Théorème de décomposition2a. Le $i^{ème}$pit est fonction des $(m_{max} + 1)$ premiers chiffres de x*

$$z_i = f(x_0, x_1, \ldots, x_{m_{max}}) \tag{3.2.40}$$

Une valeur doit être préalablement attribuée à m_{max} et p.
Posons m_{max}, une puissance de p.

```
> decomposition2a := proc(F,i)
>    add(
>        pit(F(m),i) *
>        mul((x[j]-x[j]^p)/(x[j]-pit(m,j)),j=0..NbPits(mmax)-1),
>        m=0..mmax-1);
>    simplify(%);
>    expand(%);
>    modp(%,p);
>    return sort(%);
> end proc:
```

Exemple 3.2.8 *Si $p = 2$ et $z = f(x) = x + 1$, alors le programme decomposition2a donne*

```
> p := 2:
> F := x -> x + 1:
> for i from 0 to 4 do
>   print(z[i] = decomposition2a(F,i));
> end do;
```

$$z[0] = x[0] + 1$$

$$z[1] = x[0] + x[1]$$

$$z[2] = x[0]\ x[1] + x[2]$$

$$z[3] = x[0]\ x[1]\ x[2] + x[3]$$

$$z[4] = x[0]\ x[1]\ x[2]\ x[3] + x[4]$$

Exemple 3.2.9 *Si $p = 2$ et $z = f(x) = 2 * x$, alors le programme decomposition2a donne*

```
> F := x -> 2*x:
> for i from 0 to 5 do
>   print(z[i] = decomposition2a(F,i));
> end do;
```

$$z[0] = 0$$

$$z[1] = x[0]$$

$$z[2] = x[1]$$

$$z[3] = x[2]$$

$$z[4] = x[3]$$

$$z[5] = x[4]$$

Décomposition 2 b

En second lieu, un Corollaire relatif à la décomposition 2 affirme que

$$z_i \equiv_p \sum_{m\geq 0} f(m)_i \prod_{j\geq 0} 1-(x_j-m_j)^{p-1}. \tag{3.2.41}$$

De façon pratique, pour rendre cette dernière formule finitaire, les indices doivent aussi être bornés comme démontré ci-dessous

$$z_i \equiv_p \sum_{m=0}^{m_{max}} f(m)_i \prod_{j=0}^{|m|-1} 1-(x_j-m_j)^{p-1}. \tag{3.2.42}$$

où $|m|$ représente le nombre de pits de m.

Programme 3.2.5 (DÉCOMPOSITION2B) *La fonction «décomposition2b» retourne le $i^{ème}$pit d'une fonction (à une seule variable x) passée en entrée en utilisant le Théorème de décomposition2b. Le $i^{ème}$pit est fonction des $(m_{max}+1)$ premiers chiffres de x*

$$z_i = f(x_0, x_1, \ldots, x_{m_{max}}) \tag{3.2.43}$$

Une valeur doit être préalablement attribuée à m_{max} et p. Posons m_{max}, une puissance de p.

```
decomposition2b := proc(F,i)
>    add(
>        pit(F(m),i) *
>        mul(1-(x[j]-pit(m,j))^(p-1),j=0..NbPits(mmax)-1),
>        m=0..mmax-1);
>    simplify(%);
>    expand(%);
>    modp(%,p);
>    return sort(%);
> end proc:
```

Exemple 3.2.10 *Si $p = 2$ et $z = f(x) = x + 1$, alors le programme decomposition2b donne*

```
> p := 2:
> F := x -> x + 1:
> for i from 0 to 4 do
>   print(z[i] = decomposition2b(F,i));
> end do;
```

$$z[0] = x[0] + 1$$

$$z[1] = x[0] + x[1]$$

$$z[2] = x[0]\ x[1] + x[2]$$

$$z[3] = x[0]\ x[1]\ x[2] + x[3]$$

$$z[4] = x[0]\ x[1]\ x[2]\ x[3] + x[4]$$

Exemple 3.2.11 *Si $p = 2$ et $z = f(x) = 2x$, alors le programme decomposition2b donne*

```
> p := 2:
> F := x -> 2*x:
> for i from 0 to 5 do
>   print(z[i] = decomposition2b(F,i));
> end do;
```

$$z[0] = 0$$

$$z[1] = x[0]$$

$$z[2] = x[1]$$

$$z[3] = x[2]$$

$$z[4] = x[3]$$

$$z[5] = x[4]$$

3.2.3 Décomposition 3

Le Théorème de décomposition 3 dit que si p est un nombre premier et $z = f(x)$, alors

$$z_i \equiv_p \sum_{m\geq 0} \beta_m(i) \prod_{j\geq 0} x_j^{m_j}, \qquad \beta_m(i) \equiv_p \sum_{\mu=0}^{m} \left\{ {m \atop \mu} \right\} f(\mu)_i. \tag{3.2.44}$$

Définissons quelques petites fonctions accommodantes. Noter que $\left\{ {m \atop \mu} \right\}$ désigne :

$$\left\{ {m_0 \atop \mu_0} \right\} \left\{ {m_1 \atop \mu_1} \right\} \left\{ {m_2 \atop \mu_2} \right\} \ldots \tag{3.2.45}$$

La fonction B permet de calculer ce produit des éléments de la matrice B.

Programme 3.2.6 (B) *La fonction B permet de calculer* $\left\{ {m \atop \mu} \right\}$*, qui correspond au produit des éléments de la matrice B.*

$$\left\{ {m_0 \atop \mu_0} \right\} \left\{ {m_1 \atop \mu_1} \right\} \left\{ {m_2 \atop \mu_2} \right\} \ldots \tag{3.2.46}$$

Antécédents : le nombre premier p doit être initialisé et les symboles, m et μ, doivent être des entiers non-négatifs.

```
> B := proc(m,mu)
>   local k, Lm, Lmu;
>   if (m = 0 and mu = 0) then
>     RETURN(1)
>   end if;
>   if (m = 0 and mu > 0) then
>     RETURN(0)
>   end if;
>   if (m > 0 and mu = 0) then
>     Lm := convert(m,base,p);
>     RETURN(modp(mul(Bpits[Lm[i],0],i=1..nops(Lm)),p))
>   end if;
>   if (m > 0 and mu > 0) then
>     Lm := convert(m,base,p);
```

```
>     Lmu := convert(mu,base,p);
>     if (nops(Lm) < nops(Lmu)) then
>       RETURN(0)
>     else
>       RETURN( modp( mul( Bpits[Lm[i],Lmu[i]], i=1..nops(Lmu) )
>                     *mul( Bpits[Lm[j],0], j=nops(Lmu)+1..nops(Lm)),p));
>     end if;
>   end if;
> end proc:
```

où la fonction[2] *Bpits* retourne le m-ème coefficient du polynôme de $\delta_{m_j=\mu}^{x_j}$.

```
for m from 0 to p-1 do
>   for mu from 0 to p-1 do
>     Bpits[m, mu] :=
>         coeff(modp(expand(modp(1-(x-mu)^(p-1),p)),p), x, m);
>   end do;
> end do;
```

Dans l'équation du Théorème de décomposition 3

$$z_i \equiv_p \sum_{m\geq 0} \beta_m(i) \prod_{j\geq 0} x_j^{m_j}, \tag{3.2.47}$$

l'indice du produit j, parcourt les chiffres de m et x. Lorsque j est plus grand ou égal au nombre de chiffres de m, alors

$$m_j = 0, \tag{3.2.48}$$

et

$$x_j^0 = 1. \tag{3.2.49}$$

L'indice j peut être borné par le nombre de chiffres de m moins un.

---

[2] matrice.

De façon pratique, dans le Théorème de décomposition 3, bornons l'indice de sommation m par un nombre «très élevé», m_{max}.

$$z_i \equiv_p \sum_{m=0}^{m_{max}} \beta_m(i) \prod_{j=0}^{|m|-1} x_j^{m_j}, \qquad \beta_m(i) \equiv_p \sum_{\mu=0}^{m} \left\{ \begin{matrix} m \\ \mu \end{matrix} \right\} f(\mu)_i. \tag{3.2.50}$$

où $|m|$ désigne le nombre de chiffres de m.

Programme 3.2.7 (DÉCOMPOSITION3) *La fonction «décomposition3» retourne le $i^{\text{ème}}$pit d'une fonction (à une seule variable x) passée en entrée en utilisant le Théorème de décomposition3. Le $i^{\text{ème}}$pit est fonction des $(m_{max}+1)$ premiers chiffres de x*

$$z_i = f(x_0, x_1, \ldots, x_{m_{max}}) \tag{3.2.51}$$

Une valeur doit être préalablement attribuée à m_{max} et p.

```
decomposition3 := proc(f, i)
>   local j, m, n, mu, aux, Lm;
>   aux := pit(f(0),i);
>   for m from 1 to mmax do
>     Lm := convert(m,base,p);
>     aux := aux + modp(add( pit(f(mu),i)*B(m,mu),
>                       mu=0..p^nops(Lm)-1),p)
>                       *mul(x[j]^pit(m,j), j=0..nops(Lm)-1)
>   end do;
>   RETURN(aux);
> end proc:
```

Exemple 3.2.12 *Si $p = 2$ et $z = f(x) = x + 1$, alors le programme decomposition3 donne*

```
> f := x -> x+1:
> for i from 0 to 3 do
>    print(z[i] = decomposition3(f,i));
> end do;
                              z[0] = 1 + x[0]
```

```
z[1] = x[0] + x[1]

z[2] = x[0] x[1] + x[2]

z[3] = x[0] x[1] x[2] + x[3]
```

Exemple 3.2.13 *Si $p = 2$ et $z = f(x) = 3x$, alors le programme decomposition3 donne*

```
> f := x -> 3*x:
> for i from 0 to 5 do
>    print(z[i] = decomposition3(f,i));
> end do;
                          z[0] = x[0]

                       z[1] = x[0] + x[1]

                   z[2] = x[1] + x[0] x[1] + x[2]

    z[3] = x[0] x[1] + x[2] + x[1] x[2] + x[0] x[1] x[2] + x[3]

  z[4] = x[1] x[2] + x[3] + x[0] x[1] x[3] + x[2] x[3]

         + x[1] x[2] x[3] + x[0] x[1] x[2] x[3] + x[4]

  z[5] = x[0] x[1] x[3] + x[2] x[3] + x[0] x[1] x[2] x[3] + x[4]

         + x[1] x[2] x[4] + x[3] x[4] + x[0] x[1] x[3] x[4]

         + x[2] x[3] x[4] + x[1] x[2] x[3] x[4]

         + x[0] x[1] x[2] x[3] x[4] + x[5]
```

3.3 Généralisations

Diverses généralisations de formules suggérées par Maple sont prouvées dans cette section.

3.3.1 Incrémentation

Il a été mentionné que pour $z = f(x) = x + 1$

$$z_0 \equiv_2 1 + \binom{x_0}{1}, \tag{3.3.1}$$

$$z_1 \equiv_2 \binom{x_0}{1} + \binom{x_1}{1}, \tag{3.3.2}$$

$$z_2 \equiv_2 \binom{x_0}{1}\binom{x_1}{1} + \binom{x_2}{1}, \tag{3.3.3}$$

$$z_3 \equiv_2 \binom{x_0}{1}\binom{x_1}{1}\binom{x_2}{1} + \binom{x_3}{1}, \tag{3.3.4}$$

$$z_4 \equiv_2 \binom{x_0}{1}\binom{x_1}{1}\binom{x_2}{1}\binom{x_3}{1} + \binom{x_4}{1}. \tag{3.3.5}$$

Mais, il y a moyen de généraliser cette formule pour un z_i et un nombre premier p quelconque.

Théorème 3.3.1 *Si p est un nombre premier et $z = f(x) = x + 1$ alors,*

$$z_i \equiv_p x_i + \binom{x_0}{p-1}\binom{x_1}{p-1}\cdots\binom{x_{i-1}}{p-1} \tag{3.3.6}$$

$$\equiv_p x_i + (-1)^i x_0^{(p-1)} x_1^{(p-1)} \cdots x_{i-1}^{(p-1)}. \tag{3.3.7}$$

Démonstration du Théorème :

Par les propriétés du triangle de Pascal,

$$\binom{x+1}{k} = \binom{x}{k} + \binom{x}{k-1}. \tag{3.3.8}$$

En posant que $k = p^i$ dans cette dernière équation,

$$\binom{x+1}{p^i} = \binom{x}{p^i} + \binom{x}{p^i-1}. \tag{3.3.9}$$

Or, il y a un Lemme au chapitre 2 qui affirme que

$$a_i \equiv_p \binom{a}{p^i}. \tag{3.3.10}$$

D'où,

$$z_i = x_i + \binom{x}{p^i - 1}. \tag{3.3.11}$$

Par le Théorème de Lucas,

$$\binom{x}{p^i - 1} \equiv_p \binom{x_0}{p-1}\binom{x_1}{p-1}\cdots\binom{x_{i-1}}{p-1}. \tag{3.3.12}$$

Donc,

$$z_i \equiv_p x_i + \binom{x_0}{p-1}\binom{x_1}{p-1}\cdots\binom{x_{i-1}}{p-1}. \tag{3.3.13}$$

Par le Théorème de Wilson,

$$z_i \equiv_p x_i + (-1)^i x_0^{(p-1)} x_1^{(p-1)} \cdots x_{i-1}^{(p-1)}. \tag{3.3.14}$$

■

3.3.2 Addition

La décomposition en pits de la fonction $z = x + y$ fait appel à Vandermonde.

Théorème 3.3.2 (VANDERMONDE, *(Abramowitz et Stegun, 1974)*) *Si* $0 \leq k \leq m + n$ *alors,*

$$\binom{m+n}{k} = \sum_{i+j=k} \binom{m}{i}\binom{n}{j} \tag{3.3.15}$$

Démonstration combinatoire du Théorème :
Soit A et B, deux ensembles disjoints tels que $|A| = m$ et $|B| = n$. Le nombre de façons de choisir k éléments dans l'ensemble $A \cup B$ est égal à la sommation sur les i, j tels que $i + j = k$ du nombre de façons de choisir i éléments dans A et j éléments dans B . ■

Corollaire 3.3.3 *Si p est un nombre premier et $z = f(x, y) = x + y$, alors*

$$z_i \equiv_p \sum_{a+b=p^i} \binom{x}{a}\binom{y}{b} \tag{3.3.16}$$

Par Lucas,

$$z_i \equiv_p \sum_{a+b=p^i} \binom{x_0}{a_0}\binom{x_1}{a_1}\binom{x_2}{a_2}\cdots\binom{y_0}{b_0}\binom{y_1}{b_1}\binom{y_2}{b_2}\cdots \tag{3.3.17}$$

Théorème 3.3.4 *Si p est un nombre premier et $z = f(x, y) = x + y$, alors*

$$z_i \equiv_p (x_i + y_i) + r_i, \tag{3.3.18}$$

où, r_i est définie par la récurrence

$$r_{i+1} \equiv_p \binom{x_i + y_i}{p} - (x_i + y_i)^{(p-1)} r_i, \qquad r_0 \equiv_p 0. \tag{3.3.19}$$

Démonstration du Théorème :

Par le Corollaire 3.3.3,

$$z_i \equiv_p \sum_{a+b=p^i} \binom{x}{a}\binom{y}{b}. \tag{3.3.20}$$

Si les éléments $(a = 0, b = p^i)$ et $(a = p^i, b = 0)$ de la sommation sont retirés,

$$z_i \equiv_p x_i + y_i + \sum_{a+b=p^i,\ 0<a<p^i,\ 0<b<p^i} \binom{x}{a}\binom{y}{b}. \tag{3.3.21}$$

Autrement dit,

$$z_i = (x_i + y_i) + r_i, \qquad \text{avec} \quad r_i = \sum_{a+b=p^i,\ 0<a<p^i,\ 0<b<p^i} \binom{x}{a}\binom{y}{b}. \tag{3.3.22}$$

En posant les ensembles

$$E = \{(a, b) \ : \ a + b = p^i,\ 0 < a < p^i,\ 0 < b < p^i\}, \tag{3.3.23}$$

la variable r_i est écrite de manière abrégée avec

$$r_i = \sum_{(a,b)\in E} \binom{x}{a}\binom{y}{b}. \tag{3.3.24}$$

Posons l'équation

$$E_k = \{(a,b) \in E : 0 = a_0 = a_1 \cdots = a_{k-1}, 0 = b_0 = b_1 \cdots = b_{k-1}, a_k \neq 0 \neq b_k\}. \quad (3.3.25)$$

Le couple $(a,b) \in E_k$ si et seulement si (a,b) satisfait les conditions suivantes :

$$a = a_{i-1} \ldots a_{k+1} a_k \overbrace{00\ldots0}^{k}, \quad (3.3.26)$$

$$b = b_{i-1} \ldots b_{k+1} b_k \overbrace{00\ldots0}^{k}, \quad (3.3.27)$$

avec $a_k \neq 0 \neq b_k$ et $a + b = p^i$.

Mais,

$$E = E_0 \cup E_1 \cup \cdots \cup E_{i-1}. \quad (3.3.28)$$

D'où,

$$r_i = \sum_{k=0}^{i-1} \sum_{(a,b)\in E_k} \binom{x}{a}\binom{y}{b}. \quad (3.3.29)$$

Cependant,

$$\begin{aligned}
&\textstyle\sum_{(a,b)\in E_k} \binom{x}{a}\binom{y}{b} \\
\equiv_p \quad & \sum_{(a,b)\in E_k} \binom{x_0}{0}\binom{x_1}{0}\cdots\binom{x_{k-1}}{0}\binom{x_k}{a_k}\cdots\binom{x_{i-1}}{a_{i-1}}\binom{y_0}{0}\binom{y_1}{0}\cdots\binom{y_{k-1}}{0}\binom{y_k}{b_k}\cdots\binom{y_{i-1}}{b_{i-1}} \\
\equiv_p \quad & \sum_{(a,b)\in E_k} \binom{x_k}{a_k}\binom{x_{k+1}}{a_{k+1}}\cdots\binom{x_{i-1}}{a_{i-1}}\binom{y_k}{b_k}\binom{y_{k+1}}{b_{k+1}}\cdots\binom{y_{i-1}}{b_{i-1}} \\
\equiv_p \quad & \left(\sum_{a_k+b_k=p, a_k\neq 0, b_k\neq 0} \binom{x_k}{a_k}\binom{y_k}{b_k}\right)\left(\sum_{a_{k+1}+b_{k+1}=p-1} \binom{x_{k+1}}{a_{k+1}}\binom{y_{k+1}}{b_{k+1}}\right)\cdots\left(\sum_{a_{i-1}+b_{i-1}=p-1} \binom{x_{i-1}}{a_{i-1}}\binom{y_{i-1}}{b_{i-1}}\right) \\
\equiv_p \quad & \left(\binom{x_k+y_k}{p} - \binom{x_{k+1}}{0}\binom{y_{k+1}}{p^i} - \binom{x_{k+1}}{p^i}\binom{y_{k+1}}{0}\right)\binom{x_{k+1}+y_{k+1}}{p-1}\cdots\binom{x_{i-1}+y_{i-1}}{p-1} \\
\equiv_p \quad & \binom{x_k+y_k}{p}\binom{x_{k+1}+y_{k+1}}{p-1}\cdots\binom{x_{i-1}+y_{i-1}}{p-1}.
\end{aligned}$$

Donc,

$$r_i \equiv_p \sum_{k=0}^{i-1} \binom{x_k+y_k}{p}\binom{x_{k+1}+y_{k+1}}{p-1}\cdots\binom{x_{i-1}+y_{i-1}}{p-1}. \quad (3.3.30)$$

En développant cette dernière équation, il en résulte que

$$\begin{aligned} r_i \equiv_p & \binom{x_0+y_0}{p}\binom{x_1+y_1}{p-1}\cdots\binom{x_{i-1}+y_{i-1}}{p-1} \\ & + \binom{x_1+y_1}{p}\binom{x_2+y_2}{p-1}\cdots\binom{x_{i-1}+y_{i-1}}{p-1} \\ & + \binom{x_2+y_2}{p}\binom{x_3+y_3}{p-1}\cdots\binom{x_{i-1}+y_{i-1}}{p-1} \\ & \vdots \\ & + \binom{x_{i-1}+y_{i-1}}{p-1}. \end{aligned}$$

Notons qu'avec le Théorème de Wilson, il est clair que

$$\binom{x_j+y_j}{p-1} \equiv_p \frac{(x_j+y_j)^{(p-1)}}{(p-1)!} \equiv_p -(x_j+y_j)^{(p-1)}. \tag{3.3.31}$$

En examinant les r_i pour $i = 0$, 1 et 2,

$$r_0 \equiv_p 0, \tag{3.3.32}$$

$$r_1 \equiv_p \binom{x_0+y_0}{p}, \tag{3.3.33}$$

$$r_2 \equiv_p \binom{x_0+y_0}{p}\binom{x_1+y_1}{p-1} + \binom{x_1+y_1}{p} \tag{3.3.34}$$

$$\equiv_p \binom{x_1+y_1}{p-1} r_1 + \binom{x_1+y_1}{p}. \tag{3.3.35}$$

De manière générale, la formule suivante peut être démontrée par récurrence

$$r_{i+1} \equiv_p \binom{x_i+y_i}{p-1} r_i + \binom{x_i+y_i}{p}, \qquad r_0 \equiv_p 0. \tag{3.3.36}$$

Donc,

$$z_i \equiv_p (x_i+y_i) + r_i, \tag{3.3.37}$$

où r_i est définie par la récurrence

$$r_{i+1} \equiv_p \binom{x_i+y_i}{p} - (x_i+y_i)^{(p-1)} r_i, \qquad r_0 \equiv_p 0. \tag{3.3.38}$$

■

3.3.3 Produit

En ce qui concerne le produit, on a le résultat suivant.

Théorème 3.3.5 *Si p est un nombre premier et $z = f(x,y) = xy$, alors*

$$z_i \equiv_p \binom{xy}{p^i} \equiv_p \sum_{m,n\geq 0} U_{p^i}^{m,n} \binom{x_0}{m_0}\binom{x_1}{m_1}\cdots\binom{y_0}{n_0}\binom{y_1}{n_1}\cdots \tag{3.3.39}$$

où les nombres $U_k^{m,n}$ satisfont la récurrence

$$U_{k+1}^{m,n} = \frac{(mn-k)U_k^{m,n} + mnU_k^{m-1,n} + mnU_k^{m,n-1} + mnU_k^{m-1,n-1}}{k+1} \quad avec \ U_0^{m,n} = \delta_0^m \delta_0^n \tag{3.3.40}$$

avec $1 \leq m \leq p^i$ et $1 \leq n \leq p^i$.

Sans entrer dans les détails, Gilbert Labelle mentionne dans son article (Labelle, 1978) que cette formule se trouve à partir de la décomposition 1 en faisant appel à l'inversion de Möbius en combinatoire. Il mentionne aussi que le nombre $U_k^{m,n}$ est égal au nombre de parties à k éléments du rectangle $\{1,\ldots,m\}\times\{1,\ldots,n\}$ qui sont saturées. Une partie $S \subseteq \{1,\ldots,m\}\times\{1,\ldots,n\}$ étant déclarée saturée si ses deux projections reçoivent complètement les axes $\{1,\ldots,m\}$ et $\{1,\ldots,n\}$.

En utilisant Maple, les premiers z_i du produit $z = xy$ ont été calculés pour $p = 2$.

Exemple 3.3.1 *Soit M_i, la matrice contenant les éléments $U_{p^i}^{m,n}$ où m indice les lignes de 1 à p^i et n indice les colonnes de 1 à p^i. Pour $p = 2$ et $i = 0, 1$ et 2, la matrice M_i peut être calculée avec Maple.*

$$M_0 \equiv_2 \begin{bmatrix} U_1^{1,1} \end{bmatrix} \equiv_2 \begin{bmatrix} 1 \end{bmatrix}, \quad M_1 \equiv_2 \begin{bmatrix} U_2^{1,1} & U_2^{1,2} \\ U_2^{2,1} & U_2^{2,2} \end{bmatrix} \equiv_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{3.3.41}$$

$$M_2 \equiv_2 \begin{bmatrix} U_4^{1,1} & U_4^{1,2} & U_4^{1,3} & U_4^{1,4} \\ U_4^{2,1} & U_4^{2,2} & U_4^{2,3} & U_4^{2,4} \\ U_4^{3,1} & U_4^{3,2} & U_4^{3,3} & U_4^{3,4} \\ U_4^{4,1} & U_4^{4,2} & U_4^{4,3} & U_4^{4,4} \end{bmatrix} \equiv_2 \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.3.42}$$

De ces matrices, on peut alors déduire que :

$$z_0 \equiv_2 1 \cdot \binom{x}{1}\binom{y}{1} \equiv_2 x_0 y_0 \tag{3.3.43}$$

$$z_1 \equiv_2 1 \cdot \binom{x}{1}\binom{y}{2} + 1 \cdot \binom{x}{2}\binom{y}{1} \equiv_2 x_0 y_1 + x_1 y_0 \tag{3.3.44}$$

$$z_2 \equiv_2 \binom{x}{1}\binom{y}{4} + \binom{x}{2}\binom{y}{2} + \binom{x}{3}\binom{y}{3} + \binom{x}{4}\binom{y}{1} \tag{3.3.45}$$

$$\equiv_2 x_0 y_2 + x_1 y_1 + x_0 x_1 y_0 y_1 + x_2 y_0. \tag{3.3.46}$$

Pour $p = 2$,

$$z = z_0 + z_1 2 + z_2 + 2^2 + \ldots \tag{3.3.47}$$

Il en résulte que

$$z = x \cdot y = (x_0 y_0) + (x_0 y_1 + x_1 y_0)2 + (x_0 y_2 + x_1 y_1 + x_0 x_1 y_0 y_1 + x_2 y_0)2^2 + \ldots \tag{3.3.48}$$

Exemple 3.3.2 *Pour $p = 3$, les matrices M_i pour $i = 0, 1$ et 2, sont trouvées.*

$$M_0 \equiv_3 \begin{bmatrix} 1 \end{bmatrix}, \quad M_1 \equiv_3 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad M_2 \equiv_3 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.3.49}$$

À partir de ces informations, il est possible de trouver de manière similaire à l'exemple précédent la décomposition en pits de la fonction produit pour $p = 3$.

3.3.4 Monôme

Gilbert Labelle mentionne aussi dans son article (Labelle, 1978) une décomposition en pits pour la fonction monôme

$$z = f(x, y, z, \dots) = ax^r y^s z^t \cdots . \tag{3.3.50}$$

Il est possible de trouver les décompositions pour ce genre de fonctions.

3.4 Applications

Étant donné que les opérations logiques «ou exclusif» et «et» peuvent être représentées par des portes logiques et que le nombre deux est un nombre premier, certains ingénieurs seraient peut-être intéressés à développer des diagrammes logiques afin de bâtir via des systèmes électroniques des fonctions mathématiques. Par exemple, l'opération d'incrémentation ($z = x + 1$) est largement utilisée en informatique, en particulier lorsqu'il y a des boucles. Il est possible de construire un diagramme de portes logiques satisfaisant

$$z_i \equiv_2 (x + 1)_i \equiv_2 x_0 x_1 \cdots x_{i-1} + x_i. \tag{3.4.1}$$

Il y a cependant des opérations mathématiques qui sont beaucoup plus coûteuses en termes de temps de calculs. L'utilisation de ces décompositions de fonctions peut dans certains cas alléger le temps de calculs de ces fonctions d'autant plus qu'une implémentation parallèle efficace de ces fonctions est possible. Certains ordinateurs à plusieurs processeurs pourrait bénéficier des avantages qu'offre les décompositions de fonctions. Comme les pits peuvent être calculés

de manière indépendante, il est alors possible de répartir les calculs de fonctions définies sur $\mathbb{N}$ sur plusieurs processeurs.

CONCLUSION

Nous avons présenté différentes représentations de tables de vérité. Le polynôme (modulo deux) d'une table de vérité peut être trouvé à l'aide de la fonction chapeau de Labelle : celle-ci permet de traduire une représentation de la forme descriptive en une représentation polynomiale et réciproquement via les propriétés involutives et bijectives de la fonction. Notons que nous n'avons pas généralisé la fonction chapeau pour une base supérieure à deux. Il serait intéressant de développer une fonction chapeau permettant de trouver le polynôme (modulo k) associé à une table de vérité à k états, pour $k > 2$. Pour ce faire, il est nécessaire d'exprimer les variables et les chiffres de manière ensembliste pour une base de numération qui est supérieure à 2. Cette extension de la fonction chapeau, n'étant pas essentielle pour les objectifs de ce mémoire, mériterait qu'on s'y intéresse ultérieurement.

En second lieu, pour une base p, où p est un nombre premier, nous avons vu qu'il était possible de traduire une fonction définie sur $\mathbb{N}$ en des fonctions définies sur les chiffres en utilisant des décompositions de fonctions en pits.

- La première décomposition a été prouvée à l'aide de la méthode de Newton et du Théorème de Lucas.
- La seconde a été démontrée avec le delta de Kronecker et deux formes sous-jacentes à cette décomposition ont été explicitées.
- La troisième utilise les coefficients du polynôme (modulo p) exprimant le delta de Kronecker.

Ces décompositions nous ont suggéré des fonctions définies sur les pits pour les fonctions : incrémentation, somme et produit, que nous avons par la suite démontrées.

Quelques programmes relatifs à ces décompositions ont été présentés. Cependant, les décompositions de fonctions en pits pour des fonctions à plusieurs variables $f(x, y, \dots)$ sont difficiles à établir. Néanmoins, une fois que ces formules sont trouvées, elles peuvent avoir une importance capitale. Le souci des détails et des performances est très important dans le domaine de l'informatique. Puisque les décompositions vues dans ce mémoire s'appliquent aux pits et en particulier aux bits, le calcul des fonctions peut avoir une implémentation parallèle. Il serait alors intéressant pour des ingénieurs en informatique d'appliquer ces méthodes à la conception de circuits d'ordinateurs. Avec la venue des ordinateurs à plusieurs processeurs, les programmeurs pourraient aussi bénéficier des avantages qu'offrent les décompositions de fonctions dans le but d'implémenter efficacement des fonctions définies sur $\mathbb{N}$.

ANNEXE A

PROGRAMMES MAPLE SUR LA FONCTION CHAPEAU

Voici quelques exemples relatifs à la fonction chapeau.

```
> with(combinat):
Warning, the protected name Chi has been redefined and unprotected

Chapeau - version 1
Permet de trouver la fonction chapeau.
Entrées: E est un ensemble de petits ensembles (forme descriptive)
et n est le nombre de variables (valeur maximale des petits ensembles).
Sortie: Retourne un ensemble d'ensembles (forme polynomiale).
Algorithme: t contient un nombre impair de s dans E.

> Chapeau1 := proc(E,n)
>    local a, L;
>    L := {};
>    for a in powerset(n) do
>      map(`subset`,[op(E)],a);
>      convert(%,`+`);
>      if coeff(%,true) mod 2 = 1 then
>         L := {op(L),a};
>      end if;
>    end do;
>    return L;
> end proc:
Exemples
> Chapeau1({},3);
                                    {}
> Chapeau1({{}},3);
```

```
        {{}, {3}, {2}, {2, 3}, {1, 3}, {1, 2}, {1}, {1, 2, 3}}

> Chapeau1({{},{2,3},{1,3},{1,2}},3);
                          {{}, {3}, {2}, {1}}

> Chapeau1({{2},{1,2},{1,2,3}},3);
                        {{2}, {2, 3}, {1, 2, 3}}

> Chapeau1({{},{1},{3},{2}},3);
                      {{}, {2, 3}, {1, 3}, {1, 2}}

> Chapeau1({{},{2,3},{1,3},{1,2}},4);
            {{}, {4}, {1, 4}, {2, 4}, {3}, {2}, {1}, {3, 4}}
```

Chapeau - version 2
Permet de trouver la fonction chapeau.
Entrées: E est un ensemble de petits ensembles (forme descriptive)
et n est le nombre de variables (valeur maximale des petits ensembles).
Sortie: Retourne un ensemble d'ensembles (forme polynomiale).
Alogrithme: Développement de la forme descriptive en polynôme.

```
> Chapeau2 := proc(E,n)
>   local a, A, terme, Formule;
>   Formule := 0;
>   for A in E do
>      terme := 1;
>      for a in {seq(i,i=1..n)} do
>         if member(a,A) then
>            terme := terme*(x[a])
>         else
>            terme := terme*(1+x[a])
>         end if;
>      end do;
>      Formule := Formule + terme;
>   end do;
>   Formule := expand(Formule);
>   Formule := convert(Formule,mod2);
>   Formule := convert(Formule,set);
>   if Formule = {0} then
>     return {};
>   end if;
```

```
>   if member(1,Formule) then
>     Formule := (Formule minus {1}) union {{}};
>   end if;
>   Formule := map(convert,Formule,set);
>   Formule := subs( seq(x[i] = i,i=1..n),Formule);
>
>   return Formule;
> end proc:

> Chapeau2({},3);
                                    {}
> Chapeau2({{}},3);
        {{}, {3}, {2}, {2, 3}, {1, 3}, {1, 2}, {1}, {1, 2, 3}}

> Chapeau2({{},{2,3},{1,3},{1,2}},3);
                          {{}, {3}, {2}, {1}}

> Chapeau2({{},{1},{3},{2}},3);
                     {{}, {2, 3}, {1, 3}, {1, 2}}

> Chapeau2({},3);
                                    {}
```

Abreger
Permet d'abréger un ensemble de petits ensembles en lui trouvant un ensemble qui lui est associé.
Entrées: E est un ensemble de petits ensembles (forme descriptive ou polynomiale).
Sortie: Retourne un l'ensemble qui est associé à l'entrée.
Alogrithme: Transformer les petits ensembles en nombres binaires. Retourner l'ensemble contenant les nombres décimaux associés à ces nombres binaires.

```
> Abreger := proc(E)
>   local a,Dec;
>   Dec := {};
>   for a in E do
>     map2('^'/2,2,a);
>     Dec := {op(Dec),convert(%,'+')};
>   end do;
>   return Dec;
```

```
> end proc:

> Abreger({{},{1},{2,3},{1,3},{1,2}});
                              {0, 1, 3, 5, 6}

Desabreger
> with(StringTools):
> with(ListTools):

Permet de désabréger un ensemble abrégé en ensemble de
petits ensembles.
Entrées: A est un ensemble Abrégé.
Sortie: l'ensemble de petits ensembles associé à
un ensemble abrégé.
Alogrithme: Transformer les nombres décimaux en
nombres binaires. Trouver les petits ensembles
associés à ces nombre binaires. Retourner l'ensemble
de petits ensembles.

> Desabreger := proc(A)
>   local a,E,i,Ltmp;
>   E := {};
>   for a in A do
>     convert(a,binary);
>     convert(%,string);
>     convert(%,list);
>     Ltmp := map(Ord-48,%);
>     for i from 1 to nops(Ltmp) do
>       Ltmp[i] := Ltmp[i]*(nops(Ltmp)-i+1);
>     end do;
>     E := {op(E),{op(Ltmp)} minus {0}};
>   end do;
>   return E;
> end proc:

> Desabreger({0,1,3,5,6});
                  {{}, {2, 3}, {1, 3}, {1, 2}, {1}}

> Desabreger({0,1,2,3,4});
                     {{}, {3}, {2}, {1, 2}, {1}}
```

```
> Desabreger({});
                                    {}

ChapeauAbreger1
Permet de trouver la fonction chapeau.
Entrées: A est un ensemble abrégé (ensemble de
petits ensembles)
et n est le nombre de variables (valeur maximale
des petits ensembles).
Sortie: Retourne l'ensemble abrégé de la
fonction chapeau  (forme polynomiale).
Algorithme: Désabréger, appliquer chapeau1, abréger.

> ChapeauAbreger1 := proc(A,n)
>   Desabreger(A);
>   Chapeau1(%,n);
>   Abreger(%);
> end proc:

> ChapeauAbreger1({0,3,5,6},3);
                              {0, 1, 2, 4}

> ChapeauAbreger1({0,1,2,4},3);
                              {0, 3, 5, 6}

ChapeauAbreger2
Permet de trouver la fonction chapeau.
Entrées: A est un ensemble abrégé
(ensemble de petits ensembles)
et n est le nombre de variables
(valeur maximale des petits ensembles).
Sortie: Retourne l'ensemble abrégé de la
fonction chapeau  (forme polynomiale).
Algorithme: Désabréger, appliquer chapeau2, abréger.

> ChapeauAbreger2 := proc(Dec,n)
>   Desabreger(Dec);
>   Chapeau2(%,n);
>   Abreger(%);
> end proc:
```

```
> ChapeauAbreger2({0,3,5,6},3);
                              {0, 1, 2, 4}

> ChapeauAbreger2({0,1,2,4},3);
                              {0, 3, 5, 6}
```

PhiInverse
Trouver la forme descriptive d'un ensemble de petits ensembles.

```
> PhiInverse := proc(E,n)
>   local e, i, terme, total;
>   total := 0;
>   for e in E do
>     terme := 1;
>     for i from 1 to n do
>       if member(i,e) then
>          terme := terme*x[i]^[1];
>       else
>          terme := terme*x[i]^[0];
>       end if;
>     end do;
>     total := total + terme;
>   end do;
>   total;
> end proc:

> PhiInverse({},3);
                                    0

> PhiInverse({{2},{1,2},{1,2,3}},3);

      [0]     [1]     [0]       [1]     [1]     [0]
  x[1]    x[2]    x[3]    + x[1]    x[2]    x[3]

               [1]     [1]     [1]
         + x[1]    x[2]    x[3]

> PhiInverse({{},{2,3},{1,3},{1,2},{1}},3);

      [0]     [0]     [0]       [0]     [1]     [1]
  x[1]    x[2]    x[3]    + x[1]    x[2]    x[3]
```

```
                [1]     [0]     [1]       [1]     [1]     [0]
        + x[1]    x[2]    x[3]    + x[1]    x[2]    x[3]

                [1]     [0]     [0]
        + x[1]    x[2]    x[3]
```

PsiInverse
Trouver la forme polynomiale d'un ensemble de petits ensembles.

```
> PsiInverse := proc(E,n)
>   local e, i, terme, total;
>   total := 0;
>   for e in E do
>     terme := 1;
>     for i from 1 to n do
>       if member(i,e) then
>          terme := terme*x[i];
>       end if;
>     end do;
>     total := total + terme;
>   end do;
>   total;
> end proc:

> PsiInverse({},3);
                                    0

> PsiInverse({{2},{1,2},{1,2,3}},3);
                  x[2] + x[1] x[2] + x[1] x[2] x[3]

> PsiInverse({{},{2,3},{1,3},{1,2},{1}},3);
             1 + x[2] x[3] + x[1] x[3] + x[1] x[2] + x[1]
```

ANNEXE B

ÉNUMÉRATION DES CHAPEAUX

Une énumération des chapeaux pour toutes les tables de vérité à une, deux et trois variables sont faites ici.

```
> with(combinat):

IemeSousEnsemble
Permet de trouver le i_ième sous-ensemble de E, où n
est la valeur entière maximale pour les éléments
des petits ensembles.

> IemeSousEnsemble := proc(i,E,n)
>   local L, P, j;
>   L := convert(i,base,2);
>   P := [op(E)];
>   for j from 1 to nops(P) do
>      if j <= nops(L) then
>         P[j] := P[j]*L[j];
>      else
>         P[j] := 0;
>      end if;
>   end do;
>   return {op(P)} minus {0};
> end proc:


> n := 2:
> E := powerset(n):
```

```
> for i from 0 to 2^(2^n)-1 do
>   IemeSousEnsemble(i,E,n);
> end do;
```

```
                                {}
                               {{}}
                               {{2}}
                              {{}, {2}}
                               {{1, 2}}
                             {{}, {1, 2}}
                            {{2}, {1, 2}}
                          {{}, {2}, {1, 2}}
                               {{1}}
                              {{}, {1}}
                              {{2}, {1}}
                            {{}, {2}, {1}}
                            {{1, 2}, {1}}
                          {{}, {1, 2}, {1}}
                          {{2}, {1, 2}, {1}}
                        {{}, {2}, {1, 2}, {1}}
```

```
Énumération des chapeaux
Énumérer de tous les ensembles ayant n variables et
présenter les informations sur chaque chapeau
(i-ième sous-ensemble, entrée et sortie).

Exemple d'affichage:
"
3
{{}, {1}}
{{}}
"
Le 3 correspond au 3-ième sous-ensemble de P(n).
L'entrée de la fonction chapeau qui est {{}, {1}}
est ce 3-ième sous-ensemble.
L'ensemble {{}} est le résultat de la fonction chapeau
lorsque l'entrée est {{}, {1}}.

> EnumerationDesChapeaux := proc(n)
>    local E,i,P;
>    P := powerset(n);
>    for i from 0 to 2^(2^n)-1 do
```

```
>       E := IemeSousEnsemble(i,P,n);
>       lprint(i);
>       lprint(E);
>       lprint(Chapeau1(E,n));
>       printf(%c,10);          # saut de ligne
>    end do;
> end proc:

> EnumerationDesChapeaux(1):
0
{}
{}

1
{{}}
{{}, {1}}

2
{{1}}
{{1}}

3
{{}, {1}}
{{}}

> EnumerationDesChapeaux(2):
0
{}
{}

1
{{}}
{{}, {1}, {2}, {1, 2}}

2
{{1}}
{{1}, {1, 2}}

3
{{}, {1}}
{{}, {2}}
```

```
4
{{2}}
{{2}, {1, 2}}

5
{{}, {2}}
{{}, {1}}

6
{{1}, {2}}
{{1}, {2}}

7
{{}, {1}, {2}}
{{}, {1, 2}}

8
{{1, 2}}
{{1, 2}}

9
{{}, {1, 2}}
{{}, {1}, {2}}

10
{{1}, {1, 2}}
{{1}}

11
{{}, {1}, {1, 2}}
{{}, {2}, {1, 2}}

12
{{2}, {1, 2}}
{{2}}

13
{{}, {2}, {1, 2}}
{{}, {1}, {1, 2}}
```

```
14
{{1}, {2}, {1, 2}}
{{1}, {2}, {1, 2}}

15
{{}, {1}, {2}, {1, 2}}
{{}}

> EnumerationDesChapeaux(3):
0
{}
{}

1
{{}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}

2
{{1, 2, 3}}
{{1, 2, 3}}

3
{{}, {1, 2, 3}}
{{}, {3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}

4
{{3}}
{{1, 2, 3}, {3}, {1, 3}, {2, 3}}

5
{{}, {3}}
{{}, {1}, {2}, {1, 2}}

6
{{1, 2, 3}, {3}}
{{3}, {1, 3}, {2, 3}}

7
{{}, {1, 2, 3}, {3}}
{{}, {1, 2, 3}, {1}, {2}, {1, 2}}
```

```
8
{{1, 3}}
{{1, 2, 3}, {1, 3}}

9
{{}, {1, 3}}
{{}, {3}, {2, 3}, {1}, {2}, {1, 2}}

10
{{1, 2, 3}, {1, 3}}
{{1, 3}}

11
{{}, {1, 2, 3}, {1, 3}}
{{}, {1, 2, 3}, {3}, {2, 3}, {1}, {2}, {1, 2}}

12
{{3}, {1, 3}}
{{3}, {2, 3}}

13
{{}, {3}, {1, 3}}
{{}, {1, 2, 3}, {1, 3}, {1}, {2}, {1, 2}}

14
{{1, 2, 3}, {3}, {1, 3}}
{{1, 2, 3}, {3}, {2, 3}}

15
{{}, {1, 2, 3}, {3}, {1, 3}}
{{}, {1, 3}, {1}, {2}, {1, 2}}

16
{{2, 3}}
{{1, 2, 3}, {2, 3}}

17
{{}, {2, 3}}
{{}, {3}, {1, 3}, {1}, {2}, {1, 2}}

18
```

```
{{1, 2, 3}, {2, 3}}
{{2, 3}}

19
{{}, {1, 2, 3}, {2, 3}}
{{}, {1, 2, 3}, {3}, {1, 3}, {1}, {2}, {1, 2}}

20
{{3}, {2, 3}}
{{3}, {1, 3}}

21
{{}, {3}, {2, 3}}
{{}, {1, 2, 3}, {2, 3}, {1}, {2}, {1, 2}}

22
{{1, 2, 3}, {3}, {2, 3}}
{{1, 2, 3}, {3}, {1, 3}}

23
{{}, {1, 2, 3}, {3}, {2, 3}}
{{}, {2, 3}, {1}, {2}, {1, 2}}

24
{{1, 3}, {2, 3}}
{{1, 3}, {2, 3}}

25
{{}, {1, 3}, {2, 3}}
{{}, {1, 2, 3}, {3}, {1}, {2}, {1, 2}}

26
{{1, 2, 3}, {1, 3}, {2, 3}}
{{1, 2, 3}, {1, 3}, {2, 3}}

27
{{}, {1, 2, 3}, {1, 3}, {2, 3}}
{{}, {3}, {1}, {2}, {1, 2}}

28
{{3}, {1, 3}, {2, 3}}
```

```
{{1, 2, 3}, {3}}

29
{{}, {3}, {1, 3}, {2, 3}}
{{}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}

30
{{1, 2, 3}, {3}, {1, 3}, {2, 3}}
{{3}}

31
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}}
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}

32
{{1}}
{{1, 2, 3}, {1, 3}, {1}, {1, 2}}

33
{{}, {1}}
{{}, {3}, {2, 3}, {2}}

34
{{1, 2, 3}, {1}}
{{1, 3}, {1}, {1, 2}}

35
{{}, {1, 2, 3}, {1}}
{{}, {1, 2, 3}, {3}, {2, 3}, {2}}

36
{{3}, {1}}
{{3}, {2, 3}, {1}, {1, 2}}

37
{{}, {3}, {1}}
{{}, {1, 2, 3}, {1, 3}, {2}}

38
{{1, 2, 3}, {3}, {1}}
{{1, 2, 3}, {3}, {2, 3}, {1}, {1, 2}}
```

```
39
{{}, {1, 2, 3}, {3}, {1}}
{{}, {1, 3}, {2}}

40
{{1, 3}, {1}}
{{1}, {1, 2}}

41
{{}, {1, 3}, {1}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {2}}

42
{{1, 2, 3}, {1, 3}, {1}}
{{1, 2, 3}, {1}, {1, 2}}

43
{{}, {1, 2, 3}, {1, 3}, {1}}
{{}, {3}, {1, 3}, {2, 3}, {2}}

44
{{3}, {1, 3}, {1}}
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {1, 2}}

45
{{}, {3}, {1, 3}, {1}}
{{}, {2}}

46
{{1, 2, 3}, {3}, {1, 3}, {1}}
{{3}, {1, 3}, {2, 3}, {1}, {1, 2}}

47
{{}, {1, 2, 3}, {3}, {1, 3}, {1}}
{{}, {1, 2, 3}, {2}}

48
{{2, 3}, {1}}
{{1, 3}, {2, 3}, {1}, {1, 2}}
```

```
49
{{}, {2, 3}, {1}}
{{}, {1, 2, 3}, {3}, {2}}

50
{{1, 2, 3}, {2, 3}, {1}}
{{1, 2, 3}, {1, 3}, {2, 3}, {1}, {1, 2}}

51
{{}, {1, 2, 3}, {2, 3}, {1}}
{{}, {3}, {2}}

52
{{3}, {2, 3}, {1}}
{{1, 2, 3}, {3}, {1}, {1, 2}}

53
{{}, {3}, {2, 3}, {1}}
{{}, {1, 3}, {2, 3}, {2}}

54
{{1, 2, 3}, {3}, {2, 3}, {1}}
{{3}, {1}, {1, 2}}

55
{{}, {1, 2, 3}, {3}, {2, 3}, {1}}
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {2}}

56
{{1, 3}, {2, 3}, {1}}
{{1, 2, 3}, {2, 3}, {1}, {1, 2}}

57
{{}, {1, 3}, {2, 3}, {1}}
{{}, {3}, {1, 3}, {2}}

58
{{1, 2, 3}, {1, 3}, {2, 3}, {1}}
{{2, 3}, {1}, {1, 2}}

59
```

```
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2}}

60
{{3}, {1, 3}, {2, 3}, {1}}
{{3}, {1, 3}, {1}, {1, 2}}

61
{{}, {3}, {1, 3}, {2, 3}, {1}}
{{}, {1, 2, 3}, {2, 3}, {2}}

62
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}}
{{1, 2, 3}, {3}, {1, 3}, {1}, {1, 2}}

63
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}}
{{}, {2, 3}, {2}}

64
{{2}}
{{1, 2, 3}, {2, 3}, {2}, {1, 2}}

65
{{}, {2}}
{{}, {3}, {1, 3}, {1}}

66
{{1, 2, 3}, {2}}
{{2, 3}, {2}, {1, 2}}

67
{{}, {1, 2, 3}, {2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {1}}

68
{{3}, {2}}
{{3}, {1, 3}, {2}, {1, 2}}

69
{{}, {3}, {2}}
```

```
{{}, {1, 2, 3}, {2, 3}, {1}}

70
{{1, 2, 3}, {3}, {2}}
{{1, 2, 3}, {3}, {1, 3}, {2}, {1, 2}}

71
{{}, {1, 2, 3}, {3}, {2}}
{{}, {2, 3}, {1}}

72
{{1, 3}, {2}}
{{1, 3}, {2, 3}, {2}, {1, 2}}

73
{{}, {1, 3}, {2}}
{{}, {1, 2, 3}, {3}, {1}}

74
{{1, 2, 3}, {1, 3}, {2}}
{{1, 2, 3}, {1, 3}, {2, 3}, {2}, {1, 2}}

75
{{}, {1, 2, 3}, {1, 3}, {2}}
{{}, {3}, {1}}

76
{{3}, {1, 3}, {2}}
{{1, 2, 3}, {3}, {2}, {1, 2}}

77
{{}, {3}, {1, 3}, {2}}
{{}, {1, 3}, {2, 3}, {1}}

78
{{1, 2, 3}, {3}, {1, 3}, {2}}
{{3}, {2}, {1, 2}}

79
{{}, {1, 2, 3}, {3}, {1, 3}, {2}}
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1}}
```

```
80
{{2, 3}, {2}}
{{2}, {1, 2}}

81
{{}, {2, 3}, {2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}}

82
{{1, 2, 3}, {2, 3}, {2}}
{{1, 2, 3}, {2}, {1, 2}}

83
{{}, {1, 2, 3}, {2, 3}, {2}}
{{}, {3}, {1, 3}, {2, 3}, {1}}

84
{{3}, {2, 3}, {2}}
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {2}, {1, 2}}

85
{{}, {3}, {2, 3}, {2}}
{{}, {1}}

86
{{1, 2, 3}, {3}, {2, 3}, {2}}
{{3}, {1, 3}, {2, 3}, {2}, {1, 2}}

87
{{}, {1, 2, 3}, {3}, {2, 3}, {2}}
{{}, {1, 2, 3}, {1}}

88
{{1, 3}, {2, 3}, {2}}
{{1, 2, 3}, {1, 3}, {2}, {1, 2}}

89
{{}, {1, 3}, {2, 3}, {2}}
{{}, {3}, {2, 3}, {1}}
```

```
90
{{1, 2, 3}, {1, 3}, {2, 3}, {2}}
{{1, 3}, {2}, {1, 2}}

91
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {2}}
{{}, {1, 2, 3}, {3}, {2, 3}, {1}}

92
{{3}, {1, 3}, {2, 3}, {2}}
{{3}, {2, 3}, {2}, {1, 2}}

93
{{}, {3}, {1, 3}, {2, 3}, {2}}
{{}, {1, 2, 3}, {1, 3}, {1}}

94
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {2}}
{{1, 2, 3}, {3}, {2, 3}, {2}, {1, 2}}

95
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {2}}
{{}, {1, 3}, {1}}

96
{{1}, {2}}
{{1, 3}, {2, 3}, {1}, {2}}

97
{{}, {1}, {2}}
{{}, {1, 2, 3}, {3}, {1, 2}}

98
{{1, 2, 3}, {1}, {2}}
{{1, 2, 3}, {1, 3}, {2, 3}, {1}, {2}}

99
{{}, {1, 2, 3}, {1}, {2}}
{{}, {3}, {1, 2}}

100
```

```
{{3}, {1}, {2}}
{{1, 2, 3}, {3}, {1}, {2}}

101
{{}, {3}, {1}, {2}}
{{}, {1, 3}, {2, 3}, {1, 2}}

102
{{1, 2, 3}, {3}, {1}, {2}}
{{3}, {1}, {2}}

103
{{}, {1, 2, 3}, {3}, {1}, {2}}
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1, 2}}

104
{{1, 3}, {1}, {2}}
{{1, 2, 3}, {2, 3}, {1}, {2}}

105
{{}, {1, 3}, {1}, {2}}
{{}, {3}, {1, 3}, {1, 2}}

106
{{1, 2, 3}, {1, 3}, {1}, {2}}
{{2, 3}, {1}, {2}}

107
{{}, {1, 2, 3}, {1, 3}, {1}, {2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {1, 2}}

108
{{3}, {1, 3}, {1}, {2}}
{{3}, {1, 3}, {1}, {2}}

109
{{}, {3}, {1, 3}, {1}, {2}}
{{}, {1, 2, 3}, {2, 3}, {1, 2}}

110
{{1, 2, 3}, {3}, {1, 3}, {1}, {2}}
```

```
{{1, 2, 3}, {3}, {1, 3}, {1}, {2}}

111
{{}, {1, 2, 3}, {3}, {1, 3}, {1}, {2}}
{{}, {2, 3}, {1, 2}}

112
{{2, 3}, {1}, {2}}
{{1, 2, 3}, {1, 3}, {1}, {2}}

113
{{}, {2, 3}, {1}, {2}}
{{}, {3}, {2, 3}, {1, 2}}

114
{{1, 2, 3}, {2, 3}, {1}, {2}}
{{1, 3}, {1}, {2}}

115
{{}, {1, 2, 3}, {2, 3}, {1}, {2}}
{{}, {1, 2, 3}, {3}, {2, 3}, {1, 2}}

116
{{3}, {2, 3}, {1}, {2}}
{{3}, {2, 3}, {1}, {2}}

117
{{}, {3}, {2, 3}, {1}, {2}}
{{}, {1, 2, 3}, {1, 3}, {1, 2}}

118
{{1, 2, 3}, {3}, {2, 3}, {1}, {2}}
{{1, 2, 3}, {3}, {2, 3}, {1}, {2}}

119
{{}, {1, 2, 3}, {3}, {2, 3}, {1}, {2}}
{{}, {1, 3}, {1, 2}}

120
{{1, 3}, {2, 3}, {1}, {2}}
{{1}, {2}}
```

```
121
{{}, {1, 3}, {2, 3}, {1}, {2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1, 2}}

122
{{1, 2, 3}, {1, 3}, {2, 3}, {1}, {2}}
{{1, 2, 3}, {1}, {2}}

123
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1}, {2}}
{{}, {3}, {1, 3}, {2, 3}, {1, 2}}

124
{{3}, {1, 3}, {2, 3}, {1}, {2}}
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {2}}

125
{{}, {3}, {1, 3}, {2, 3}, {1}, {2}}
{{}, {1, 2}}

126
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {2}}
{{3}, {1, 3}, {2, 3}, {1}, {2}}

127
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {2}}
{{}, {1, 2, 3}, {1, 2}}

128
{{1, 2}}
{{1, 2, 3}, {1, 2}}

129
{{}, {1, 2}}
{{}, {3}, {1, 3}, {2, 3}, {1}, {2}}

130
{{1, 2, 3}, {1, 2}}
{{1, 2}}
```

```
131
{{}, {1, 2, 3}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {2}}

132
{{3}, {1, 2}}
{{3}, {1, 3}, {2, 3}, {1, 2}}

133
{{}, {3}, {1, 2}}
{{}, {1, 2, 3}, {1}, {2}}

134
{{1, 2, 3}, {3}, {1, 2}}
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1, 2}}

135
{{}, {1, 2, 3}, {3}, {1, 2}}
{{}, {1}, {2}}

136
{{1, 3}, {1, 2}}
{{1, 3}, {1, 2}}

137
{{}, {1, 3}, {1, 2}}
{{}, {1, 2, 3}, {3}, {2, 3}, {1}, {2}}

138
{{1, 2, 3}, {1, 3}, {1, 2}}
{{1, 2, 3}, {1, 3}, {1, 2}}

139
{{}, {1, 2, 3}, {1, 3}, {1, 2}}
{{}, {3}, {2, 3}, {1}, {2}}

140
{{3}, {1, 3}, {1, 2}}
{{1, 2, 3}, {3}, {2, 3}, {1, 2}}

141
```

```
{{}, {3}, {1, 3}, {1, 2}}
{{}, {1, 3}, {1}, {2}}

142
{{1, 2, 3}, {3}, {1, 3}, {1, 2}}
{{3}, {2, 3}, {1, 2}}

143
{{}, {1, 2, 3}, {3}, {1, 3}, {1, 2}}
{{}, {1, 2, 3}, {1, 3}, {1}, {2}}

144
{{2, 3}, {1, 2}}
{{2, 3}, {1, 2}}

145
{{}, {2, 3}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {1}, {2}}

146
{{1, 2, 3}, {2, 3}, {1, 2}}
{{1, 2, 3}, {2, 3}, {1, 2}}

147
{{}, {1, 2, 3}, {2, 3}, {1, 2}}
{{}, {3}, {1, 3}, {1}, {2}}

148
{{3}, {2, 3}, {1, 2}}
{{1, 2, 3}, {3}, {1, 3}, {1, 2}}

149
{{}, {3}, {2, 3}, {1, 2}}
{{}, {2, 3}, {1}, {2}}

150
{{1, 2, 3}, {3}, {2, 3}, {1, 2}}
{{3}, {1, 3}, {1, 2}}

151
{{}, {1, 2, 3}, {3}, {2, 3}, {1, 2}}
```

```
{{}, {1, 2, 3}, {2, 3}, {1}, {2}}

152
{{1, 3}, {2, 3}, {1, 2}}
{{1, 2, 3}, {1, 3}, {2, 3}, {1, 2}}

153
{{}, {1, 3}, {2, 3}, {1, 2}}
{{}, {3}, {1}, {2}}

154
{{1, 2, 3}, {1, 3}, {2, 3}, {1, 2}}
{{1, 3}, {2, 3}, {1, 2}}

155
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1}, {2}}

156
{{3}, {1, 3}, {2, 3}, {1, 2}}
{{3}, {1, 2}}

157
{{}, {3}, {1, 3}, {2, 3}, {1, 2}}
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1}, {2}}

158
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1, 2}}
{{1, 2, 3}, {3}, {1, 2}}

159
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1, 2}}
{{}, {1, 3}, {2, 3}, {1}, {2}}

160
{{1}, {1, 2}}
{{1, 3}, {1}}

161
{{}, {1}, {1, 2}}
{{}, {1, 2, 3}, {3}, {2, 3}, {2}, {1, 2}}
```

```
162
{{1, 2, 3}, {1}, {1, 2}}
{{1, 2, 3}, {1, 3}, {1}}

163
{{}, {1, 2, 3}, {1}, {1, 2}}
{{}, {3}, {2, 3}, {2}, {1, 2}}

164
{{3}, {1}, {1, 2}}
{{1, 2, 3}, {3}, {2, 3}, {1}}

165
{{}, {3}, {1}, {1, 2}}
{{}, {1, 3}, {2}, {1, 2}}

166
{{1, 2, 3}, {3}, {1}, {1, 2}}
{{3}, {2, 3}, {1}}

167
{{}, {1, 2, 3}, {3}, {1}, {1, 2}}
{{}, {1, 2, 3}, {1, 3}, {2}, {1, 2}}

168
{{1, 3}, {1}, {1, 2}}
{{1, 2, 3}, {1}}

169
{{}, {1, 3}, {1}, {1, 2}}
{{}, {3}, {1, 3}, {2, 3}, {2}, {1, 2}}

170
{{1, 2, 3}, {1, 3}, {1}, {1, 2}}
{{1}}

171
{{}, {1, 2, 3}, {1, 3}, {1}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {2}, {1, 2}}
```

```
172
{{3}, {1, 3}, {1}, {1, 2}}
{{3}, {1, 3}, {2, 3}, {1}}

173
{{}, {3}, {1, 3}, {1}, {1, 2}}
{{}, {1, 2, 3}, {2}, {1, 2}}

174
{{1, 2, 3}, {3}, {1, 3}, {1}, {1, 2}}
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}}

175
{{}, {1, 2, 3}, {3}, {1, 3}, {1}, {1, 2}}
{{}, {2}, {1, 2}}

176
{{2, 3}, {1}, {1, 2}}
{{1, 2, 3}, {1, 3}, {2, 3}, {1}}

177
{{}, {2, 3}, {1}, {1, 2}}
{{}, {3}, {2}, {1, 2}}

178
{{1, 2, 3}, {2, 3}, {1}, {1, 2}}
{{1, 3}, {2, 3}, {1}}

179
{{}, {1, 2, 3}, {2, 3}, {1}, {1, 2}}
{{}, {1, 2, 3}, {3}, {2}, {1, 2}}

180
{{3}, {2, 3}, {1}, {1, 2}}
{{3}, {1}}

181
{{}, {3}, {2, 3}, {1}, {1, 2}}
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {2}, {1, 2}}

182
```

```
{{1, 2, 3}, {3}, {2, 3}, {1}, {1, 2}}
{{1, 2, 3}, {3}, {1}}

183
{{}, {1, 2, 3}, {3}, {2, 3}, {1}, {1, 2}}
{{}, {1, 3}, {2, 3}, {2}, {1, 2}}

184
{{1, 3}, {2, 3}, {1}, {1, 2}}
{{2, 3}, {1}}

185
{{}, {1, 3}, {2, 3}, {1}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2}, {1, 2}}

186
{{1, 2, 3}, {1, 3}, {2, 3}, {1}, {1, 2}}
{{1, 2, 3}, {2, 3}, {1}}

187
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1}, {1, 2}}
{{}, {3}, {1, 3}, {2}, {1, 2}}

188
{{3}, {1, 3}, {2, 3}, {1}, {1, 2}}
{{1, 2, 3}, {3}, {1, 3}, {1}}

189
{{}, {3}, {1, 3}, {2, 3}, {1}, {1, 2}}
{{}, {2, 3}, {2}, {1, 2}}

190
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {1, 2}}
{{3}, {1, 3}, {1}}

191
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {1, 2}}
{{}, {1, 2, 3}, {2, 3}, {2}, {1, 2}}

192
{{2}, {1, 2}}
```

```
{{2, 3}, {2}}

193
{{}, {2}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {1}, {1, 2}}

194
{{1, 2, 3}, {2}, {1, 2}}
{{1, 2, 3}, {2, 3}, {2}}

195
{{}, {1, 2, 3}, {2}, {1, 2}}
{{}, {3}, {1, 3}, {1}, {1, 2}}

196
{{3}, {2}, {1, 2}}
{{1, 2, 3}, {3}, {1, 3}, {2}}

197
{{}, {3}, {2}, {1, 2}}
{{}, {2, 3}, {1}, {1, 2}}

198
{{1, 2, 3}, {3}, {2}, {1, 2}}
{{3}, {1, 3}, {2}}

199
{{}, {1, 2, 3}, {3}, {2}, {1, 2}}
{{}, {1, 2, 3}, {2, 3}, {1}, {1, 2}}

200
{{1, 3}, {2}, {1, 2}}
{{1, 2, 3}, {1, 3}, {2, 3}, {2}}

201
{{}, {1, 3}, {2}, {1, 2}}
{{}, {3}, {1}, {1, 2}}

202
{{1, 2, 3}, {1, 3}, {2}, {1, 2}}
{{1, 3}, {2, 3}, {2}}
```

```
203
{{}, {1, 2, 3}, {1, 3}, {2}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1}, {1, 2}}

204
{{3}, {1, 3}, {2}, {1, 2}}
{{3}, {2}}

205
{{}, {3}, {1, 3}, {2}, {1, 2}}
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1}, {1, 2}}

206
{{1, 2, 3}, {3}, {1, 3}, {2}, {1, 2}}
{{1, 2, 3}, {3}, {2}}

207
{{}, {1, 2, 3}, {3}, {1, 3}, {2}, {1, 2}}
{{}, {1, 3}, {2, 3}, {1}, {1, 2}}

208
{{2, 3}, {2}, {1, 2}}
{{1, 2, 3}, {2}}

209
{{}, {2, 3}, {2}, {1, 2}}
{{}, {3}, {1, 3}, {2, 3}, {1}, {1, 2}}

210
{{1, 2, 3}, {2, 3}, {2}, {1, 2}}
{{2}}

211
{{}, {1, 2, 3}, {2, 3}, {2}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {1, 2}}

212
{{3}, {2, 3}, {2}, {1, 2}}
{{3}, {1, 3}, {2, 3}, {2}}
```

```
213
{{}, {3}, {2, 3}, {2}, {1, 2}}
{{}, {1, 2, 3}, {1}, {1, 2}}

214
{{1, 2, 3}, {3}, {2, 3}, {2}, {1, 2}}
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {2}}

215
{{}, {1, 2, 3}, {3}, {2, 3}, {2}, {1, 2}}
{{}, {1}, {1, 2}}

216
{{1, 3}, {2, 3}, {2}, {1, 2}}
{{1, 3}, {2}}

217
{{}, {1, 3}, {2, 3}, {2}, {1, 2}}
{{}, {1, 2, 3}, {3}, {2, 3}, {1}, {1, 2}}

218
{{1, 2, 3}, {1, 3}, {2, 3}, {2}, {1, 2}}
{{1, 2, 3}, {1, 3}, {2}}

219
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {2}, {1, 2}}
{{}, {3}, {2, 3}, {1}, {1, 2}}

220
{{3}, {1, 3}, {2, 3}, {2}, {1, 2}}
{{1, 2, 3}, {3}, {2, 3}, {2}}

221
{{}, {3}, {1, 3}, {2, 3}, {2}, {1, 2}}
{{}, {1, 3}, {1}, {1, 2}}

222
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {2}, {1, 2}}
{{3}, {2, 3}, {2}}

223
```

```
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {2}, {1, 2}}
{{}, {1, 2, 3}, {1, 3}, {1}, {1, 2}}

224
{{1}, {2}, {1, 2}}
{{1, 2, 3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}

225
{{}, {1}, {2}, {1, 2}}
{{}, {3}}

226
{{1, 2, 3}, {1}, {2}, {1, 2}}
{{1, 3}, {2, 3}, {1}, {2}, {1, 2}}

227
{{}, {1, 2, 3}, {1}, {2}, {1, 2}}
{{}, {1, 2, 3}, {3}}

228
{{3}, {1}, {2}, {1, 2}}
{{3}, {1}, {2}, {1, 2}}

229
{{}, {3}, {1}, {2}, {1, 2}}
{{}, {1, 2, 3}, {1, 3}, {2, 3}}

230
{{1, 2, 3}, {3}, {1}, {2}, {1, 2}}
{{1, 2, 3}, {3}, {1}, {2}, {1, 2}}

231
{{}, {1, 2, 3}, {3}, {1}, {2}, {1, 2}}
{{}, {1, 3}, {2, 3}}

232
{{1, 3}, {1}, {2}, {1, 2}}
{{2, 3}, {1}, {2}, {1, 2}}

233
{{}, {1, 3}, {1}, {2}, {1, 2}}
```

```
{{}, {1, 2, 3}, {3}, {1, 3}}

234
{{1, 2, 3}, {1, 3}, {1}, {2}, {1, 2}}
{{1, 2, 3}, {2, 3}, {1}, {2}, {1, 2}}

235
{{}, {1, 2, 3}, {1, 3}, {1}, {2}, {1, 2}}
{{}, {3}, {1, 3}}

236
{{3}, {1, 3}, {1}, {2}, {1, 2}}
{{1, 2, 3}, {3}, {1, 3}, {1}, {2}, {1, 2}}

237
{{}, {3}, {1, 3}, {1}, {2}, {1, 2}}
{{}, {2, 3}}

238
{{1, 2, 3}, {3}, {1, 3}, {1}, {2}, {1, 2}}
{{3}, {1, 3}, {1}, {2}, {1, 2}}

239
{{}, {1, 2, 3}, {3}, {1, 3}, {1}, {2}, {1, 2}}
{{}, {1, 2, 3}, {2, 3}}

240
{{2, 3}, {1}, {2}, {1, 2}}
{{1, 3}, {1}, {2}, {1, 2}}

241
{{}, {2, 3}, {1}, {2}, {1, 2}}
{{}, {1, 2, 3}, {3}, {2, 3}}

242
{{1, 2, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{1, 2, 3}, {1, 3}, {1}, {2}, {1, 2}}

243
{{}, {1, 2, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{}, {3}, {2, 3}}
```

```
244
{{3}, {2, 3}, {1}, {2}, {1, 2}}
{{1, 2, 3}, {3}, {2, 3}, {1}, {2}, {1, 2}}

245
{{}, {3}, {2, 3}, {1}, {2}, {1, 2}}
{{}, {1, 3}}

246
{{1, 2, 3}, {3}, {2, 3}, {1}, {2}, {1, 2}}
{{3}, {2, 3}, {1}, {2}, {1, 2}}

247
{{}, {1, 2, 3}, {3}, {2, 3}, {1}, {2}, {1, 2}}
{{}, {1, 2, 3}, {1, 3}}

248
{{1, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{1, 2, 3}, {1}, {2}, {1, 2}}

249
{{}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{}, {3}, {1, 3}, {2, 3}}

250
{{1, 2, 3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{1}, {2}, {1, 2}}

251
{{}, {1, 2, 3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}}

252
{{3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}

253
{{}, {3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{}, {1, 2, 3}}
```

```
254
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}

255
{{}, {1, 2, 3}, {3}, {1, 3}, {2, 3}, {1}, {2}, {1, 2}}
{{}}
```

BIBLIOGRAPHIE

M. Abramowitz et I. A. Stegun. 1974. "Handbook of Mathematical Functions", *National Bureau of Standards*, Dover, New York.

P. Camion. 1960. "Une méthode de résolution par l'algèbre de Boole des problèmes combinatoires où interviennent des entiers", *Cahier centre d'études rech. opér.* 2, p. 234-289.

L. E. Dickson. 1971. "History of the theory of numbers", *Dover Publications*, New York, vol. 1.

P. Fermat. 1640. "Oeuvres de Fermat" (1894), Paris, 2, p. 198.

N. J. Fine. 1946. "Binomial coefficients modulo a prime", *Amer. Math. Monthly 54*, p. 589-592.

A. O. Guelfond. 1963. "Calcul des différences finies", *Coll. Univ. de mathématiques*, vol. 12, Dunod, Paris.

A. Goupil. 2003. "Mathématiques interactives avec Maple", *Groupe modulo.*

G. H. Hardy. 1960. "An introduction to the Theory of numbers", *Oxford Univ. Press, Fourth Edition.*

G. Labelle. 1971. "Tableaux Propositionnels", *Bulletin de l'A.M.Q.*, vol. XII, 4, p.53-60.

G. Labelle. 1978. "Sur la décomposition des opérations en bits et en pits", *Ann. Sc. Math. Québec*, vol. II, no. 2, p. 289-304.

A. M. Legendre. 1808. "Théorie des nombres", edition 2, p. 8.

E. Lucas. 1878. "Sur les congruences des nombres eulériens et des coefficients différentiels", *Bull. Soc. Math. de France*, 6, p. 49-54.

E. Kummer. 1878. *Jour. für Math. 44*, p. 115-116.

H. M. Sheffer. 1913. "A set of five independent postulates for Boolean algebras, with application to logical constants", *Transactions of the American Mathematical Society 14*, p. 481-488.