Thèse

en vue de l'obtention du grade de

Docteur de l'École Normale Supérieure de Lyon – Université de Lyon

Spécialité : informatique

Laboratoire de l'Informatique du Parallélisme

École Doctorale Informatique et Mathématiques

présentée et soutenue publiquement le 29 novembre 2012 par

Bruno Grenet

Représentations des polynômes, algorithmes et bornes inférieures

Directeur de thèse :	Pascal Koiran	
Co-directrice :	Natacha Portier	
Après avis de :	Nitin Saxena	
	Éric Schost	
Devant la commision d'examen formée de :		
	Jean-Guillaume Dumas	Membre
	Arnaud Durand	Membre
	Pascal Koiran	Directeur
	Claire Mathieu	Présidente
	Natacha Portier	Co-directrice
	Nitin Saxena	Rapporteur

Titre : Représentations des polynômes, algorithmes et bornes inférieures

Résumé : La complexité algorithmique est l'étude des ressources nécessaires — le temps, la mémoire, ... — pour résoudre un problème de manière algorithmique. Dans ce cadre, la théorie de la complexité algébrique est l'étude de la complexité algorithmique de problèmes de nature algébrique, concernant des polynômes. Dans cette thèse, nous étudions différents aspects de la complexité algébrique.

D'une part, nous nous intéressons à l'expressivité des déterminants de matrices comme représentations des polynômes dans le modèle de complexité de Valiant. Nous montrons que les matrices symétriques ont la même expressivité que les matrices quelconques dès que la caractéristique du corps est différente de deux, mais que ce n'est plus le cas en caractéristique deux. Nous construisons également la représentation la plus compacte connue du permanent par un déterminant.

D'autre part, nous étudions la complexité algorithmique de problèmes algébriques. Nous montrons que la détection de racines dans un système de n polynômes homogènes à n variables est NP-difficile. En lien avec la question « $\mathsf{VP} = \mathsf{VNP}$? », version algébrique de « $\mathsf{P} = \mathsf{NP}$? », nous obtenons une borne inférieure pour le calcul du permanent d'une matrice par un circuit arithmétique, et nous exhibons des liens unissant ce problème et celui du test d'identité polynomiale. Enfin nous fournissons des algorithmes efficaces pour la factorisation des polynômes lacunaires à deux variables.

Mots-clés : complexité algébrique, déterminant, permanent, théorie de Valiant, bornes inférieures, corps finis, test d'identité polynomiale, factorisation, circuits arithmétiques, systèmes polynomiaux.

Title: Representations of polynomials, algorithms and lower bounds

Abstract: Computational complexity is the study of the resources – time, memory, ... – needed to algorithmically solve a problem. Within these settings, algebraic complexity theory is the study of the computational complexity of problems of algebraic nature, concerning polynomials. In this thesis, we study several aspects of algebraic complexity.

On the one hand, we are interested in the expressiveness of the determinants of matrices as representations of polynomials in Valiant's model of complexity. We show that symmetric matrices have the same expressiveness as the ordinary matrices as soon as the characteristic of the underlying field in different from two, but that this is not the case anymore in characteristic two. We also build the smallest known representation of the permanent by a determinant.

On the other hand, we study the computational complexity of algebraic problems. We show that the detection of roots in a system of n homogeneous polynomials in n variables in NP-hard. In line with the "VP = VNP?" question, which is the algebraic version of "P = NP?", we obtain a lower bound for the computation of the permanent of a matrix by an arithmetic circuit, and we point out the links between this problem and the polynomial identity testing problem. Finally, we give efficient algorithms for the factorization of lacunary bivariate polynomials.

Keywords: algebraic complexity, determinant, permanent, Valiant's theory, lower bounds, finite fields, polynomial identity testing, factorization, arithmetic circuits, polynomial systems.

Remerciements

Mes deux rapporteurs, Éric Schost et Nitin Saxena, ont accepté de lire en détail ce manuscrit. Leurs remarques m'ont apporté un éclairage nouveau sur mes propres travaux, et leurs conseils me guideront dans les années à venir. Je tiens à les en remercier chaleureusement. Je suis également reconnaissant envers Claire Mathieu pour avoir accepté de présider mon jury de soutenance, et Jean-Guillaume Dumas et Arnaud Durand pour y avoir pris part. Recueillir l'avis de spécialistes couvrant un large spectre de l'informatique théorique a été très enrichissant. Pascal et Natacha, merci infiniment d'avoir accepté d'encadrer ma thèse alors que votre départ au Canada ne facilitait pas les choses. Je souhaite à tous les thésards de pouvoir bénéficier d'un encadrement comme le vôtre.

En trois années passées au sein du LIP à l'École Normale Supérieure de Lyon et du Department of Computer Science de l'Université de Toronto, j'ai eu la chance de rencontrer de nombreuses personnes. Je pense en particulier à mes quelques co-bureaux, par ordre d'apparition : Irénée, Santiago, Marcelo, Carlos, Lee, Yu, Marek, Kévin, Mathilde, Adrien, Yann, Sébastien, Théophile, Pierre et Aurélie. Je tiens à remercier tous les membres du LIP, et en particulier l'équipe MC2, pour les discussions autour d'un café ou d'une bière. Many thanks are due to the members of the Theory Group of the Department of Computer Science at the University of Toronto, and in particular to Yuval, Kaveh and Dai, who made my stay in Toronto so enjoyable. Before I switch back to French, I remember how lucky I was to meet Avner, his smile and his pairs of shorts. Mes coauteurs m'ont apporté leur vision des sujets abordés et de la recherche en informatique théorique plus généralement. Merci donc à Erich Kaltofen, Stéphan Thomassé, Thierry Monteil, Yann Strozecki et Arkadev Chattopadhyay.

Régulièrement, je me suis appuyé sur le service d'assistantes du LIP pour toute sorte de tâches administratives. Un immense merci à Marie pour les réponses aux innombrables questions que je lui ai posées et pour l'organisation de mes missions, à Damien pour avoir atténué autant qu'il le pouvait la lourdeur des procédures d'inscription et de réinscription en thèse, et à Catherine, Évelyne, Laetitia, Sèverine et Sylvie pour avoir toutes, à un moment donné, su répondre à mes demandes. Avant de commencer ma thèse, j'ai eu le plaisir de découvrir l'informatique, et plus particulièrement

l'informatique théorique, au département informatique de l'ÉNS Lyon. Pour tout ce qu'ils m'ont appris, je souhaite remercier autant mes enseignants — Daniel, Anne, Florent, Éric (×2), Yves, Nicolas, Guillaume, Sylvain (×2), Damien, Victor, ... — que mes camarades — BoolS, Pascal, Mathilde, ...

Comme une thèse ne se résume pas au temps passé au labo, je souhaite remercier tous ceux qui m'ont accompagné durant ces trois années. À l'ÉNS Lyon, j'ai toujours pu compter sur les Zébus. Merci à Pierre, Fifi, Anto, Tom, Ju, Bleue, AnSo, Olive, Marion, Adrien, Bapt, Marc, Fred pour les officiels, mais également à Charlène, Tamara et Lise. Merci en particulier à Pierre et Charlène de m'avoir permis d'éviter la visite des ponts lyonnais. Il m'est parfois arrivé de quitter Gerland. Merci à Théo, Delph, Maëlle, Nico, Antoine, Gaston, Soline, Margot, Beune, et tous les autres pour les moments passés ensemble. Et parce qu'ils ne m'ont pas demandé si souvent que cela à quoi servait ma thèse, je remercie mes parents, mes frangins, mes belles-sœurs et bien sûr Arthur et Alice !

Il y aurait mille raisons de te remercier, Laurie. Pour garder un rapport avec ma thèse, ta décision de venir passer six mois à Toronto a été le plus beau cadeau que tu pouvais me faire.

Table des matières

Remerciements i

Table des matières iii

Introduction vii

1 Prolégomènes 1
- 1.1 Polynômes, déterminants et graphes . . . 1
  - 1.1.1 Polynômes . . . 1
  - 1.1.2 Déterminants . . . 2
  - 1.1.3 Graphes . . . 3
- 1.2 Représenter les polynômes . . . 4
  - 1.2.1 Représentations dense, creuse et lacunaire . . . 4
  - 1.2.2 Les circuits arithmétiques . . . 6
  - 1.2.3 Les programmes à branchements . . . 10
- 1.3 Modèles de calcul et complexité . . . 11
  - 1.3.1 Le modèle booléen . . . 11
  - 1.3.2 Le modèle de Valiant . . . 13

I Résolution des systèmes polynomiaux 17

2 Complexité du résultant multivarié 19
- 2.1 Complexité du résultant en caractéristique nulle . . . 22
  - 2.1.1 Borne supérieure . . . 22
  - 2.1.2 Borne inférieure . . . 23
- 2.2 NP-difficulté en caractéristique quelconque . . . 26
  - 2.2.1 Une réduction probabiliste . . . 27
  - 2.2.2 Deux réductions déterministes . . . 28
- 2.3 Matrices de Macaulay . . . 33
  - 2.3.1 Représentation des matrices de Macaulay . . . 34
  - 2.3.2 Déterminant d'une matrice représentée par un circuit 36

II Représentations déterminantielles de polynômes **39**

3 Complexité déterminantielle **41**
3.1 Taille réduite . . . 42
3.2 Circuits et programmes à branchements . . . 44
3.2.1 Formules et programmes à branchements . . . 44
3.2.2 Circuits asymétriques . . . 46
3.3 Complexité du déterminant . . . 52
3.3.1 Expressivité du déterminant . . . 53
3.3.2 Un programme à branchements pour le déterminant . 54
3.4 Complexité déterminantielle du permanent . . . 60

4 Représentations symétriques **63**
4.1 Représentation symétrique des programmes à branchements 64
4.2 Comparaisons avec des résultats existants . . . 68

5 Représentations symétriques en caractéristique deux **73**
5.1 Prérequis algébrique . . . 74
5.1.1 Polynômes, déterminants et graphes en caractéristique 2 74
5.1.2 Anneaux quotient . . . 75
5.2 Polynômes représentables . . . 76
5.3 Obstructions aux représentations . . . 80
5.3.1 Condition nécessaire . . . 80
5.3.2 Exemple . . . 86
5.3.3 Polynômes multilinéaires . . . 86
5.3.4 Vers une caractérisation complète ? . . . 87
5.4 Représentation et factorisation des polynômes multilinéaires 89
5.4.1 Résultats préliminaires . . . 89
5.4.2 Test de factorisabilité . . . 94
5.4.3 Algorithme de représentation . . . 96
5.5 Représentations déterminantielles alternées . . . 98

III Polynômes de type creux **101**

6 Autour de la τ-conjecture réelle **103**
6.1 La τ-conjecture réelle . . . 103
6.1.1 Travaux existants . . . 106
6.1.2 Notre approche . . . 107
6.2 Racines réelles des sommes de produits de polynômes creux 109
6.2.1 Définitions . . . 109
6.2.2 Une généralisation de la règle de Descartes . . . 111
6.2.3 Affinement de l'analyse . . . 114
6.3 Borne inférieure pour le permanent . . . 116

6.4 Tests d'identité polynomiale 118
6.5 Conclusion 121

7 Factorisation des polynômes lacunaires à deux variables 123
7.1 Valuation et théorème de lacune 125
7.1.1 Preuve du théorème 7.1 126
7.1.2 Des améliorations possibles ? 130
7.1.3 Un théorème de lacune 133
7.2 Algorithmes 134
7.3 Généralisations 138
7.3.1 Une borne générale sur la valuation 138
7.3.2 Généralisations des algorithmes 141
7.4 Caractéristique positive 143

Bibliographie 147

Introduction

La complexité algorithmique est l'étude des ressources nécessaires — le temps, la mémoire, ... — pour résoudre un problème de manière algorithmique. Quoique très ancienne, la notion d'algorithme n'a été formalisée que dans les années 1930. Différents modèles ont été proposés, notamment par Alan M. Turing, Alonzo Church, Stephen C. Kleene, Emil L. Post, Jacques Herbrand et Kurt Gödel. Bien que de natures très différentes, ces modèles se sont trouvés équivalents : ils sont capables d'exprimer exactement les mêmes problèmes. Cette équivalence a renforcé la conviction que ces modèles capturent bien la notion d'algorithme. Ce postulat, sur lequel se fonde l'informatique théorique, est appelé *thèse de Church* ou *thèse de Church-Turing*.

La question de l'efficacité des algorithmes, et plus précisément la formalisation de la notion d'algorithme *efficace*, est venue bien plus tard. Juris Hartmanis et Richard E. Stearns ont introduit dans les années 1960 l'idée que l'efficacité d'un algorithme doit être mesurée en fonction de la taille de son entrée. Leur article est considéré comme fondateur de la théorie de la complexité [1]. Il y a deux aspects dans cette théorie. D'une part, il s'agit de déterminer quel est le meilleur algorithme possible — le plus rapide, celui utilisant le moins de mémoire, ... — pour résoudre un problème donné. On peut par exemple se demander si l'algorithme appris à l'école pour effectuer la multiplication de deux nombres entiers est le plus efficace possible — la réponse est non. D'autre part, on peut renverser le point de vue et classifier les problèmes en fonction de leur complexité en regroupant ceux qui nécessitent les mêmes ressources pour être résolus. Alors qu'il semble intuitif qu'additionner deux nombres entiers est une tâche plus facile que de les multiplier, il n'existe à l'heure actuelle aucune preuve de ce fait. Par contre, on sait qu'effectuer une division requiert le même temps de calcul qu'une multiplication, allant à l'encontre de l'intuition.

Pour la classification des problèmes algorithmiques, Alan Cobham et Jack Edmonds ont indépendamment proposé dans les années 1960 de considérer qu'un algorithme efficace est un algorithme dont le temps de calcul est une fonction polynomiale de la taille de l'entrée, menant à la définition de la

1. Le titre de leur article, *On the computational complexity of algorithms*, « De la complexité calculatoire des algorithmes », a d'ailleurs donné son nom au domaine.

classe P des problèmes admettant un algorithme polynomial. Par exemple, trouver l'itinéraire le plus court entre deux villes données sur une carte est un problème de la classe P. Dans une certaine mesure, la classe P est toujours considérée comme la classe des problèmes que l'on sait résoudre efficacement, même si d'autres classes peuvent revendiquer ce titre, comme la classe BPP des problèmes admettant un algorithme probabiliste polynomial [2], ou la classe BQP dans cadre du calcul quantique.

Parmi les problèmes pour lesquels on ne connaît pas d'algorithme polynomial, certains ont la propriété d'être *vérifiables* en temps polynomial : si une solution est fournie, on peut vérifier que cette solution est valide en temps polynomial. Cette notion conduit à la définition de la classe NP [3] des problèmes facilement vérifiables. Par exemple, étant donné un certain nombre de villes sur une carte, déterminer s'il existe un trajet de moins de mille kilomètres passant par chacune de ces villes est un problème de la classe NP. En effet, si un trajet est fourni, il est aisé de vérifier s'il passe par chacune des villes et s'il fait moins de mille kilomètres. D'autre part, tout problème de la classe P est également dans la classe NP puisque si l'on peut trouver une solution, on peut également vérifier une solution. Au début des années 1970, Stephen A. Cook au Canada et Leonid A. Levin en URSS ont indépendamment identifié une sous-classe de NP formée des problèmes *les plus difficiles* à résoudre. Si l'un de ces problèmes, qui sont dits NP-complets, admet un algorithme polynomial alors tous les problèmes de la classe NP en admettent un. C'est la fameuse question « P = NP ? » dont la solution a été mise à prix à un million de dollars par l'institut Clay de mathématiques. Les spécialistes du sujet sont convaincus que ces deux classes sont différentes. Cependant, malgré quarante années [4] d'efforts, la solution semble toujours hors de portée.

Dans ce cadre, la théorie de la complexité algébrique est l'étude de la complexité de problèmes de nature algébrique. En particulier, de nombreuses questions de complexité se posent lorsque l'on souhaite manipuler des polynômes. Naturellement, la question de la représentation des polynômes est primordiale. D'un côté, la complexité de problèmes comme le calcul des racines d'un polynôme — c'est-à-dire les valeurs pour lesquelles le polynôme s'annule — dépend de manière essentielle de la représentation du polynôme en entrée. D'un autre côté, on peut voir le polynôme comme étant le problème en lui-même, et la taille de sa plus petite représentation comme étant sa complexité. Leslie G. Valiant a introduit en 1979 un modèle de calcul, aujourd'hui connu sous le nom de *modèle de Valiant*, pour étudier ce genre de questions. En particulier, il a défini des analogues des classes P et NP,

2. De nombreux spécialistes sont cependant persuadés que les classes BPP et P coïncident.

3. Les initiales NP signifient *Nondeterministic Polynomial time* en anglais.

4. En réalité, la question avait déjà été posée en des termes très proches par Kurt Gödel dans une lettre adressée à John von Neumann en 1956, mais cette lettre ne fut découverte qu'en 1988.

notées VP et VNP, respectivement constituées des polynômes admettant une représentation de taille polynomiale, et des polynômes admettant une description (implicite) de taille polynomiale. On parle de modèle de Valiant pour ce modèle de calcul, en opposition au modèle booléen classique. Il se pose alors la même question « VP = VNP ? » que dans le cadre booléen. Au delà de l'étude des problèmes de nature algébrique, la raison d'être de ce modèle est de proposer une approche pour résoudre la question « P = NP ? ». L'espoir est de pouvoir utiliser la simplicité de ce modèle et la structure riche des anneaux de polynômes pour résoudre la question « VP = VNP ? ». Une réponse à cette question, que l'on pense également négative, donnerait un éclairage nouveau sur la question « P = NP ? ». Cependant, comme souvent en complexité, cette question n'a pas encore été résolue.

Dans cette thèse, nous étudions les différents aspects de la complexité algébrique. En particulier, nous nous intéressons aussi bien au problème de la représentation des polynômes dans le modèle de Valiant qu'à la complexité algorithmique de problèmes concernant les polynômes — détection de racines, test de nullité, factorisation, ... De plus, nous manipulons les polynômes sous plusieurs formes, qu'il s'agisse des représentations dense, creuse ou lacunaire comme des représentations sous forme de circuits de différents types.

Le premier chapitre est un rappel des notions utilisées dans la suite. Les notions les plus classiques de complexité booléennes sont très rapidement survolées. Le vocabulaire et les définitions de complexité algébrique sont donnés avec un peu plus de détail puisqu'elles sont sans doute un peu moins connues. Ce chapitre ne contient quasiment que des définitions. De nombreuses illustrations le parsèment pour tenter de le rendre un peu moins aride. La suite du manuscrit est découpée en trois grandes parties.

La première partie contient un unique chapitre. Il y est question de la complexité du calcul du résultant. En particulier, nous montrons que pour tout corps décider si un système de n polynômes homogènes à n variables admet une racine non triviale est un problème NP-difficile. Il est intéressant de noter que ce résultat ne fait pas appel aux réductions probabilistes même pour les corps finis, contrairement à un grand nombre de résultats du même acabit.

La deuxième partie est constituée de trois chapitres, et concerne les représentations déterminantielles de polynômes, c'est-à-dire les représentations des polynômes par des déterminants. Le chapitre 3 se veut une introduction à ce sujet. En particulier, on passe en revue les liens entre les représentations déterminantielles des polynômes et les représentations par circuits ou programmes à branchements. Nous faisons une présentation unifiée de ces résultats qui n'existe pas à ce jour dans la littérature, et donnons pour plusieurs résultats de nouvelles preuves qui améliorent légèrement les bornes connues. Enfin, nous étudions la complexité déterminantielle du permanent

en donnant une nouvelle borne supérieure pour ce problème. Dans les chapitres 4 et 5, nous nous intéressons aux représentations des polynômes par des déterminants de matrices symétriques. Dans le chapitre 4, nous montrons que les déterminants de matrices symétriques sont aussi expressifs que les déterminants de matrices quelconque lorsque que la caractéristique du corps est différente de deux. En particulier, nous donnons des constructions de matrices symétriques de dimension polynomiale pour représenter certains polynômes donnés sous forme de circuit. On en déduit qu'on peut transformer tout déterminant en un déterminant de matrice symétrique en dimension polynomiale. Dans le chapitre 5, nous montrons que le panorama est tout à fait différent lorsque la caractéristique du corps est deux. En particulier, certains polynômes n'admettent dans ce cas aucune représentation déterminantielle symétrique. Pour prouver ce fait, nous montrons une équivalence avec le fait de pouvoir factoriser le polynôme dans certains anneaux quotient. Nous développons ensuite des algorithmes de factorisation pour ces anneaux quotients, qui nous permettent *in fine* d'obtenir des algorithmes pour trouver des représentations déterminantielles symétriques lorsqu'il en existe.

La troisième partie, constituée de deux chapitres, traite de problèmes algorithmiques dont les entrées sont des polynômes de type creux. Dans le chapitre 6, on donne une borne supérieure sur le nombre de racines réelles de polynômes donnés sous la forme de sommes de produits de polynômes creux. Cette borne nous permet de déduire d'une part un algorithme polynomial de test d'identité polynomiale pour des polynômes de cette forme, et d'autre part une borne inférieure pour la taille de la plus petite représentation pour le permanent. Ces résultats fournissent une nouvelle illustration des liens profonds qui unissent ces deux problèmes. Dans le chapitre 7, on étudie la complexité de la factorisation des polynômes à deux variables représentés sous forme lacunaire. On donne des algorithmes polynomiaux permettant de trouver les facteurs linéaires de tels polynômes. En caractéristique positive, on ne trouve que certains facteurs. Cependant on montre que les facteurs que l'on trouve sont tous ceux qui peuvent l'être en temps polynomial si $\mathsf{P} \neq \mathsf{NP}$. En d'autres termes, on identifie finement la limite pour ce problème entre ce qui est faisable *efficacement* et ce qui ne l'est pas.

Chapitre 1

Prolégomènes

Dans ce chapitre, nous introduisons les notions utiles pour la compréhension de cette thèse. Ces notions sont classiques pour la plupart à l'exception de certaines définitions concernant les circuits arithmétiques et le modèle de Valiant. Ce chapitre n'a pas la prétention de constituer une introduction aux domaines abordés – complexité booléenne, complexité algébrique à la Valiant – mais plutôt un rappel des notions les plus importantes et une présentation de quelques notions peut-être moins centrales mais récurrentes dans la suite. Il est sans doute plus intéressant de ne pas lire en détail ce chapitre mais d'y revenir lorsqu'une définition fait défaut au lecteur.

Dans une première partie, nous fixons un peu de vocabulaire concernant les polynômes, les matrices et leurs déterminants, ainsi que les graphes. Nous introduisons ensuite différentes façons de représenter les polynômes. Enfin, nous présentons très brièvement la complexité booléenne et avec un peu plus de détail la complexité algébrique à la Valiant.

1.1 Polynômes, déterminants et graphes

1.1.1 Polynômes

Soit f un polynôme à une variable sur un anneau $\mathbb{A}$, c'est-à-dire une expression de la forme

$$f(X) = c_0 + c_1 X + \cdots + c_d X^d = \sum_{i=0}^{d} c_i X^i$$

où $c_i \in \mathbb{A}$ pour tout i et $c_d \neq 0$. L'entier d est appelé le *degré* du polynôme et noté $\deg(f)$. Le polynôme est une somme de *termes* de la forme $c_i X^i$. Un terme est le produit d'un *coefficient* c_i et d'un *monôme* X^i. L'ensemble des polynômes à une variable sur l'anneau $\mathbb{A}$ est noté $\mathbb{A}[X]$.

Soit maintenant f un polynôme à n variables $X_1, \ldots, X_n$ sur un anneau $\mathbb{A}$:

$$f(X_1, \ldots, X_n) = \sum_{\alpha \in \{0,\ldots,d\}^n} c_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}$$

où $\alpha = (\alpha_1, \ldots, \alpha_n)$ et $c_\alpha \in \mathbb{A}$ pour tout n-uplet α. Le *degré total* du polynôme est

$$\deg(f) = \max \{\alpha_1 + \cdots + \alpha_n : c_\alpha \neq 0\}$$

et le *degré de f en X_j* est la plus grande puissance de X_j qui apparaît dans f. Formellement,

$$\deg_{X_j}(f) = \max \left\{\alpha_j : \exists(\alpha_1, \ldots, \alpha_{j-1}, \alpha_{j+1}, \ldots, \alpha_n), c_\alpha \neq 0\right\}.$$

Un polynôme est dit *multilinéaire* lorsque son degré en chacune de ses variables est au plus 1. L'ensemble des polynômes à n variables $X_1, \ldots, X_n$ sur l'anneau $\mathbb{A}$ est noté $\mathbb{A}[X_1, \ldots, X_n]$.

On note que pour tout n, $\mathbb{A}[X_1, \ldots, X_n]$ est un anneau, et qu'il est isomorphe à $\mathbb{A}[X_1, \ldots, X_{n-1}][X_n]$.

Si $\mathbb{A}$ est factoriel, une *fraction rationnelle* sur $\mathbb{A}$ est une expression de la forme $\phi = f/g$ où f et g sont deux polynômes sur $\mathbb{A}$. Le degré de ϕ est $\deg(\phi) = \deg(f) - \deg(g)$. On peut toujours supposer que ϕ est sous forme irréductible, c'est-à-dire que f et g sont premiers entre eux.

Une *racine* d'un polynôme f est un élément $a \in \mathbb{A}^n$ (où n est le nombre de variables de f) tel que $f(a) = 0$. Une racine est dite *non triviale* si $a \neq \bar{0}$. Si f est un polynôme à une variable, la multiplicité μ d'une racine a de f est l'exposant maximal tel que $(X - a)^\mu$ divise f. Une racine d'une fraction rationnelle ϕ est un élément a tel que $\phi(a) = 0$, et un *pôle* de ϕ est un élément a tel que $\phi(a)$ n'est pas définie. Si $\phi = f/g$ où f et g sont premiers entre eux, les racines de ϕ sont les racines de f et les pôles de ϕ sont les racines de g.

1.1.2 Déterminants

Une matrice $\mathcal{M}$ de dimensions (m, n) sur un anneau $\mathbb{A}$ est un élément de $\mathbb{A}^{m \times n}$. Les coordonnées de $\mathcal{M}$ sont appelées ses *coefficients* et sont indexées par $\{1, \ldots, m\} \times \{1, \ldots, n\}$. On note habituellement

$$\mathcal{M} = (m_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$$

c'est-à-dire que m_{ij} est le coefficient d'indice (i, j) de $\mathcal{M}$. Pour i fixé, l'ensemble des coefficients m_{ij} est la *ligne i* de $\mathcal{M}$, et similairement l'ensemble des m_{ij} pour j fixé est la *colonne j* de $\mathcal{M}$.

Si $\mathcal{M}$ est de dimensions (n, n) pour un certain n, on dit que $\mathcal{M}$ est une *matrice carrée de dimension n*.

Une *permutation* est une bijection d'un ensemble fini dans lui-même, typiquement de $\{1, \dots, n\}$ dans lui-même. On note $\mathfrak{S}_n$ l'ensemble des permutations de $\{1, \dots, n\}$. Le nombre d'*inversions* d'une permutation $\sigma \in \mathfrak{S}_n$ est le nombre de couples (i, j), tels que $i < j$ et $\sigma(i) > \sigma(j)$. La *signature* d'une permutation σ est

$$\epsilon(\sigma) = (-1)^{\text{Nombre d'inversions de } \sigma}.$$

Le déterminant d'une matrice carrée $\mathcal{M} = (m_{ij})_{1 \leq i,j \leq n}$ est

$$\det(\mathcal{M}) = \sum_{\sigma \in \mathfrak{S}_n} \epsilon(\sigma) \prod_{i=1}^{n} m_{i,\sigma(i)}.$$

Le déterminant de $\mathcal{M}$ est donc un polynôme multilinéaire de degré n en ses n^2 coefficients.

Une matrice carrée est dite *symétrique* si pour tout (i, j), $m_{ij} = m_{ji}$.

1.1.3 Graphes

Un graphe orienté $\mathcal{G} = (S, A)$ est la donnée d'un ensemble fini de sommets S et d'un ensemble d'arcs $A \subseteq S \times S$. Un arc (u, v) est souvent noté $u \to v$. On peut considérer une fonction de poids w sur les arcs d'un graphe, qui associe à chaque arc (u, v) un poids $w(u, v)$ dans un ensemble E quelconque, et $w(u, v) = 0$ si $(u, v) \notin A$. Par exemple, pour un anneau $\mathbb{A}$, on peut considérer une fonction de poids $w : A \to \mathbb{A}$.

La *matrice d'adjacence* de $\mathcal{G}$ est la matrice $\mathcal{M}(\mathcal{G})$ définie par $m_{uv} = w(u, v)$ où w est une fonction de poids sur les arcs de $\mathcal{G}$. Si $\mathcal{G}$ n'a pas de fonction de poids explicite, on considère la fonction de poids $w(u, v) = 1$ pour $(u, v) \in A$ et $w(u, v) = 0$ pour $(u, v) \notin A$.

Notons $\{1, \dots, n\}$ les sommets d'un graphe $\mathcal{G}$. Une permutation de $\{1, \dots, n\}$ peut se décomposer en cycles disjoints. De tels cycles correspondent à des cycles éventuels dans $\mathcal{G}$. Les permutations de $\{1, \dots, n\}$ sont donc en bijection avec les *couvertures par cycles* de $\mathcal{G}$, c'est-à-dire les ensembles d'arcs de $\mathcal{G}$ qui forment un ensemble de cycles disjoints et tels que chaque sommet de $\mathcal{G}$ appartienne exactement à un cycle. La figure 1.1 donne un exemple d'une telle couverture avec quatre cycles (colorés). Le *poids d'une couverture par cycles* est le produit des poids des arcs qui la composent. On note $w(C)$ le poids d'une couverture par cycles C.

Soit C une couverture par cycles d'un graphe $\mathcal{G}$, et σ la permutation de $\{1, \dots, n\}$ correspondante. On définit la signature de C par $\epsilon(C) = \epsilon(\sigma)$. Si C possède k cycles, alors $\epsilon(C) = (-1)^{n+k}$. D'autre part, un cycle est dit *pair* s'il possède un nombre pair de sommets, et *impair* sinon. Alors la signature de C peut également être définie par $\epsilon(C) = (-1)^{\ell}$ où ℓ est le nombre de cycles pairs de C.

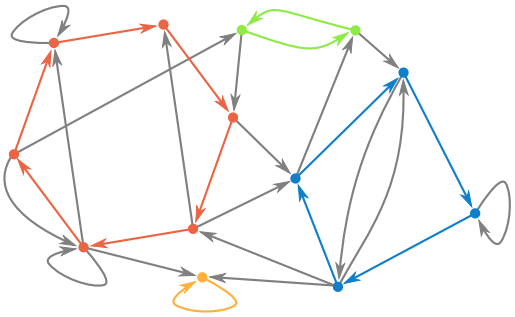


FIGURE 1.1 – Un exemple de couverture par quatre cycles.

Avec ces définitions, on obtient que

$$\det(\mathcal{M}(\mathcal{G})) = \sum_{C} \epsilon(C) w(C)$$

où la somme porte sur toutes les couvertures par cycles de $\mathcal{G}$.

On peut également s'intéresser à des graphes non orientés $\mathcal{G} = (S, A)$. Dans ce cas, A est un ensemble d'*arêtes* de la forme $\{u, v\}$ pour $u, v \in S$. On peut voir un graphe non orienté comme un graphe orienté symétrique. Dans ce cas, on considère l'arête $\{u, v\}$ comme l'union des deux arcs (u, v) et (v, u). Une fonction de poids pour un graphe non orienté doit respecter $w(u, v) = w(v, u)$ pour tout $u, v \in S$. La matrice d'adjacence de $\mathcal{G}$ est donc une matrice symétrique. Quand on parle de couvertures par cycles d'un graphe non orienté, il s'agit toujours de couverture par cycles *orientés* du graphe orienté symétrique correspondant. Ainsi, avec trois sommets u, v et w, on distingue les cycles $u \to v \to w \to u$ et $u \to w \to v \to u$. En d'autres termes, on associe à chaque cycle son sens de parcours.

1.2 Représenter les polynômes

Nous nous intéressons ici aux différentes manières de représenter les polynômes. On fixe un corps $\mathbb{K}$ quelconque, et on s'intéresse aux représentations de polynômes de $\mathbb{K}[X]$ ou $\mathbb{K}[X_1, \ldots, X_n]$.

1.2.1 *Représentations dense, creuse et lacunaire*

Dans cette partie, on s'intéresse aux représentations des polynômes sous forme de listes. Il y a essentiellement deux possibilités. Soit tous les coefficients du polynôme sont représentés, soit seulement les coefficients non nuls sont représentés, mais il faut alors leur adjoindre l'exposant (ou le vecteur d'exposants) correspondant. Selon les situations, l'une ou l'autre de ces représentations est préférable. On s'intéresse à la taille de chacune de ces représentations. La taille est définie en fonction de la taille des coefficients

c du polynôme, notée taille(c). On ne précise pas de valeur pour taille(c) puisqu'elle dépend du corps $\mathbb{K}$ dans lequel vivent les coefficients.

La taille définie ci-après est une approximation du nombre de bits nécessaires pour écrire complètement le polynôme. Nous nous contentons à dessein d'une approximation pour faire un compromis entre simplicité de la définition et précision.

On commence par le cas des polynômes à une variable.

Définition 1.1

La *représentation dense* d'un polynôme $f(X) = c_0 + c_1X + \cdots + c_dX^d$ est la liste $[c_0, c_1, \ldots, c_d]$. La taille de la représentation est $\sum_{i=0}^{d} \text{taille}(c_i)$.

Cette représentation est adaptée lorsque le polynôme est effectivement dense, c'est-à-dire que peu de ses coefficients s'annulent. Cependant, il est clair qu'elle n'est par exemple pas très adaptée au polynôme $(X^{500} - 1)$ puisqu'elle suppose de stocker 501 coefficients.

Définition 1.2

La *représentation creuse* de $f(X) = c_0X^{\delta_0} + c_1X^{\delta_1} + \cdots + c_kX^{\delta_k}$ est la liste $[(c_0, \delta_0), (c_1, \delta_1), \cdots, (c_k, \delta_k)]$. La taille de cette représentation est $\sum_{i=0}^{k}(\text{taille}(c_i) + \log(1 + \delta_i))$.

Le polynôme précédent est représenté de manière efficace en représentation creuse puisqu'il suffit de stocker les deux couples $(1, 500)$ et $(1, 0)$. Cette représentation n'est pas *meilleure* ou plus compacte que la précédente dans le cas général. En effet, pour un polynôme dont aucun coefficient n'est nul, la suite des exposants du polynôme est superflue.

On s'intéresse maintenant aux polynômes à plusieurs variables. Dans ce cas, on n'a plus deux mais trois représentations distinctes utilisées dans la littérature. Le nombre de monômes à n variables de degré au plus d est $\binom{n+d}{d}$. Ainsi, la représentation dense atteint rapidement une taille très importante, même pour des polynômes de petit degré.

Définition 1.3

La *représentation dense* d'un polynôme

$$f(X_1, \ldots, X_n) = \sum_{e \in \{0,\ldots,d\}^n} c_e X_1^{e_1} \cdots X_n^{e_n},$$

où $e = (e_1, \ldots, e_n)$, est la liste de tous les coefficients c_e pour e parcourant $\{0, \ldots, d\}^n$. La taille de la représentation est la somme des tailles de tous les coefficients.

Cette représentation est de taille exponentielle en le degré et le nombre de variables du polynôme. Pour obtenir une représentation de taille polynomiale en ces paramètres, on utilise une représentation à mi-chemin entre les représentations dense et creuse du cas univarié. Cette représentation est

dite creuse malgré sa proximité avec la représentation dense des polynômes à une variable. Cette terminologie peut être source de confusions, mais elle est très largement répandue dans la littérature.

Définition 1.4

La *représentation creuse* d'un polynôme

$$f(X_1, \ldots, X_n) = \sum_{e \in \mathcal{E}} c_e X_1^{e_1} \cdots X_n^{e_n},$$

où $e = (e_1, \ldots, e_n)$ et $\mathcal{E} \subseteq \{0, \ldots, d\}^n$, est la liste de tous les $(n+1)$-uplets $(c_e, 1^{e_1}, \ldots, 1^{e_n})$ pour $e \in \mathcal{E}$. La taille de cette représentation est $\sum_{e \in \mathcal{E}} (\text{taille}(c_e) + |e|)$ où $|e| = \sum_{i=1}^{n} e_i$.

La représentation creuse d'un polynôme à plusieurs variables consiste donc à écrire les exposants en unaire. L'équivalent de la représentation creuse des polynômes à une variable est appelée *supercreuse* ou lacunaire dans le cas des polynômes à plusieurs variables.

Définition 1.5

La *représentation lacunaire* d'un polynôme

$$f(X_1, \ldots, X_n) = \sum_{e \in \mathcal{E}} c_e X_1^{e_1} \cdots X_n^{e_n},$$

où $e = (e_1, \ldots, e_n)$ et $\mathcal{E} \subseteq \{0, \ldots, d\}^n$, est la liste de tous les $(n+1)$-uplets $(c_e, e_1, \ldots, e_n)$ pour $e \in \mathcal{E}$. La taille de cette représentation est $\sum_{e \in \mathcal{E}} (\text{taille}(c_e) + \log(1+e_1) + \cdots + \log(1+e_n))$.

Par abus de langage, on utilisera souvent les termes de polynôme dense, creux ou lacunaire pour parler d'un polynôme représenté sous forme dense, creuse ou lacunaire. De plus, dans le cas des polynômes à une variable, on pourra utiliser le terme de polynôme lacunaire à la place de polynôme creux.

1.2.2 Les circuits arithmétiques

Les représentations de la partie précédente sont toutes des représentations d'un polynôme sous forme développée. Par exemple, les représentations dense et creuse du polynôme $(X+1)^{500}$ sont des listes de taille 500. Il est bien entendu beaucoup plus avantageux de garder le polynôme sous forme factorisée pour le représenter. Pour cela, on utilise la notion de circuit arithmétique.

Définition 1.6

Un *circuit arithmétique* sur le corps $\mathbb{K}$ est un multigraphe orienté sans cycle dont les sommets, appelés *portes*, sont étiquetés. Les portes de degré entrant 0 sont les *entrées* du circuit. Elles sont étiquetées soit par une

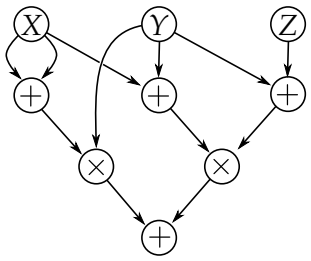


FIGURE 1.2 – Le polynôme $2XY + (X + Y)(Y + Z)$ représenté par un circuit arithmétique.

variable X_i, soit par une constante $c \in \mathbb{K}$. Les autres portes sont de degré entrant 2 et sont les *portes de calcul*. Elles sont de deux types : les *portes d'addition*, étiquetées par $+$, et les *portes de multiplication*, étiquetées par $\times$. Il y a une unique porte de degré sortant 0 que l'on nomme *sortie* du circuit. Les arcs d'un circuit sont appelés *flèches*.

Le polynôme représenté par un circuit arithmétique est défini de manière inductive. Une entrée d'étiquette $c \in \mathbb{K}$ représente le polynôme constant égal à c et une entrée d'étiquette X_i représente le polynôme X_i. Une porte d'addition recevant des flèches de deux portes représentant respectivement des polynômes f et g représente le polynôme $f + g$. De même, une porte de multiplication recevant des flèches de portes représentant respectivement f et g représente le polynôme $f \times g$. Le polynôme représenté par le circuit est le polynôme représenté par sa porte de sortie.

Dans certaines preuves, on utilise une généralisation évidente de cette définition en autorisant plusieurs portes de sortie. Il y a alors autant de polynômes représentés par un tel circuit que de portes de sortie.

L'ensemble des portes β telles qu'il existe une flèche $\gamma \to \beta$ est l'ensemble des *successeurs* de γ. Inversement, si γ est une porte de calcul, les deux portes dont sont issues les flèches dirigées vers γ sont les *arguments* de γ. Ces deux arguments ne sont pas nécessairement distincts.

Le sous-graphe induit par l'ensemble des portes β telles qu'il existe un chemin (orienté) de β vers γ dans $\mathcal{C}$ est le *sous-circuit* associé à γ, noté $\mathcal{C}_\gamma$. Ainsi, le polynôme représenté par une porte γ dans $\mathcal{C}$ est précisément le polynôme représenté par le circuit $\mathcal{C}_\gamma$.

Définition 1.7

Soit $\mathcal{C}$ un circuit arithmétique. Sa *taille* $t(\mathcal{C})$ est son nombre de portes de calcul et sa *profondeur* $p(\mathcal{C})$ est la longueur du plus long chemin reliant une entrée à une sortie. Son nombre d'entrées est noté $e(\mathcal{C})$.

Par exemple, le circuit de la figure 1.2 est de taille 6, de profondeur 3 et a 3 entrées.

La notion de taille d'un circuit peut varier d'un auteur à l'autre. Certains auteurs considèrent par exemple le nombre total de portes (incluant les entrées) comme notion de taille. On rencontre l'appellation *taille fine* ou *complexité* pour $t(\mathcal{C})$, et *taille grossière* pour la quantité $m(\mathcal{C}) = t(\mathcal{C}) + e(\mathcal{C})$. La proposition suivante exprime les liens entre tailles fine et grossière d'un circuit.

Proposition 1.8

Soit $\mathcal{C}$ un circuit dont le graphe sous-jacent est connexe. Alors $1 \leq e(\mathcal{C}) \leq t(\mathcal{C}) + 1$.

Démonstration : On raisonne par induction sur la taille de $\mathcal{C}$. Un circuit connexe de taille 0 est réduit à une entrée, donc $e(\mathcal{C}) = 1 \leq t(\mathcal{C}) + 1$. Soit $\mathcal{C}$ un circuit connexe de taille $t > 0$ et γ une de ses portes de sortie (qui n'est donc pas une entrée). Si $\mathcal{C}' = \mathcal{C} \setminus \{\gamma\}$ est connexe, alors $e(\mathcal{C}) = e(\mathcal{C}') \leq t(\mathcal{C}') + 1 = t(\mathcal{C})$. Si $\mathcal{C} \setminus \{\gamma\}$ est non connexe, soit $\mathcal{C}'$ et $\mathcal{C}''$ ses deux composantes connexes. Par hypothèse d'induction, $e(\mathcal{C}') \leq t(\mathcal{C}') + 1$ et $e(\mathcal{C}'') \leq t(\mathcal{C}'') + 1$. Donc $e(\mathcal{C}) = e(\mathcal{C}') + e(\mathcal{C}'') \leq t(\mathcal{C}') + t(\mathcal{C}'') + 2 = t(\mathcal{C}) + 1$. □

Les circuits arithmétiques sont parfois présentés sous la forme équivalente des programmes sans boucle [1]. On se donne un ensemble d'entrées, constitué de constantes et de variables, ainsi qu'une suite ordonnée d'instructions. Pour $i \geq 1$, l'instruction i est de la forme $j \star k$ où $\star \in \{+, \times\}$ et j et k sont chacun soit une entrée, soit un entier strictement inférieur à i. On définit le polynôme calculé par le programme de manière inductive. Le polynôme calculé par une entrée est l'entrée elle-même. Pour une instruction de la forme $j \star k$ telle que le polynôme calculé par j est f_j et celui calculé par k est f_k, le polynôme calculé est $f_j \star f_k$.

Il est aisé de voir qu'un programme sans boucle avec t instructions et e entrées est simplement un circuit arithmétique de taille t avec e entrées dans lequel on a fixé un ordre d'évaluation des portes.

On peut imposer des restrictions aux circuits pour obtenir des représentations moins compactes mais plus faciles à manipuler.

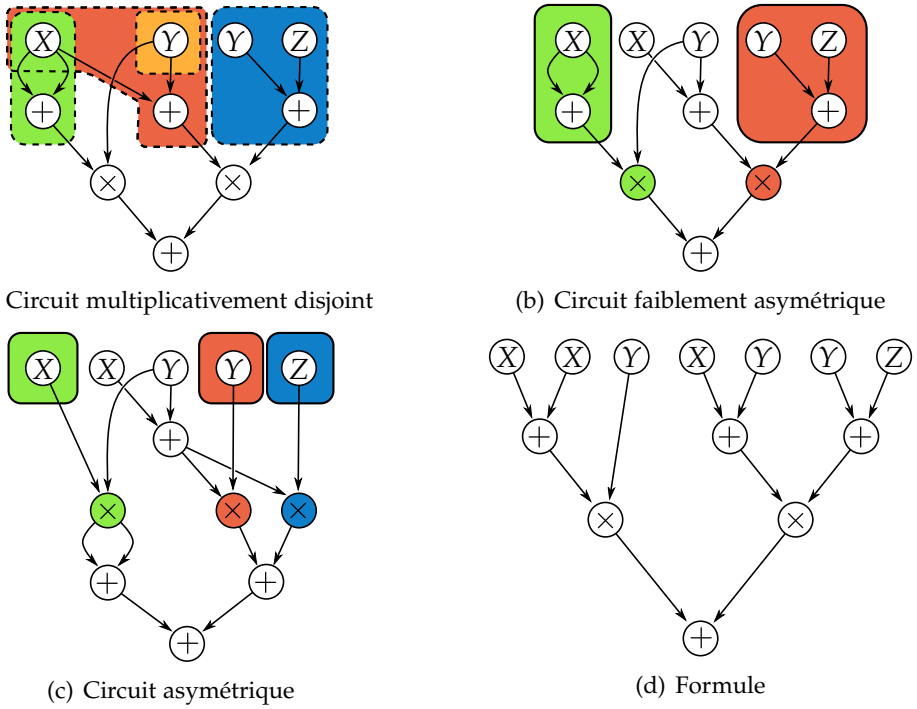
Définition 1.9

Un circuit est dit *multiplicativement disjoint* si pour chaque porte de multiplication recevant des flèches des portes α et β, les sous-circuits $\mathcal{C}_\alpha$ et $\mathcal{C}_\beta$ sont disjoints.

Un circuit est dit *faiblement asymétrique* si chaque porte de multiplication γ possède au moins un argument β tel que l'arc $\beta \to \gamma$ est le seul arc connectant $\mathcal{C}_\beta$ à $\mathcal{C} \setminus \mathcal{C}_\beta$.

Un circuit est dit *asymétrique* si l'un (au moins) des arguments de chaque porte de multiplication est une entrée du circuit, et que cette

1. *Straight-line program*, en anglais.



(a) Circuit multiplicativement disjoint

(b) Circuit faiblement asymétrique

(c) Circuit asymétrique

(d) Formule

FIGURE 1.3 – Le polynôme $2XY + (X + Y)(Y + Z)$ représenté par les quatre types de circuits de la définition 1.9.

entrée n'émet qu'une seule flèche.

Une *formule* est un circuit dont chaque porte est de degré sortant au plus 1.

Ces notions sont illustrées par la figure 1.3. Pour le circuit multiplicativement disjoint, les sous-circuits disjoints sont identifiés par des cadres en pointillés. Les deux sous-circuits disjoints de la porte de multiplication de gauche sont en vert et jaune. Ceux de la porte de multiplication de droite sont en bleu et orange. Pour les circuits faiblement asymétriques et asymétriques, les sous-circuits disjoints du reste du circuit sont matérialisés par des cadres en trait plein.

Toutes ces restrictions sont des cas particuliers de circuit. On note aussi qu'une formule est un circuit faiblement asymétrique particulier (mais ce n'est pas nécessairement un circuit asymétrique), et qu'un circuit faiblement asymétrique est un circuit multiplicativement disjoint particulier. Enfin, un circuit asymétrique est un circuit faiblement asymétrique particulier. Ces inclusions sont représentées par la figure 1.4.

Une formule est un circuit dont le graphe sous-jacent est un arbre et non un graphe orienté acyclique quelconque. Cette définition est équivalente à la notion naturelle et intuitive de *formule bien parenthésée*. Les parenthèses dans

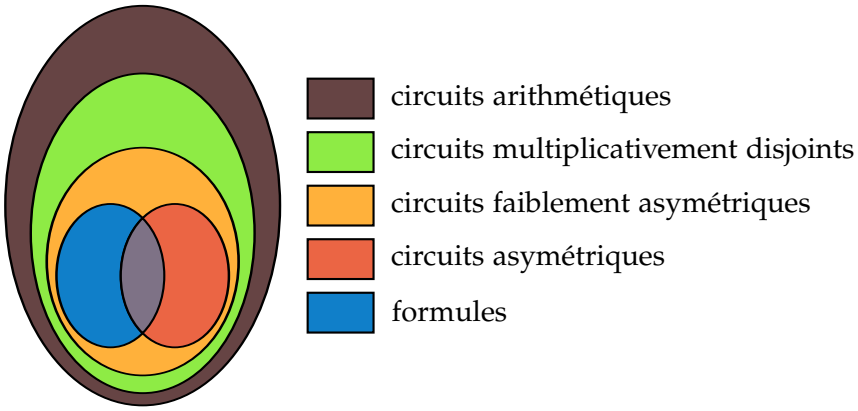


FIGURE 1.4 – Les inclusions entre les différents types de circuits.

une telle formule forment la structure d'arbre.

Dans un circuit asymétrique, les seules multiplications autorisées sont celles par une constante ou par une variable. Dans un circuit faiblement asymétrique, la restriction sur les multiplications est un peu moins forte. Tout ce qui est demandé est que l'un des deux arguments de la multiplication soit un polynôme qui a été calculé spécifiquement pour cette porte. Dans ce cas, on dira que le sous-circuit qui calcule cet argument est un sous-circuit *clos* : en effet, aucun des calculs intermédiaires ne peut être réutilisé à l'extérieur de ce sous-circuit. À l'inverse, une porte n'apparaissant pas dans un sous-circuit clos sera dite *réutilisable*.

Par extension, on dira qu'une porte de multiplication est asymétrique si l'un de ses arguments est une entrée, et faiblement asymétrique si l'un de ses arguments est un sous-circuit clos.

1.2.3 *Les programmes à branchements*

On s'intéresse ici à la représentation de polynômes par des graphes orientés acycliques, appelés programmes arithmétiques à branchements [2].

Définition 1.10

Un *programme arithmétique à branchements* sur un corps $\mathbb{K}$ est un graphe orienté acyclique dont les arcs sont étiquetés par des variables et des éléments de $\mathbb{K}$, et qui possède un sommet *source* s de degré entrant 0 et un sommet *puits* p de degré sortant 0.

Le *poids* d'un chemin de s à p est le produit des étiquettes des arcs utilisés. Le polynôme représenté par un programme à branchements est la somme des poids de tous les chemins de s à p.

2. *Arithmetic Branching Program*, en anglais.

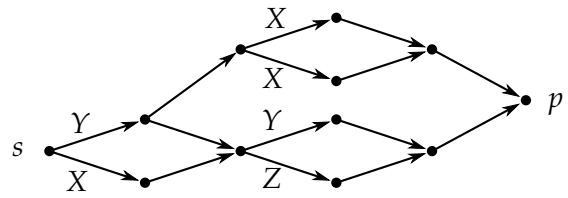


FIGURE 1.5 – Le polynôme $2XY + (X + Y)(Y + Z)$ représenté par un programme à branchements.

On appelle poids les étiquettes portées par les arcs d'un programme à branchements. Par convention, un arc dont le poids n'est pas précisé est un arc de poids 1. Le polynôme représenté par un sommet v quelconque est la somme des poids des chemins de s à v, et le polynôme représenté par un programme à branchements est le polynôme représenté par son puits. De même que pour les circuits, on peut s'autoriser l'existence de plusieurs puits, obtenant ainsi un programme à branchements représentant plusieurs polynômes.

1.3 Modèles de calcul et complexité

L'étude de la complexité des polynômes peut se faire dans différents cadres, avec différents modèles de calcul. Nous utiliserons le cadre booléen dont les deux grands modèles de calcul sont la machine de Turing et les circuits booléens, et le cadre de la théorie de Valiant qui utilise des circuits arithmétiques et leurs variantes comme modèles. Un troisième modèle, le modèle de Blum, Shub et Smale [10, 11] est souvent utilisé pour parler de complexité de problèmes concernant les polynômes mais il n'en sera pas question dans cette thèse.

1.3.1 Le modèle booléen

On suppose le lecteur familier du modèle de la machine de Turing et des classes de complexité les plus classiques qui s'y rattachent. Pour plus de détails sur ce sujet, on consultera les chapitres 1 à 4 de l'ouvrage de Sanjeev Arora et Boaz Barak [5]. On commence par rappeler quelques notions tout à fait basiques, puis on se penche sur quelques définitions un peu plus exotiques qui sont utilisées dans la suite.

Un langage booléen une partie de $\{0,1\}^\star$, c'est-à-dire un ensemble de mots finis sur l'alphabet $\{0,1\}$. Pour une fonction $f : \mathbb{N} \to \mathbb{N}$, la classe $\mathsf{DTIME}(f(n))$ représente l'ensemble des langages reconnus par une machine de Turing déterministe en temps $\mathcal{O}(f(n))$. De même, $\mathsf{NTIME}(f(n))$ est la classe des langages reconnus par une machine de Turing non déterministe en temps $\mathcal{O}(f(n))$. Pour $s : \mathbb{N} \to \mathbb{N}$, $\mathsf{DSPACE}(s(n))$ (resp. $\mathsf{NSPACE}(s(n))$) est la

classe des langages reconnus par une machine de Turing déterministe (resp. non déterministe) utilisant un espace $\mathcal{O}(s(n))$.

Les classes les plus classiques sont, par ordre d'inclusion,
- $\mathsf{L} = \mathsf{DSPACE}(\log n)$,
- $\mathsf{P} = \bigcup_k \mathsf{DTIME}(n^k)$,
- $\mathsf{NP} = \bigcup_k \mathsf{NTIME}(n^k)$,
- $\mathsf{PSPACE} = \bigcup_k \mathsf{DSPACE}(n^k)$ et
- $\mathsf{EXP} = \bigcup_k \mathsf{DTIME}(2^{n^k})$.

On dit qu'un langage $\mathcal{L}$ se réduit en temps polynomial à un langage $\mathcal{L}'$ s'il existe une fonction f, calculable en temps polynomial, telle que $x \in \mathcal{L}$ si et seulement si $f(x) \in \mathcal{L}'$. On dit qu'un langage $\mathcal{L}$ est NP-difficile (resp. PSPACE-difficile) si tout langage de NP (resp. de PSPACE) se réduit à $\mathcal{L}$ en temps polynomial. Un langage est NP-complet (resp. PSPACE-complet) s'il est NP-difficile (resp. PSPACE-difficile) et qu'il appartient à la classe NP (resp. PSPACE).

On peut considérer des machines de Turing *probabilistes*. Ces machines peuvent, à chaque étape de calcul, tirer une pièce à pile ou face et utiliser la réponse comme aide à la décision. On peut alors parler de *réduction probabiliste* : un langage $\mathcal{L}$ se réduit en temps polynomial probabiliste à un langage $\mathcal{L}'$ s'il existe une machine de Turing probabiliste fonctionnant en temps polynomial telle que la probabilité que $M(x) \in \mathcal{L}'$ si $x \in \mathcal{L}$ est supérieure à $2/3$, ainsi que la probabilité que $M(x) \notin \mathcal{L}'$ si $x \notin \mathcal{L}'$. Cette notion de réduction probabiliste permet de donner l'une des définitions de la classe AM[3] : c'est la classe des langages $\mathcal{L}$ qui se réduisent en temps polynomial probabiliste à un problème NP-complet. Puisque toute machine déterministe peut être vue comme une machine probabiliste qui ignore les résultats de ses tirages aléatoires, tout problème de NP se réduit de manière probabiliste à un problème NP-complet. En d'autres termes, $\mathsf{NP} \subseteq \mathsf{AM}$.

On peut définir la notion de circuit booléen par analogie avec celle de circuit arithmétique. Un *circuit booléen* est un graphe orienté acyclique dont les sommets de degré entrant nul sont appelés *entrées* et sont étiquetés par des variables, et les autres sommets sont appelés des portes et étiquetés soit par $\wedge$, soit par $\vee$, soit par $\neg$. Une porte $\wedge$ est de degré entrant 2 et calcule le « et » de ses deux arguments. Une porte $\vee$ est également de degré entrant 2, et calcule le « ou » (inclusif) de ses arguments. Enfin une porte $\neg$ est de degré entrant 1 et calcule la négation de son argument. Un circuit à n entrées accepte un mot $w = w_1 \cdots w_n$ si son évaluation sur le mot w est 1. Un langage $\mathcal{L}$ est décidé par une suite de circuits $(\mathcal{C}_n)$ si pour tout n, l'ensemble des mots acceptés par $\mathcal{C}_n$ est exactement $\mathcal{L} \cap \{0,1\}^n$.

On peut définir des classes de complexité à partir de la notion de circuit. Par exemple pour tout k, la classe NC^k est l'ensemble des langages décidés par une suite de circuits de taille polynomiale et de profondeur $\mathcal{O}(\log^k n)$.

3. Les initiales AM signifient Arthur-Merlin.

On définit également $\mathsf{NC} = \bigcup_k \mathsf{NC}^k$.

1.3.2 Le modèle de Valiant

Cette partie est un survol rapide des bases de la théorie de Valiant [122]. Une référence pour ce chapitre est l'ouvrage de Peter Bürgisser [17], ainsi que la thèse de Guillaume Malod pour les variantes des classes de base [96].

On définit une notion de complexité pour un polynôme basée sur les circuits arithmétiques définis précédemment.

Définition 1.11

Soit $f \in \mathbb{K}[X_1, \ldots, X_n]$. Alors la *complexité arithmétique* de f est la taille $L(f)$ du plus petit circuit arithmétique représentant f.

De même, $L_{\mathrm{md}}(f)$ est la taille du plus petit circuit multiplicativement disjoint représentant f, $L_{\mathrm{ws}}(f)$ est la taille du plus petit circuit faiblement asymétrique représentant f et $L_{\mathrm{e}}(f)$ est la taille de la plus petite formule représentant f.

On ne définit pas de notion de complexité associée aux circuits asymétriques et aux programmes à branchements puisqu'on verra dans le chapitre 3 que ces modèles sont équivalents aux circuits faiblement asymétriques.

On définit ensuite les classes de complexité du modèle de Valiant. Dans ce modèle de calcul, les langages sont des *familles de polynômes* indexées habituellement par les entiers naturels. On notera $(f_n)_{n\in\mathbb{N}}$ une telle famille, ou simplement (f_n), où $f_n \in \mathbb{K}[X_1, \ldots, X_{v(n)}]$ pour tout n. Une famille de polynômes est représentée par une famille de circuits $(\mathcal{C}_n)_{n\in\mathbb{N}}$ si pour tout n, le circuit $\mathcal{C}_n$ représente le polynôme f_n.

Les deux premières classes définies sont les classes VP et VNP définies par Leslie G. Valiant [122]. La classe VP est le pendant algébrique de la classe booléenne P. Quant à VNP, on peut la voir comme la version algébrique de NP ou de la classe de comptage #P. On note que ces classes dépendent en réalité du corps $\mathbb{K}$ dans lequel vivent les coefficients des polynômes considérés. Il y a donc autant de classes VP et VNP que de corps.

Une suite (u_n) est dite *polynomialement bornée* s'il existe une fonction polynomiale p telle que $u_n \leq p(n)$ pour tout n.

Définition 1.12

Soit $\mathbb{K}$ un corps.

La classe VP est l'ensemble des familles de polynômes (f_n) telles $(\deg(f_n))$ et $(L(f_n))$ sont polynomialement bornées.

La classe VNP est l'ensemble des familles de polynômes (f_n) telles qu'il existe une famille $(g_n) \in \mathsf{VP}$ telle que pour tout n

$$f_n(X) = \sum_{\epsilon \in \{0,1\}^{w(n)}} g_n(X, \epsilon)$$

où $X = (X_1, \ldots, X_{v(n)})$ et $\epsilon = (\epsilon_1, \ldots, \epsilon_{w(n)})$.

On peut définir VP de la manière équivalente suivante : une famille (f_n) appartient à VP si $(L_{\mathrm{md}}(f_n))$ est polynomialement bornée. De la même manière, on définit trois variantes de VP et une variante de VNP.

Définition 1.13

Soit (f_n), $f_n \in \mathbb{K}[X_1, \ldots, X_{v(n)}]$. Alors
- $(f_n) \in \mathsf{VP}_{\mathrm{nb}}$ si $(L(f_n))$ est polynomialement bornée ;
- $(f_n) \in \mathsf{VP}_{\mathrm{ws}}$ si $(L_{\mathrm{ws}}(f_n))$ est polynomialement bornée ;
- $(f_n) \in \mathsf{VP}_e$ si $(L_{\mathrm{e}}(f_n))$ est polynomialement bornée.

La classe $\mathsf{VNP}_{\mathrm{nb}}$ est définie par analogie à VNP en remplaçant VP par $\mathsf{VP}_{\mathrm{nb}}$.

On pourrait définir de même $\mathsf{VNP}_{\mathrm{ws}}$ et $\mathsf{VNP}_{\mathrm{e}}$, mais Valiant a prouvé que $\mathsf{VNP}_{\mathrm{e}} = \mathsf{VNP}$, et donc $\mathsf{VNP}_{\mathrm{ws}}$ est aussi égal à VNP [122]. On a les inclusions

$$\mathsf{VP}_e \subseteq \mathsf{VP}_{\mathrm{ws}} \subseteq \mathsf{VP} \subsetneq \mathsf{VP}_{\mathrm{nb}}.$$

L'inclusion stricte vient du fait que $\mathsf{VP}_{\mathrm{nb}}$ contient des familles de polynômes dont le degré n'est pas polynomialement borné comme par exemple la famille (X^{2^n}), contrairement aux autres. On a également

$$\mathsf{VP} \subseteq \mathsf{VNP} \subsetneq \mathsf{VNP}_{\mathrm{nb}}.$$

On définit ensuite la notion de circuit *sans constante*, ainsi qu'une notion de complexité associée.

Définition 1.14

Un circuit arithmétique *sans constante* est un circuit dont les entrées sont étiquetées soit par une variable, soit par l'entier relatif -1. Un circuit sans constante représente un polynôme $f \in \mathbb{Z}[X_1, \ldots, X_n]$.

Pour $f \in \mathbb{Z}[X_1, \ldots, X_n]$, la τ-complexité de f, notée $\tau(f)$, est la taille du plus petit circuit sans constante représentant f. Les mesures de complexité $\tau_{\mathrm{md}}(f)$, $\tau_{\mathrm{ws}}(f)$ et $\tau_{\mathrm{e}}(f)$ sont définies de manière analogue.

On remarque que pour tout $f \in \mathbb{Z}[X_1, \ldots, X_n]$, $L(f) \leq \tau(f)$. Guillaume Malod a défini des versions sans constante des classes de Valiant [96].

Définition 1.15

La classe VP^0 est l'ensemble des familles (f_n) de polynômes à coefficients entiers telles que $(\deg(f_n))$ et $(\tau(f_n))$ sont polynomialement bornées. De manière équivalente, $(\tau_{\mathrm{md}}(f_n))$ est polynomialement bornée.

Par analogie, on peut définir les classes VNP^0, VP^0_e, $\mathsf{VP}^0_{\mathrm{ws}}$, $\mathsf{VP}^0_{\mathrm{nb}}$ et $\mathsf{VNP}^0_{\mathrm{nb}}$.

Une notion importante en complexité est celle de réduction. Dans le cadre de la complexité de Valiant, la notion la plus communément utilisée est celle de p-projection.

Définition 1.16

Un polynôme $g \in \mathbb{K}[Y_1, \ldots, Y_m]$ est une *projection* d'un polynôme $f \in \mathbb{K}[X_1, \ldots, X_n]$ s'il existe une fonction de projection

$$\pi : \{X_1, \ldots, X_n\} \to \mathbb{K} \cup \{Y_1, \ldots, Y_m\}$$

telle que $g(Y_1, \ldots, Y_m) = f(\pi(X_1), \ldots, \pi(X_n))$.

Une famille (g_n) de polynômes est une p-projection d'une famille (f_n) s'il existe une fonction polynomiale p telle que pour tout n, g_n soit une projection de $f_{p(n)}$.

Soit VC une classe de complexité de Valiant. Une famille (f_n) est dite VC-complète si $(f_n) \in \mathsf{VC}$ et si toute famille $(g_n) \in \mathsf{VC}$ est une p-projection de (f_n).

Pour les classes sans constante, on impose de plus que la projection soit *bornée*, c'est-à-dire que chaque constante utilisée doit pouvoir être calculée à partir de la constante -1 par un circuit multiplicativement disjoint de taille polynomiale.

On donne maintenant les deux exemples de familles de polynômes que l'on va principalement rencontrer par la suite. Soit $\mathcal{X} = (X_{ij})_{1 \leq i,j \leq n}$ la matrice dont les coefficients sont n^2 indéterminées. La famille $(\textsc{Det}_n)$ est définie par $\textsc{Det}_n = \det(\mathcal{X})$ pour tout n. Soit $\mathcal{M} = (m_{ij})_{1 \leq i,j \leq n}$ une matrice. Alors son permanent est

$$\operatorname{per}(\mathcal{M}) = \sum_{\sigma \in \mathfrak{S}_n} \prod_{j=1}^{n} m_{i\sigma(i)}.$$

Le permanent est donc un objet très proche du déterminant, mais il est en réalité bien plus difficile à calculer. La famille $(\textsc{Per}_n)$ est définie par $\textsc{Per}_n = \operatorname{per}(\mathcal{X})$ pour tout n.

La famille $(\textsc{Det}_n)$ est $\mathsf{VP}_{\mathrm{ws}}$-complète [120, 96, 97] et la famille $(\textsc{Per}_n)$ est VNP-complète [122]. Bien entendu, on ne sait pas séparer les classes VP et VNP, et la question « $\mathsf{VP} = \mathsf{VNP}$? » est vue comme l'analogue algébrique de « P=NP ? ». On note qu'on ne sait pas non plus séparer VP_{e} ou $\mathsf{VP}_{\mathrm{ws}}$ de VNP.

Première partie

Résolution des systèmes polynomiaux

Chapitre 2

Complexité du résultant multivarié

Étant donnés deux polynômes à une variable, leur *matrice de Sylvester* est une matrice construite à partir des coefficients des polynômes et dont le déterminant s'annule si et seulement si les polynômes ont une racine commune. La dimension de la matrice de Sylvester est la somme des degrés des polynômes. On peut donc facilement tester de cette manière l'existence d'une racine commune à deux polynômes à une variable donnés sous forme dense.

L'étude des généralisations de ce résultat est l'objet de la théorie de l'élimination [123, 92, 33, 119, 34]. Si on considère un système de $(n+1)$ polynômes *homogènes* $f_0, \ldots, f_n \in \mathbb{K}[X_0, \ldots, X_n]$, un résultat central de cette théorie est l'existence d'un polynôme $\mathrm{res}(f_0, \ldots, f_n)$ en les coefficients des f_i qui s'annule si et seulement si les f_i ont une racine commune non triviale dans la clôture algébrique de $\mathbb{K}$, c'est-à-dire différente de $(0, \ldots, 0)$. Ce polynôme est appelé le *résultant multivarié* dans la littérature. En particulier, le déterminant de la matrice de Sylvester de deux polynômes à une variable $f, g \in \mathbb{K}[X]$ est égal au résultant des polynômes homogènes à deux variables $Y^{\deg(f)} f(X/Y)$ associé à f et $Y^{\deg(g)} g(X/Y)$ associé à g. On s'intéresse dans ce chapitre à la complexité du résultant multivarié, que l'on appellera simplement *résultant* dans la suite.

Le résultant est souvent utilisé pour la résolution des systèmes polynomiaux [81, 108, 23, 25] et pour l'élimination des quanteurs dans les corps algébriquement ou réellement clos [113, 56]. Plus récemment, le résultant a été étudié aussi bien d'un point de vue applicatif que purement théorique. Par exemple, le problème de la planification du mouvement des robots repose de manière essentielle sur le résultant [22, 24] et plus généralement le résultant est ubiquitaire en géométrie algébrique réelle [21, 69]. Enfin, dans le domaine du calcul formel, des progrès ont été réalisés dans la recherche

de formulations explicites du résultant [68, 31, 19, 25, 58], voir aussi [61].

Le but de ce chapitre est l'étude de la complexité du résultant. On s'intéresse d'une part au test de sa nullité, c'est-à-dire au test d'existence d'une racine non triviale pour un système de $(n+1)$ polynômes homogènes à $(n+1)$ variables, et d'autre part à l'évaluation de ce résultant. Dans tout ce chapitre, on considère les racines situées dans la clôture algébrique du corps ambiant.

Soit $\mathbb{K}$ un corps. Sa *clôture algébrique* est notée $\bar{\mathbb{K}}$. Pour tout nombre premier p et entier s, $\mathbb{F}_{p^s}$ est le corps à p^s éléments. On étend cette notation en posant $\mathbb{F}_0 = \mathbb{Q}$. On notera $X = (X_0, \ldots, X_n)$ et $X^\alpha = X_0^{\alpha_0} \cdots X_n^{\alpha_n}$.

Définition 2.1

Soit $\mathbb{K}$ un corps et $f_0, \ldots, f_n \in \mathbb{K}[X_0, \ldots, X_n]$ des polynômes homogènes. Le *résultant (multivarié)* $\mathrm{res}(f_0, \ldots, f_n)$ est un polynôme irréductible en les coefficients de $f_0, \ldots, f_n$ qui s'annule si et seulement s'il existe $a \in \bar{\mathbb{K}}^{n+1}$, $a \neq \bar{0}$, tel que $f_0(a) = \cdots = f_n(a) = 0$.

Le résultant est unique à une constante multiplicative près.

L'existence du résultant n'est pas évidente. Une preuve peut se trouver dans les ouvrages de Bartel L. van der Waerden [123] et Serge Lang [79]. Si $\mathbb{K}$ est un corps infini, son unicité tient au fait que deux polynômes irréductibles ayant les mêmes racines sont égaux à une constante multiplicative près. Pour le cas des corps finis, on se référera également à [123, 79].

On s'intéresse au problème de la nullité du résultant, autrement dit au problème de décider si un système de $(n+1)$ polynômes homogènes à $(n+1)$ variables sur un corps $\mathbb{K}$ admet une racine non triviale dans $\bar{\mathbb{K}}$. Ce problème est lié au problème de décision de la théorie existentielle des corps algébriquement clos. Ce dernier problème est parfois appelé dans la littérature anglophone *Hilbert's Nullstellentsatz problem*.

Définition 2.2

Soit $\mathbb{K}$ un corps. On définit les trois problèmes $\mathrm{HN}(\mathbb{K})$, $\mathrm{H_2N}(\mathbb{K})$ et $\textsc{Résultant}(\mathbb{K})$ de la manière suivante :

- $\mathrm{HN}(\mathbb{K})$: étant donné un système f de s polynômes de $\mathbb{K}[X_1, \ldots, X_n]$, existe-t-il une racine de f dans $\bar{\mathbb{K}}^n$?
- $\mathrm{H_2N}(\mathbb{K})$: étant donné un système de s polynômes *homogènes* de $\mathbb{K}[X_0, \ldots, X_n]$, existe-t-il une racine *non triviale* de f dans $\bar{\mathbb{K}}^{n+1}$?
- $\textsc{Résultant}(\mathbb{K})$: étant donné un système de $(n+1)$ polynômes homogènes de $\mathbb{K}[X_0, \ldots, X_n]$, existe-t-il une racine non triviale de f dans $\bar{\mathbb{K}}^{n+1}$?

Le problème $\textsc{Résultant}(\mathbb{K})$ est donc la restriction $\mathrm{H_2N}(\mathbb{K})$ aux systèmes *carrés* ayant autant de polynômes que de variables.

Pour définir tout à fait formellement les trois problèmes $\mathrm{HN}(\mathbb{K})$, $\mathrm{H_2N}(\mathbb{K})$ et $\textsc{Résultant}(\mathbb{K})$, il faut préciser la représentation des polynômes en entrée.

Sauf mention contraire, les polynômes seront représentés sous forme dense. Cependant, pour le théorème 2.11, il faudra considérer des polynômes creux.

D'autre part, il est nécessaire de préciser la représentation d'un élément de $\mathbb{K}$. Dans ce chapitre, $\mathbb{K}$ désignera toujours une extension finie de $\mathbb{F}_p$ où p est soit nul soit premier. En d'autres termes, $\mathbb{K}$ sera soit un corps fini soit un corps de nombre. La représentation utilisée sera $\mathbb{K} = \mathbb{F}_p[\xi]/\langle\phi\rangle$ où $\phi \in \mathbb{F}_p[\xi]$ est un polynôme irréductible unitaire. Par abus de langage, l'expression « pour tout corps $\mathbb{K}$ » signifiera « pour tout corps $\mathbb{K}$ de la forme $\mathbb{F}_p[\xi]/\langle\phi\rangle$ ».

Lorsque $\mathbb{K} = \mathbb{Q}$, il est plus naturel de considérer les problèmes $\mathrm{HN}(\mathbb{Z})$, $\mathrm{H_2N}(\mathbb{Z})$ et $\textsc{Résultant}(\mathbb{Z})$ où les polynômes sont à coefficients entiers. Dans ce cas, John Canny a donné un algorithme fonctionnant en espace polynomial pour calculer le résultant [22]. À notre connaissance, c'est la meilleure borne supérieure connue à ce jour pour l'évaluation exacte du résultant. Dans ce chapitre, nous montrons que pour tout corps $\mathbb{K}$, $\textsc{Résultant}(\mathbb{K})$ est NP-difficile. Nous obtenons aussi des résultats concernant l'évaluation du résultant. Plus précisément, nous montrons qu'améliorer la complexité obtenue par Canny requiert de nouvelles techniques.

Le résultat de NP-difficulté dans le cas d'un corps de caractéristique 0 semble faire partie du « folklore » du calcul formel, et une preuve peut se trouver dans un article de Joos Heintz et Jacques Morgenstern [51]. On peut obtenir un résultat similaire mais incomparable en considérant un système de deux polynômes à deux variables, mais donnés sous forme lacunaire. Ce résultat est une reformulation d'un résultat de David A. Plaisted [106]. Nous donnons des preuves de ces deux résultats, ce qui nous permet ensuite de voir en quoi ils sont incomparables et également pourquoi ces preuves sont inutilisables en caractéristique positive. Plus exactement, le résultat de Plaisted peut s'adapter en caractéristique positive mais uniquement à l'aide de techniques probabilistes [39, 63]. En d'autres termes, il ne permet d'obtenir que des résultats de NP-difficulté sous réduction probabiliste quand notre but est de donner des réductions déterministes.

Avec des techniques relativement standard, nous pouvons montrer que pour tout corps $\mathbb{K}$, décider si un système de polynômes homogènes à coefficients dans $\mathbb{K}$ a une racine dans $\bar{\mathbb{K}}$ est NP-difficile, c'est-à-dire $\mathrm{H_2N}(\mathbb{K})$ est NP-difficile. Les systèmes construits pour prouver ce résultat ont plus de polynômes que de variables. Il y a alors deux stratégies pour obtenir une preuve de NP-difficulté pour les systèmes carrés : soit diminuer le nombre de polynômes, soit augmenter le nombre de variables. Pour la première stratégie, l'idée est de remplacer le système polynomial par des combinaisons linéaires des polynômes de départ. Par des techniques assez standard, on peut montrer que si les combinaisons linéaires sont aléatoires, alors les racines du nouveau système sont exactement celles de l'ancien. Ce résultat se base sur un « théorème de Bertini ». Ceci nous permet de donner une preuve de NP-difficulté de $\textsc{Résultant}(\mathbb{K})$ sous réduction probabiliste. Pour obtenir une réduction déterministe, on utilise la seconde stratégie. La principale difficulté consiste à

ne pas créer de racines artificielles dans le nouveau système qui n'existaient pas dans le système de départ. En réalité, on peut voir ce résultat comme un résultat de *dérandomisation*. En effet, de même que pour la première stratégie, choisir aléatoirement les coefficients des nouvelles variables suffit à assurer l'absence de racine artificielle. Nous montrons comment éviter ce recours à l'aléatoire.

La partie 2.3 traite de la difficulté de l'évaluation du résultant. L'algorithme de Canny repose sur des calculs de déterminants de matrices dites *de Macaulay*. Nous montrons que ces matrices peuvent être représentées de manière succincte par des circuits, et que le calcul du déterminant de telles matrices succinctes est PSPACE-difficile dans le cas général. Ceci implique que la borne obtenue par Canny ne peut être améliorée qu'en utilisant de nouvelles techniques spécifiques au résultant et non en utilisant simplement un algorithme efficace de calcul de déterminant.

Ce chapitre reprend les articles [45] et [44].

2.1 Complexité du résultant en caractéristique nulle

2.1.1 *Borne supérieure*

Nous donnons dans cette partie une borne supérieure pour le test de nullité du résultant en caractéristique 0. Pour cela, le problème $\mathrm{H_2N}(\mathbb{K})$ est réduit au problème $\mathrm{HN}(\mathbb{K})$. Cette réduction est valable pour n'importe quel corps $\mathbb{K}$. Ceci nous permet d'en déduire que pour des polynômes à coefficients entiers, $\mathrm{H_2N}(\mathbb{Z})$ est dans la classe AM. De manière évidente, une instance de $\textsc{Résultant}(\mathbb{K})$ est une instance particulière de $\mathrm{H_2N}(\mathbb{K})$ donc ces résultats s'étendent au problème $\textsc{Résultant}(\mathbb{K})$.

Théorème 2.3

> Pour tout corps $\mathbb{K}$, le problème $\mathrm{H_2N}(\mathbb{K})$ se réduit polynomialement à $\mathrm{HN}(\mathbb{K})$.

Démonstration : Soit f une instance de $\mathrm{H_2N}(\mathbb{K})$, c'est-à-dire un système de s polynômes homogènes $f_1, \ldots, f_s \in \mathbb{K}[X_0, \ldots, X_n]$. On peut voir les polynômes $f_1, \ldots, f_s$ comme éléments de $\mathbb{K}[X_0, \ldots, X_n, Y_0, \ldots, Y_n]$ où les Y_i sont de nouvelles variables qui n'apparaissent pas dans les f_i. Soit g le système constitué des f_i et du nouveau polynôme inhomogène $\sum_{i=0}^{n} X_i Y_i - 1$. Le système g est une instance de $\mathrm{HN}(\mathbb{K})$. Il reste à montrer que f et g sont équivalents, c'est-à-dire que f a une racine non triviale si et seulement si g a une racine.

Soit $(a_0, \ldots, a_n, b_0, \ldots, b_n)$ une racine de g. Puisque $\sum_i a_i b_i - 1 = 0$ par définition, l'un au moins des a_i est non nul. Donc $(a_0, \ldots, a_n)$ est une racine non triviale de f. Réciproquement, si $(a_0, \ldots, a_n)$ est une racine

non triviale de f, soit i un indice tel que $a_i \neq 0$. Alors si on pose $b_i = 1/a_i$ et $b_j = 0$ pour $j \neq i$, $(a_0, \ldots, a_n, b_0, \ldots, b_n)$ est une racine de g. □

Pascal Koiran a montré que sous l'hypothèse de Riemann généralisée, $\mathrm{HN}(\mathbb{Z}) \in \mathsf{AM}$ [75]. Le même résultat est donc vrai pour $\mathrm{H_2N}(\mathbb{Z})$.

Corollaire 2.4

> Sous l'hypothèse de Riemann généralisée, $\mathrm{H_2N}(\mathbb{Z})$ est dans la classe AM, et donc Résultant$(\mathbb{Z})$ également.

Il est intéressant de noter qu'en caractéristique positive, la meilleure borne supérieure connue est PSPACE pour les trois problèmes $\mathrm{HN}(\mathbb{K})$, $\mathrm{H_2N}(\mathbb{K})$ et Résultant$(\mathbb{K})$. On ne sait en particulier pas si ces problèmes appartiennent à la hiérarchie polynomiale.

2.1.2 *Borne inférieure*

Nous nous intéressons maintenant aux bornes inférieures connues en caractéristique 0. La première partie de la proposition suivante se déduit aisément de [51, proposition 10]. La preuve de la seconde partie utilise une technique classique de *simulation* des grands entiers.

Proposition 2.5

> Le problème Résultant$(\mathbb{Z})$ est NP-difficile, et il le reste même si le système ne contient que des polynômes de degré au plus 2 et dont les coefficients sont bornés par 2 en valeur absolue.

Démonstration : La réduction est effectuée à partir du problème NP-complet Partition (voir [38, problème SP12]) : étant donné un ensemble fini S et un poids positif $w(s)$ pour chaque $s \in S$, existe-t-il un sous-ensemble S' de S tel que $\sum_{s\in S'} w(s) = \sum_{s\in S\setminus S'} w(s)$?

Soit $S = \{s_1, \ldots, s_n\}$ une instance de Partition. Pour $1 \leq i \leq n$, on définit $f_i(X) = X_0^2 - X_i^2$. On pose ensuite $f_0(X) = w(s_1)X_1 + \cdots + w(s_n)X_n$. Supposons que $a = (a_0, \ldots, a_n)$ soit racine de f. Alors puisque $f_i(a) = 0$ pour $1 \leq i \leq n$, $a_i = \pm a_0$ pour tout i. Sans perte de généralité, on peut supposer que $a_0 = 1$ et $a_i = \pm 1$ pour $i \geq 1$. Alors il est clair que $f_0(a) = 0$ si et seulement s'il existe S' tel que $\sum_{s\in S'} w(s) = \sum_{s\in S\setminus S'} w(s)$. Ceci prouve la première partie de la proposition.

Pour la seconde partie, il reste simplement à montrer qu'on peut supposer que les coefficients de f_0 peuvent être bornés par 2 en valeur absolue. Pour cela, on remplace les $w(s_i)$ par des variables. On peut écrire $w(s_i)$ en base 2 sous la forme $w(s_i) = \sum_{j=0}^{p} w_{ij}2^j$. Pour chaque w_{ij}, on ajoute une nouvelle variable W_{ij}. Pour tout i, on définit la valeur des W_{ij}

par la récurrence

$$\begin{cases} W_{ip} - w_{ip}X_0 & = 0, \quad \text{et} \\ W_{ij} - (2W_{i,j+1} + w_{ij}X_0) & = 0 \quad \text{pour } 0 \leq j < p. \end{cases} \tag{2.1}$$

Ces égalités impliquent que pour tous i et j, $W_{ij} = \sum_{\ell=j}^{p} w_{ij} 2^{j-\ell} X_0$. On remplace alors f_0 par $W_{1,0}X_1 + W_{2,0}X_2 + \cdots + W_{n,0}X_n$ et on ajoute les polynômes de l'équation (2.1) au système. Celui-ci reste bien carré et vérifie les propriétés voulues. □

Cette preuve n'est pas utilisable en caractéristique positive pour une raison simple : le problème $\textsc{Partition}_p$ consistant à décider l'existence d'un sous-ensemble S' tel que $\sum_{s\in S'} w(s) \equiv \sum_{s\in S\setminus S'} w(s) \mod p$ n'est pas NP-complet en caractéristique positive mais peut se résoudre en temps polynomial à l'aide d'un algorithme de programmation dynamique.

Si on encode les polynômes sous forme lacunaire, on peut prouver un résultat analogue pour un système de taille 2, c'est-à-dire de deux polynômes à deux variables. Ce résultat est la reformulation dans notre vocabulaire d'un résultat de David A. Plaisted qui dit que le calcul du plus grand diviseur commun (pgcd) de deux polynômes lacunaires à une variable et à coefficients entiers est NP-difficile [106]. Comme indiqué en début de ce chapitre, ce problème peut se résoudre en temps polynomial si les polynômes sont donnés sous forme dense en calculant le déterminant de leur matrice de Sylvester. Nous esquissons la preuve pour pouvoir en discuter par la suite. Pour une preuve complète incluant la correction de la réduction, on réfère le lecteur à [106, Theorem 5.1].

Proposition 2.6

> Étant donnés deux polynômes lacunaires homogènes de $\mathbb{Z}[X, Y]$, décider s'ils ont une racine commune dans $\mathbb{C}^2$ est NP-difficile.

Ébauche de démonstration : La réduction est effectuée à partir du problème Sat. Considérons donc une formule ϕ sous forme normale conjonctive sur les variables $v_1, \ldots, v_n$. Pour tout j, on associe à v_j un nombre premier p_j de telle sorte que les p_j sont distincts deux à deux. Soit $M = \prod_j p_j$. On construit un polynôme $f_\phi \in \mathbb{Z}[X]$ par induction sur ϕ. On associe à une variable v_j le polynôme $f_{v_j}(X) = X^{M/p_j} - 1$. Pour la négation d'une variable $\neg v_j$, on construit $f_{\neg v_j} = 1 + X^{m/p_j} + X^{2M/p_j} + \cdots + X^{(p_j-1)M/p_j}$. À une clause $C = \ell_1 \vee \cdots \vee \ell_m$ où les ℓ_i sont des littéraux, on associe le plus petit multiple commun (ppcm) f_C des f_{ℓ_i}. Enfin, $\phi = \bigwedge_i C_i$ est une conjonction de clause, et on lui associe le polynôme

$$f_\phi = X^M \sum_i P_{C_i}(X) P_{C_i}(1/X).$$

Le système est alors $(X^M - Y^M, Y^{\deg(f_\phi)} f_\phi(X/Y))$. La preuve que cette réduction s'effectue en temps polynomial et que le système a une racine dans $\mathbb{C}^2$ si et seulement si ϕ est satisfiable est omise. □

Les propositions 2.5 et 2.6 semblent être incomparables. En particulier, il n'est pas évident que l'on puisse directement obtenir la proposition 2.5 à partir de la proposition 2.6. Une idée naturelle est de tenter d'effectuer une réduction des degrés dans le résultat de Plaisted, de la même manière qu'on a réduit la valeur des coefficients pour la proposition 2.5. Nous allons en fait voir que cette technique classique qui fonctionne dans un cadre inhomogène n'est pas applicable dans un cadre homogène.

Pour diminuer le degré d'un polynôme, on pourrait vouloir remplacer pour $d > 1$ les occurrences de X^d par une nouvelle variable X_d et ajouter des polynômes pour imposer l'égalité de X_d et X^d. Par exemple, on pourrait ajouter les variables X_2 et X_4 ainsi que les polynômes $X_2 - X^2$ et $X_4 - X_2^2$ au système pour supprimer les occurrences de X^4. Pour garder le système homogène, ces polynômes peuvent être homogénéisés sous la forme $X_2X_0 - X^2$ et $X_4X_0 - X_2^2$. Cependant, cela crée la racine non triviale artificielle dans laquelle toutes les variables prennent la valeur 0 sauf X_4 qui prend une valeur arbitraire.

On peut donner un exemple explicite du problème qui a lieu si on cherche à diminuer les degrés dans la construction de Plaisted. Considérons la formule $(v \vee w) \wedge (\neg v) \wedge (\neg w)$ qui n'est pas satisfiable. Associons le nombre premier 2 à v et 3 à w. Alors $M = 6$. Dans construction de Plaisted, on construit $f_v = X^3 - 1$, et $f_w = X^2 - 1$, d'où $f_{v\vee w} = (X^2 - 1)(X^2 + X + 1)$. De plus, $f_{\neg v} = 1 + X^3$ et $f_{\neg w} = 1 + X^2 + X^4$. Les deux polynômes sont donc $X^6 - 1$ et $-X^3 + X^4 + 2X^5 + 9X^6 + 2X^7 + X^8 - X^9$. On peut vérifier qu'ils n'ont pas de racine commune.

Plutôt qu'homogénéiser ces deux polynômes comme dans la preuve précédente, on cherche à diminuer leurs degrés tout en obtenant un système homogène. Avec l'idée esquissée auparavant, le système obtenu est

$$\begin{cases} -X_3 + X_4 + 2X_5 + 9X_6 + 2X_7 + X_8 - X_9 \\ X_6 - X_0; \quad X_0X_2 - X^2 \quad ; \quad X_0X_3 - X_2X \\ X_0X_4 - X_2^2; \quad X_0X_5 - X_4X; \quad X_0X_6 - X_2X_4 \\ X_0X_7 - X_4X_3; \quad X_0X_8 - X_4^2 \quad ; \quad X_0X_9 - X_8X \end{cases}$$

où les deux premiers polynômes sont ceux de départ, et les autres sont les nouveaux polynômes du système. On vérifie aisément qu'en fixant X_8 et X_9 à la même valeur non nulle et toutes les autres variables à 0, on obtient une racine du système qui ne correspond pas à une racine du système de départ.

À notre connaissance, il n'y a pas de solution pour éviter cette création de racines artificielles. De plus, le résultat de Plaisted utilise de manière essentielle le fait qu'en caractéristique 0, une somme de termes positifs est nulle si et seulement si chaque terme est nul. Pour pouvoir étendre le résultat

à la caractéristique positive, il est nécessaire d'ajouter de l'aléa [39, 63]. Ainsi, même si on étudiait le problème Résultant avec en entrée des polynômes lacunaires, on ne pourrait obtenir de résultat de NP-difficulté sous réduction déterministe à l'aide du résultat de Plaisted.

Cette discussion justifie une approche nouvelle pour prouver le résultat de NP-difficulté pour les corps de caractéristique quelconque.

2.2 NP-difficulté en caractéristique quelconque

Dans cette partie, nous montrons la NP-difficulté du test de nullité du résultant en caractéristique quelconque. Deux stratégies sont étudiées. En partant d'un système de s polynômes homogènes de $\mathbb{K}[X_0, \ldots, X_n]$ où $s > n$, on souhaite construire un système *carré*, c'est-à-dire ayant autant de polynômes que de variables. La première stratégie consiste à diminuer le nombre de polynômes en effectuant des combinaisons linéaires de ceux-ci, et la seconde à augmenter le nombre de variables tout en s'assurant de ne pas créer de racines artificielles correspondant à la racine triviale du système de départ. La première stratégie fournit une preuve de NP-difficulté sous réduction probabiliste, tandis que la seconde fournit deux preuves de NP-difficulté sous réduction déterministe pour deux variantes du problème. On en déduit que le problème Résultant($\mathbb{K}$) est NP-difficile pour tout corps $\mathbb{K}$. Ces réductions sont basées sur la NP-difficulté de $\mathrm{H_2N}(\mathbb{K})$, dont la preuve donnée ici est une très légère modification de celle donnée par Pascal Koiran [75].

La preuve que l'on donne est une réduction depuis le problème Boolsys consistant à décider la satisfiabilité d'un système d'équations booléennes sur les variables $v_1, \ldots, v_n$ où les équations sont de trois types possibles :

(1) $v_i = \textbf{Vrai}$;

(2) $v_i = \neg v_j$;

(3) $v_i = v_j \vee v_k$.

Il est assez facile de montrer la NP-complétude de ce problème par réduction à partir de 3-Sat.

Proposition 2.7

> Soit $\mathbb{K}$ un corps de caractéristique quelconque. Le problème $\mathrm{H_2N}(\mathbb{K})$ consistant à décider si un système de s polynômes homogènes de $\mathbb{K}[X_0, \ldots, X_n]$ admet une racine non triviale dans $\bar{\mathbb{K}}^{n+1}$ est NP-difficile.

Démonstration : Soit $\mathbb{K}$ un corps de caractéristique p, où p est soit nul soit un nombre premier. On suppose dans un premier temps que $p \neq 2$. Considérons une instance $\mathcal{B}$ de Boolsys. On va définir un système polynomial homogène f ayant une racine non triviale si et seulement si $\mathcal{B}$ est satisfiable. Les variables du système f sont $X_0, \ldots, X_n$ où pour $1 \leq i \leq n$, X_i correspond à la variable booléenne v_i et X_0 est une variable d'homogénéisation. Le système f contient quatre types de polynômes, les

trois derniers correspondant aux trois types d'équations booléennes du langage Boolsys :

(0) $X_0^2 - X_i^2$, pour tout $i > 0$;

(1) $X_0 \cdot (X_i + X_0)$, pour toute équation $v_i =$ **Vrai** ;

(2) $X_0 \cdot (X_i + X_j)$, pour toute équation $v_i = \neg v_j$;

(3) $(X_i + X_0)^2 - (X_j + X_0) \cdot (X_k + X_0)$, pour toute équation $v_i = v_j \vee v_k$.

Les polynômes de type (0) impliquent qu'une racine $a = (a_0, \ldots, a_n)$ de f vérifie $a_0^2 = a_1^2 = \cdots = a_n^2$. En particulier, si a n'est pas la racine triviale, $a_i = \pm a_0 \neq 0$ pour tout $i > 0$. On vérifie aisément que l'assignation $v_i =$ **Vrai** si $a_i = -a_0$, et $v_i =$ **Faux** sinon satisfait le système booléen $\mathcal{B}$. Réciproquement, si $(v_1, \ldots, v_n)$ est une assignation valide de $\mathcal{B}$, tout $(n+1)$-uplet $a = (a_0, \ldots, a_n)$ où $a_0 \neq 0$ et $a_i = -a_0$ si $v_i =$ **True** et $a_i = a_0$ sinon est une racine non triviale de f. Ceci prouve la NP-difficulté du problème $\mathrm{H_2N}(\mathbb{K})$ pour tout corps $\mathbb{K}$ de caractéristique différente de 2.

Le cas de la caractéristique 2 est traité à part. On construit également quatre types de polynômes :

(0) $X_0 X_i - X_i^2$, pour tout $i > 0$;

(1) $X_0 \cdot (X_i + X_0)$, pour toute équation $v_i =$ **Vrai** ;

(2) $X_0 \cdot (X_i + X_j + X_0)$, pour toute équation $v_i = \neg v_j$;

(3) $X_i^2 + X_j X_k + X_0 \cdot (X_j + X_k)$, pour toute équation $v_i = v_j \vee v_k$.

Une racine $a = (a_0, \ldots, a_n)$ de f vérifie $a_i \in \{0, a_0\}$ pour tout $i > 0$. Donc toute racine non triviale de f vérifie $a_0 \neq 0$. On se convainc aisément que les assignations valides pour $\mathcal{B}$ sont en bijection avec les racines de f vérifiant $a_0 \neq 0$ et $a_i = a_0$ si $v_i =$ **Vrai** et $a_i = 0$ sinon. Ceci permet de conclure que $\mathrm{H_2N}(\mathbb{K})$ est également NP-difficile en caractéristique 2. □

2.2.1 *Une réduction probabiliste*

Cette partie est dédiée à une réduction probabiliste du problème $\mathrm{H_2N}(\mathbb{K})$ au problème Résultant$(\mathbb{K})$ en caractéristique quelconque. On note cependant qu'en caractéristique 0, la proposition 2.5 est préférable en ce sens que sa preuve est plus simple et le résultat plus fort. Pour plus de détails concernant les réductions probabilistes, on réfère le lecteur à [5].

On commence avec un résultat sur les variétés algébriques dans les corps algébriquement clos. Ce résultat est souvent appelé un théorème de Bertini effectif.

Proposition 2.8

Soit $\mathbb{K}$ un corps algébriquement clos, et V_f une variété algébrique de $\mathbb{K}^{n+1}$ définie par un ensemble $f_1, \ldots, f_s \in \mathbb{K}[X_0, \ldots, X_n]$ de polynômes homogènes de degré d. Pour $0 \leq i \leq n$, on définit $g_i = \sum_{j=1}^{s} \alpha_{ij} f_j$ où

$(\alpha_{ij}) \in \mathbb{K}^{s(n+1)}$, et on note V_g la variété définie par $g_0, \ldots, g_n$.

Alors il existe un polynôme F de degré au plus $(d+1)^{n+1}$ tel que pour tout $(n+1)$-uplet $\alpha \in \mathbb{K}^{s\times(n+1)}$ n'annulant pas F, $V_f = V_g$.

Ébauche de démonstration : D'après [116, Theorem A.8.7], l'adhérence pour la topologie de Zariski de l'ensemble des (α_{ij}) tels que $V_f \neq V_g$ forme un sous-espace A de dimension au plus $s(n+1)-1$ de $\mathbb{K}^{s(n+1)}$ (voir aussi [74, proposition 1]). L'existence du polynôme F et la borne sur son degré se déduisent du théorème de Bézout. □

On peut utiliser ce résultat pour donner un premier résultat de NP-difficulté en caractéristique quelconque.

Théorème 2.9

Soit p un entier soit nul soit premier, et $\mathbb{K}$ un corps de caractéristique p. Alors Résultant($\mathbb{K}$) est NP-difficile sous réduction probabiliste dès que $\mathbb{K}$ possède suffisamment d'éléments.

Démonstration : La réduction est effectuée à partir de $\mathrm{H_2N}(\mathbb{F}_p)$ qui est NP-difficile d'après la proposition 2.7. Soit f une instance de $\mathrm{H_2N}(\mathbb{F}_p)$, c'est-à-dire un système de s polynômes homogènes $f_1, \ldots, f_s \in \mathbb{F}_p[X_0, \ldots, X_n]$. On peut supposer que les f_j ont degré 2. On construit à partir de f un système carré g équivalent, en temps probabiliste polynomial.

Soit $(\alpha_{ij}) \in \mathbb{K}^{(n+1)\times s}$ et pour $0 \leq i \leq n$,

$$g_i = \sum_{j=1}^{s} \alpha_{ij} f_j.$$

Clairement, quelque soit (α_{ij}), une racine de f est également racine de g. D'après la proposition 2.8, la réciproque est vraie dès que les α_{ij} évitent les racines d'un polynôme de degré au plus 3^{n+1}.

On considère alors une extension $\mathbb{K}$ de $\mathbb{F}_p$ ayant au moins 3^{n+2} éléments. D'après le lemme de Schwartz-Zippel [32, 112, 126], il suffit de tirer les α_{ij} de manière aléatoire uniforme dans $\mathbb{K}$ pour assurer que α n'est pas racine de F avec probabilité au moins $2/3$. On note que $\mathbb{K}$ peut être construit en temps polynomial à l'aide de l'algorithme de Shoup [114] lorsque p est un nombre premier. Pour $p = 0$, $\mathbb{K} = \mathbb{Q}$ suffit.

Ainsi, on construit en temps polynomial un système carré g qui possède les mêmes racines que f avec probabilité $2/3$. □

2.2.2 *Deux réductions déterministes*

On peut maintenant utiliser la deuxième stratégie, augmenter le nombre de variables, pour améliorer le résultat de NP-difficulté de la partie précédente. Non seulement le résultat est plus fort, mais la preuve est plus

élémentaire puisqu'elle ne fait pas appel à l'élimination effective des quanteurs.

Deux résultats sont présentés. Dans le premier, la NP-difficulté du problème Résultant($\mathbb{K}$) est prouvée pour $\mathbb{K} = \mathbb{Q}$ (ou $\mathbb{Z}$) ou $\mathbb{K} = \mathbb{F}_q$ où q est une puissance d'un nombre premier. Plus précisément, on montre que pour tout nombre premier p, il existe une extension finie $\mathbb{F}_q$ de $\mathbb{F}_p$ telle que Résultant($\mathbb{F}_q$) est NP-difficile. De plus, le résultat reste vrai en se restreignant à des systèmes dont les polynômes sont de degré constant (au plus 2). Dans un deuxième temps, on montre que pour tout p, Résultant($\mathbb{F}_p$) est NP-difficile. Pour la preuve de ce résultat plus fort, on perd cependant le fait que les polynômes du système sont de degré constant. Leurs degrés sont en effet linéaires en le nombre de variables (ou le nombre de polynômes, qui est égal). Autrement dit, les polynômes doivent être donnés en représentation creuse et non plus dense.

Théorème 2.10

Soit p un entier soit nul soit premier. Alors il existe une extension finie $\mathbb{F}_q$ de $\mathbb{F}_p$ telle que Résultant($\mathbb{F}_q$) est NP-difficile. De plus, le résultat reste vrai si les polynômes considérés sont de degré au plus 2.

Démonstration : Pour tout p, on réduit le problème $H_2N(\mathbb{F}_p)$ au problème Résultant($\mathbb{F}_q$) pour une extension $\mathbb{F}_q$ de $\mathbb{F}_p$. Plus précisément, on part d'une instance de $H_2N(\mathbb{F}_p)$ produite par la réduction effectuée dans la preuve de la proposition 2.7. On considère donc un système f de s polynômes homogènes de $\mathbb{F}_p[X_0, \ldots, X_n]$ dont les composantes sont comme suit. Les polynômes f_1 à f_n sont les n polynômes de type (o) : pour $1 \leq i \leq n$, $f_i = X_0^2 - X_i^2$ (si $p \neq 2$) ou $f_i = X_0X_i - X_i^2$ (si $p = 2$). Les autres composantes, c'est-à-dire les polynômes f_{n+1} à f_s, sont chacun de l'un des types (1) à (3) définis dans la preuve de la proposition 2.7. On construit à partir de f un nouveau système g de s polynômes à s variables, équivalent à f. Les variables du système g sont les $(n+1)$ variables X_0, …, X_n de f et $(s-n-1)$ nouvelles variables Y_1, …, Y_{s-n-1}. Soit $\lambda \neq 0$ un paramètre à fixer ultérieurement. On définit g par

$$g(X,Y) = \begin{pmatrix} f_1(X) & & \\ \vdots & & \\ f_n(X) & & \\ f_{n+1}(X) & & +\lambda Y_1^2 \\ f_{n+2}(X) & -Y_1^2 & +\lambda Y_2^2 \\ & \vdots & \\ f_{n+i}(X) & -Y_{i-1}^2 & +\lambda Y_i^2 \\ & \vdots & \\ f_{s-1}(X) & -Y_{s-n-2}^2 & +\lambda Y_{s-n-1}^2 \\ f_s(X) & -Y_{s-n-1}^2 & \end{pmatrix}.$$

Clairement, f a une racine non triviale $a = (a_0, \ldots, a_n)$ si et seulement si $(a, \bar{0})$ est une racine non triviale de g. Il reste à montrer qu'il existe λ tel que si g a une racine non triviale (a, b), alors $b = \bar{0}$. On note également que λ doit pouvoir être calculé en temps polynomial.

Soit $(a, b) = (a_1, \ldots, a_n, b_1, \ldots, b_{s-n-1})$ une racine non triviale de g. En particulier, $f_i(a) = 0$ pour $1 \leq i \leq n$. Pour $1 \leq i \leq s-n$, on pose $\epsilon_i = f_{n+i}(a) \in \mathbb{F}_p$. Comme (a, b) est une racine de g, les b_i^2 satisfont le système linéaire

$$\begin{cases} \epsilon_1 & & & + & \lambda Z_1 & = 0, \\ \epsilon_2 & - & Z_1 & + & \lambda Z_2 & = 0, \\ & & & \vdots & & \\ \epsilon_{s-n-1} & - & Z_{s-n-2} & + & \lambda Z_{s-n-1} & = 0, \\ \epsilon_{s-n} & - & Z_{s-n-1} & & & = 0. \end{cases}$$

La variable Z_i représente Y_i^2 pour $1 \leq i \leq s-n-1$. On peut homogénéiser le système en remplaçant chaque ϵ_i par $\epsilon_i Z_0$ où Z_0 est une variable fraîche. On obtient alors un système linéaire homogène carré. Le déterminant de la matrice du système est égal à

$$(-1)^{s-n-1} \left(\epsilon_1 + \epsilon_2 \lambda + \cdots + \epsilon_{s-n} \lambda^{s-n-1} \right).$$

Ce déterminant est un polynôme de degré au plus $(s-n-1)$ en λ. Si λ n'est pas racine de ce polynôme, le déterminant est non nul et la seule solution du système linéaire est la solution triviale $Z_0 = \cdots = Z_{s-n-1} = 0$. Dans ce cas, $b = \bar{0}$. De plus, si tous les ϵ_i sont nuls, on a également $b = \bar{0}$. Il reste à prouver qu'il existe λ tel que le déterminant s'annule si et seulement si tous les ϵ_i sont nuls.

Si les polynômes sont à coefficients dans $\mathbb{Z}$, il suffit de prendre $\lambda = 3$ (ou toute autre entier $\lambda > 2$). En effet, comme $f_i(a) = 0$ pour $1 \leq i \leq n$, on a $a_i = \pm a_0$ pour tout i. Si $a_0 = 0$, alors la seule racine de g est la racine triviale. Ainsi, (a, b) est une racine non triviale de g si et seulement si $(1, a_1/a_0, \ldots, a_n/a_0, b_1/a_0, \ldots, b_{s-n-1}/a_0)$ est racine de g. On vérifie aisément que si $a_i = \pm 1$ pour tout i, $\epsilon_i \in \{-4, 0, 2, 4\}$. Le déterminant est alors nul si et seulement si $\epsilon'_1 + \epsilon'_2 \lambda + \cdots + \epsilon'_{s-n} \lambda^{s-n-1} = 0$ où $\epsilon'_i = \epsilon_i/2 \in \{-2, 0, 1, 2\}$ pour tout i. Posons alors pour tout i, $\epsilon_i^+ = \max(\epsilon'_i, 0)$ et $\epsilon_i^- = \max(-\epsilon'_i, 0)$. Alors $\epsilon'_i = \epsilon_i^+ - \epsilon_i^-$, et $0 \leq \epsilon_i^+, \epsilon_i^- \leq 2$. Puisque $\lambda = 3$, le déterminant du système linéaire est nul si et seulement si $\sum_i \epsilon_i^+ 3^{i+1} = \sum_i \epsilon_i^- 3^{i+1}$. Par unicité de la représentation des entiers en base 3, $\epsilon_i^+ = \epsilon_i^-$ pour tout i et $\epsilon_i = \epsilon'_i = 0$.

Cette stratégie ne fonctionne pas en caractéristique positive. On cherche simplement λ tel que λ ne soit racine d'aucun polynôme de degré $(s-n-1)$ dans $\mathbb{F}_p$. Ceci nous empêche en particulier de prendre λ

dans le corps premier $\mathbb{F}_p$. Considérons donc une extension $\mathbb{F}_q$ de $\mathbb{F}_p$, donnée sous la forme de $\mathbb{F}_p[\xi]/\langle\varphi\rangle$ où φ est un polynôme irréductible dans $\mathbb{F}_p$ de degré $(s-n)$. Si on pose $\lambda = \xi \in \mathbb{F}_q \simeq \mathbb{F}_p[\xi]/\langle\varphi\rangle$, alors par définition $h(\lambda) \neq 0$ pour tout polynôme h de degré strictement inférieur à $(s-n)$. En particulier, le déterminant du système linéaire ne peut s'annuler que si tous les ϵ_i sont nuls. On obtient donc un système g de polynômes homogènes à coefficients dans $\mathbb{F}_q$ dont toute racine (a, b) vérifie $b = \bar{0}$. En particulier, a est racine de f. Pour pouvoir conclure, il faut que λ soit constructible en temps polynomial. Ceci est assuré par l'algorithme de Shoup qui construit pour tout entier N un polynôme irréductible de degré N à coefficients dans $\mathbb{F}_p$ en temps polynomial [114].

En conclusion, pour tout corps $\mathbb{F}_p$ on peut construire en temps polynomial une instance g de Résultant$(\mathbb{F}_q)$ à partir d'une instance f de $\mathrm{H_2N}(\mathbb{F}_p)$, où $\mathbb{F}_q$ est une extension finie de $\mathbb{F}_p$, telle que f admet une racine non triviale si et seulement si g également. Donc Résultant$(\mathbb{F}_q)$ est NP-difficile. □

On montre maintenant comment obtenir une preuve de NP-difficulté de Résultant$(\mathbb{F}_p)$ plutôt que Résultant$(\mathbb{F}_q)$. Dans la réduction suivante, on a besoin de recourir à des polynômes de degré linéaire en le nombre de variables et non plus constant comme précédemment. Il n'est pas clair que l'on puisse obtenir le même résultat pour des polynômes de degré 2. Par exemple, réduire le degré des polynômes en ajoutant de nouvelles variables crée des racines artificielles comme on l'a vu à la fin de la partie 2.1.2.

L'idée de la preuve du théorème 2.11 est relativement simple. Le polynôme irréductible φ utilisé pour définir une extension de $\mathbb{F}_p$ fait maintenant partie du système et λ est considéré comme une variable. Ceci ne va pas sans poser quelques difficultés. En effet, si λ est considéré maintenant comme une variable, le système n'est plus homogène. Il faut donc trouver une façon de l'homogénéiser sans créer de racine artificielle.

Théorème 2.11

> Soit p un entier premier. Alors Résultant$(\mathbb{F}_p)$ est NP-difficile lorsque les polynômes sont donnés en représentation creuse.

Démonstration : Considérons le système g construit dans la preuve du théorème 2.10, où λ est vu comme une variable. On ajoute le polynôme φ au système. On l'homogénéise de manière canonique à l'aide de la variable X_0 en posant $\varphi(\lambda, X_0) = X_0^{\deg(\varphi)}\varphi(\lambda/X_0)$. On souhaite ensuite homogénéiser le reste du système. En particulier, il faut homogénéiser des polynômes de la forme $f_{n+i}(X) - Y_{i-1}^2 + \lambda Y_i^2$. Une première idée consiste à homogénéiser un tel polynôme sous la forme $X_0 f_{n+i}(X) - X_0 Y_{i-1}^2 + \lambda Y_i^2$. Cependant, les variables Y_i n'apparaissent dans ce cas jamais seules dans un monôme, mais toujours multipliées à une autre variable. Une racine non triviale de g peut alors être construite en mettant toutes les variables

sauf les Y_i à 0, puis les Y_i à des valeurs quelconques. Ce genre de racine n'est bien entendu pas souhaité. De même, on ne peut pas simplement remplacer les monômes λY_i^2 par λY_i, car la preuve précédente utilise le fait que les Y_i apparaissent tous avec degré 2 pour considérer le système comme étant linéaire en les Y_i^2. Le système g_h construit est finalement un peu plus compliqué :

$$g_h(X,Y,\lambda) = \begin{pmatrix} f_1(X) & & \\ \vdots & & \\ f_n(X) & & \\ X_0^{s-n-1} f_{n+1}(X) & & +\lambda Y_1^{s-n} \\ X_0^{s-n-2} f_{n+2}(X) & -Y_1^{s-n} & +\lambda Y_2^{s-n-1} \\ & \vdots & \\ X_0^{s-n-i} f_{n+i}(X) & -Y_{i-1}^{s-n-i+2} & +\lambda Y_i^{s-n-i+1} \\ & \vdots & \\ X_0 f_{s-1}(X) & -Y_{s-n-2}^3 & +\lambda Y_{s-n-1}^2 \\ f_s(X) & -Y_{s-n-1}^2 & \\ \varphi(\lambda, X_0) & & \end{pmatrix}.$$

Les Y_i n'apparaissent plus toutes à la même puissance. Cependant, toutes les occurrences d'une même variable Y_i ont le même degré.

Une partie des observations pour le système g dans la preuve du théorème 2.10 restent valables pour g_h. En particulier, il suffit de montrer que si un $(s+1)$-uplet (a,b,ℓ) est une racine non triviale de g_h vérifiant $a_0 = 1$, alors $b = \bar{0}$. Puisque $\varphi(0,1) \neq 0$, on peut supposer que $\ell \neq 0$.

Soit donc (a,b,ℓ) un $(s+1)$-uplet vérifiant $a_0 = 1$, $a_i = \pm a_0$ pour tout i (si $p \neq 2$) ou $a_i \in \{0, a_0\}$ (si $p = 2$), et $\ell \neq 0$. Nous souhaitons montrer que si (a,b,ℓ) est une racine de g_h, alors $b = \bar{0}$. Définissons, par analogie à la preuve précédente, $\epsilon_i = a_0^{s-n-i} f_{n+i}(a) \in \mathbb{F}_p$. On remarque que dans g_h, la variable Y_i apparaît toujours à la puissance $(s-n-i+1)$. Ainsi, pour toutes valeurs de a et ℓ telles que $f_i(a) = 0$ pour $1 \leq i \leq n$ et $\varphi(\ell, a_0) = 0$, (a,b,ℓ) est une racine de g_h si et seulement si les $b_i^{s-n-i+1}$ satisfont le système linéaire

$$\begin{cases} \epsilon_1 & & & + & \ell Z_1 & = 0, \\ \epsilon_2 & - & Z_1 & + & \ell Z_2 & = 0, \\ & & & \vdots & & \\ \epsilon_{s-n-1} & - & Z_{s-n-2} & + & \ell Z_{s-n-1} & = 0, \\ \epsilon_{s-n} & - & Z_{s-n-1} & & & = 0. \end{cases}$$

C'est le même système que dans la preuve précédente. Puisque ℓ est racine de φ qui est un polynôme irréductible dans $\mathbb{F}_p$ de degré $(s-n)$, ℓ ne peut être racine d'aucun polynôme de degré strictement inférieur

à $(s-n)$. Le déterminant du système linéaire est un tel polynôme, et donc ce déterminant est non nul sauf si tous les ϵ_i sont nuls. En d'autres termes, (a, b, ℓ) est une racine de g_h si et seulement si a est une racine de f. □

2.3 Matrices de Macaulay

La définition 2.1 affirme, pour $(n+1)$ polynômes homogènes $f_0, \ldots, f_n \in \mathbb{K}[X_0, \ldots, X_n]$, l'existence et l'unicité d'un polynôme $\mathrm{res}(f_0, \ldots, f_n)$ qui s'annule si et seulement si le système $f = (f_0, \ldots, f_n)$ admet une racine non triviale. Les formulations explicites de ce polynôme sont toutes basées sur un ou des calculs de déterminant. Souvent, les polynômes calculés ne sont que des multiples du résultant qui peuvent donc s'annuler sans que le système n'ait de racine non triviale [68, 31, 19, 25, 58]. Il existe cependant des formulations exactes basées sur des déterminants de matrices dites *de Macaulay*. Le résultant peut être calculé comme pgcd de déterminants de matrices de Macaulay (vus comme des polynômes) [22], ou comme le quotient de deux déterminants de matrices de Macaulay [123, 79]. Enfin, ce quotient peut être transformé en un unique déterminant à l'aide de la méthode de Strassen d'élimination des divisions [61]. Dans toutes ces formulations exactes, il est nécessaire d'évaluer le déterminant d'une ou plusieurs matrices de Macaulay. John F. Canny a donné un algorithme pour effectuer le calcul du résultant à l'aide de déterminants de matrices de Macaulay qui fonctionne en espace polynomial [22]. À notre connaissance, il s'agit de la meilleure borne supérieure connue à ce jour. On peut également noter que les déterminants de matrices de Macaulay forment une famille de la classe VPSPACE [77], qui est un équivalent de PSPACE dans le modèle de Valiant.

Les matrices de Macaulay sont de dimension exponentielle en la taille de la représentation dense des f_i. On ne peut donc pas les représenter explicitement avant de calculer leur déterminant. Il faut passer par des représentations implicites et calculer le déterminant à partir de ces représentations implicites. Canny a précisément montré que l'on pouvait effectuer ce calcul en espace polynomial. Le but de cette partie est de montrer qu'une amélioration de la borne supérieure donnée par Canny nécessite soit d'utiliser une formulation totalement nouvelle du résultant, soit d'exploiter finement la structure des matrices Macaulay et d'utiliser un algorithme *ad hoc* pour calculer leur déterminant. En effet, on montre que le calcul du déterminant d'une matrice représentée implicitement sous forme d'un circuit booléen est PSPACE-difficile (quelque que soit la caractéristique du corps), et que simplement tester la nullité d'un tel déterminant est PSPACE-complet. Un résultat similaire a été obtenu récemment par Malod qui montre que pour une notion proche de représentation implicite, le déterminant d'une matrice donnée sous forme implicite est un polynôme VPSPACE-complet dans le modèle de Valiant [98].

2.3.1 Représentation des matrices de Macaulay

On commence par définir les matrices de Macaulay, en suivant les présentations de [91] et [22]. Soit f un système de $(n+1)$ polynômes homogènes de $\mathbb{K}[X_0, \ldots, X_n]$ de degrés respectifs d_0 ,..., d_n. Soit $d = 1 + \sum_{i=0}^{n}(d_i - 1)$ le *degré du système*. Notons enfin $\text{Mon}_d = \{X^\alpha : \alpha_0 + \cdots + \alpha_n = d\}$ l'ensemble des monômes de degré d. On remarque que $|\text{Mon}_d| = \binom{n+d}{d}$.

Pour chaque ordre mis sur les variables X_0, ..., X_n, on peut définir une matrice de Macaulay. Supposons que $X_0 < \cdots < X_n$. La matrice $\text{Mac}(f_0, \ldots, f_n)$ est définie comme suit. Ses lignes et colonnes sont indexées par les éléments de Mon_d, ordonnés par ordre lexicographique inverse sur les multiplets α : $\alpha < \beta$ s'il existe i tel que $\alpha_i < \beta_i$ et $\alpha_j = \beta_j$ pour tout $j > i$. La ligne d'indice X^α représente alors le polynôme

$$\frac{X^\alpha}{X_i^{d_i}} f_i, \text{ où } i = \min\{j : d_j \leq \alpha_j\}. \tag{2.2}$$

Le minimum est pris pour l'ordre sur les variables, c'est-à-dire l'ordre naturel sur les entiers d'après la façon dont sont ordonnées les variables. L'indice i est bien défini grâce à la définition de d. Les autres matrices de Macaulay sont définies de manière similaire mais en changeant l'ordre sur les variables (et donc l'ordre pour le calcul du minimum). En particulier, Francis S. Macaulay a montré que le résultant est le pgcd des déterminants des n matrices de Macaulay que l'on obtient en prenant toutes les permutations circulaires de l'ordre $X_0 < \cdots < X_n$ [91].

On va montrer que les matrices de Macaulay admettent une représentation concise sous forme de circuit booléen. La définition suivante est une adaptation directe de la notion de *représentation par petit circuit* [1] définie pour les graphes [36], également connue sous le nom de *représentation succincte*. Différents auteurs ont étudié la différence de complexité pour des problèmes sur les graphes selon que le graphe est donné de manière explicite ou sous forme succincte [36, 103, 90, 6]. Des variantes de cette notion ont également été étudiées [124, 7, 35], dont une version pour les problèmes de comptage [121] et une pour la machine BSS [16].

Définition 2.12

> Une *représentation par circuit* d'une matrice $\mathcal{M}$ à coefficients entiers de dimensions $(m \times n)$ est un circuit booléen à plusieurs sorties $\mathcal{C}_\mathcal{M}$ ayant deux entrées de $\lceil \log m \rceil$ et $\lceil \log n \rceil$ bits respectivement, tel que sur l'entrée (i, j) l'évaluation du circuit est le coefficient $\mathcal{M}_{ij}$.
>
> La représentation par circuit d'un graphe est la représentation de sa matrice d'adjacence.

1. *Small circuit representation*, en anglais.

Théorème 2.13

Pour un système homogène $f = (f_0, \ldots, f_n) \in \mathbb{K}[X_0, \ldots, X_n]^{n+1}$ de degré d, la matrice $\mathrm{Mac}(f_0, \ldots, f_n)$ a une représentation par circuit de taille polynomiale en n et d.

Démonstration : Notons simplement $\mathrm{Mac} = \mathrm{Mac}(f_0, \ldots, f_n)$. Il suffit de donner un algorithme polynomial en n et d qui sur l'entrée (n, d, i, j) calcule la valeur de Mac_{ij}.

La première étape de l'algorithme est de trouver les monômes X^α et X^β indexant la ligne i et la colonne j respectivement. Notons $A_d = \{\alpha : \alpha_0 + \cdots + \alpha_n = d\}$ l'ensemble des multiplets tel que $\mathrm{Mon}_d = \{X^\alpha : \alpha \in A_d\}$. Il faut trouver les éléments i et j de A_d dans l'ordre lexicographique inverse.

Dans l'ordre lexicographique inverse, les premiers éléments de A_d sont ceux dont la dernière coordonnée vaut 0, puis viennent ceux dont la dernière coordonnée vaut 1, et le dernier est $(0, \ldots, 0, d)$. Étant donné i, on cherche la valeur α_n de l'élément i. Définissons pour cela

$$A_d^k = \{\alpha : \alpha_0 + \cdots + \alpha_k = d \text{ et } \alpha_{k+1} = \cdots = \alpha_n = 0\}$$

pour $0 \leq k \leq n$. En particulier, $A_d = A_d^n$, $A_d^k \subseteq A_d^{k+1}$ pour tout $k < n$, et tous les éléments de A_d^k sont plus petits que les éléments de $A_d \setminus A_d^k$ pour l'ordre lexicographique inverse. Remarquons ensuite que les éléments de la forme $(\alpha_0, \ldots, \alpha_{n-1}, v)$ sont en bijection avec A_{d-v}^{n-1}. Donc si

$$\sum_{\ell=0}^{d-v-1} |A_\ell^{n-1}| < i \leq \sum_{\ell=0}^{d-v} |A_\ell^{n-1}|$$

alors $\alpha_n = v$. De plus, si l'élément i de A_d vérifie $\alpha_n = v$, alors c'est l'élément d'indice $(i - \sum_{\ell=0}^{d-v-1} |A_\ell^{n-1}|)$ dans A_{d-v}^{n-1}. On peut alors calculer récursivement chaque composante de l'élément α d'indice i de A_d. Pour vérifier que l'algorithme est bien de complexité polynomiale, on note que

$$|A_d^k| = \binom{d+k}{d} = |A_{d-1}^k| \frac{d+k}{d}.$$

On peut donc calculer les sommes $\sum_{\ell=0}^{d-v-1} |A_\ell^{n-1}|$ pour toutes les valeurs de v en temps linéaire en n et d. Ainsi, on peut déterminer X^α et X^β en temps quadratique.

Pour calculer Mac_{ij}, il suffit alors d'utiliser l'équation (2.2). En effet, on cherche le coefficient du monôme X^β dans le polynôme $(X^\alpha / X_i^{d_i}) f_i$, qui est égal au coefficient de $X^\beta / (X^\alpha / X_i^{d_i})$ dans f_i. Ce coefficient se calcule donc en temps polynomial. □

2.3.2 Déterminant d'une matrice représentée par un circuit

On cherche à prouver qu'en toute caractéristique, tester si le déterminant d'une matrice représentée par un circuit est nul est un problème PSPACE-complet. La preuve est basée sur le fait que tester si deux sommets s et p d'un graphe représenté par un circuit sont reliés par un chemin est PSPACE-complet, même avec la promesse que ce chemin est unique s'il existe.

La PSPACE-complétude du problème de connectivité dans un graphe donné sous forme de circuit (sans la promesse d'unicité) est prouvée par une réduction au problème canonique PSPACE-complet TQBF dans [90]. Puisque la preuve de PSPACE-complétude de TQBF utilise elle-même une réduction vers un problème de connectivité dans un graphe, il semble naturel de vouloir obtenir une preuve directe du résultat qui ne passe pas par TQBF. Nous fournissons ici une telle preuve directe.

Lemme 2.14

Soit $\mathcal{C}$ un circuit booléen représentant un graphe orienté $\mathcal{G}$ ayant deux sommets distingués s et p, avec la promesse qu'il existe au plus un chemin de s à p dans $\mathcal{G}$. Alors décider si un tel chemin existe est un problème PSPACE-complet.

Démonstration : L'appartenance du problème à PSPACE découle de l'algorithme en espace non déterministe logarithmique pour décider la connectivité dans un graphe (voir par exemple [5]). On utilise le même algorithme, en évaluant le circuit sur deux sommets u et v quand on souhaite savoir si les deux sommets sont reliés par une arête ou non. On obtient un algorithme qui fonctionne en espace non déterministe logarithmique en la taille du graphe, c'est-à-dire polynomial en la taille de l'entrée. On conclut grâce au théorème de Savitch qui montre que NPSPACE = PSPACE.

Considérons maintenant un problème $\mathcal{L} \in$ PSPACE, décidé par une machine de Turing déterministe M en espace polynomial. Sur l'entrée x, on peut considérer le graphe des configurations $\mathcal{G}_M^x$ de la machine M. On peut décrire ce graphe par un circuit booléen de taille polynomiale qui étant données deux configurations c et c' de M décide si c' est atteignable depuis c en un pas de calcul. Ce circuit peut être construit en temps polynomial (voir à nouveau [5]). De plus, il existe un chemin de la configuration initiale de $\mathcal{G}_M^x$ à sa configuration acceptante si et seulement si $x \in \mathcal{L}$.

Pour conclure, on remarque que M étant déterministe, il existe au plus un chemin entre les configurations initiale et acceptante de $\mathcal{G}_M^x$. □

Corollaire 2.15

Soit $\mathcal{C}$ un circuit représentant une forêt (orientée des feuilles vers les

racines), c'est-à-dire un graphe acyclique $\mathcal{F}$ dont chaque sommet est de degré sortant au plus 1, et deux sommets distingués s et p dans $\mathcal{F}$. Alors décider l'existence d'un chemin de s à p dans $\mathcal{F}$ est PSPACE-complet.

Démonstration : Il suffit de remarquer que dans la preuve précédente, le graphe des configurations $\mathcal{G}_M^x$ est une forêt puisque M est déterministe. □

On peut en déduire le résultat principal de cette partie.

Théorème 2.16

Soit $\mathcal{C}$ un circuit représentant une matrice $\mathcal{M}$ à coefficients dans $\{0,1\}$ avec la promesse que $\det(\mathcal{M}) \in \{-1,0,1\}$. Alors décider si le déterminant est nul est PSPACE-complet.

Démonstration : On peut déduire un algorithme de complexité spatiale polynomiale du fait que le problème appartient à la classe NC (uniforme) si la matrice est donnée explicitement en entrée [29]. On peut également prouver que le problème appartient à PSPACE en utilisant le programme à branchements pour le déterminant décrit au chapitre 3 (proposition 3.12) et remarquer qu'il permet de définir un algorithme de programmation dynamique calculant le déterminant en espace logarithmique.

Soit $\mathcal{C}$ un circuit représentant une forêt $\mathcal{F}$ avec deux sommets distingués s et p. Soit $\mathcal{G}$ le graphe obtenu en rajoutant un arc de p vers s, et des boucles sur tous les autres sommets. Alors la seule couverture par cycles potentielle de $\mathcal{G}$ est faite d'un *grand* cycle et de boucles, où le grand cycle est un chemin de s à p suivi de l'arc de p vers s. Ainsi, le déterminant de la matrice d'adjacence de $\mathcal{G}$ est non nul si et seulement s'il existe un chemin de s à p dans $\mathcal{G}$.

Pour terminer la preuve, il faut remarquer que le déterminant de la matrice d'adjacence de $\mathcal{G}$ ne peut prendre que les trois valeurs -1, 0 et 1 et qu'il est facile de construire à partir de $\mathcal{C}$ un circuit représentant $\mathcal{F}$. □

La technique utilisée pour transformer la forêt $\mathcal{F}$ en le graphe $\mathcal{G}$ dans la preuve précédente est utilisée à plusieurs reprises dans les chapitres 3 et 4.

Corollaire 2.17

Pour tout $n \geq 2$, décider si le déterminant d'une matrice à coefficients dans $\{0,1\}$ représentée par un circuit est nul *modulo* n est PSPACE-complet.

Démonstration : Pour tout $n \geq 2$, les entiers 1 et -1 ne sont pas congrus à 0 *modulo* n. Le théorème précédent implique donc ce résultat. □

De ce corollaire, on déduit que le calcul du déterminant d'une matrice donnée par circuit est PSPACE-complet quelle que soit la caractéristique du corps sur lequel on travaille.

Deuxième partie

Représentations déterminantielles de polynômes

Chapitre 3

Complexité déterminantielle

Une représentation déterminantielle d'un polynôme f est une matrice dont les coefficients sont soit des constantes, soit des variables, et dont le déterminant égale f. Ce type de représentation est utilisé par Leslie G. Valiant pour étudier les classes de complexité algébriques dites *classes de Valiant* [122]. En particulier, il prouve l'*universalité du déterminant* pour les formules arithmétiques : toute formule arithmétique peut être représentée par le déterminant d'une matrice dont les coefficients sont des variables et des constantes du corps de base, de dimension linéaire en la taille de la formule. Ce résultat a ensuite été indépendamment étendu par Seinosuke Toda [120] et Guillaume Malod [96, 97] aux circuits (faiblement) asymétriques. D'autre part, Hong Liu et Kenneth W. Regan ont amélioré la borne obtenue par Valiant en utilisant des méthodes tout à fait différentes [88]. En particulier, les résultats de Valiant, Toda et Malod utilisent des constructions à base de graphes (qui sont en fait, plus ou moins déguisés, des programmes à branchements) quand Liu et Regan donnent des constructions purement matricielles. Ces résultats sont une manière d'étudier finement la complexité du déterminant.

Dans ce chapitre, nous faisons une présentation unifiée de ces résultats basés sur la représentation des formules et circuits (faiblement) asymétriques par des programmes à branchements d'une part [1], et sur la représentation des programmes à branchements par des déterminants d'autre part. Ceci nous permet d'améliorer légèrement les bornes précédentes. Pour ce faire, nous introduisons la notion de *taille réduite* qui comme son nom l'indique est une notion de taille pour les circuits arithmétiques, plus petite que la taille classique. En particulier nous montrons que l'on peut construire de manière graphique des représentations déterminantielles de formules aussi compactes que celles de Liu et Regan.

1. Ceci permet également de corriger une légère erreur de l'article de Valiant qui n'avait jamais été signalée dans la littérature.

À l'aide de techniques similaires, nous nous penchons sur la complexité déterminantielle du permanent. Il est conjecturé que cette complexité est $2^{\Omega(n)}$. Cette question est une formulation du problème « $\mathsf{VP}^0_{\mathrm{ws}} = \mathsf{VNP}^0$? » dans la théorie de Valiant, qui est une version particulière de la question principale « $\mathsf{VP} = \mathsf{VNP}$? » de cette théorie. Nous donnons ici une borne supérieure $(2^n - 1)$ obtenue par une construction élémentaire qui est à notre connaissance la meilleure borne connue pour le problème.

Ce chapitre reprend une partie de l'article [42] pour les notions de taille réduite et la représentation des formules, et l'article [41] pour la complexité déterminantielle du permanent.

3.1 Taille réduite

On considère une généralisation des circuits arithmétiques que l'on appelle *circuits décorés*. Un circuit décoré est un circuit arithmétique sur un corps $\mathbb{K}$ dont les flèches sont étiquetées par des éléments de $\mathbb{K}$, que l'on appelle des poids. Le polynôme représenté par une porte de calcul γ dont les arguments sont α et β, et tels que les flèches de α et β vers γ sont de poids respectifs w_α et w_β, est défini par induction par $f_\gamma = w_\alpha f_\alpha \star w_\beta f_\beta$ où $\star \in \{+, \times\}$ est l'étiquette de γ. On peut de plus supposer que les entrées constantes d'un circuit décoré sont étiquetées par la constante 1. L'ensemble des poids portés par les flèches d'un circuit décoré est appelé l'*ensemble des constantes* du circuit.

Soit $v(\mathcal{C})$ le nombre d'entrée d'un circuit $\mathcal{C}$ qui sont étiquetées par une variable. On dit qu'une porte est *constante* lorsqu'elle représente un polynôme constant, et qu'elle est *non constante* sinon. On définit similairement les *arguments constants et non constants* d'une porte de calcul.

Lemme 3.1

Tout circuit arithmétique $\mathcal{C}$ ayant au moins une porte d'addition non constante peut être représenté par un circuit décoré $\mathcal{C}_r$ tel que
- $t(\mathcal{C}_r) \leq t(\mathcal{C})$ et $v(\mathcal{C}_r) = v(\mathcal{C})$;
- les entrées de $\mathcal{C}_r$ sont étiquetées soit par une variable soit par la constante 1, et les entrées constantes ont degré sortant 1 ;
- les portes d'additions ont au plus un argument constant, et cet argument est une entrée ;
- les portes de multiplication ont deux arguments non constants.

De plus, le circuit $\mathcal{C}_r$ est de même type que $\mathcal{C}$: si $\mathcal{C}$ est une formule, alors $\mathcal{C}_r$ est une formule, et si $\mathcal{C}$ est (faiblement) asymétrique, alors $\mathcal{C}_r$ est (faiblement) asymétrique.

Démonstration : Le circuit réduit $\mathcal{C}_r$ est obtenu par une série de transformations appliquées à $\mathcal{C}$. Chaque porte d'entrée étiquetée par une constante c est remplacée par une entrée étiquetée 1, et les flèches émises par cette porte ont poids c. Si l'entrée émet plusieurs flèches, elle est dupliquée

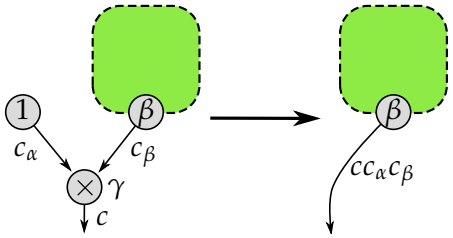


FIGURE 3.1 – Suppression d'une porte de multiplication de degré sortant non nul.

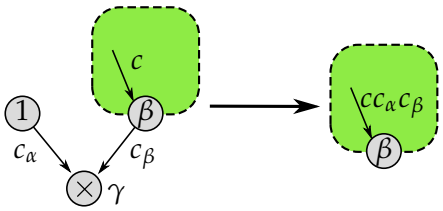


FIGURE 3.2 – Suppression d'une porte de multiplication de degré sortant nul.

autant de fois que nécessaire pour avoir degré sortant 1. Ainsi, les entrées satisfont le lemme. Toute porte de calcul γ de valeur constante c est remplacée par une porte d'entrée étiquetée 1 et les flèches émises par γ ont leur poids multiplié par c. De même que précédemment, on duplique la porte γ si plusieurs flèches sont émises. Toutes les portes de calcul ont donc au moins un argument non constant. On s'intéresse maintenant aux portes de multiplication. Soit γ une porte de multiplication ayant un argument constant α. L'entrée α est étiquetée par 1, et la flèche $\alpha \to \gamma$ a un poids c_α. Soit β l'autre argument de γ et c_β le poids de la flèche $\beta \to \gamma$. Alors les portes α et γ sont supprimées et toutes les flèches émises par γ proviennent maintenant de β. Si γ émet au moins une flèche, le poids de chacune de ces flèches est multiplié par $c_\alpha c_\beta$ (Fig. 3.1). Sinon, la porte γ est la sortie de $\mathcal{C}$. En la supprimant, la porte β devient la sortie du nouveau circuit $\mathcal{C}_r$. Puisqu'il y a au moins une porte d'addition dans le circuit, β ne peut pas être une entrée et c'est donc une porte de calcul. Alors si β est une porte d'addition, le poids des deux flèches pointant sur β est multiplié par $c_\alpha c_\beta$. Si β est une porte de multiplication, le poids d'une des deux flèches pointant sur β est multiplié par $c_\alpha c_\beta$ (Fig. 3.2). Ainsi, on vérifie aisément que la valeur du nouveau circuit $\mathcal{C}_r$ est la même que celle de $\mathcal{C}$.

Pour finir, on supprime toutes les portes qui se retrouvent isolées, c'est-à-dire depuis lesquelles il n'existe pas de chemin vers la sortie de $\mathcal{C}_r$. □

Définition 3.2

Soit $\mathcal{C}$ un circuit arithmétique. Son *circuit réduit associé* est le circuit $\mathcal{C}_r$ obtenu en appliquant le lemme 3.1. La *taille réduite* de $\mathcal{C}$, notée $t_r(\mathcal{C})$, est définie par $t_r(\mathcal{C}) = t(\mathcal{C}_r)$.

La taille réduite s'apparente à la notion de taille de Liu et Regan [88]. Elle est en fait légèrement plus petite.

3.2 Circuits et programmes à branchements

Cette partie exhibe les liens entre les formules, les circuits (faiblement) asymétriques, et les programmes à branchements.

3.2.1 *Formules et programmes à branchements*

Le premier résultat est essentiellement une reformulation (et une correction) d'un résultat de Leslie G. Valiant [122]. Nous l'étendons au cas des formules décorées. Dans la suite, on identifiera une formule et le polynôme qu'elle représente.

Théorème 3.3

Soit φ une formule décorée de taille t. Alors il existe un programme à branchements $\mathcal{B}$ à $(t+2)$ sommets et une constante c tels que $\varphi = cf_{\mathcal{B}}$ où $f_{\mathcal{B}}$ est le polynôme représenté par $\mathcal{B}$.

Si φ possède au moins une porte d'addition, il existe dans $\mathcal{B}$ un sommet de degré sortant 1 et dont le seul arc sortant a un poids constant.

Démonstration : On prouve la première partie du lemme par induction sur les formules.

Si φ est une entrée d'étiquette x (constante ou variable), alors $\mathcal{B}$ possède deux sommets s et p et un arc $s \to p$ de poids x. La constante c vaut 1.

Si $\varphi = w_1\varphi_1 \star w_2\varphi_2$, $\star \in \{+, \times\}$, soit $\mathcal{B}_1$ et c_1 (resp. $\mathcal{B}_2$ et c_2) le programme à branchements et la constante associés à φ_1 (resp. φ_2) par induction. Notons également f_1 et f_2 les polynômes respectivement représentés par $\mathcal{B}_1$ et $\mathcal{B}_2$ de telle sorte que $\varphi_1 = c_1f_1$ et $\varphi_2 = c_2f_2$.

Si $\star = \times$, soit $c = c_1c_2w_1w_2$, et $\mathcal{B}$ l'union de $\mathcal{B}_1$ et $\mathcal{B}_2$ dans laquelle le puits de $\mathcal{B}_2$ et la source de $\mathcal{B}_1$ sont identifiés (Fig. 3.3). Les chemins de la source de $\mathcal{B}$ vers son puits sont composés de deux parties, l'une de la source au puits de $\mathcal{B}_1$ et l'autre de la source au puits de $\mathcal{B}_2$. Ainsi, le polynôme $f_{\mathcal{B}}$ représenté par $\mathcal{B}$ est égal à f_1f_2. Alors $cf_{\mathcal{B}} = (w_1c_1f_1) \times (w_2c_2f_2)$, donc $cf_{\mathcal{B}} = w_1\varphi_1 \times w_2\varphi_2 = \varphi$. D'autre part, $|\mathcal{B}| = |\mathcal{B}_1| + |\mathcal{B}_2| - 1 \leq t(\varphi_1) + t(\varphi_2) + 3 = t + 2$.

Si $\star = +$, $\mathcal{B}$ est l'union de $\mathcal{B}_1$ et $\mathcal{B}_2$ dans laquelle les sources de $\mathcal{B}_1$ et $\mathcal{B}_2$ sont identifiées. De plus, on ajoute un arc de poids $\frac{c_2w_2}{c_1w_1}$ du puits de

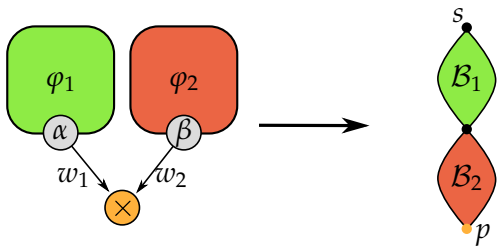


FIGURE 3.3 – Construction pour un produit de deux sous-formules.

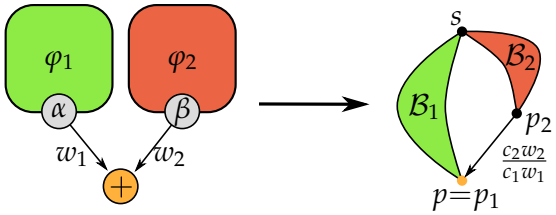


FIGURE 3.4 – Construction pour une somme de deux sous-formules.

$\mathcal{B}_2$ vers le puits de $\mathcal{B}_1$ qui devient le puits de $\mathcal{B}$ (Fig. 3.4). Les chemins de la source au puits dans $\mathcal{B}$ passent soit par $\mathcal{B}_1$, soit par $\mathcal{B}_2$ puis le nouvel arc. Ainsi,

$$f_{\mathcal{B}} = f_1 + \frac{c_2 w_2}{c_1 w_1} f_2 = \frac{1}{c_1 w_1}(w_1 \varphi_1 + w_2 \varphi_2) = \frac{\varphi}{c_1 w_1}.$$

Il suffit donc de poser $c = c_1 w_1$. De plus, $|\mathcal{B}| = |\mathcal{B}_1| + |\mathcal{B}_2| - 1 = t + 2$ comme pour le cas de la multiplication.

La deuxième partie du lemme est assurée par la construction effectuée pour les portes d'addition. En effet, le puits de $\mathcal{B}_2$ est de degré sortant 1 et l'arc de ce puits vers le puits de $\mathcal{B}_1$ a une constante comme poids. □

On peut appliquer le théorème à une formule non décorée et à la formule réduite d'une formule φ pour obtenir les résultats suivants. En particulier, on vérifie que pour une formule non décorée, les constantes dans la preuve précédente valent toutes 1.

Corollaire 3.4

Toute formule de taille t peut être représentée par un programme à branchements de taille $(t + 2)$.

Soit φ une formule de taille réduite t_r. Alors il existe un programme à branchements $\mathcal{B}$ à $(t_r + 2)$ sommets et une constante c tels que $\varphi = c f_{\mathcal{B}}$ où $f_{\mathcal{B}}$ est le polynôme représenté par $\mathcal{B}$. De plus, si φ possède au moins une porte d'addition, il existe dans $\mathcal{B}$ un sommet de degré sortant 1 dont le seul arc sortant a un poids constant.

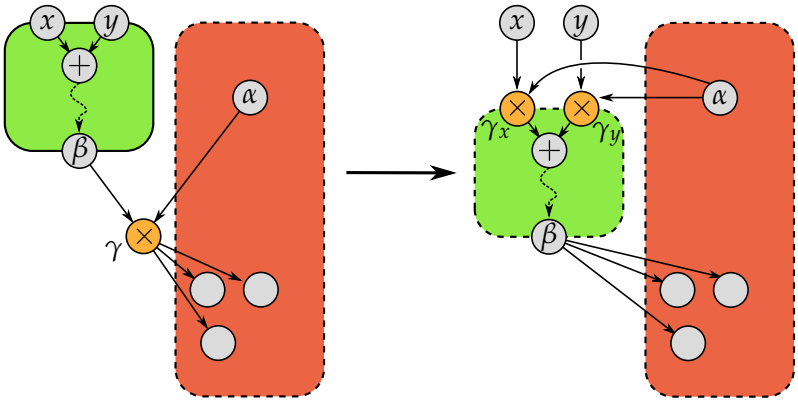


FIGURE 3.5 – Élimination d'une porte de multiplication faiblement asymétrique.

3.2.2 *Circuits asymétriques*

La prochaine proposition fait le lien entre les circuits asymétriques et faiblement asymétriques. Cette proposition est due à Erich L. Kaltofen et Pascal Koiran [61], et indépendamment à Maurice Jansen [57]. La preuve de Kaltofen et Koiran passe par les programmes à branchements. Celle de Jansen est une construction directe d'un circuit asymétrique à partir d'un circuit faiblement asymétrique mais la borne obtenue est moins bonne que celle de Kaltofen et Koiran. Nous donnons ici une nouvelle preuve, directe, de ce résultat. La borne obtenue est plus précise que celle de Kaltofen et Koiran. On remarque également que la proposition suivante s'étend facilement au cas des circuits décorés.

Proposition 3.5

> Tout circuit faiblement asymétrique de taille t avec e entrées peut être représenté par un circuit asymétrique possédant les mêmes entrées et de taille au plus $(t + e - 1)$.

Démonstration : Nous donnons un algorithme permettant de transformer tout circuit faiblement asymétrique $\mathcal{C}$ en un circuit asymétrique $\mathcal{C}'$ équivalent. Soit γ une porte de multiplication faiblement asymétrique, et α et β ses arguments. Supposons que $\mathcal{C}_\beta$ est un sous-circuit clos. De manière inductive, on peut supposer de plus que $\mathcal{C}_\beta$ est un circuit asymétrique. On construit alors $\mathcal{C}'$ en supprimant γ et en ne rajoutant que des portes asymétriques. Ainsi, par induction toutes les portes faiblement asymétriques sont supprimées et le circuit résultant est lui-même asymétrique. La figure 3.5 représente la transformation effectuée.

Formellement, le circuit $\mathcal{C}'$ est obtenu de la manière suivante. Tout d'abord, on supprime γ. Les successeurs de γ dans $\mathcal{C}$ deviennent successeurs de β, c'est-à-dire que toute flèche de la forme $\gamma \to \delta$ dans $\mathcal{C}$ est

remplacée par $\beta \to \delta$. Ainsi, $\mathcal{C}'_\beta$ n'est pas un sous-circuit clos. Ensuite, on considère le sous-circuit $\mathcal{C}_\beta$. Chaque porte de multiplication a au moins un argument qui est une entrée n'émettant qu'une seule flèche puisque $\mathcal{C}_\beta$ est supposé asymétrique. Nous appellerons ce type d'entrée des *entrées isolées*. Si les deux arguments ont cette propriété, seul l'un des deux, choisi arbitrairement, est dit isolé. On note qu'une entrée reliée à une porte d'addition n'est donc jamais isolée. Pour chaque entrée non isolée d'étiquette x (que l'on appellera porte x dans la suite), on ajoute une porte de multiplication γ_x dont les deux arguments sont les portes x et α (on rappelle que α était le deuxième argument de γ dans $\mathcal{C}$). Enfin, les flèches du type $x \to \delta$ sont remplacées par des flèches $\gamma_x \to \delta$. Clairement, les portes de multiplications rajoutées sont asymétriques puisqu'un de leurs arguments est l'entrée x, et que cette entrée n'émet qu'une seule flèche. De plus, les portes asymétriques de $\mathcal{C}_\beta$ restent asymétriques.

Le polynôme $f_\gamma^{\mathcal{C}}$ représenté par la porte γ dans $\mathcal{C}$ vaut $f_\alpha^{\mathcal{C}} f_\beta^{\mathcal{C}}$. Dans $\mathcal{C}'$, la porte α représente le même polynôme que dans $\mathcal{C}$, autrement dit $f_\alpha^{\mathcal{C}'} = f_\alpha^{\mathcal{C}}$. Il faut donc montrer que le polynôme $f_\beta^{\mathcal{C}'}$ représenté par β dans $\mathcal{C}'$ est $f_\alpha^{\mathcal{C}} f_\beta^{\mathcal{C}}$. Une induction facile sur la profondeur montre en fait que pour toute porte de calcul δ dans $\mathcal{C}_\beta$, le polynôme représenté par δ dans $\mathcal{C}'$ est $f_\delta^{\mathcal{C}'} = f_\alpha^{\mathcal{C}} f_\delta^{\mathcal{C}}$.

Il ne reste plus qu'à estimer la taille du circuit obtenu. Cette estimation est faite de manière globale sur $\mathcal{C}'$ et non à chaque étape. Les seules portes rajoutées sont les portes de multiplications γ_x pour les entrées non isolées d'un sous-circuit $\mathcal{C}_\beta$. Ce faisant, ces entrées non isolées deviennent des entrées isolées. De plus, aucune entrée non isolée n'est créée. Cela signifie que pour chaque entrée du circuit $\mathcal{C}$, une seule porte de multiplication peut être ajoutée au cours de l'algorithme. Ainsi, au plus e nouvelles portes sont ajoutées. De plus, si au moins une nouvelle porte est ajoutée, c'est qu'une porte de multiplication γ a été supprimée. Le nouveau nombre total de portes de calcul est donc au plus $(t + e - 1)$. □

Cette analyse de la taille de la formule obtenue est un peu plus fine que celle effectuée dans [63]. Il y est indiqué qu'un circuit faiblement asymétrique de taille grossière $m = t + e$ peut être transformé en un circuit asymétrique de taille grossière au plus $2m$. Puisque le nombre d'entrées d'un circuit est borné par le nombre de portes de calcul, nous obtenons ici la borne $3m/2$.

Notre borne peut en fait être un peu affinée. Supposons que l'on veuille traiter une porte faiblement asymétrique γ telle que le sous-circuit clos $\mathcal{C}_\beta$ possède un très grand nombre d'entrées, mais que $\mathcal{C}_\alpha$ soit un très petit sous-circuit. Il est alors préférable de dupliquer $\mathcal{C}_\alpha$ pour qu'une des deux copies devienne un argument clos de γ puis d'appliquer la construction précédente à $\mathcal{C}_\alpha$. Si $\mathcal{C}_\alpha$ possède e_α entrées et t_α portes de calcul, le circuit obtenu possédera $(e + e_\alpha)$ portes d'entrée et $(t + t_\alpha + e_\alpha)$ portes de calcul. De plus, les entrées rajoutées sont toutes isolées, donc cette augmentation

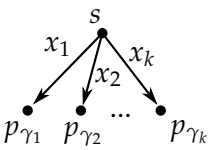


FIGURE 3.6 – Cas de base pour représenter un circuit asymétrique par un programme à branchements.

du nombre d'entrées n'a pas d'incidence sur la suite de l'algorithme. Cette solution est avantageuse dès que $t_\alpha + e_\alpha < e_\beta$ où e_β est le nombre d'entrées de $\mathcal{C}_\beta$. Cet affinement permet d'obtenir un circuit un peu plus compact que celui construit dans la preuve de la proposition 3.5 mais a l'inconvénient de ne pas conserver la structure du circuit original.

On a vu précédemment que les circuits asymétriques et faiblement asymétriques ont la même expressivité. Les programmes à branchements ont également ce même pouvoir d'expression. Les résultats suivants sont basés sur ceux de Malod et Portier [96, 97].

Théorème 3.6

> Tout circuit décoré (faiblement) asymétrique de taille t avec e entrées ayant au moins une porte d'addition peut être représenté par un programme à branchements ayant $(t + e + 1)$ sommets et $(2t + e)$ arcs.

Démonstration : La preuve est effectuée pour les circuits faiblement asymétrique, et reste donc valide pour les circuits asymétriques. Soit $\mathcal{C}$ un circuit faiblement asymétrique. On cherche à construire un programme à branchements $\mathcal{B}$ représentant le même polynôme que $\mathcal{C}$. Dans la suite, $f_\gamma^{\mathcal{C}}$ est le polynôme représenté par une porte γ dans $\mathcal{C}$ et $w_{\mathcal{B}}(s, p_\gamma)$ est la somme des poids des chemins entre la source s et un sommet p_γ de $\mathcal{B}$. C'est donc le polynôme représenté par le sommet p_γ dans $\mathcal{B}$. Par induction sur la taille t de $\mathcal{C}$, on construit $\mathcal{B}$ tel que pour toute porte réutilisable γ de $\mathcal{C}$, il existe un sommet p_γ dans $\mathcal{B}$ et une constante c_γ tels que $f_\gamma^{\mathcal{C}} = c_\gamma w_{\mathcal{B}}(s, p_\gamma)$. On montre finalement que si $\mathcal{C}$ a au moins une porte d'addition, on peut supprimer cette constante c_γ pour la porte de sortie. Les circuits considérés ont donc plusieurs sorties, et les programmes à branchements plusieurs puits.

Si $t = 0$, le circuit consiste simplement en un ensemble d'entrées $\{\gamma_1, \ldots, \gamma_k\}$ étiquetées respectivement x_1, x_2, …, x_k. Alors $\mathcal{B}$ possède une source s ainsi qu'un sommet p_{γ_i} et un arc $s \to p_{\gamma_i}$ de poids x_i pour $1 \leq i \leq k$ (Fig. 3.6). On vérifie aisément l'hypothèse d'induction pour $\mathcal{B}$ avec $c_{\gamma_i} = 1$ pour tout i.

Soit $t > 0$, et γ une porte de sortie de $\mathcal{C}$. Notons α et β ses arguments. Alors $f_\gamma^{\mathcal{C}} = w_\alpha f_\alpha^{\mathcal{C}} \star w_\beta f_\beta^{\mathcal{C}}$ où $\star$ est l'étiquette de γ et w_α et w_β sont les poids des flèches de α et β vers γ.

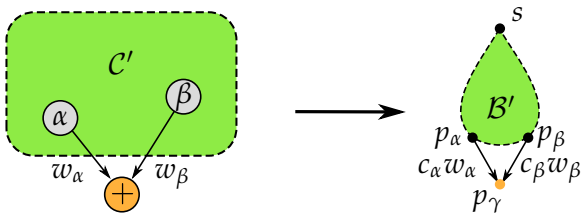


FIGURE 3.7 – Construction pour une porte d'addition dont les deux arguments sont distincts.

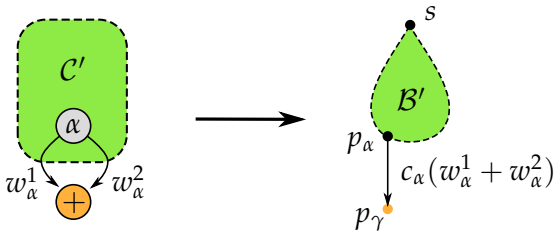


FIGURE 3.8 – Construction pour une porte d'addition dont les deux arguments sont confondus.

Si γ est une porte d'addition, soit $\mathcal{C}'$ le circuit obtenu en supprimant γ de $\mathcal{C}$ et $\mathcal{B}'$ le programme à branchements représentant $\mathcal{C}'$. Soit p_α et p_β les sommets de $\mathcal{B}'$ correspondant à α et β. Puisque γ est une sortie, α et β sont des portes réutilisables, donc il existe c_α et c_β telles que $c_\alpha w_{\mathcal{B}'}(s, p_\alpha) = f_\alpha^{\mathcal{C}'}$ et $c_\beta w_{\mathcal{B}'}(s, p_\beta) = f_\beta^{\mathcal{C}'}$. Pour construire $\mathcal{B}$, on ajoute à $\mathcal{B}'$ un sommet p_γ et deux arcs de poids respectifs $c_\alpha w_\alpha$ et $c_\beta w_\beta$ de p_α à p_γ et de p_β à p_γ (Fig. 3.7). Si $\alpha = \beta$, on met un unique arc de poids $c_\alpha(w_\alpha^1 + w_\alpha^2)$ (Fig. 3.8). Alors les chemins de s à p_γ sont l'union disjointe des chemins de s à p_α suivis de l'arc $p_\alpha \to p_\gamma$ et de ceux de s à p_β suivis de l'arc $p_\beta \to p_\gamma$. Ainsi,

$$w_{\mathcal{B}}(s, p_\gamma) = c_\alpha w_\alpha w_{\mathcal{B}}(s, p_\alpha) + c_\beta w_\beta w_{\mathcal{B}}(s, p_\beta) = f_\gamma^{\mathcal{C}}.$$

La constante $c_\gamma = 1$ convient donc. Le circuit $\mathcal{C}'$ ayant $(t-1)$ portes de calcul et e entrées, $\mathcal{B}'$ a $(t+e)$ sommets et $(2t+e-2)$ arcs. Donc $\mathcal{B}$ a $(t+e+1)$ sommets et $(2t+e)$ arcs.

Si γ est une porte de multiplication, l'un de ses arguments, disons α, est l'unique sortie d'un sous-circuit clos $\mathcal{C}_\alpha$. Le circuit obtenu à partir de $\mathcal{C}$ en enlevant γ est donc constitué de deux composantes connexes, $\mathcal{C}_\alpha$ d'une part, et un circuit $\mathcal{C}'$ de l'autre (qui contient le deuxième argument β de γ). Notons t_α et e_α la taille et le nombre d'entrées de $\mathcal{C}_\alpha$, et similairement t' et e' les paramètres de $\mathcal{C}'$. Ainsi, $t_\alpha + t' = t - 1$ et $e_\alpha + e' = e$. Par hypothèse d'induction, il existe deux programmes à branchements $\mathcal{B}_\alpha$ et $\mathcal{B}'$ représentant respectivement $\mathcal{C}_\alpha$ et $\mathcal{C}'$. Dans $\mathcal{C}_\alpha$, la porte α est une sortie, et donc réutilisable. Ainsi, $\mathcal{B}_\alpha$ possède une source s_α et un sommet p_α

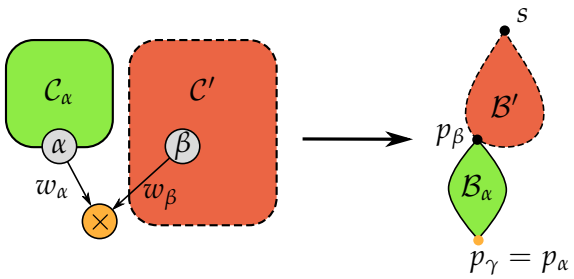


FIGURE 3.9 – Construction pour une porte de multiplication.

tels que $c_\alpha w_{\mathcal{B}_\alpha}(s_\alpha, p_\alpha) = f_\alpha^{\mathcal{C}}$. De même, il existe une source s' et sommet p_β dans $\mathcal{B}'$ tels que $c_\beta w_{\mathcal{B}'}(s', p_\beta) = f_\beta^{\mathcal{C}}$. On construit $\mathcal{B}$ comme l'union de $\mathcal{B}_\alpha$ et $\mathcal{B}'$ dans laquelle les sommets s_α et p_β sont identifiés (Fig. 3.9). Le sommet p_γ est alors le sommet p_α et la source s de $\mathcal{B}$ est la source s' de $\mathcal{B}'$. Les portes réutilisables de $\mathcal{C}$ sont γ et celles de $\mathcal{C}'$. Si δ est une porte réutilisable de $\mathcal{C}'$, les chemins de s à p_δ ne sont pas modifiés. Les chemins de s à p_γ sont constitués de deux parties : la première de s à p_β, puis la seconde de s_α (qui est confondu avec p_β) à p_γ. Donc $w_{\mathcal{B}}(s, p_\gamma) = w_{\mathcal{B}}(s, p_\beta) w_{\mathcal{B}}(s_\alpha, p_\alpha)$. Si on pose $c_\gamma = c_\alpha w_\alpha c_\beta w_\beta$, on a donc $c_\gamma w_{\mathcal{B}}(s, p_\gamma) = f_\gamma^{\mathcal{C}}$. La taille de $\mathcal{B}_\alpha$ est $(t_\alpha + e_\alpha + 1)$ et celle de $\mathcal{B}'$ est $(t' + e' + 1)$ donc celle de $\mathcal{B}$ est $(t_\alpha + e_\alpha + 1) + (t' + e' + 1) - 1 = t + e - 1$. De même, le nombre d'arcs est $(2t_\alpha + e_\alpha) + (2t' + e') = 2t + e - 2$.

Pour conclure, on a obtenu un programme à branchements $\mathcal{B}$ et une constante c tels que $c w_{\mathcal{B}}(s, p) = f_{\mathcal{C}}$. Supposons que $\mathcal{C}$ a au moins une porte d'addition. Considérons la porte d'addition γ la plus proche de la sortie de $\mathcal{C}$, et $\mathcal{B}_\gamma$ le programme à branchements construit pour le sous-circuit $\mathcal{C}_\gamma$. Puisqu'il n'existe après γ que des portes de multiplication, tous les chemins de s à p dans $\mathcal{B}$ passent par $\mathcal{B}_\gamma$. Ainsi, en multipliant les poids des arcs incidents à p_γ dans $\mathcal{B}_\gamma$ par la constante c, les poids de tous les chemins de s à p dans $\mathcal{B}$ sont multipliés par c. En d'autres termes, on obtient un nouveau programme à branchements $\mathcal{B}$ tel que $w_{\mathcal{B}}(s, p) = f_{\mathcal{C}}$. □

Corollaire 3.7

Tout circuit faiblement asymétrique (non décoré) de taille t avec e entrées peut être représenté par un programme à branchements ayant $(t + e + 1)$ sommets et $(2t + e)$ arcs.

Tout circuit faiblement asymétrique de taille réduite t_r avec v variables et ayant au moins une porte d'addition non constante peut être représenté par un programme à branchements ayant $(t_r + v + 1)$ sommets et $(2t_r + v)$ arcs.

Ébauche de démonstration : Pour la première partie, on applique le théorème précédent en remarquant que si $\mathcal{C}$ n'est pas décoré, alors toutes les

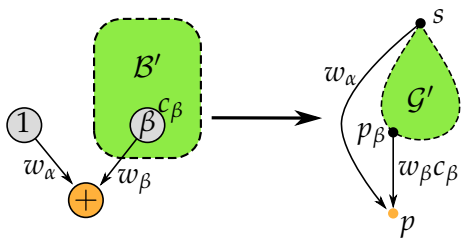


FIGURE 3.10 – Représentation d'une porte d'addition ayant une entrée constante dans un circuit réduit.

constantes c_γ manipulées valent 1. En particulier, il n'est pas nécessaire de supposer l'existence d'une porte d'addition.

Pour la seconde partie, on applique le théorème précédent au circuit réduit d'un circuit faiblement asymétrique avec une légère modification. Tout d'abord, on note que si un circuit $\mathcal{C}$ a au moins une porte d'addition non constante, alors son circuit réduit $\mathcal{C}_r$ a au moins une porte d'addition. Si une entrée est étiquetée 1, on ne la traite pas dans le cas de base (autrement dit, on se restreint dans le cas de base aux entrées étiquetées par des variables). Soit γ une porte d'addition dont un argument est une entrée étiquetée 1 et l'autre argument est une porte β. Alors le circuit obtenu en supprimant γ et l'entrée 1 peut être représenté par un programme à branchements $\mathcal{B}'$ ayant $(t_r + v)$ sommets et $(2t_r + v - 2)$ arcs, de source s et puits p_β. On construit un programme à branchements $\mathcal{B}$ en ajoutant un sommet p à $\mathcal{B}$, et deux arcs $s \to p$ et $p_\beta \to p$ avec les poids appropriés (Fig. 3.10). Alors $\mathcal{B}$ a bien $(t_r + v + 1)$ sommets et $(2t_r + v)$ arcs. □

Nous nous intéressons maintenant à la réciproque du théorème précédent, à savoir comment transformer un programme à branchements en circuit asymétrique équivalent.

Proposition 3.8

> Tout programme à branchements ayant n sommets et m arcs peut être représenté par un circuit asymétrique de taille $(2m - n)$ avec m entrées.

Démonstration : On raisonne par induction sur le nombre n de sommets. Soit donc $\mathcal{B}$ un programme à branchements de taille n que l'on souhaite représenter par un circuit asymétrique $\mathcal{C}$. De même que dans la preuve précédente, on s'intéresse à des circuits et des programmes à branchements ayant plusieurs sorties. Ainsi, pour chaque puits p de $\mathcal{B}$, $\mathcal{C}$ possède une porte γ_p telle que $w_\mathcal{B}(s, p) = f^\mathcal{C}_{\gamma_p}$.

Le plus petit programme à branchements possible a deux sommets et un arc les reliant. Alors $\mathcal{C}$ est réduit à une entrée dont l'étiquette est le poids de l'arc. C'est bien un circuit de taille 0 avec 1 entrée.

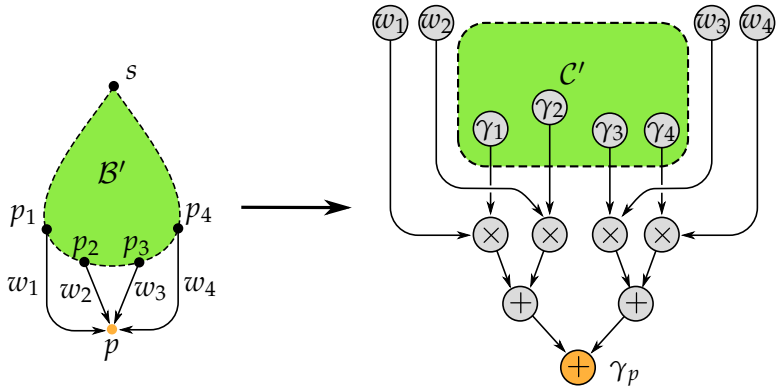


FIGURE 3.11 – Traitement d'un sommet de degré entrant 4.

Supposons maintenant que $\mathcal{B}$ a $n > 2$ sommets. Soit p un puits de $\mathcal{B}$ et $\mathcal{B}' = \mathcal{B} \setminus \{p\}$. Notons k le degré entrant de p. Alors $\mathcal{B}'$ possède $(n-1)$ sommets et $(m-k)$ arcs. Par induction, il est représenté par un circuit $\mathcal{C}'$ de taille $(2(m-k)-n+1)$ ayant $(m-k)$ entrées. Notons $p_1, \ldots, p_k$ les sommets de $\mathcal{B}'$ dont sont issus les arcs entrants de p, et pour $1 \leq i \leq k$ notons w_i le poids de l'arc $p_i \to p$. On construit $\mathcal{C}$ en ajoutant pour tout i une porte de multiplication dont un argument est l'entrée w_i et l'autre la porte γ_i représentant le puits p_i, puis $(k-1)$ portes d'addition pour sommer ces k portes de multiplication (Fig. 3.11). On note γ_p la plus profonde de ces portes d'addition. Puisqu'on a $w_{\mathcal{B}}(s, p_i) = f^{\mathcal{C}}_{\gamma_i}$, on obtient

$$f^{\mathcal{C}}_{\gamma_p} = \sum_{i=1}^{k} w_i f^{\mathcal{C}}_{\gamma_i} = \sum_i w_i w_{\mathcal{B}}(s, p_i) = w_{\mathcal{B}}(s, p).$$

La taille de $\mathcal{C}$ est $t(\mathcal{C}') + 2k - 1 = 2m - n$ et le nombre d'entrées est $e(\mathcal{C}') + k = m$. □

On remarque qu'on pourrait diminuer un peu la taille du circuit obtenu en autorisant le circuit à être décoré.

3.3 Complexité du déterminant

Pour étudier la complexité du déterminant, on est amené à voir le déterminant comme une nouvelle représentation des polynômes. La définition 1.16 introduit la notion de projection entre polynômes. On rappelle cette définition dans le cas particulier du polynôme Det_m.

Un polynôme $f \in \mathbb{K}[X_1, \ldots, X_n]$ est une *projection* du déterminant Det_m s'il existe une matrice carrée $\mathcal{M}$ de dimension m dont les coefficients appartiennent à $\mathbb{K} \cup \{X_1, \ldots, X_n\}$ telle que $f = \det(\mathcal{M})$.

3.3.1 Expressivité du déterminant

On montre d'abord comment transformer un programme à branchements en un déterminant.

Proposition 3.9

Tout programme à branchements ayant n sommets est une projection de $\textsc{Det}_n$. De plus, cette projection n'utilise comme élément de $\mathbb{K}$ que les constantes 0, ± 1 et les poids des arcs du programme à branchements.

Démonstration : Soit $\mathcal{B}$ un programme à branchements à n sommets dont sa source s et son puits p. On considère le graphe $\mathcal{G}$ obtenu en ajoutant un arc de poids $(-1)^{n+1}$ de p vers s et une boucle de poids -1 sur tous les sommets différents de s et p. Les couvertures par cycles de $\mathcal{G}$ sont en bijection avec les chemins de s à p dans $\mathcal{B}$. Le poids d'une couverture par cycles associée à un chemin π de s à p est $(-1)^{n+1}(-1)^{n-|\pi|}w(\pi) = (-1)^{|\pi|+1}w(\pi)$. Et sa signature est $(-1)^{|\pi|+1}$. Ainsi, une telle couverture apporte la contribution $w(\pi)$ au calcul du déterminant de la matrice $\mathcal{M}$ d'adjacence de $\mathcal{G}$. Donc $\det(\mathcal{M}) = w_{\mathcal{B}}(s,p)$ et $\mathcal{M}$ représente le programme à branchements $\mathcal{B}$. □

Corollaire 3.10

Tout circuit faiblement asymétrique de taille t avec e entrées est une projection de $\textsc{Det}_m$ avec $m = t + e + 1$.

Tout circuit faiblement asymétrique de taille réduite t_r avec v variables et ayant au moins une porte d'addition non constante est une projection de $\textsc{Det}_m$ avec $m = t_r + v + 1$.

On s'intéresse maintenant à l'expressivité du déterminant pour les formules. On pourrait appliquer directement la proposition précédente avec le corollaire 3.4 mais le résultat suivant donne une meilleure borne, qui égale en particulier celle de Liu et Regan [88].

Théorème 3.11

Toute formule φ de taille réduite t_r ayant au moins une porte d'addition non constante est une projection de $\textsc{Det}_m$ avec $m = t_r + 1$.

Démonstration : Soit $\mathcal{B}$ le programme à branchements représentant la formule φ construit grâce au corollaire 3.4. En utilisant la proposition 3.9, on obtiendrait une matrice $\mathcal{M}$, de dimension $(t_r + 2)$ dont le déterminant égale la valeur de $\mathcal{B}$. La construction faite dans la proposition 3.9 consiste à rajouter un arc du puits vers la source de $\mathcal{B}$ et des boucles sur les autres sommets. Pour obtenir la borne voulue, on identifie à la place le puits et la source, et on ajoute des boucle de poids -1 sur tous les autres sommets. Le déterminant de la matrice d'adjacence du graphe ainsi obtenu vaut $\pm f_{\mathcal{B}}$ où $f_{\mathcal{B}}$ est le polynôme représenté par $\mathcal{B}$. Le corol-

laire 3.4 assure l'existence d'un sommet v de degré sortant 1. On peut alors multiplier le poids de la boucle sur v et le poids de l'arc sortant de v par $\pm c$ où c est la constante associée à $\mathcal{B}$ dans le corollaire 3.4. Dans la matrice d'adjacence $\mathcal{M}$ du graphe obtenu, il n'y a que deux coefficients non nuls dans la ligne v, tous les deux multipliés par $\pm c$. Ainsi, $\det(\mathcal{M}) = \pm c \times (\pm f_{\mathcal{B}}) = c f_{\mathcal{B}} = \varphi$. □

Si φ est une formule sans addition, alors elle est de la forme $cx_1 \cdots x_n$. Sa taille réduite est donc $(n-1)$. On peut prendre pour $\mathcal{M}$ la matrice diagonale contenant c et les variables x_i. Si $c = 1$, on peut ne pas l'inclure dans la matrice et on obtient une matrice de dimension $n = t_r + 1$. Sinon, on obtient de cette manière une matrice de dimension $n + 1 = t_r + 2$. Cette méthode n'est pas nécessairement optimale comme le prouve l'exemple de la formule $2xyz$ représentée par la matrice

$$\begin{pmatrix} 0 & x & y \\ x & 0 & z \\ y & z & 0 \end{pmatrix}.$$

Cependant, la borne du théorème n'est pas toujours atteignable lorsqu'il n'y a pas d'addition. Ainsi, il n'existe pas de matrice de dimension 2×2 représentant la formule $2xy$.

3.3.2 *Un programme à branchements pour le déterminant*

Nous présentons dans cette partie un programme à branchements pour le déterminant $\textsc{Det}_n$. Cette proposition est due à Meena Mahajan et V Vinay [93]. Plus précisément, ils montrent l'existence d'un programme à branchements pour représenter le déterminant ayant $(2n^3 + 3)$ sommets et $4n^4$ arcs. Nous obtenons ici des bornes plus fines, pour deux raisons : la première est que l'on peut un peu diminuer la taille du programme à branchements qu'ils obtiennent, et la deuxième est que l'on procède ici à une analyse plus fine de cette taille.

La proposition peut être facilement étendue pour calculer tout le polynôme caractéristique [93]. De plus, Mahajan et Vinay ont explicité les liens entre leur construction et les algorithmes de Samuelson-Berkowitz, Csansky et Chistov [95]. Enfin, Mahajan, Subramanya et Vinay ont également étendu la construction au calcul du Pfaffien d'une matrice antisymétrique, montrant l'appartenance de ce problème à la classe NC [94].

Proposition 3.12

> Le polynôme $\textsc{Det}_n$ peut être représenté par un programme à branchements ayant $(\frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n + 3)$ sommets et $(\frac{1}{4}n^4 - \frac{1}{4}n^2 + n + 1)$ arcs.

La figure 3.12 présente le programme à branchements obtenu en appliquant la proposition 3.12 avec $n = 3$.

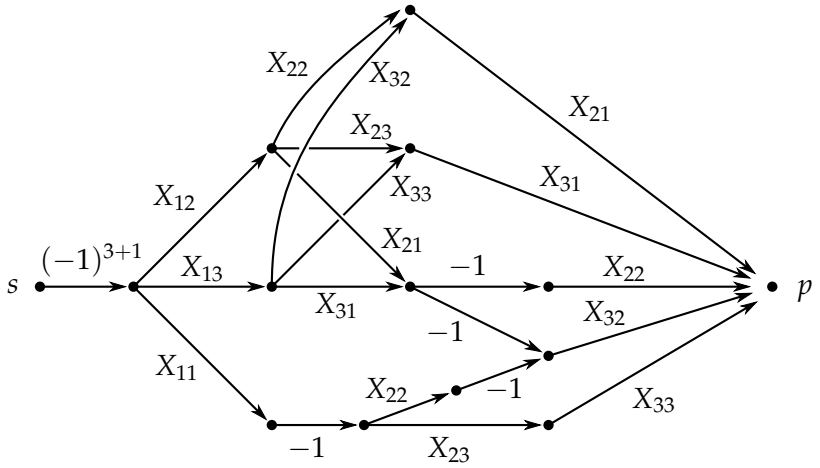


FIGURE 3.12 – Programme à branchements représentant le déterminant DET_3.

Pour prouver la proposition, il nous faut introduire les notions de *promenade*[2] et de *randonnée*[3].

Définition 3.13

Soit $\mathcal{G} = (V, A)$ un graphe orienté où $V = \{v_1, \ldots, v_n\}$ est ordonné selon l'ordre $v_1 < v_2 < \cdots < v_n$.

Une *promenade au départ de* v_i est une suite de sommets $(u_1, \ldots, u_\ell)$ telle que $u_1 = u_\ell = v_i$, $u_j > v_i$ pour $1 < j < \ell$, et $(u_j, u_{j+1}) \in A$ pour tout j. Le sommet v_i est le *point de départ* de la promenade.

Une *randonnée* est une suite ordonnée de promenades $\mathcal{P}_1, \ldots, \mathcal{P}_k$ telle que pour tout $i < k$, le point de départ de $\mathcal{P}_i$ est strictement inférieur au point de départ de $\mathcal{P}_{i+1}$, et telle que le nombre total d'arc utilisés (comptés avec multiplicité) est exactement n.

On remarque que les couvertures par cycles sont des cas particuliers de randonnées. Si $\mathcal{G}$ possède des poids sur les arcs, on définit le poids $w(\mathcal{R})$ d'une randonnée $\mathcal{R}$ comme le produit des poids des arcs utilisés (toujours avec multiplicité). On définit aussi la signature $\epsilon(\mathcal{R})$ d'une randonnée, compatible avec celle des couvertures par cycles, par $(-1)^{n+k}$ où n est le nombre total de sommets de $\mathcal{G}$, et k est le nombre de promenades dans la randonnée.

On peut alors exprimer le déterminant en terme de promenades. La

2. *Closed walk*, ou *clow*, en anglais.
3. *Clow sequence*, en anglais.

proposition suivante est également due à Mahajan et Vinay [93].

Proposition 3.14

Soit $\mathcal{G}$ un graphe orienté et $\mathcal{M}$ sa matrice d'adjacence. Alors

$$\det(\mathcal{M}) = \sum_{\mathcal{R}} \epsilon(\mathcal{R}) w(\mathcal{R}),$$

où la somme porte sur toutes les randonnées du graphe $\mathcal{G}$.

Démonstration : Comme indiqué précédemment, les couvertures par cycles sont des randonnées particulières. Il suffit donc de montrer que les contributions des randonnées qui ne sont pas des couvertures par cycles s'annulent les unes les autres. Ainsi, il ne reste dans la somme que les couvertures par cycles, d'où le résultat.

Pour cela, on définit une involution sur l'ensemble des promenades, c'est-à-dire une bijection ϕ telle que $\phi \circ \phi$ est l'identité. Cette involution a les propriétés suivantes : $\phi(\mathcal{R}) = \mathcal{R}$ si et seulement si $\mathcal{R}$ est une couverture par cycles, et $w(\phi(\mathcal{R})) = w(\mathcal{R})$ et $\epsilon(\phi(\mathcal{R})) = -\epsilon(\mathcal{R})$ si $\mathcal{R}$ n'est pas une couverture par cycles. De cette manière, la contribution d'une randonnée $\mathcal{R}$ qui n'est pas une couverture par cycles s'annule avec celle de $\phi(\mathcal{R})$. On commence par fixer $\phi(\mathcal{R}) = \mathcal{R}$ pour les couvertures par cycles.

Soit $\mathcal{R} = (\mathcal{P}_1, \ldots, \mathcal{P}_k)$ une randonnée qui n'est pas une couverture par cycles. Alors il existe nécessairement un sommet apparaissant deux fois dans la randonnée. Soit i le plus petit indice tel que $(\mathcal{P}_{i+1}, \ldots, \mathcal{P}_k)$ est un ensemble de cycles disjoints. Posons $\mathcal{P}_i = (u_1, \ldots, u_\ell)$ et soit u_j le premier sommet de $\mathcal{P}_i$ qui est égal soit à un autre sommet $u_{j'}$ de cette promenade avec $j' < j$, soit à un sommet d'un des cycles $\mathcal{P}_{i+1}$, …, $\mathcal{P}_k$.

Dans le premier cas, la randonnée $\phi(\mathcal{R})$ est obtenue en supprimant le cycle $(u_{j'}, \ldots, u_j)$ de la promenade $\mathcal{P}_i$ et en l'ajoutant, à la bonne place, comme nouvelle promenade. Dans le second cas, on supprime le cycle rencontré de $\mathcal{R}$ et on l'insère dans la promenade $\mathcal{P}_i$ (Fig. 3.13). Dans les deux cas, $\phi(\mathcal{R})$ a même poids que $\mathcal{R}$ et signature opposée puisque le nombre de promenades dans la randonnée change de 1. On vérifie qu'on a bien défini une involution. Dans le premier cas, on ajoute à la fin de la randonnée un nouveau cycle qui est bien disjoint de $\mathcal{P}_{i+1}$, …, $\mathcal{P}_k$. Ainsi, dans $\phi(\mathcal{R})$, la première collision a lieu quand $\mathcal{P}_i$ rencontre ce nouveau cycle et $\phi^2(\mathcal{R}) = \mathcal{R}$. Dans le deuxième cas, on a enlevé un cycle à la fin de la promenade pour l'insérer dans $\mathcal{P}_i$. Dans $\phi(\mathcal{R})$, la première collision a donc lieu à l'endroit où le cycle a été inséré dans $\mathcal{P}_i$. □

On peut définir la notion de ℓ-randonnée pour $1 \leq \ell \leq n$: ce sont les suites ordonnées de promenades utilisant exactement ℓ arcs, de sorte qu'une n-randonnée est une randonnée. Dans la proposition 3.14, si on somme sur les ℓ-randonnées au lieu des randonnées, on obtient alors le coefficient du

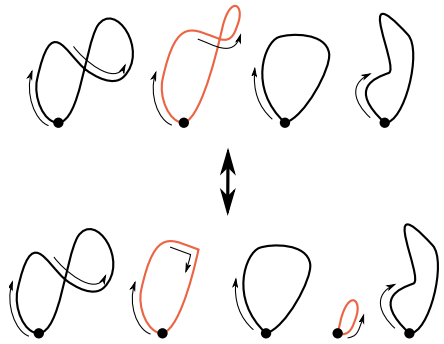


FIGURE 3.13 – Illustration de l'involution pour les promenades.

monôme de degré $(n - \ell)$ du polynôme caractéristique de $\mathcal{M}$ [93].

Cette nouvelle caractérisation du déterminant permet de définir un programme à branchements pour le représenter.

Démonstration de la proposition 3.12 : Notre construction est adaptée de celle de Mahajan et Vinay [93]. Nous commençons par donner une construction relativement semblable à la leur, puis nous montrons qu'on peut en fait légèrement diminuer la taille du programme à branchements obtenu. Ceci nous permet de faire ensuite une analyse précise de la taille pour obtenir une borne légèrement meilleure que la borne originale. Pour diminuer la taille du programme obtenu, nous montrons en particulier que dans la proposition 3.14, la somme peut être restreinte à certaines randonnées particulières.

Soit $\mathcal{X} = (X_{ij})_{1 \leq i,j \leq n}$ la matrice dont tous les coefficients sont des indéterminées. Pour définir un programme à branchements $\mathcal{B}$ représentant $\det(\mathcal{X})$, on s'intéresse au graphe orienté $\mathcal{G}$ dont $\mathcal{X}$ est la matrice d'adjacence. Les sommets de $\mathcal{G}$ sont numérotés de 1 à n et on confond un sommet et son numéro. On souhaite associer à chaque randonnée de $\mathcal{G}$ un unique chemin de la source s au puits p de $\mathcal{B}$. Pour cela, on fait porter aux sommets de $\mathcal{B}$ des informations. Celles-ci sont essentiellement le sommet de $\mathcal{G}$ courant, la promenade sur laquelle on se trouve qu'on identifie par son point de départ, ainsi que le nombre d'arcs empruntés. De plus, lorsqu'on visite le point de départ d'une promenade, on souhaite savoir si on a déjà parcouru la promenade ou pas. Ainsi, l'ensemble des sommets de $\mathcal{B}$ est

$$\begin{aligned} S = \{s, p\} &\cup \{(u, v, i) : 1 \leq u < v \leq n, 1 \leq i \leq n-1\} \\ &\cup \{(u, u, i) : 1 \leq u \leq n, 0 \leq i \leq n-1\} \\ &\cup \{(u, u, i)^\star : 1 \leq u \leq n, 1 \leq i \leq n\}. \end{aligned}$$

Un sommet d'étiquette (u, v, i) avec $v > u$ porte l'information : le sommet courant est v, la promenade a pour point de départ u et i arcs ont été

utilisés. Nécessairement, pour atteindre v depuis u, il y a eu au moins un arc utilisé, et pour finir la promenade il faudra encore au moins un arc vers u, donc $1 \leq i \leq n-1$. Un sommet (u,u,i) porte l'information : le sommet courant est u, qui est le point de départ de la promenade courante, cette promenade n'a pas été encore parcourue et i arcs ont été utilisés. La promenade ayant au moins un arc, $i \leq n-1$. Enfin $(u,u,i)^\star$ porte la même information que (u,u,i) mais la promenade a été parcourue. Ainsi, au moins un arc a été utilisé et $i \geq 1$.

Pour associer un chemin de s à p à une randonnée $\mathcal{R} = (\mathcal{P}_1, \ldots, \mathcal{P}_k)$ de $\mathcal{G}$, on parcourt les promenades dans l'ordre. Les arcs sont les suivants :

(1) $s \to (u,u,0)$ où $1 \leq u \leq n$, de poids $(-1)^{n+1}$, pour initier une randonnée ;

(2) $(u,v,i) \to (u,w,i+1)$ où $v,w \geq u$, de poids X_{vw}, pour représenter l'arc $v \to w$ de $\mathcal{G}$ utilisé dans une promenade de point de départ u ;

(3) $(u,v,i) \to (u,u,i+1)^\star$ où $v \geq u$, de poids X_{vu}, pour représenter l'arc $v \to u$ dans une promenade de point de départ u ;

(4) $(u,u,i)^\star \to (v,v,i)$ où $v > u$, de poids -1, pour représenter le passage de la promenade de point de départ u à celle de point de départ v ;

(5) $(u,u,n)^\star \to p$, de poids 1, pour terminer la randonnée.

Ainsi définis, les chemins de s à p dans $\mathcal{B}$ sont en bijection avec les randonnées de $\mathcal{G}$. Pour une randonnée $\mathcal{R}$ de k promenades, le nombre d'arcs $(u,u,i)^\star \to (v,v,i)$ utilisés est $(k-1)$. Le signe du poids du chemin de s à p correspondant est donc $(-1)^{n+1}(-1)^{k-1} = \epsilon(\mathcal{R})$. Pour chaque arc $v \to w$ de $\mathcal{G}$ utilisé au cours de la randonnée $\mathcal{R}$, il y a un arc de type $(u,v,i) \to (u,w,i+1)$ ou $(w,v,i) \to (w,w,i+1)^\star$ de poids X_{vw}. Ceci montre que le poids du chemin de s à p est $\epsilon(\mathcal{R})w(\mathcal{R})$. Donc $\mathcal{B}$ représente $\textsc{Det}_n$. À ce stade, on a retrouvé le résultat de Mahajan et Vinay. On peut estimer le nombre de sommets et d'arcs pour retrouver leurs bornes. On montre maintenant comment obtenir les bornes plus précises de l'énoncé.

Pour diminuer la taille de $\mathcal{B}$, on fait une remarque simple. Les seules randonnées qui sont réellement utiles sont les couvertures par cycles. Celles-ci ont la propriété que le point de départ de leur première randonnée est le sommet 1. De plus, l'involution ϕ définie dans la preuve de la proposition 3.14 préserve le premier point de départ, c'est-à-dire que les points de départ des premières promenades de $\mathcal{R}$ et $\phi(\mathcal{R})$ sont identiques. Se restreindre aux randonnées commençant par le sommet 1 ne change donc pas la conclusion précédente. Ceci prouve qu'on peut ne considérer que l'arc $s \to (1,1,0)$ et supprimer de $\mathcal{B}$ les sommets $(u,u,0)$ et $(u,u,1)^\star$ pour $u > 1$, ainsi que $(1,1,i)$ pour $i > 0$.

De la même façon, les sommets (u,u,i) où $i < u-1$ sont inutiles. En effet, de tels sommets représentent des points de départ de promenade. Si $i < u-1$, il reste $(n-i)$ arcs à emprunter, et il y a $n-u+1 < n-i$ sommets distincts. L'un au moins des sommets $v \geq u$ est donc visité deux fois. Cette randonnée ne peut donc pas être une couverture par cycles. Il reste ensuite à se convaincre que l'involution ϕ préserve bien cette propriété, c'est-à-dire que si $\mathcal{R}$ a un point de départ u visité après avoir traversé $i < u-1$ arcs, alors il en est de même pour $\phi(\mathcal{R})$ (c'est suffisant puisque ϕ est une involution). Soit $\mathcal{R} = (\mathcal{P}_1, \ldots, \mathcal{P}_k)$ une randonnée et j l'indice maximal tel que le point de départ u de $\mathcal{P}_j$ vérifie $u > 1 + \sum_{\ell<j} |\mathcal{P}_\ell|$. Puisqu'il y a plus d'arcs à visiter que de sommets dans les promenades $\mathcal{P}_j, \ldots, \mathcal{P}_k$, il y a une collision parmi ces promenades et les promenades modifiées par l'involution ϕ ont un indice supérieur ou égal à j. En tout état de cause, le point de départ de $\mathcal{P}_j$ reste le même, et la propriété reste vérifiée pour $\phi(\mathcal{R})$. Ceci prouve qu'en se restreignant aux (u,u,i) où $i \geq u-1$, la proposition 3.14 reste valable. De plus, un chemin vers un sommet $(u,u,i)^\star$ passe nécessairement par $(u,u,i-1)$, donc les sommets $(u,u,i)^\star$ conservés sont ceux tels que $i \geq u$. De même, les sommets (u,v,i) sont tels que $i \geq u$. Enfin, on peut remplacer les arcs $(u,v,n-1) \to (u,u,n)^\star$ par des arcs $(u,v,n-1) \to p$ et supprimer les $(u,u,n)^\star$.

L'ensemble des sommets de $\mathcal{B}$ est donc constitué de s, p, $(1,1,0)$, (u,u,i) pour $2 \leq u \leq i+1 \leq n$, $(u,u,i)^\star$ pour $1 \leq u \leq i \leq n-1$, et (u,v,i) pour $1 \leq u \leq i \leq n-1$ et $1 \leq u < v \leq n$. Le programme à branchements $\mathcal{B}$ possède donc

$$3 + \frac{1}{2}n(n-1) + \frac{1}{2}n(n-1) + \frac{1}{6}n(n-1)(2n-1) = \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n + 3$$

sommets.

Il y a cinq sortes d'arcs différents décrits précédemment. Il y a un seul arc de type (1), puisque seul $s \to (1,1,0)$ est conservé. Les arcs de type (2) peuvent être classés en trois catégories. Il y en a $(n-1)$ de la forme $(1,1,0) \to (1,w,1)$. Il y a ensuite ceux de la forme $(u,u,i) \to (u,w,i+1)$ où $u \geq 2$. Ils vérifient $i+1 \geq u$ et $w > u$ et sont donc au nombre de $\frac{1}{3}n(n-1)(n-2)$. Enfin, ceux de la forme $(u,v,i) \to (u,w,i+1)$ avec $v,w > u$ vérifient $i \geq u$ et $i+1 \leq n-1$. Il y en a $\frac{1}{12}n(n-1)(n-2)(3n-1)$. Pour les arcs de type (3), il y a $(1,1,0) \to (1,1,1)^\star$, ceux de la forme $(u,u,i) \to (u,u,i+1)^\star$ avec $2 \leq u \leq i+1 \leq n$, et ceux de la forme $(u,v,i) \to (u,u,i+1)^\star$ avec $v > u$ et $i+1 \leq n$ — on compte ici les arcs vers p comme s'ils allaient toujours vers $(u,u,n)^\star$, et on ignorera donc les arcs de type (5). Leurs nombres sont respectivement 1, $\frac{1}{2}n(n-1)$ et $\frac{1}{6}n(n-1)(2n-1)$. Enfin, les arcs de type (4) sont de la forme $(u,u,i)^\star \to (v,v,i)$ et vérifient $1 \leq u < v \leq i+1 \leq n$. Il y en a donc $\frac{1}{6}n(n-1)(n+1)$.

Au total, le nombre d'arcs de $\mathcal{B}$ est

$$\frac{1}{4}n^4 - \frac{1}{4}n^2 + n + 1.$$

□

Corollaire 3.15

Le polynôme $\textsc{Det}_n$ peut être représenté par un circuit asymétrique de taille $(\frac{1}{4}n^4 - \frac{1}{3}n^3 + \frac{11}{6}n - 2)$ ayant $(\frac{1}{4}n^4 - \frac{1}{4}n^2 + n + 1)$ entrées.

Démonstration : C'est simplement l'application de la proposition 3.8 au programme à branchements obtenu précédemment. □

La complexité du déterminant est capturée de manière assez précise par les circuits asymétriques et les programmes à branchements. Le corollaire précédent donne le plus petit circuit asymétrique connu pour le déterminant, ainsi que le plus petit circuit faiblement asymétrique. On peut noter qu'on peut obtenir une meilleure taille de circuit si on ne se restreint pas aux circuits asymétriques. Ainsi, Erich L. Kaltofen et Gilles Villard ont montré qu'il existe des circuits de taille $\mathcal{O}(n^3)$ représentant le déterminant $\textsc{Det}_n$ [65]. Un tel circuit correspond à ce qu'on appelle souvent un algorithme *sans division*. Si on s'autorise les divisions, le meilleur algorithme connu actuellement est la version améliorée de l'algorithme de Don Coppersmith et Shmuel Winograd [28] proposée par Virginia Vassilevska Williams [125] (voir aussi la thèse d'Andrew Stothers [118]) qui est de complexité $\mathcal{O}(n^{2,3727})$ environ. Enfin, il est intéressant de noter qu'en se restreignant aux formules, on perd la polynomialité puisque la plus petite formule connue pour le déterminant est de taille $n^{\mathcal{O}(\log n)}$.

3.4 Complexité déterminantielle du permanent

On s'intéresse maintenant à la complexité déterminantielle du permanent. Il est conjecturé que le permanent $\textsc{Per}_n$ ne peut être représenté par le déterminant d'une matrice de dimension polynomiale en n, et même que sa complexité déterminantielle est $2^{\Omega(n)}$. Actuellement, la meilleure borne inférieure connue est due à Thierry Mignon et Nicolas Ressayre qui ont montré que la complexité déterminantielle de $\textsc{Per}_n$ est au moins $n^2/2$ en caractéristique 0 [99]. Cette borne a été étendue à tout corps de caractéristique différente de 2 par Jin-Yi Cai, Xi Cheng et Dong Li [20]. On remarque qu'en caractéristique 2, le déterminant et le permanent coïncident et donc la complexité déterminantielle de $\textsc{Per}_n$ est exactement n.

Les bornes supérieures connues se déduisent des formules de taille $\mathcal{O}(n2^n)$ de Herbert J. Ryser [110] et David G. Glynn [40], à l'aide du théorème 3.11. On obtient alors un déterminant de dimension $\mathcal{O}(n2^n)$ pour représenter $\textsc{Per}_n$. Cai, Cheng et Li affirment qu'avec un peu de travail supplémentaire, on peut obtenir une représentation déterminantielle du permanent

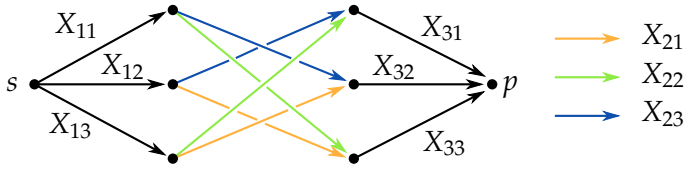


FIGURE 3.14 – Programme à branchements représentant le permanent PER_3.

de dimension $\mathcal{O}(2^n)$ mais sans donner de preuve [20]. Nous donnons une nouvelle construction d'une matrice de dimension $(2^n - 1)$ pour représenter PER_n. Par exemple, le permanent (3×3) peut être représenté par le déterminant d'une matrice de dimension 7 de la manière suivante :

$$\text{PER}_3 = \det \begin{pmatrix} 0 & X_{11} & X_{21} & X_{31} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & X_{33} & X_{23} & 0 \\ 0 & 0 & 1 & 0 & 0 & X_{13} & X_{33} \\ 0 & 0 & 0 & 1 & X_{13} & 0 & X_{23} \\ X_{22} & 0 & 0 & 0 & 1 & 0 & 0 \\ X_{32} & 0 & 0 & 0 & 0 & 1 & 0 \\ X_{12} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Notre construction est basée sur la construction d'un programme à branchements de taille 2^n pour le permanent. Pour le permanent (3×3), on obtient le programme à branchements présenté à la figure 3.14.

À notre connaissance, il s'agit de la meilleure borne supérieure connue pour ce problème.

Théorème 3.16

> Pour tout n, le permanent PER_n est une projection du déterminant DET_m avec $m = 2^n - 1$. De plus, cette projection n'utilise que les constantes 0, 1 et -1.

Lemme 3.17

> Pour tout n, il existe un programme à branchements à 2^n sommets et $n2^{n-1}$ arcs représentant PER_n.

Démonstration : Soit $\mathcal{B}$ un programme à branchements dont les sommets sont étiquetés par les sous-ensembles de $\{1, \ldots, n\}$. Le programme à branchements $\mathcal{B}$ possède donc 2^n sommets. Les sommets distingués s et p sont respectivement l'ensemble vide et l'ensemble $\{1, \ldots, n\}$ tout entier. Pour tout sommet d'étiquette $S \subseteq \{1, \ldots, n\}$ où $|S| = i$, il y a un arc de poids $X_{i+1,j}$ de S vers $S \cup \{j\}$ pour tout $j \notin S$. Ce sont les seuls arcs de $\mathcal{B}$.

Un sommet d'étiquette S tel que $|S| = i$ est de degré sortant $(n - i)$.

Le nombre total d'arcs dans $\mathcal{B}$ est donc

$$\sum_{i=0}^{n}(n-i)\binom{n}{i} = n\sum_{i=0}^{n}\binom{n}{i} - \sum_{i=0}^{n} i\binom{n}{i} = n2^n - n2^{n-1} = n2^{n-1}.$$

Le poids d'un chemin de s à p est de la forme $X_{1j_1}X_{2j_2}\cdots X_{nj_n}$. De plus, puisque les arcs sont de la forme $S \to S \cup \{j\}$ pour $j \notin S$, les j_i sont tous distincts. En d'autres termes, l'application $i \mapsto j_i$ est une permutation. Puisqu'il existe un arc $S \to S \cup \{j\}$ pour tout $j \notin S$, l'ensemble des chemins de s à p est en bijection avec les permutations de $\{1,\dots,n\}$. Donc $\mathcal{B}$ représente bien $\textsc{Per}_n$. □

Démonstration du théorème 3.16 : On pourrait se contenter d'appliquer la proposition 3.9. On obtiendrait ainsi une matrice $\mathcal{M}$ de dimension 2^n telle que $\det(\mathcal{M}) = \textsc{Per}_n$. Pour obtenir une dimension $(2^n - 1)$, on modifie un peu la construction de $\mathcal{M}$. Partant du programme à branchements $\mathcal{B}$ obtenu par le lemme 3.17, on construit $\mathcal{G}$ en identifiant les sommets s et p et en ajoutant une boucle de poids 1 sur tous les autres sommets. Tous les chemins de s à p dans $\mathcal{B}$ ont même longueur n. Ainsi, les cycles dans $\mathcal{G}$ sont soit de longueur n, soit de longueur 1. Chaque couverture par cycle a donc la même signature. Ainsi, le déterminant de la matrice d'adjacence de $\mathcal{G}$ vaut $(-1)^{n+1}\textsc{Per}_n$. Si n est pair, on remplace la boucle de poids 1 par une boucle de poids -1 sur chacun des n sommets étiquetés par des singletons. Pour chaque couverture par cycles du graphe, exactement un de ces sommets appartient au cycle passant par s. En d'autres termes, il y a $(n-1)$ boucles de poids -1 qui sont utilisées à chaque fois. Le déterminant de la matrice d'adjacence du graphe est donc multipliée par $(-1)^{n-1} = -1$ puisque n est pair. □

Corollaire 3.18

> Pour tout n, il existe un circuit asymétrique de taille $(n-1)(2^n-1)$ avec $n2^{n-1}$ entrées représentant $\textsc{Per}_n$.

Démonstration : Il suffit d'appliquer la proposition 3.8 au programme à branchements obtenu au lemme 3.17. □

Chapitre 4

Représentations symétriques

Quelle est l'expressivité des matrices symétriques ? Ce chapitre et le suivant abordent cette question. Au lieu de s'intéresser aux représentations déterminantielles des polynômes, on s'intéresse aux représentations déterminantielles symétriques. En d'autres termes, on cherche maintenant à représenter un polynôme par le déterminant d'une matrice symétrique dont les coefficients sont toujours des variables et des constantes du corps de base.

Il s'agit donc de comparer l'expressivité du déterminant d'une matrice symétrique par rapport au déterminant d'une matrice quelconque. On montre dans ce chapitre que si le corps de base est de caractéristique différente de 2, alors l'expressivité du déterminant symétrique est sensiblement la même que celui du déterminant quelconque. En particulier, on montre le résultat nouveau à notre connaissance que le déterminant d'une matrice quelconque $\mathcal{M}$ de dimension n peut être représenté par le déterminant d'une matrice symétrique $\mathcal{N}$ de dimension $\mathcal{O}(n^3)$ dont les coefficients sont ceux de $\mathcal{M}$ et les constantes $\{0, 1, -1, 1/2\}$. Le chapitre suivant est consacré au cas de la caractéristique 2 dans lequel certains polynômes n'admettent pas de représentation sous forme de déterminant de matrice symétrique.

Une motivation externe pour cette étude vient du domaine de l'optimisation convexe. Une notion centrale est celle d'expression linéaire matricielle [1] qui est simplement une matrice symétrique dont les coefficients sont des fonctions affines réelles des variables $X_1, \ldots, X_n$. Une expression linéaire matricielle est souvent écrite sous la forme

$$A(X_1, \ldots, X_n) = A_0 + A_1 X_1 + \cdots + A_n X_n$$

où les A_i sont des matrices réelles symétriques. Parmi ces expressions, celles qui vérifient $A_0 \succeq 0$, c'est-à-dire que A_0 est semi-définie positive, sont d'un grand intérêt pour la programmation semi-définie et sont appelées *inégalités*

1. *Linear matrix expression*, ou *symmetric linear matrix form*, ou encore *affine symmetric matrix pencil* sont les termes rencontrés dans la littérature anglophone.

linéaires matricielles. Elles ont des liens forts avec les polynômes RZ[2], qui sont tels que pour tous $X \in \mathbb{R}^n$ et $\mu \in \mathbb{C}$, $f(\mu X) = 0$ implique $\mu \in \mathbb{R}$. En particulier, la conjecture de Lax [80], prouvée par J. William Helton et Victor Vinnikov [54] (voir aussi [86]), affirme qu'un polynôme f à deux variables est RZ si et seulement s'il est déterminant d'une inégalité linéaire matricielle dont la dimension est le degré de f. Une généralisation naturelle de ce résultat consiste à s'intéresser au cas des polynômes RZ à un nombre quelconque de variables. Par un argument de comptage, Helton et Vinnikov ont tout de suite montré que la conjecture de Lax était fausse dans ce cadre plus général. Ils ont posé la question d'un résultat similaire en abandonnant la condition sur la dimension de l'inégalité linéaire matricielle ou en cherchant une représentation du polynôme f élevé à une certaine puissance. Ces deux nouvelles généralisations de la conjecture de Lax ont été réfutées par Petter Brändén [14] (voir également [102, 101]).

Une autre relaxation de la conjecture consiste à abandonner la condition $A_0 \succeq 0$ et donc à chercher une représentation déterminantielle symétrique des polynômes. Des résultats en ce sens ont été obtenus par J. William Helton, Scott A. McCullough et Victor Vinnikov [53] puis simplifiés par Ronan Quarez [107]. Les résultats que nous présentons dans ce chapitre améliorent les résultats cités puisqu'il aboutissent à des représentations par des déterminants de matrices symétriques de dimensions moindres. Une comparaison est effectuée dans la partie 4.2.

Les résultats de ce chapitre sont issus dans une certaine mesure des articles [43] et [42]. Cependant, les preuves sont simplifiées et les résultats légèrement améliorés.

4.1 Représentation symétrique des programmes à branchements

Nous donnons dans cette partie une construction d'une représentation déterminantielle symétrique pour tout polynôme représenté par un programme à branchements. Cette construction simplifie et unifie celles des articles [43] et [42]. Elle est inspirée de la construction de représentation déterminantielle symétrique du déterminant présentée dans [42] due à Meena Mahajan et Prajakta Nimbhorkar.

Théorème 4.1

> Tout programme à branchements $\mathcal{B}$ à n sommets peut être représenté par le déterminant d'une matrice symétrique de dimension $(2n-1)$ dont chaque coefficient est soit le poids d'un arc de $\mathcal{B}$, soit un élément de $\{0, 1, -1, 1/2\}$.

Pour prouver ce théorème, le graphe sous-jacent au programme à branchements $\mathcal{B}$ est rendu symétrique (non orienté) en le transformant en un

2. *Real zero polynomials*, en anglais.

graphe biparti.

Définition 4.2

Soit $\mathcal{G}$ un graphe orienté. On définit son biparti (non orienté) associé $\mathcal{G}_b$ de la manière suivante : chaque sommet v de $\mathcal{G}$ est dupliqué en un sommet *entrant* v^e et un sommet *sortant* v^s reliés par une arête, et chaque arc (u, v) de $\mathcal{G}$ est transformé en l'arête $\{u^s, v^e\}$.

Lemme 4.3

Soit $\mathcal{G}$ un graphe orienté acyclique et $\mathcal{G}_b$ son biparti associé. Alors l'unique couverture par cycles de $\mathcal{G}_b$ est le couplage parfait associant chaque sommet entrant au sommet sortant correspondant.

Démonstration : Dans $\mathcal{G}$, on définit la *profondeur* d'un sommet par induction : si v est de degré entrant nul, il est de profondeur nulle, et sinon, sa profondeur est un de plus que la profondeur maximale des sommets dont il reçoit un arc. On associe à un sommet de $\mathcal{G}_b$ la profondeur de son antécédent dans $\mathcal{G}$.

Dans $\mathcal{G}_b$, les deux types d'arêtes sont $\{v^e, v^s\}$, et $\{u^s, v^e\}$ où v^e est de profondeur strictement supérieure à u^s.

Considérons une couverture par cycles (orientés) de $\mathcal{G}_b$ et un cycle c qui ne soit pas de la forme (v^e, v^s). Soit v^s un sommet de profondeur minimale dans c. C'est nécessairement un sommet sortant car un sommet entrant a l'un au moins de ses voisins qui est de profondeur moindre. Le sommet entrant v^e correspondant à v^s doit lui aussi être couvert par un cycle c'. Ce cycle ne peut pas être (v^e, v^s) puisque v^s appartient au cycle c. Les deux voisins de v^e dans c' sont de profondeur strictement moindre que v^s. Il existe donc dans c' un sommet w^s de profondeur minimale, qui est de profondeur strictement inférieure à v^s. Par induction, on construit une suite de sommets de profondeur strictement décroissante. Puisque la profondeur d'un sommet est toujours positive, c'est une contradiction. Autrement dit, tous les cycles de la couverture par cycles sont de la forme (v^e, v^s). □

Démonstration du théorème 4.1 : L'idée est de construire à partir d'un programme à branchements $\mathcal{B}$ un graphe biparti $\mathcal{G}$ de telle sorte que tout chemin de s à p dans $\mathcal{B}$ corresponde à une couverture par cycles de $\mathcal{G}$ faite d'un *grand* cycle correspondant au chemin et d'un couplage parfait des sommets restants.

Soit f un polynôme calculé par un programme à branchements $\mathcal{B}$ de taille n, avec s sa source et p son puits. Soit $\mathcal{B}_b$ le biparti associé à $\mathcal{B}$, et $\mathcal{G}$ le graphe obtenu en supprimant les sommets s^e et p^s à $\mathcal{B}_b$ et en attribuant un poids -1 aux arêtes de la forme (v^s, v^e) et le poids de l'arc (u, v) de $\mathcal{B}$ aux arêtes de la forme $\{u^s, v^e\}$. On renomme s le sommet s^s et p le sommet p^e.

Par le lemme précédent, la seule couverture par cycles de $\mathcal{B}_b$ est le couplage parfait. La seule couverture par cycle potentielle de $\mathcal{G}$ consiste donc en un couplage parfait utilisant l'arête $\{s, p\}$ si elle existe et des arêtes de la forme $\{v^e, v^s\}$. Soit π un chemin entre s et p dans $\mathcal{G}$. On dit qu'il est *acceptable* s'il est de la forme

$$(s, v_1^e, v_1^s, v_2^e, v_2^s, \ldots, v_k^e, v_k^s, p).$$

Le chemin réduit à (s, p) est également considéré comme acceptable. Si un chemin π est acceptable, alors $\mathcal{G} \setminus \pi$ admet le couplage parfait constitué d'arêtes de la forme $\{v^e, v^s\}$ comme une unique couverture par cycles d'après le lemme précédent. En effet, $\mathcal{G} \setminus \pi$ est alors le biparti associé à $\mathcal{B} \setminus \{s, v_1, \ldots, v_k, p\}$. On cherche à montrer que si un chemin π n'est pas acceptable, alors $\mathcal{G} \setminus \pi$ n'admet aucune couverture par cycles.

Supposons que π ne soit pas un chemin acceptable. Cela signifie qu'il existe une suite de trois sommets (u^s, v^e, w^s) dans π, et v est alors de profondeur strictement supérieure à celles de u et w. Le sommet v^e est donc un maximum local en terme de profondeur. Considérons une telle suite de trois sommets avec v^e de profondeur maximale. Alors v^s ne peut pas appartenir à π. En effet, si c'était le cas, v^s serait entouré de deux sommets entrants, donc de profondeurs strictement supérieures à la sienne. Ainsi, il existerait sur la partie de π comprise entre v^e et v^s des sommets entrants de profondeur supérieure à celle de v^e. Il y aurait donc un maximum local entre v^e et v^s de profondeur strictement supérieure à celle de v^e, ce qui est contradictoire. Ceci prouve que si un chemin π n'est pas acceptable, il existe un sommet v^e dans π tel que v^s n'est pas dans π.

Supposons que $\mathcal{G} \setminus \pi$ admette une couverture par cycles. Le sommet v^s doit être couvert par un cycle c, qui n'est pas (v^e, v^s). Il existe dans c un sommet w^e de profondeur maximale. Le sommet w^s correspondant ne peut pas appartenir à c puisque sinon, un des voisins de w^s dans c est un sommet entrant de profondeur strictement supérieure à w^e. On peut alors construire par induction une suite infinie de sommets sortants de profondeur strictement croissante, ce qui est contradictoire. Donc si π n'est pas acceptable, $\mathcal{G} \setminus \pi$ n'admet pas de couverture par cycles.

Considérons maintenant le graphe $\mathcal{G}^+$ obtenu en ajoutant un sommet u_0 et les arêtes $\{u_0, s\}$ de poids α et $\{u_0, p\}$ de poids β à $\mathcal{G}$. Les poids α et β seront fixées ultérieurement. Le graphe $\mathcal{G}^+$ a un nombre impair de sommets. Ses couvertures par cycles doivent donc toutes avoir au moins un cycle de longueur impaire. Comme $\mathcal{G}$ est biparti, il ne possède pas de cycle impair et les seuls cycles impairs de $\mathcal{G}^+$ sont donc ceux passant par le nouveau sommet u_0. Ces cycles sont en bijection avec les chemins entre s et p dans $\mathcal{G}$. De plus, d'après la discussion précédente, les seuls chemins entre s et p menant à une couverture par cycles de $\mathcal{G}^+$ sont les chemins acceptables. Une fois choisi un tel chemin, le reste de la couverture par

cycles est nécessairement un couplage parfait n'utilisant que des arêtes de la forme $\{v^e, v^s\}$. Notons de plus que tout chemin acceptable est associé de manière unique au chemin $(s, v_1, v_2, \ldots, v_k, p)$ dans $\mathcal{B}$. Ceci montre que les couvertures par cycles de $\mathcal{G}^+$ dont le grand cycle est parcouru dans le sens s vers p sont en bijection avec les chemin de s à p dans $\mathcal{B}$. Pour chaque couverture par cycles de ce type, il existe la couverture correspondante parcourant le grand cycle en sens inverse puisque $\mathcal{G}^+$ n'est pas orienté.

Soit $\mathcal{M}$ la matrice d'adjacence de $\mathcal{G}^+$. Alors

$$\det(\mathcal{M}) = \sum_{\mathcal{C}} \epsilon(\mathcal{C}) w(\mathcal{C})$$

où la somme porte sur les couvertures par cycles de $\mathcal{G}^+$. Pour un chemin π de s à p donné, le poids de la couverture correspondante est $w(\pi)(-1)^{\frac{|\pi|}{2}}\alpha\beta$. La signature est donné par le nombre de cycles pairs utilisés, et elle vaut donc $(-1)^{|\mathcal{G}^+ \setminus \pi|/2}$. Une telle couverture par cycles ajoute donc au déterminant le terme $(-1)^{|\mathcal{G}^+ \setminus \pi|/2}(-1)^{|\pi|/2}\alpha\beta w(\pi) = (-1)^{|\mathcal{G}^+|/2}\alpha\beta w(\pi)$. Ainsi,

$$\det(\mathcal{M}) = 2(-1)^{|\mathcal{G}^+|/2}\alpha\beta \sum_{\pi : s \rightsquigarrow p} w(\pi) = 2(-1)^{|\mathcal{G}^+|/2}\alpha\beta f.$$

Il suffit donc de poser $\alpha = 1/2$ et $\beta = (-1)^{|\mathcal{G}^+|/2}$ pour obtenir $f = \det(\mathcal{M})$. □

On peut alors utiliser les résultats du chapitre précédent pour obtenir des représentations déterminantielles symétriques des formules, des circuits (faiblement) asymétriques et du déterminant.

Corollaire 4.4

Toute formule de taille t peut être représentée par le déterminant d'une matrice symétrique de dimension $(2t + 1)$ dont chaque coefficient est soit une variable, soit une constante de la formule, soit un élément de $\{0, 1, -1, 1/2\}$.

Tout circuit faiblement asymétrique de taille t ayant e entrées peut être représenté par le déterminant d'une matrice symétrique de dimension $2(t + e) + 1$ dont chaque coefficient est soit une variable, soit une constante du circuit, soit un élément de $\{0, 1, -1, 1/2\}$.

On peut obtenir des résultats similaires en remplaçant la taille et le nombre d'entrées par la taille réduite et le nombre de variables, de la même façon que dans le corollaire 3.10 et le théorème 3.11.

Corollaire 4.5

Le déterminant $\textsc{Det}_n$ de la matrice $\mathcal{X} = (X_{ij})_{1 \leq i,j \leq n}$ peut être représenté par le déterminant d'une matrice symétrique de dimension $\frac{2}{3}n^3 + o(n^3)$ à coefficients dans $\{X_{ij} : 1 \leq i, j \leq n\} \cup \{0, 1, -1, 1/2\}$.

4.2 Comparaisons avec des résultats existants

Des représentations déterminantielles symétriques ont été données par J. William Helton, Scott A. McCullough et Victor Vinnikov [53] et Ronan Quarez [107]. D'une part, ces constructions ne sont valables que pour des polynômes à coefficients dans $\mathbb{R}$. D'autre part, la construction de Quarez est une construction effective, mais produit des matrices de taille exponentielle. Nous nous attachons ici à comparer ces méthodes et les nôtres. Il est à noter que certains polynômes à n variables et de degré d n'admettent pas de formule arithmétique de taille inférieure à $\mathcal{O}(\binom{n+d}{d}/\log\binom{n+d}{d})$ [55]. Ainsi, nos constructions qui sont linéaires en la taille d'une formule représentant le polynôme ne sont pas toujours polynomiales en le nombre de variables ou le degré. On peut obtenir de meilleures bornes en s'intéressant à des circuits et non des formules. Cependant, certains polynômes n'admettent pas de circuit de taille inférieure à $\mathcal{O}(\sqrt{\binom{n+d}{d}})$. Ce résultat est un résultat classique de la complexité arithmétique, et Pavel Hrubeš et Amir Yehudayoff ont montré que cette borne inférieure reste valable si l'on ne s'intéresse qu'aux polynômes ayant uniquement 0 ou 1 comme coefficients [55]. Shachar Lovett a montré que cette borne est essentiellement la bonne en exhibant un circuit de taille $(nd)^{\mathcal{O}(1)}\sqrt{\binom{n+d}{d}}$ pour tout polynôme de degré d à n variables [89].

Dans cette partie, nous donnons une preuve légèrement remaniée de ce dernier résultat, qui nous permet de conclure que le circuit obtenu est faiblement asymétrique. Plus exactement, nous montrons comment obtenir un programme à branchements en suivant la preuve de Lovett, duquel on peut déduire un circuit asymétrique. Ceci nous permet ensuite d'obtenir des bornes sur les représentations déterminantielles symétriques dans le pire cas qui se trouvent être de dimensions légèrement inférieures à celles de Quarez [107]. On note cependant que ce pire est loin d'être atteint dans un grand nombre de cas, ce qui signifie que nos représentations peuvent être sensiblement plus petites que celles de Quarez.

Théorème 4.6

> Tout polynôme de degré d à n variables peut être représenté par un programme à branchements ayant au plus
>
> $$(n+1)(d+3)\binom{\lceil\frac{n+1}{2}\rceil+\lceil\frac{d}{2}\rceil}{\lceil\frac{d}{2}\rceil}+2 \leq (nd)^{\mathcal{O}(1)}\sqrt{\binom{n+d}{d}}$$
>
> sommets.

Démonstration : Soit $X^\alpha = X_1^{\alpha_1}\cdots X_n^{\alpha_n}$ un monôme. On considère dans toute cette preuve les indices des variables *modulo* n. Supposons pour simplifier que n est impair et d est pair. Pour un indice i, on définit l'*intervalle à*

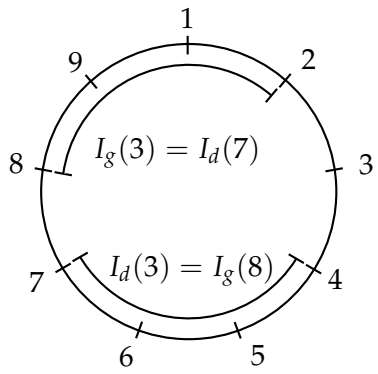


FIGURE 4.1 – Illustration des intervalles à gauche et à droite pour $n = 9$.

gauche de i par

$$I_g(i) = \{i - (n-1)/2, i - (n-1)/2 + 1, \ldots, i - 1\}$$

et similairement l'*intervalle à droite de i* par

$$I_d(i) = \{i + 1, i + 2, \ldots, i + (n-1)/2\}.$$

Les indices dans ces intervalles sont pris *modulo n*. On remarque que $I_g(i)$, $I_d(i)$ et $\{i\}$ forment une partition de $\{1, \ldots, n\}$. De plus, $I_g(i) = I_d(i + (n-1)/2)$ pour tout i (Fig. 4.1).

Nous allons montrer que si X^α est de degré au plus d, alors il existe un indice i tel que $\sum_{j \in I_g(i)} \alpha_j \leq d/2$ et $\sum_{j \in I_d(i)} \alpha_j \leq d/2$. Notons respectivement $S_g(i)$ et $S_d(i)$ ces deux sommes. Considérons d'abord l'indice 1. Soit il vérifie la propriété et c'est l'indice voulu. Sinon, supposons sans perte de généralité que $S_d(1) > d/2$, et donc $S_g(1) < d/2$. Puisque $I_g(1) = I_d(1 + (n-1)/2)$, on en déduit l'existence d'un indice $i \in \{2, \ldots, 1 + (n-1)/2\}$ tel que $S_d(i) \leq d/2$ et $S_d(i-1) > d/2$. De plus, $S_g(i) = S_g(i-1) + \alpha_{i-1} - \alpha_{i-1-(n-1)/2}$. Comme $S_d(i-1) + S_g(i-1) + \alpha_{i-1} \leq d$, $S_g(i) \leq d/2$ et i vérifie la propriété voulue. Cet indice n'est pas nécessairement unique, mais soit i_c un indice arbitraire vérifiant $S_g(i), S_d(i) \leq d/2$, que l'on appelle *indice central de* X^α. On peut définir $\alpha^g_{i_c}$ et $\alpha^d_{i_c}$ de telle sorte que $\alpha^g_{i_c} + \alpha^d_{i_c} = \alpha_{i_c}$, $\alpha^g_{i_c} + S_g(i_c) \leq d/2$ et $\alpha^d_{i_c} + S_d(i_c) \leq d/2$. On définit ensuite $\alpha^g = (\alpha_{i_c-(n-1)/2}, \ldots, \alpha_{i_c-1}, \alpha^g_{i_c}, \bar{0})$ et $\alpha^d = (\alpha^d_{i_c}, \alpha_{i_c+1}, \ldots, \alpha_{i_c+(n-1)/2}, \bar{0})$. Alors $X^\alpha = X^{\alpha^g} X^{\alpha^d}$. Les monômes X^{α^g} et X^{α^d} sont donc de degré au plus $d/2$, et ont $(n+1)/2$ variables chacun. Leurs variables sont de plus consécutives. On note A l'ensemble des n-uplets α qui ont au moins $(n-1)/2$ composantes consécutives égales à 0 et tels que $\sum_i \alpha_i \leq d/2$. Cette somme est appelée le degré de α.

On construit alors le programme à branchements de la manière suivante. On définit tout d'abord sa source s et son puits p. Pour tout $\alpha^g \in A$

de degré δ, on crée un programme à branchements à $(\delta+1)$ sommets, de source s_{α^g} et de puits p_{α^g}, qui calcule X^{α^g}. Ce programme à branchements est simplement un chemin de longueur δ de s_{α^g} à p_{α^g} dont chaque arc porte comme poids l'une des variables du monôme X^{α^g}. De même pour tout $\alpha^d \in A$, on crée un programme à branchement calculant X^{α^d}, de source s_{α^d} et de puits p_{α^d}. On relie s à chaque s_{α^g} par un arc de poids 1, et chaque p_{α^d} à p par un arc de poids 1.

Soit

$$f = \sum_{\alpha} c_\alpha X^\alpha$$

le polynôme que l'on cherche à représenter, où la somme parcourt tous les n-uplets de degré au plus d. Pour tout α, on fixe un indice central et on définit α^g et α^d en fonction de cet indice. On relie alors le puits p_{α^g} à la source s_{α^d} par un arc de poids c_α.

Le programme à branchements ainsi défini calcule bien le polynôme f. En effet, l'ensemble des couples (c_α, X^α) est en bijection avec les chemins de la forme $s \to s_{\alpha^g} \rightsquigarrow p_{\alpha^g} \to s_{\alpha^d} \rightsquigarrow p_{\alpha^d} \to p$, dont le poids est $c_\alpha X^\alpha$.

Il reste maintenant à calculer la taille du programme à branchements obtenu. Le nombre de monômes de degré au plus d à n variables est $\binom{n+d}{d}$. Ainsi, le nombre de monômes de degré au plus $d/2$ sur $(n+1)/2$ variables consécutives est $n\binom{(n+d+1)/2}{d/2}$. Pour chaque n-uplet de A, on crée deux programmes à branchements de taille au plus $d/2+1$ chacun. En ajoutant la source s et le puits p, le programme à branchements pour f est de taille au plus

$$n(d+2)\binom{(n+d+1)/2}{d/2} + 2.$$

Comme on a supposé n impair et d pair, on borne n par $(n+1)$, d par $(d+1)$, $\frac{n+1}{2}$ par $\lceil \frac{n+1}{2} \rceil$ et $\frac{d}{2}$ par $\lceil \frac{d}{2} \rceil$ pour obtenir la borne du théorème. Enfin, la borne asymptotique se déduit de la formule de Stirling. □

On peut alors construire à partir de ce programme à branchements soit un circuit asymétrique ayant au plus $(nd)^{\mathcal{O}(1)}\sqrt{\binom{n+d}{d}}$ portes de multiplications, soit directement une matrice symétrique de même dimension. En comparaison, Quarez [107] construit des matrices de dimension fixe $\binom{n+\lfloor \frac{d}{2} \rfloor}{n}$. Nos constructions sont donc, même dans le pire cas, plus petites que les siennes. On peut s'intéresser par exemple à un polynôme tel que le permanent pour lequel on ne connaît pas de circuit de taille polynomiale. D'après le lemme 3.17, le permanent admet un programme à branchements de taille 2^n. Ceci nous permet de représenter le permanent d'une matrice carrée de dimension n par le déterminant d'une matrice symétrique de dimension $(2^{n+1}-2)$. Puisque le permanent est un polynôme de degré n en n^2 variables, Quarez construit pour le représenter une matrice symétrique de dimension

$\binom{\lfloor n/2 \rfloor + n^2}{\lfloor n/2 \rfloor} \geq (2n+1)^{n/2}$. Ainsi, nos constructions permettent d'obtenir une représentation déterminantielle symétrique du permanent plus compacte que celle de Quarez.

Puisque dans le cadre de l'optimisation convexe, les polynômes étudiés sont les polynômes RZ, il serait intéressant d'étudier dans quelle mesure la structure imposée à ces polynômes se traduit en terme de taille du plus petit programme à branchements pour les représenter afin d'obtenir éventuellement de meilleures bornes que celles présentées ici.

Chapitre 5

Représentations symétriques en caractéristique deux

Après avoir montré dans le chapitre précédent qu'en caractéristique différente de deux, les représentations déterminantielles symétriques sont aussi expressives que les représentations déterminantielles classiques, on s'aperçoit dans ce chapitre que la situation change drastiquement en caractéristique deux. Les constructions du chapitre précédent ne sont pas valables ici puisqu'elles utilisent la constante $1/2$ qui n'est plus disponible. Nous allons en fait voir qu'il n'y a en quelque sorte aucun moyen de se débarrasser de cette constante dans le cas général, et que certains polynômes ne sont pas représentables comme déterminant d'une matrice symétrique en caractéristique deux.

On fixe pour tout ce chapitre un corps quelconque $\mathbb{F}$ de caractéristique 2. Pour simplifier, on dira qu'un polynôme $f \in \mathbb{F}[X_1, \ldots, X_m]$ est *représentable* s'il existe une matrice carrée $\mathcal{M}$ à coefficients dans $\mathbb{F} \cup \{X_1, \ldots, X_m\}$ telle que $f = \det(\mathcal{M})$. Dans ce cas, la matrice $\mathcal{M}$ *représente* le polynôme f.

Par exemple, le polynôme $XY + YZ + XZ$ est représentable par le déterminant de la matrice de dimension 4

$$\begin{pmatrix} X & 0 & 0 & 1 \\ 0 & Y & 0 & 1 \\ 0 & 0 & Z & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}. \tag{5.1}$$

Une relaxation évidente consiste à autoriser les coefficients de $\mathcal{M}$ à prendre des valeurs dans une extension $\mathbb{G}$ de $\mathbb{F}$. Cependant, nous montrons dans ce chapitre que c'est inutile, au moins pour les polynômes multilinéaires. Plus précisément, nous montrons qu'un polynôme multilinéaire est représentable si et seulement s'il l'est par une matrice qui utilise comme constantes uniquement des éléments du corps engendré par les coefficients du polynôme.

Ce chapitre correspond à l'article [47]. Les algorithmes présentés, en particulier ceux de la partie 5.4, ont été implémentés à l'aide du logiciel de calcul formel *Sage* [117] et sont disponibles à l'adresse `http://perso.ens-lyon.fr/bruno.grenet/publis/SymDetReprChar2.sage`.

5.1 Prérequis algébrique

5.1.1 *Polynômes, déterminants et graphes en caractéristique* 2

Soit $\mathbb{F}$ un corps de caractéristique 2, et soit $\mathbb{F}[X_1, \dots, X_m]$ l'anneau des polynômes en m indéterminées sur $\mathbb{F}$. Le fait de travailler en caractéristique 2 induit certaines spécificités pour les polynômes et les déterminants. Premièrement, l'endomorphisme de Frobenius assure que pour tous polynômes f_1 et f_2, $(f_1 + f_2)^2 = f_1^2 + f_2^2$. Le déterminant d'une matrice symétrique admet quant à lui une expression plus simple que dans le cas général.

Proposition 5.1

Soit $\mathcal{M} = (m_{ij})$ une matrice symétrique de dimension n à coefficients dans $\mathbb{F}[X_1, \dots, X_m]$. Alors son déterminant est

$$\det(\mathcal{M}) = \sum_{\sigma} \prod_{i=1}^{n} m_{i,\sigma(i)}$$

où la somme porte sur les *involutions* σ de $\{1, \dots, n\}$, c'est-à-dire les permutations vérifiant $\sigma^{-1} = \sigma$.

Démonstration : D'après la formule de Leibniz,

$$\det(\mathcal{M}) = \sum_{\sigma \in \mathfrak{S}_n} \epsilon(\sigma) \prod_{i=1}^{n} m_{i,\sigma(i)}. \tag{5.2}$$

Comme $\epsilon(\sigma) \in \{-1, 1\}$, et que les deux éléments -1 et 1 coïncident en caractéristique 2, on peut supprimer la signature de l'expression (5.2).

Considérons maintenant $P_\sigma = \prod_i m_{i,\sigma(i)}$, avec $\sigma^{-1} \neq \sigma$. Alors puisque $\mathcal{M}$ est symétrique, $m_{i,\sigma^{-1}(i)} = m_{\sigma^{-1}(i),i}$, et donc $P_{\sigma^{-1}} = P_\sigma$. Ces deux produits s'annulent donc dans l'expression (5.2). Ainsi, on peut restreindre la somme aux involutions. □

Cette expression du déterminant se traduit de manière immédiate en terme de graphes. En effet, on a vu au chapitre 1 qu'un déterminant de $\mathcal{M}$ a une interprétation en terme de couvertures par cycles du graphe $\mathcal{G}$ dont $\mathcal{M}$ est la matrice d'adjacence. Restreindre les permutations aux involutions pour une matrice symétrique signifie restreindre les couvertures par cycles du graphe non orienté $\mathcal{G}$ à celles constituées de cycles (orientés) de longueur 1 (les *boucles*) et 2 (les *arêtes*). Ce type de couverture par cycles est connu sous le nom de *couplage partiel*, ou parfois simplement *couplage*, en opposition

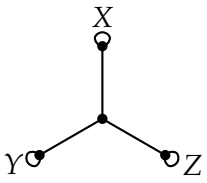


FIGURE 5.1 – Graphe représentant le polynôme $XY + YZ + XZ$.

aux *couplages parfaits* qui n'utilisent que des arêtes. On rencontre également le terme de *couverture par monomères-dimères* issu de la physique statistique. Notons que le poids d'un cycle de longueur 2 est le carré du poids de l'arête correspondante. En effet, lorsque l'on considère une couverture par cycles d'un graphe non orienté, ces cycles sont orientés. Un cycle de longueur 2, représenté par une arête $\{u, v\}$ de poids w, est en réalité constitué des deux arcs $u \to v$ et $v \to u$. Chacun de ces arcs ayant comme poids w, le poids du cycle est w^2.

Un couplage partiel d'un graphe $\mathcal{G}$ peut être vu comme un ensemble μ d'arêtes tel qu'aucun sommet du graphe n'appartienne à deux arêtes distinctes. En notant $\mathcal{M}$ la matrice d'adjacence de $\mathcal{G}$, la discussion précédente peut être résumée par l'identité

$$\det(\mathcal{M}) = \sum_{\mu} \left(\prod_{e \in \mu} w(e)^2 \times \prod_{v \notin \mu} w(v) \right),$$

où la somme porte sur tous les couplages partiels μ de $\mathcal{G}$, $w(e)$ est le poids de l'arête e, $w(v)$ le poids de la boucle sur le sommet v, et $v \notin \mu$ est un léger abus de notation pour signifier que v n'est pas couvert par le couplage partiel μ. Un exemple est donné par le graphe de la figure 5.1 dont la matrice d'adjacence est donnée par l'équation (5.1), page 73. Par convention, une arête ou une boucle dont le poids n'est pas spécifié est de poids 1. Pour obtenir un couplage partiel du graphe, il faut nécessairement que le sommet central appartienne à l'une des trois arêtes. Il ne reste alors qu'une manière de compléter le couplage avec les boucles sur les deux sommets restants.

Par abus de langage, on dira qu'un graphe $\mathcal{G}$ représente un polynôme f si sa matrice d'adjacence représente f. On rappelle qu'on note $\mathcal{M}(\mathcal{G})$ la matrice d'adjacence d'un graphe $\mathcal{G}$, et inversement $\mathcal{G}(\mathcal{M})$ le graphe dont $\mathcal{M}$ est la matrice d'adjacence. Dans le même esprit, on pourra écrire $\det(\mathcal{G})$ à la place de $\det(\mathcal{M}(\mathcal{G}))$ pour simplifier les notations.

5.1.2 Anneaux quotient

Soit $p_1, \ldots, p_k \in \mathbb{F}[X_1, \ldots, X_m]$. On note $\langle p_1, \ldots, p_k \rangle$ l'idéal qu'ils génèrent, c'est-à-dire

$$\langle p_1, \ldots, p_k \rangle = \left\{ \sum_{i=1}^{k} p_i q_i : q_i \in \mathbb{F}[X_1, \ldots, X_m] \right\}.$$

Étant donné un multiplet $\ell = (\ell_1, \ldots, \ell_m) \in \mathbb{F}^m$, on définit l'idéal

$$\mathcal{I}(\ell) = \langle X_1^2 + \ell_1, \ldots, X_m^2 + \ell_m \rangle.$$

On définit aussi l'anneau quotient $\mathcal{R}(\ell) = \mathbb{F}[X_1, \ldots, X_m]/\mathcal{I}(\ell)$ et on note π_ℓ, ou simplement π, la projection canonique $\mathbb{F}[X_1, \ldots, X_m] \to \mathcal{R}(\ell)$. La restriction de cette projection à $\mathbb{F}$ est une injection. Il existe donc un plongement naturel de $\mathbb{F}$ dans $\mathcal{R}(\ell)$ et les éléments de $\mathbb{F} \subseteq \mathcal{R}(\ell)$ sont les *constantes* de l'anneau. On étend la projection aux matrices en posant $\pi(\mathcal{M})_{ij} = \pi(\mathcal{M}_{ij})$. On remarque que $\pi(\det(\mathcal{M})) = \det(\pi(\mathcal{M}))$. On dit qu'un élément de $\mathcal{R}(\ell)$ est *linéaire* si c'est la projection d'un polynôme linéaire de $\mathbb{F}[X_1, \ldots, X_m]$.

Le quotient identifie les carrés des variables avec des constantes. On en déduit que tout élément $r \in \mathcal{R}(\ell)$ possède un unique représentant multilinéaire $f \in \mathbb{F}[X_1, \ldots, X_m]$ que l'on note $\rho_\ell(r)$, ou $\rho(r)$. Clairement, $\pi_\ell \circ \rho_\ell(r) = r$ pour tout $r \in \mathcal{R}(\ell)$. On note $\textsc{Mult}_\ell$, ou $\textsc{Mult}$, l'application $\rho_\ell \circ \pi_\ell$ qui envoie tout polynôme sur le polynôme multilinéaire obtenu en remplaçant chaque x_i^2 par ℓ_i.

Le carré de tout élément de $\mathcal{R}(\ell)$ est une constante de l'anneau. En particulier, un élément de $\mathcal{R}(\ell)$ est inversible si et seulement si son carré est non nul. Par exemple, $\pi(X_1X_2 + 1)$ est inversible si et seulement si $\ell_1\ell_2 \neq 1$.

Étant donné un multiplet $\ell = (\ell_1, \ldots, \ell_m) \in \mathbb{F}^m$, on note ℓ^2 le multiplet $(\ell_1^2, \ldots, \ell_m^2)$, et on dit que c'est un *multiplet de carrés*. Si ℓ^2 est un multiplet de carrés, le carré d'un élément $r \in \mathcal{R}(\ell^2)$ est le carré d'un unique élément $c \in \mathbb{F}$ que l'on note $|r|_{\ell^2}$, ou $|r|$. On appelle cet élément la *valeur absolue* de r. On remarque que $|r_1 r_2| = |r_1|\,|r_2|$, et $|r_1 + r_2| = |r_1| + |r_2|$ pour tous $r_1, r_2 \in \mathcal{R}(\ell^2)$. De plus, r est inversible si et seulement si $|r| \neq 0$.

5.2 Polynômes représentables

Dans cette partie, on prouve quelques résultats positifs en exhibant une classe de polynômes représentables. Bien que la majeure partie de ce chapitre concerne des résultats d'impossibilité, ces résultats positifs nous sont utiles pour donner une caractérisation des polynômes multilinéaires représentables.

Lemme 5.2

> Soit f et g deux polynômes représentables. Alors le polynôme fg est représentable.

Démonstration : Si les matrices $\mathcal{M}$ et $\mathcal{N}$ représentent respectivement f et g, alors il suffit de construire la matrice diagonale par blocs contenant les deux blocs $\mathcal{M}$ et $\mathcal{N}$. La vision en terme de graphe consiste à dire que le graphe représentant fg est l'union disjointe de $\mathcal{G}(\mathcal{M})$ et $\mathcal{G}(\mathcal{N})$. □

Dans le lemme suivant, nous montrons que le carré de tout polynôme est représentable. Il existe plusieurs constructions pour ce résultat. Deux

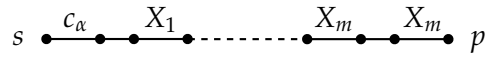


FIGURE 5.2 – Graphe $\mathcal{G}_\alpha$ associé à un terme $c_\alpha X^\alpha$ avec $\alpha_1 \geq 1$ et $\alpha_m \geq 2$.

constructions différentes sont données dans les articles [42] et [47]. Nous reproduisons ici celle de l'article [47].

Lemme 5.3

Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$. Alors f^2 est représentable.

De plus, il existe un graphe $\mathcal{G}$ représentant f^2 ayant deux sommets distingués s et p et tel que $\det(\mathcal{G} \setminus \{s, p\}) = 1$ et $\det(\mathcal{G} \setminus \{s\}) = \det(\mathcal{G} \setminus \{p\}) = 0$.

Démonstration : Le graphe $\mathcal{G}$ que l'on construit a un nombre pair de sommets et aucune boucle. Ceci implique alors que $\mathcal{G} \setminus \{s\}$ et $\mathcal{G} \setminus \{t\}$ n'admettent aucun couplage partiel, donc $\det(\mathcal{G} \setminus \{s\}) = \det(\mathcal{G} \setminus \{p\}) = 0$.

On considère $f = \sum_\alpha c_\alpha X^\alpha$ où la somme porte sur un sous-ensemble fini de $\mathbb{N}^m$ et $X^\alpha = X_1^{\alpha_1} \cdots X_m^{\alpha_m}$. On représente le carré d'un monôme $c_\alpha X^\alpha$ par un graphe $\mathcal{G}_\alpha$ ayant $2(\deg(X^\alpha) + 1)$ sommets : pour cela, on voit $c_\alpha X^\alpha$ comme un produit de $(\deg(X^\alpha) + 1)$ éléments (avec répétition) de $\{c_\alpha, X_1, \ldots, X_m\}$. À chacun de ces éléments e, on associe deux sommets et une arête de poids e les reliant. Ainsi, en faisant l'union disjointe de toutes ces arêtes, on obtiendrait un graphe ayant $2(\deg(X^\alpha) + 1)$ sommets représentant le carré de $c_\alpha X^\alpha$. Pour que $\det(\mathcal{G}_\alpha \setminus \{s, p\}) = 1$, au lieu de faire l'union disjointe de ces arêtes on les agence en un chemin en ajoutant des arêtes de poids 1 entre chaque , et on note s et p les extrémités du chemin (Fig. 5.2). Clairement, $\det(\mathcal{G}_\alpha) = (c_\alpha X^\alpha)^2$ et $\det(\mathcal{G}_\alpha \setminus \{s, p\}) = 1$.

Pour obtenir $\mathcal{G}$ à partir des $\mathcal{G}_\alpha$, on crée un fuseau en faisant l'union des $\mathcal{G}_\alpha$ dans laquelle on identifie tous les sommets s ensemble, et tous les sommets p ensemble. Pour calculer $\det(\mathcal{G})$, on étudie ses couplages — qui sont nécessairement parfaits puisqu'il n'y a pas de boucle. Supposons que s soit couvert par une arête du graphe $\mathcal{G}_\alpha$ pour un certain α. Puisque $\mathcal{G}_\alpha \setminus \{p\}$ n'admet pas de couverture par cycles, alors p doit également être couvert par une arête de $\mathcal{G}_\alpha$. Un couplage de $\mathcal{G}$ est donc fait d'un couplage de $\mathcal{G}_\alpha$ et des couplages de $\mathcal{G}_\beta \setminus \{s, p\}$ pour tout $\beta \neq \alpha$. Le poids du couplage de $\mathcal{G}_\alpha$ est $(c_\alpha X^\alpha)^2$ et les couplages des $\mathcal{G}_\beta \setminus \{s, p\}$ sont de poids 1. Ainsi, $\det(\mathcal{G})$ étant la somme des poids de tous les couplages de $\mathcal{G}$, $\det(\mathcal{G}) = \sum_\alpha (c_\alpha X^\alpha)^2 = f^2$. Enfin, un couplage de $\mathcal{G} \setminus \{s, p\}$ est fait des couplages des $\mathcal{G}_\alpha \setminus \{s, p\}$. Il n'en existe donc qu'un, et il est de poids 1. □

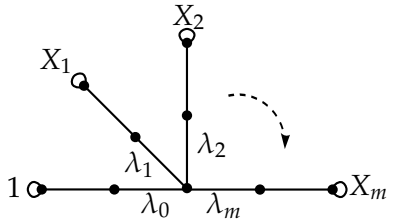


FIGURE 5.3 – Graphe représentant le polynôme $L = \lambda_0^2 + \lambda_1^2 X_1 + \cdots + \lambda_m^2 X_m$.

À l'aide de ces deux lemmes, on peut exhiber une classe relativement large de polynômes représentables.

Théorème 5.4

Soit $f = L_1 \times \cdots \times L_k \in \mathbb{F}[X_1, \ldots, X_m]$ où pour $1 \leq i \leq k$,

$$L_i(X_1, \ldots, X_m) = f_{i0}^2 + f_{i1}^2 X_1 + \cdots + f_{im}^2 X_m$$

avec $f_{i1}, \ldots, f_{im} \in \mathbb{F}[X_1, \ldots, X_m]$. Alors f est représentable.

Démonstration : D'après le lemme 5.2, il suffit de savoir représenter chaque L_i. On construit dans un premier temps un graphe représentant un polynôme

$$L(X_1, \ldots, X_m) = \lambda_0^2 + \lambda_1^2 X_1 + \cdots + \lambda_m^2 X_m$$

où $\lambda_0, \ldots, \lambda_m \in \mathbb{F}$. On représente $\lambda_i^2 X_i$ par un graphe $\mathcal{G}_i$ à trois sommets s_i, v_i et p_i avec une arête de poids λ_i entre s_i et v_i, une arête de poids 1 entre v_i et p_i et une boucle de poids X_i sur le sommet p_i. On fait de même pour λ_0^2 mais le poids de la boucle sur le sommet p_0 est alors 1. Le polynôme linéaire L est alors représenté par le graphe $\mathcal{G}_L$ consistant en l'union de tous les $\mathcal{G}_i$ dans laquelle on identifie tous les s_i (Fig. 5.3). Un couplage partiel de $\mathcal{G}_L$ utilise une arête $\{s_i, v_i\}$ pour couvrir le sommet central, puis doit couvrir le sommet p_i correspondant par la boucle de poids X_i. Il reste à couvrir tous les autres sommets v_j et p_j, ce qui ne peut se faire que par les arêtes $\{v_j, p_j\}$. Le poids d'un tel couplage est $\lambda_i^2 X_i$. Ainsi, $\det(\mathcal{G}_L) = L$.

Soit maintenant $\mathcal{G}_{f_i}$ le graphe représentant f_i^2 qui est construit à l'aide du lemme 5.3, et s et p ses sommets distingués. Par abus de langage, on désigne par λ_i l'arête $\{s_i, v_i\}$ de poids λ_i dans $\mathcal{G}_L$. On considère le graphe $\mathcal{G}_L \setminus \lambda_i$ obtenu en enlevant cette arête à $\mathcal{G}_L$ — mais en gardant les deux sommets s_i et v_i. On peut construire un nouveau graphe $\mathcal{G}'_L$ comme l'union de $\mathcal{G}_L \setminus \lambda_i$ et $\mathcal{G}_{f_i}$ dans laquelle on identifie s_i et s d'une part, et v_i et p d'autre part. En d'autres termes, $\mathcal{G}'_L$ est obtenu à partir de $\mathcal{G}_L$ en remplaçant l'arête entre s_i et v_i par le fuseau représentant f_i^2. Un couplage partiel de $\mathcal{G}'_L$ peut être soit constitué d'un couplage de $\mathcal{G}_{f_i}$, de la boucle sur p_i et des arêtes $\{v_j, p_j\}$ pour tout $j \neq i$, soit d'un couplage de

$\mathcal{G}_{f_i} \setminus \{s, p\}$ et d'un couplage de $\mathcal{G}_L \setminus \lambda_i$. Ainsi,

$$\begin{aligned}\det(\mathcal{G}'_L) &= \det(\mathcal{G}_{f_i}) \times X_i + \det(\mathcal{G}_{f_i} \setminus \{s, p\}) \times \det(\mathcal{G}_L \setminus \lambda_i) \\ &= f_i^2 X_i + 1 \times \det(\mathcal{G}_L \setminus \lambda_i).\end{aligned}$$

On peut donc remplacer dans $\mathcal{G}_L$ chaque λ_i par G_{f_i} pour obtenir un graphe représentant $f_0^2 + f_1^2 X_1 + \cdots + f_m^2 X_m$. □

Si $\mathbb{F}$ est un corps fini de caractéristique 2, un cas particulier de ce théorème est l'existence d'une représentation déterminantielle symétrique pour tout polynôme linéaire à coefficients dans $\mathbb{F}$ puisque tout élément est alors un résidu quadratique.

Le théorème précédent motive la définition de représentation déterminantielle symétrique *généralisée*.

Définition 5.5

> Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$. Une *représentation déterminantielle symétrique généralisée* de f est une matrice symétrique carrée $\mathcal{M}$ telle que $\det(\mathcal{M}) = f$ et dont les coefficients sont des éléments $\mathbb{F}[X_1, \ldots, X_m]$ tels que chaque coefficient diagonal est de la forme $f_0^2 + f_1^2 X_1 + \cdots + f_m^2 X_m$ où $f_0, \ldots, f_m \in \mathbb{F}[X_1, \ldots, X_m]$.

Par souci de concision, on parlera respectivement de *représentation généralisée* et *représentation classique* pour parler de représentation déterminantielle symétrique généralisée et de représentation déterminantielle symétrique (non généralisée).

Théorème 5.6

> Un polynôme $f \in \mathbb{F}[X_1, \ldots, X_m]$ est représentable si et seulement s'il admet une représentation généralisée.

Démonstration : Une représentation classique est un cas particulier de représentation généralisée. La définition 5.5 peut être traduite en terme de graphes en disant qu'il existe un graphe $\mathcal{G}$ dont les poids des arêtes sont des polynômes quelconques et les poids des boucles des polynômes de la forme $f_0^2 + f_1^2 X_1 + \cdots + f_m^2 X_m$ et tel que $\det(\mathcal{G}) = f$. Nous allons montrer comment le graphe $\mathcal{G}$ peut être modifié, sans changer son déterminant, de telle sorte que les poids de ses arêtes et boucles appartiennent à $\mathbb{F} \cup \{X_1, \ldots, X_m\}$.

On utilise la même technique que dans la preuve du théorème 5.4 pour remplacer une arête de poids p quelconque par un fuseau représentant p^2. On s'intéresse maintenant aux boucles.

Soit v un sommet de $\mathcal{G}$ avec une boucle de poids $L = f_0^2 + f_1^2 X_1 + \cdots + f_m^2 X_m$. Soit $\mathcal{G}_L$ le graphe représentant L construit par le théorème 5.4, et $\mathcal{G}_0$ le graphe $\mathcal{G}$ dans lequel on a supprimé la boucle sur v. Alors on remplace $\mathcal{G}$ par l'union de $\mathcal{G}_0$ et $\mathcal{G}_L$ dans laquelle v est identifié avec le

sommet central de $\mathcal{G}_L$. Notons $\mathcal{G}'$ ce nouveau graphe, et appelons v le sommet central de $\mathcal{G}_L$ par abus de notation. On remarque en premier lieu que $\det(\mathcal{G}_L \setminus \{v\}) = 1$. De plus, un couplage de $\mathcal{G}'$ couvre le sommet v soit par une arête de $\mathcal{G}_L$, soit par une arête de $\mathcal{G}$. On en déduit que

$$\begin{aligned}\det(\mathcal{G}') &= \det(\mathcal{G}_L) \times \det(\mathcal{G} \setminus \{v\}) + \det(\mathcal{G}_L \setminus \{v\}) \times \det(\mathcal{G}_0) \\ &= L \times \det(\mathcal{G} \setminus \{v\}) + 1 \times \det(\mathcal{G}_0) = \det(\mathcal{G}).\end{aligned}$$

Pour achever la preuve, il suffit d'effectuer cette opération pour toutes les boucles de $\mathcal{G}$. □

5.3 Obstructions aux représentations

On s'intéresse maintenant aux résultats d'impossibilité, déduisant en particulier que tout polynôme n'est pas représentable en caractéristique 2 contrairement au cas des autres caractéristiques.

5.3.1 Condition nécessaire

Pour donner une condition nécessaire pour qu'un polynôme soit représentable, on définit une notion de factorisation *modulo* un idéal $\mathcal{I}(\ell)$. On rappelle que $\mathcal{I}(\ell) = \langle X_1^2 + \ell_1, \ldots, X_m^2 + \ell_m \rangle$, $\mathcal{R}(\ell) = \mathbb{F}[X_1, \ldots, X_m]/\mathcal{I}(\ell)$ et π_ℓ est la projection canonique $\mathbb{F}[X_1, \ldots, X_m] \to \mathcal{R}(\ell)$.

Définition 5.7

Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$. Alors f est dit *factorisable modulo* $\mathcal{I}(\ell)$ s'il existe des éléments linéaires $L_1, \ldots, L_k \in \mathcal{R}(\ell)$ tels que

$$\pi_\ell(f) = L_1 \times \cdots \times L_k.$$

Cette notion nous permet d'exprimer une condition nécessaire pour qu'un polynôme soit représentable.

Théorème 5.8

Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$ un polynôme représentable. Alors pour tout multiplet de carrés ℓ^2, f est factorisable *modulo* $\mathcal{I}(\ell^2)$.

La preuve de ce théorème est de nature algorithmique. On donne en effet un algorithme qui, étant donnée une représentation de f sous forme de matrice symétrique $\mathcal{M}$ et un multiplet de carré ℓ^2, construit la factorisation de f *modulo* $\mathcal{I}(\ell^2)$. L'idée générale est de projeter une représentation $\mathcal{M}$ de f pour obtenir une représentation $\mathcal{N} = \pi(\mathcal{M})$ de $\pi(f)$, d'effectuer des opérations élémentaires sur les lignes et les colonnes de $\mathcal{N}$ pour *isoler* un élément diagonal $\mathcal{N}_{ii}$, c'est-à-dire annuler tous les coefficients de $\mathcal{N}$ d'indices (i, j) et (j, i) pour $j \neq i$, tout en préservant sa symétrie. On montre

alors que $\mathcal{N}_{ii}$ est un élément linéaire de $\mathcal{R}(\ell^2)$, et on obtient donc que $\pi(f) = \mathcal{N}_{ii} \det(\mathcal{N}')$ où $\mathcal{N}'$ est la matrice $\mathcal{N}$ de laquelle on a enlevé la ligne et la colonne d'indices i. Ceci nous permet de conclure par induction sur la dimension de $\mathcal{N}$.

La correction de cette démarche est justifiée par un ensemble de lemmes que l'on démontre maintenant. On fixe pour ce qui suit un multiplet de carrés ℓ^2.

On commence par étendre la définition de représentation généralisée aux éléments de $\mathcal{R}(\ell^2)$.

Définition 5.9

> Soit $r \in \mathcal{R}(\ell^2)$. Une représentation déterminantielle symétrique généralisée de r est une matrice symétrique $\mathcal{N}$ à coefficients dans $\mathcal{R}(\ell^2)$ dont les coefficients diagonaux sont linéaires et telle que $\det(\mathcal{N}) = r$.

De même que pour les polynômes, on parlera de représentation généralisée, voire de représentation tout court, pour parler de représentation déterminantielle symétrique généralisée.

Lemme 5.10

> Soit $\mathcal{M}$ une représentation généralisée d'un polynôme $f \in \mathbb{F}[X_1, \ldots, X_m]$. Alors la matrice $\pi(\mathcal{M})$ est une représentation généralisée de $\pi(f)$.

Démonstration : Une entrée diagonale de $\mathcal{M}$ est de la forme $f_0^2 + f_1^2 X_1 + \ldots + f_m^2 X_m$. Puisque $\pi(f_i^2) = \pi(f_i)^2$, $\pi(f_i^2)$ est une constante. Donc $\pi(f_0^2 + f_1^2 X_1 + \ldots + f_m^2 X_m)$ est linéaire. □

Lemme 5.11

> Soit $\mathcal{N}$ une représentation d'un élément $r \in \mathcal{R}(\ell^2)$. Alors il existe une représentation $\mathcal{N}'$ de r dont les coefficients non diagonaux sont des constantes.

Démonstration : Supposons que $\mathcal{N}_{ij} = \mathcal{N}_{ji} = \pi(f)$, $i \neq j$, pour un certain polynôme f. Puisque le déterminant de $\mathcal{N}$ égale $\sum_\sigma \prod_i \mathcal{N}_{i,\sigma(i)}$ où la somme porte sur les involutions σ (d'après la proposition 5.1), tout terme que $\mathcal{N}_{ij}$ divise dans le déterminant est également un multiple de $\mathcal{N}_{ji}$. Autrement dit, si $\pi(f)$ divise un terme du déterminant, alors $\pi(f)^2$ également. Puisque $|\pi(f)|^2 = \pi(f)^2$ par définition, remplacer $\mathcal{N}_{ij}$ et $\mathcal{N}_{ji}$ par la valeur absolue $|\pi(f)| \in \mathbb{F}$ ne change pas le déterminant de $\mathcal{N}$. On construit alors $\mathcal{N}'$ en remplaçant chaque coefficient non diagonal par sa valeur absolue. □

Grâce à ce lemme, on peut supposer dans la suite que toutes les représentations manipulées ont comme coefficients non diagonaux des constantes.

Nous définissons maintenant les ingrédients principaux utiles à la preuve du théorème. Premièrement, nous appelons CLEAN l'algorithme qui remplace

Algorithme 1: $\textsc{Add}_{i,j,\alpha}(\mathcal{N})$

```
n ← dimension de N
pour k = 1 à n faire N_jk ← N_jk + α N_ik        // L_j ← L_j + α L_i
pour k = 1 à n faire N_kj ← N_kj + α N_ki        // C_j ← C_j + α C_i
retourner Clean(N)
```

chaque coefficient non diagonal d'une matrice par sa valeur absolue comme dans la preuve du lemme 5.11. On définit deux autres algorithmes très simples que l'on applique à la matrice symétrique représentant un élément $r \in \mathcal{R}(\ell^2)$ pour la rendre diagonale sans changer son déterminant. Ces deux algorithmes, tout comme Clean, dépendent de ℓ^2 bien qu'il n'y soit pas fait explicitement mention afin d'alléger les notations.

L'algorithme $\textsc{Add}_{i,j,\alpha}$ (algorithme 1) permet d'effectuer des opérations élémentaires de lignes et de colonnes sur une représentation d'un élément $r \in \mathcal{R}(\ell^2)$, tout en gardant certaines propriétés.

Lemme 5.12

> Soit $\mathcal{N}$ une représentation d'un élément $r \in \mathcal{R}(\ell^2)$. Alors pour tout $i \neq j$ et $\alpha \in \mathcal{R}(\ell^2)$, $\textsc{Add}_{i,j,\alpha}(\mathcal{N})$ est une représentation de r.
>
> De plus, les coefficients non diagonaux de $\textsc{Add}_{i,j,\alpha}(\mathcal{N})$ sont des constantes.

Démonstration : On peut décomposer l'algorithme en deux étapes. On commence par ajouter la ligne i multipliée par α à la ligne j et la colonne i multipliée par α à la colonne j, puis on remplace les coefficients non diagonaux par leurs valeurs absolues. La première étape ne change pas le déterminant, et garde la symétrie de la matrice puisque l'on effectue des opérations symétriques sur la ligne et la colonne j. De plus, le coefficient d'indice (j, j) est modifié deux fois, et remplacé par $\mathcal{N}_{jj} + \alpha\mathcal{N}_{ij} + \alpha(\mathcal{N}_{ji} + \alpha\mathcal{N}_{ii}) = \mathcal{N}_{jj} + \alpha^2\mathcal{N}_{ii}$. Puisque $\alpha^2 \in \mathbb{F}$ pour tout $\alpha \in \mathcal{R}(\ell^2)$, $\mathcal{N}_{jj}$ est bien remplacé par un élément linéaire. Enfin, l'application finale de l'algorithme Clean assure la deuxième partie du lemme. □

L'algorithme $\textsc{Isolate}_i$ (algorithme 2) permet d'isoler le coefficient diagonal d'indice (i, i) : sous certaines conditions, il met à 0 tous les coefficients des ligne et colonne i sauf le coefficient diagonal. On rappelle qu'un élément $r \in \mathcal{R}(\ell)$ est inversible si et seulement si $|r|_\ell \neq 0$.

Lemme 5.13

> Soit $\mathcal{N}$ une représentation d'un élément $r \in \mathcal{R}(\ell)$. Si $\mathcal{N}_{ii}$ est inversible, $\textsc{Isolate}_i(\mathcal{N})$ est une représentation de r dans laquelle le seul coefficient non nul des ligne et colonne i est $\mathcal{N}_{ii}$.

Algorithme 2: $\textsc{Isolate}_i(\mathcal{N})$

```
n ← dimension de N
pour j = 1 à n faire
    si j ≠ i alors
        α ← N_ij × |N_ii|^{-1}
        A ← Add_{i,j,α}(N)
retourner N
```

Algorithme 3: $\textsc{NonZeroDiag}_i(\mathcal{N})$

```
n ← dimension de N
M ← matrice carrée de dimension (n + 1)
pour 1 ≤ ℓ, j ≤ n faire M_ℓj ← N_ℓj          // Copier N dans M
pour j = 1 à n faire M_{j,n+1} ← 0 ; M_{n+1,j} ← 0
M_{n+1,n+1} ← 1
M_{i,n+1} ← 1 ; M_{n+1,i} ← 1
M_ii ← N_ii + 1
retourner Isolate_i(M)
```

Démonstration : La matrice $\textsc{Isolate}_i(\mathcal{N})$ est une représentation de r puisque pour tous j et α, $\textsc{Add}_{i,j,\alpha}(\mathcal{N})$ en est une. Puisque $\mathcal{N}_{ii}$ est inversible, $|\mathcal{N}_{ii}| \neq 0$. Soit $\alpha = \mathcal{N}_{ij} \times |\mathcal{N}_{ii}|^{-1}$ pour un indice $j \neq i$ et considérons l'action de $\textsc{Add}_{i,j,\alpha}$ sur la ligne i de $\mathcal{N}$. Le seul coefficient à être modifié est $\mathcal{N}_{ij}$ qui est remplacé par $\mathcal{N}_{ij}(1 + |\mathcal{N}_{ii}|^{-1}\mathcal{N}_{ii})$ puis par sa valeur absolue. Comme $\left|1 + |N_{ii}|^{-1}\mathcal{N}_{ii}\right| = \left|1 + |N_{ii}|^{-1}|\mathcal{N}_{ii}|\right| = 0$, $\mathcal{N}_{ij}$ est remplacé par 0. De même pour la colonne i, le seul élément modifié est $\mathcal{N}_{ij}$ qui est remplacé par 0.

Le seul coefficient non nul dans les ligne et colonne i de $\textsc{Isolate}_i(\mathcal{N})$ est donc celui d'indice (i, i). □

Il reste à voir comment s'assurer qu'il existe bel et bien un indice i tel que $\mathcal{N}_{ii}$ soit inversible. En fait, ce n'est pas toujours le cas, et on décrit donc un nouvel algorithme $\textsc{NonZeroDiag}_i$ (algorithme 3) qui permet d'en créer s'il n'en existe pas.

Lemme 5.14

Soit $\mathcal{N}$ une représentation d'un élément $r \in \mathcal{R}(\ell^2)$ telle qu'aucun coefficient diagonal ne soit inversible. Si $\mathcal{N}_{ii} \neq 0$ et s'il existe $j \neq i$ tel que $\mathcal{N}_{ij} \neq 0$, alors $\textsc{NonZeroDiag}_i(\mathcal{N})$ est une représentation de r dont le coefficient d'indice (j, j) est inversible et celui d'indice (i, i) est isolé.

Démonstration : Sans perte de généralité, supposons que $i = 1$. La matrice

$\mathcal{M}$ construite par l'algorithme est

$$\mathcal{M} = \begin{pmatrix} \mathcal{N}_{1,1}+1 & \mathcal{N}_{1,2} & \dots & \mathcal{N}_{1,n} & 1 \\ \mathcal{N}_{2,1} & & & & 0 \\ \vdots & & \mathcal{N}' & & \vdots \\ \mathcal{N}_{n,1} & & & & 0 \\ 1 & 0 & \dots & 0 & 1 \end{pmatrix}$$

où $\mathcal{N}'$ est la matrice $\mathcal{N}$ à laquelle on a supprimé la première ligne et la première colonne. Alors en ajoutant la dernière ligne de $\mathcal{M}$ à sa première, et sa dernière colonne à sa première, on obtient la matrice

$$\mathcal{M}' = \begin{pmatrix} \mathcal{N}_{1,1} & \mathcal{N}_{1,2} & \dots & \mathcal{N}_{1,n} & 0 \\ \mathcal{N}_{2,1} & & & & 0 \\ \vdots & & \mathcal{N}' & & \vdots \\ \mathcal{N}_{n,1} & & & & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

donc $\det(\mathcal{M}) = \det(\mathcal{M}') = \det(\mathcal{N}) = r$. Ceci prouve que $\mathcal{M}$ est une représentation de r, et par suite que $\textsc{Isolate}_1(\mathcal{M})$ également.

Maintenant, lorsqu'on applique $\textsc{Isolate}_1$ à $\mathcal{M}$, puisque $\mathcal{M}_{1j} = \mathcal{N}_{1j}$ est non nul par hypothèse, le coefficient $\mathcal{M}_{jj}$ est remplacé par $\mathcal{M}_{jj} + (\mathcal{M}_{1j}\,|\mathcal{M}_{11}|^{-1})^2\mathcal{M}_{11}$. Par hypothèse $|\mathcal{M}_{jj}| = 0$ et $|\mathcal{M}_{11}| = |\mathcal{N}_{11}+1| = 1$, donc

$$\left|\mathcal{M}_{jj} + (\mathcal{M}_{1j}\,|\mathcal{M}_{11}|^{-1})^2\mathcal{M}_{11}\right| = 0 + \left|M_{1j}\right|^2.$$

Mais comme $\mathcal{M}_{1j} = \mathcal{N}_{1j}$ n'est pas diagonal, c'est une constante et $|\mathcal{N}_{1j}| \neq 0$. □

On dispose maintenant de tous les outils nécessaires pour prouver le théorème.

Démonstration du théorème 5.8 : Il suffit d'utiliser les lemmes précédents dans le bon ordre. Soit donc $f \in \mathbb{F}[X_1, \dots, X_m]$ un polynôme représenté par une matrice $\mathcal{M}$. D'après le lemme 5.10, la matrice $\mathcal{N} = \pi(\mathcal{M})$ est une représentation de $r = \pi(f)$. De plus, on peut supposer que les coefficients non diagonaux de $\mathcal{N}$ sont des constantes d'après le lemme 5.11. Il reste à prouver qu'on peut trouver des éléments linéaires $L_1, \dots, L_k \in \mathcal{R}(\ell^2)$ tels que $r = L_1 \cdots L_k$. Pour cela, on cherche à rendre la matrice $\mathcal{N}$ diagonale, ou plus exactement *pseudo-diagonale* : une telle matrice est diagonale par blocs avec deux blocs dont l'un est diagonal, et l'autre de diagonale nulle. Le déterminant d'une matrice pseudo-diagonale est un produit de facteurs linéaires. En effet, les coefficients diagonaux étant linéaires, le déterminant du bloc diagonal est un produit de facteurs linéaires. Et comme les coefficients non diagonaux sont des constantes, le déterminant du second bloc est une constante.

Algorithme 4: FACTORIZATION($\mathcal{N}$)

si $\exists (i,j), \mathcal{N}_{ii} \neq 0$ *et* $\mathcal{N}_{ij} \neq 0$ **alors**
  // Pseudo-diagonalisation :
  $I \leftarrow \{i : N_{ii} \neq 0 \text{ et } \exists j, \mathcal{N}_{ij} \neq 0\}$
  // Lemme 5.14 :
  si $\forall i \in I, |\mathcal{N}_{ii}| = 0$ **alors**
    $i \leftarrow$ indice tel que $\mathcal{N}_{ii} \neq 0$ et $\exists j, \mathcal{N}_{ij} \neq 0$
    $\mathcal{N} \leftarrow \text{NONZERODIAG}_i(\mathcal{N})$
  // Lemme 5.13 :
  $i \leftarrow$ indice tel que $|\mathcal{N}_{ii}| \neq 0$ et $\exists j, \mathcal{N}_{ij} \neq 0$
  $\mathcal{N} \leftarrow \text{ISOLATE}_i(\mathcal{N})$
  FACTORIZATION($\mathcal{N}$)
sinon
  // Extraction des facteurs linéaires :
  $J \leftarrow \{i : \mathcal{N}_{ii} = 0\}$
  retourner $\{\mathcal{N}_{ii} : i \notin J\} \cup \{\det(\mathcal{N}_J)\}$

L'algorithme FACTORIZATION (algorithme 4) effectue la pseudo-diagonalisation avant d'extraire de la matrice pseudo-diagonale les facteurs linéaires. Pour un ensemble J d'indices, on note $\mathcal{N}_J$ la sous-matrice de $\mathcal{N}$ obtenu en ne gardant que les lignes et les colonnes dont les indices appartiennent à J.

De même que dans l'algorithme, notons I l'ensemble des indices i tels que $\mathcal{N}_{ii} \neq 0$ et qu'il existe $j \neq i$ tel que $\mathcal{N}_{ij} \neq 0$. Une matrice est pseudo-diagonale si $I = \emptyset$. Nous allons montrer qu'à chaque appel récursif de FACTORIZATION, le cardinal de I diminue strictement. L'algorithme NONZERODIAG_i augmente la dimension de $\mathcal{N}$ mais d'après le lemme 5.14, l'indice i n'appartient plus à I après l'application de l'algorithme à $\mathcal{N}$. Ainsi, le cardinal de I reste constant. Ensuite, l'effet de l'algorithme ISOLATE_i est précisément de supprimer un indice de I, donc lors de l'appel récursif son cardinal a bien strictement diminué. Ceci montre qu'en temps borné par la dimension de la matrice $\mathcal{N}$ d'entrée, celle-ci est pseudo-diagonalisée. Le fait qu'elle reste bien une représentation de r est assuré par les lemmes 5.13 et 5.14.

La remarque effectuée précédemment sur les matrices pseudo-diagonales montre que la deuxième partie de l'algorithme qui extrait de $\mathcal{N}$ les facteurs linéaires de r est correcte. □

5.3.2 Exemple

On a montré dans la partie précédente qu'un polynôme est représentable si et seulement si sa projection *modulo* un multiplet de carrés est factorisable en un produit d'éléments linéaires de $\mathcal{R}(\ell^2)$. Considérons par exemple les polynômes $\mathbb{F}_2[X, Y, Z]$ où $\mathbb{F}_2$ est le corps à deux éléments. Alors l'anneau $\mathcal{R}(1, 1, 1)$ possède 256 éléments dont 136 peuvent être écrits comme produits de facteurs linéaires et 120 ne le peuvent pas. En particulier, $\pi(XY + Z)$ ne s'écrit pas sous la forme d'un produit de facteurs linéaires. D'après le théorème 5.8, on en déduit que le polynôme $XY + Z$ ne peut pas être représenté sous la forme du déterminant d'une matrice symétrique à coefficients dans $\mathbb{F}_2 \cup \{X, Y, Z\}$.

5.3.3 Polynômes multilinéaires

Une question naturelle consiste à se demander si l'on peut caractériser les polynômes représentables en caractéristique 2. Cette partie est dédiée à une étude de la caractérisation pour le cas des polynômes multilinéaires, c'est-à-dire dont le degré en chaque variable est au plus 1. On montre que la condition nécessaire du théorème 5.8 est en fait suffisante pour les polynômes multilinéaires. Ce résultat repose sur un lemme de structure des représentations déterminantielles symétriques valable pour tout polynôme, multilinéaire ou non.

Lemme 5.15

> Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$ un polynôme représentable. Alors il existe une représentation $\mathcal{M}$ de f telle que chaque variable apparaît au plus une fois sur la diagonale.

Démonstration : Soit $\mathcal{M}$ une représentation quelconque de f, c'est-à-dire une matrice symétrique à coefficients dans $\mathbb{F} \cup \{X_1, \ldots, X_m\}$ telle que $\det(\mathcal{M}) = f$. Supposons que $\mathcal{M}_{i_1 i_1} = \mathcal{M}_{i_2 i_2} = X_i$. Soit $\mathcal{M}'$ obtenue en ajoutant la ligne i_1 à la ligne i_2 puis la colonne i_1 à la colonne i_2 dans $\mathcal{M}$. Alors $\mathcal{M}'_{i_2 i_2} = 0$. De plus, $\mathcal{M}'$ est symétrique et $\det(\mathcal{M}') = \det(\mathcal{M})$. On peut effectuer cette opération autant de fois que nécessaire pour obtenir une matrice ayant les propriétés requises. □

On peut en déduire la caractérisation voulue lorsque $\mathbb{F}$ est un corps fini. Le cas des corps infinis est brièvement discutée ensuite.

Théorème 5.16

> Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$ un polynôme multilinéaire où $\mathbb{F}$ est un corps fini de caractéristique 2. Alors les trois propriétés suivantes sont équivalentes :
>
> (1) f est représentable ;
>
> (2) pour tout multiplet de carrés $\ell^2 \in \mathbb{F}^m$, f est factorisable *modulo*

$\mathcal{I}(\ell^2)$;

(3) il existe un multiplet de carrés $\ell^2 \in \mathbb{F}^m$ tel que f est factorisable *modulo* $\mathcal{I}(\ell^2)$.

Démonstration : L'implication (1) $\Longrightarrow$ (2) est un cas particulier du théorème 5.8, et l'implication (2) $\Longrightarrow$ (3) est évidente. Il nous reste à montrer (3) $\Longrightarrow$ (1).

Soit $\ell^2 \in \mathbb{F}^m$ tel que $\pi_{\ell^2}(f) = L_1 \cdots L_k$ où chaque L_i est un élément linéaire de $\mathcal{R}(\ell^2)$. Pour tout i, $\rho_{\ell^2}(L_i)$ est un polynôme linéaire, donc d'après le théorème 5.4 le polynôme $g = \rho_{\ell^2}(L_1) \cdots \rho_{\ell^2}(L_k)$ est représentable par une matrice $\mathcal{M}$. D'après le lemme 5.15, on peut supposer que chaque variable apparaît au plus une fois sur la diagonale de $\mathcal{M}$. En combinant les lemmes 5.10 et 5.11, on en déduit que $\pi_{\ell^2}(g) = \pi_{\ell^2}(f)$ admet une représentation généralisée $\mathcal{N}$ telle que chaque $\mathcal{N}_{ii}$ est linéaire et chaque $\mathcal{N}_{ij}$ une constante pour $i \neq j$. Considérons la matrice $\mathcal{M}'$ définie par $\mathcal{M}'_{ij} = \rho_{\ell^2}(\mathcal{N}_{ij})$ pour tous i et j. Puisque chaque variable apparaît au plus une fois sur la diagonale, $\det(\mathcal{M}')$ est un polynôme multilinéaire. De plus $\pi_{\ell^2}(f) = \pi_{\ell^2}(g) = \det(\mathcal{N}) = \pi_{\ell^2}(\det(\mathcal{M}'))$. Donc f et $\det(\mathcal{M}')$ étant deux polynômes multilinéaires, ils sont égaux. Ainsi f est représentable. □

Si $\mathbb{F}$ est un corps infini, on peut obtenir une caractérisation similaire. Pour cela, la conclusion du théorème 5.8 doit être renforcée de la manière suivante : si f est un polynôme représentable, alors il existe des polynômes linéaires $L_1, \ldots, L_k$ *dont les coefficients sont des résidus quadratiques dans* $\mathbb{F}$ tels que $\pi_{\ell^2}(f) = \pi_{\ell^2}(L_1 \cdots L_k)$. Il suffit de vérifier que la preuve du théorème 5.8 est en réalité une preuve de ce résultat un peu plus fort. On obtient la réciproque de cet énoncé renforcé à l'aide du théorème 5.4.

5.3.4 *Vers une caractérisation complète ?*

Le théorème 5.8 est valable pour tout polynôme. Ainsi, nous avons une condition nécessaire pour tout polynôme pour qu'il soit représentable. Dans le cas d'un polynôme multilinéaire, on utilise le fait que $\rho \circ \pi(f) = f$ pour obtenir une caractérisation de ceux qui sont représentables. Si f n'est pas multilinéaire, la projection de f *modulo* un idéal $\mathcal{I}(\ell)$ peut faire perdre toute l'information sur le polynôme. Par exemple si $f = X_1^2 Q$ où Q est un polynôme multilinéaire, $\pi_{(1,\ldots)}(f) = Q$ tandis que $\pi_{(0,\ldots)}(f) = 0$. Il est donc certainement impossible de déduire quoi que ce soit sur f à partir de sa projection *modulo* $\mathcal{I}(0, \ldots)$. Pour aborder partiellement cette difficulté, on s'intéresse aux projections *modulo* certains idéaux spécifiques. Dans cette partie, on suppose à nouveau que $\mathbb{F}$ est un corps fini. La discussion resterait valide pour des corps infinis en utilisant la remarque faite à la fin de la partie précédente.

Soit $\mathbb{F}(\xi_1, \ldots, \xi_m)$ le corps des fractions rationnelles en m indéterminées sur $\mathbb{F}$ et $\mathcal{I}(\xi^2) = \langle X_1^2 + \xi_1^2, \ldots, X_m^2 + \xi_m^2 \rangle$. Si $f \in \mathbb{F}[X_1, \ldots, X_m]$, on peut appliquer le théorème 5.16 au polynôme multilinéaire $\textsc{Mult}_{\xi^2}(f)$ où $\textsc{Mult}_{\xi^2} = \rho_{\xi^2} \circ \pi_{\xi^2}$. En particulier, $\textsc{Mult}_{\xi^2}(f)$ est représentable si et seulement s'il est factorisable *modulo* $\mathcal{I}(\ell^2)$ pour tout $\ell^2 \in \mathbb{F}^m$.

Cependant, nos constructions utilisent des inverses d'éléments du corps de base. Ceci signifie que l'on a une équivalence entre factorisation et existence d'une représentation pour les polynômes multilinéaires de l'anneau $\mathbb{F}(\xi_1, \ldots, \xi_m)[X_1, \ldots, X_m]$ mais aussi bien les facteurs linéaires de la factorisation que la matrice qui représente le polynôme peuvent avoir des coefficients qui sont des fractions rationnelles en les ξ_i. On ne peut alors pas toujours obtenir d'information sur f à partir d'informations sur $\textsc{Mult}_{\xi^2}(f)$. On prouve tout de même quelques résultats partiels.

Lemme 5.17

> Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$. Alors f est représentable si et seulement si $\textsc{Mult}_{\xi^2}(f)$ admet une représentation dont les coefficients non diagonaux appartiennent à $\mathbb{F}[\xi_1, \ldots, \xi_m]$.

Démonstration : Notons en premier lieu que $\textsc{Mult}_{\xi^2}$ est une bijection de $\mathbb{F}[X_1, \ldots, X_m]$ dans l'ensemble des polynômes multilinéaires à coefficients dans $\mathbb{F}[\xi_1, \ldots, \xi_m]$. En effet, son inverse $\textsc{Mult}_{\xi^2}^{-1}$ est la fonction qui envoie chaque ξ_i sur X_i.

Soit $\mathcal{M}$ une représentation de f. En utilisant le lemme 5.15, on peut supposer que chaque variable apparaît une unique fois sur la diagonale de f. Alors $\mathcal{M}' = \rho_{\xi^2}(\textsc{Clean}(\pi_{\xi^2}(\mathcal{M})))$, où l'algorithme Clean est utilisé avec $\ell^2 = \xi^2$, est une représentation de $\textsc{Mult}_{\xi^2}(f)$ d'après le lemme 5.11. Il est clair que ses coefficients non diagonaux appartiennent à $\mathbb{F}[\xi_1, \ldots, \xi_m]$. Réciproquement, d'une représentation $\mathcal{M}'$ de $\textsc{Mult}_{\xi^2}$ dont les coefficients non diagonaux appartiennent à $\mathbb{F}[\xi_1, \ldots, \xi_m]$, on obtient une représentation $\mathcal{M}$ de f en appliquant $\textsc{Mult}_{\xi^2}^{-1}$ à chacun des coefficients de $\mathcal{M}'$. Cette opération étant compatible avec l'addition et la multiplication, $\det(\mathcal{M}) = \textsc{Mult}_{\xi^2}^{-1}(\det(\mathcal{M}')) = f$. □

On énonce maintenant un équivalent du théorème 5.16 pour les polynômes non multilinéaires. Le résultat est plus modeste puisque ce n'est pas une caractérisation mais simplement deux conditions, nécessaire et suffisante, distinctes l'une de l'autre. On rappelle que le corps $\mathbb{F}$ se plonge naturellement dans l'anneau quotient $\mathcal{R}(\ell)$. Pour le prochain théorème, on travaille sur le corps $\mathbb{F}(\xi_1, \ldots, \xi_m)$. Pour un multiplet ℓ, on note

$$\mathcal{R}(\ell) = \mathbb{F}(\xi_1, \ldots, \xi_\ell)[X_1, \ldots, X_m]/\mathcal{I}(\ell).$$

Les éléments de $\mathcal{R}(\ell)$ sont des projections de polynômes à coefficients dans $\mathbb{F}(\xi_1, \ldots, \xi_m)$. Parmi ceux-ci, on distinguera les projections de polynômes à

coefficients dans $\mathbb{F}[\xi_1, \ldots, \xi_m]$ qui par analogie seront dits *à coefficients dans* $\mathbb{F}[\xi_1, \ldots, \xi_m]$.

Théorème 5.18

Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$.
- Si f est représentable, alors pour tout multiplet de carrés $\ell^2 \in \mathbb{F}^m$, $\textsc{Mult}_{\xi^2}(f)$ est factorisable *modulo* $\mathcal{I}(\ell^2)$ en un produit de facteurs linéaires à coefficients dans $\mathbb{F}(\xi_1, \ldots, \xi_m)$.
- S'il existe un multiplet de carrés $\ell^2 \in \mathbb{F}^m$ tel que $\textsc{Mult}_{\xi^2}(f)$ est factorisable *modulo* $\mathcal{I}(\ell^2)$ en un produit de facteurs linéaires à coefficients dans $\mathbb{F}[\xi_1, \ldots, \xi_m]$, alors f est représentable.

Ce théorème donne donc à la fois une condition nécessaire et une condition suffisante pour qu'un polynôme f soit représentable, mais ces deux conditions diffèrent. En effet, la condition nécessaire est d'être factorisable en un produit de facteurs linéaires à coefficients dans $\mathbb{F}(\xi_1, \ldots, \xi_m)$ quand la condition suffisante impose que les facteurs soient à coefficients dans $\mathbb{F}[\xi_1, \ldots, \xi_m]$.

Démonstration : Le premier point est simplement une application du théorème 5.8 à $\textsc{Mult}_{\xi^2}(f)$, avec comme corps de base $\mathbb{F}(\xi_1, \ldots, \xi_m)$.

Pour le second point, supposons que $\pi_{\ell^2}(\textsc{Mult}^2_{\xi}(f)) = L_1 \cdots L_k$ où pour tout i, L_i est un élément linéaire à coefficients dans $\mathbb{F}[\xi_1, \ldots, \xi_m]$. En utilisant le théorème 5.16, on construit alors une représentation $\mathcal{M}$ de $\textsc{Mult}_{\xi^2}(f)$. D'après la preuve de ce théorème, on vérifie que les coefficients non diagonaux de $\mathcal{M}$ appartiennent à $\mathbb{F}[\xi_1, \ldots, \xi_m]$. On conclut grâce au lemme 5.17. □

5.4 Représentation et factorisation des polynômes multilinéaires

Fort de notre caractérisation des polynômes multilinéaires représentables, nous étudions les conditions pour qu'un polynôme multilinéaire soit factorisable *modulo* un idéal $\mathcal{I}(\ell)$. On en déduit un algorithme qui étant donné un polynôme multilinéaire $f \in \mathbb{F}[X_1, \ldots, X_m]$ produit une représentation déterminantielle symétrique de f s'il en existe une. L'algorithme est polynomial en la taille lacunaire du polynôme. Dans toute cette partie, les polynômes manipulés sont multilinéaires et $\mathbb{F}$ est un corps fini de caractéristique 2.

5.4.1 *Résultats préliminaires*

Dans la partie précédente, nous avons raisonné sur les éléments de l'anneau quotient $\mathcal{R}(\ell)$ pour un certain multiplet ℓ. Dans les algorithmes présentés dans cette partie, on considérera des polynômes multilinéaires $f \in \mathbb{F}[X_1, \ldots, X_m]$. Le théorème 5.16 est l'outil de base utilisée dans cette partie. Puisque $f = \textsc{Mult}_{\ell}(f)$ pour tout ℓ, il peut être reformulé de la manière

suivante : un polynôme multilinéaire f est représentable si et seulement si pour tout multiplet de carrés ℓ^2, il existe des polynômes linéaires $L_1, \ldots, L_k$ tels que $f = \textsc{Mult}_{\ell^2}(L_1 \cdots L_k)$. Cette formulation revient également à dire que $\pi_{\ell^2}(f) = \pi_{\ell^2}(L_1) \cdots \pi_{\ell^2}(L_k)$. De plus, comme on l'a vu cette existence de polynômes linéaires ne dépend pas du multiplet ℓ^2. Ceci amène la définition suivante.

Définition 5.19

Un polynôme multilinéaire f est *factorisable* s'il existe un multiplet de carrés ℓ^2 et des polynômes linéaires $L_1, \ldots, L_k$ tels que

$$f = \textsc{Mult}_{\ell^2}(L_1 \times \cdots \times L_k).$$

Le théorème 5.16 affirme donc qu'un polynôme multilinéaire est représentable si et seulement s'il est factorisable.

Les algorithmes reposent de manière essentielle sur le fait qu'être factorisable ne dépend pas du multiplet ℓ^2. Deux multiplets sont utilisés, $\bar{0} = (0, \ldots, 0)$ et $\bar{1} = (1, \ldots, 1)$. Pour simplifier les notations, on note simplement 0 et 1 ces deux multiplets, et donc π_0, ρ_0 et $\textsc{Mult}_0$ d'une part, et π_1, ρ_1 et $\textsc{Mult}_1$ d'autre part, les fonctions définies dans la partie 5.1.2. De même, on définit

$$\mathcal{I}_0 = \mathcal{I}(\bar{0}) = \langle X_1^2, \ldots, X_m^2 \rangle \quad \text{et} \quad \mathcal{I}_1 = \mathcal{I}(\bar{1}) = \langle X_1^2 + 1, \ldots, X_m^2 + 1 \rangle,$$

et $\mathcal{R}_0$ et $\mathcal{R}_1$ par analogie.

On dira parfois qu'un polynôme est factorisable *modulo* $\mathcal{I}_0$ ou $\mathcal{I}_1$ au lieu de simplement factorisable pour expliciter l'idéal avec lequel on travaille. Pour un polynôme f, on note $\mathrm{lin}(f)$ sa *partie linéaire* constituée de ses termes de degré au plus 1. Par exemple, $\mathrm{lin}(X_1X_2X_3 + X_1X_2 + X_1 + X_3 + 1) = X_1 + X_3 + 1$. De plus, on note $\partial f / \partial X_i$ la dérivée partielle de f par rapport à une variable X_i. On remarque que pour un polynôme multilinéaire, $\partial f / \partial X_i$ est également le quotient dans la division euclidienne de f par X_i.

Pour montrer qu'on peut tester si un polynôme multilinéaire est factorisable, on procède en deux étapes. Dans la première, on montre qu'étant donné un polynôme multilinéaire de valuation 1 (c'est-à-dire sans coefficient constant mais avec des monômes de degré 1), on peut tester s'il est factorisable. On montre pour cela que f est factorisable si et seulement si $f = \textsc{Mult}_0(\mathrm{lin}(f) \times \frac{1}{\alpha_i} \frac{\partial f}{\partial X_i})$ où $\alpha_i X_i$ est l'un des termes de $\mathrm{lin}(f)$ (lemmes 5.20 et 5.21). La deuxième étape consiste à montrer qu'on peut toujours se ramener au cas où la valuation de f est 1 (lemmes 5.22 et 5.23). Cette deuxième étape, contrairement à la première, utilise l'idéal $\mathcal{I}_1$. L'algorithme découlant de ces quatre lemmes est présenté dans la partie suivante.

Dans le premier lemme, on montre que si f est factorisable *modulo* $\mathcal{I}_0$,

alors l'un des facteurs est un multiple de sa partie linéaire $\text{lin}(f)$.

Lemme 5.20

Soit f un polynôme multilinéaire de valuation 1. S'il existe des polynômes linéaires $L_1, \ldots, L_k$ tels que

$$f = \textsc{Mult}_0(L_1 \times \cdots \times L_k),$$

alors il existe un indice j et une constante $\alpha \in \mathbb{F}$ tels que $\text{lin}(f) = \alpha L_j$.

Démonstration : Supposons que $f = \textsc{Mult}_0(L_1 \cdots L_k)$ et soit $g = L_1 \cdots L_k$. En particulier, $g(0) = 0$ et $\text{lin}(f) = \text{lin}(g)$. Il existe j tel que $L_j(0) = 0$. Le polynôme L_j est une somme de termes de degré exactement 1. Puisque $\text{lin}(f) \neq 0$, alors nécessairement g/L_j a un coefficient constant $\alpha \in \mathbb{F}$ et tous les termes de degré 1 de g sont le produit de α par un terme de L_j. Donc $\text{lin}(f) = \text{lin}(g) = \alpha L_j$. □

Pour ce deuxième lemme, on n'a pas besoin de l'hypothèse que f est de valuation 1. Cependant, l'hypothèse du lemme ne peut être vérifiée que par un polynôme f de valuation 1.

Lemme 5.21

Soit f un polynôme multilinéaire et L un polynôme linéaire sans coefficient constant contenant un terme $\alpha_i X_i$ avec $\alpha_i \neq 0$. S'il existe un polynôme multilinéaire g tel que $f = \textsc{Mult}_0(L \times g)$, alors

$$f = \textsc{Mult}_0\left(L \times \frac{1}{\alpha_i}\frac{\partial f}{\partial X_i}\right).$$

Démonstration : Supposons que $f = \textsc{Mult}_0(L \times g)$. Cela signifie qu'il existe des polynômes $p_1, \ldots, p_m$ tels que

$$f = L \times g + p_1 X_1^2 + \cdots + p_m X_m^2.$$

Remarquons que $\partial(p_j X_j^2)/\partial X_i = X_j^2 \partial p_j/\partial X_i$ pour tout j. Pour $j = i$ c'est une conséquence du fait que $\partial X_i^2/\partial X_i = 2X_i = 0$. De plus, si L contient le terme $\alpha_i X_i$, $\partial L/\partial X_i = \alpha_i$. On en déduit que

$$\frac{\partial f}{\partial X_i} = \alpha_i g + L\frac{\partial g}{\partial X_i} + \frac{\partial p_1}{\partial X_i}X_1^2 + \cdots + \frac{\partial p_m}{\partial X_i}X_m^2.$$

Si $\alpha_i \neq 0$, on peut multiplier l'égalité précédente par L/α_i et obtenir

$$L \times \frac{1}{\alpha_i}\frac{\partial f}{\partial X_i} = L \times g + \frac{L^2}{\alpha_i}\frac{\partial g}{\partial X_i} + \frac{L}{\alpha_i}\left(\frac{\partial p_1}{\partial X_i}X_1^2 + \cdots + \frac{\partial p_m}{\partial X_i}X_m^2\right).$$

Puisque L^2 est une somme de carrés,

$$\textsc{Mult}_0\left(L \times \frac{1}{\alpha_i}\frac{\partial f}{\partial X_i}\right) = \textsc{Mult}_0(L \times g) = f.$$

□

Nous en venons à la deuxième étape, qui montre qu'on peut toujours se ramener à un polynôme de valuation 1. On traite pour cela deux cas distincts. Dans un premier temps, on s'intéresse aux polynômes *complets*, c'est-à-dire dont chaque coefficient est non nul. Un polynôme multilinéaire à m variables est donc complet s'il a 2^m termes non nuls. Un cas particulier est le polynôme dont tous les coefficients sont égaux à 1. Ce polynôme peut se factoriser sous la forme $\prod_i(1 + X_i)$. Cependant, une telle factorisation n'existe pas en général. À l'inverse, un polynôme dont l'un au moins des coefficients est nul sera dit *incomplet*.

Dans le lemme suivant, soit on produit un nouveau polynôme qui est incomplet, soit éventuellement on produit un nouveau polynôme complet mais avec moins de variables. En tout état de cause, le nombre de monômes diminue.

Lemme 5.22

> Soit f un polynôme multilinéaire dépendant de m variables, et ayant 2^m monômes. Alors il existe un polynôme linéaire L tel que $g = \textsc{Mult}_0(L \times f)$ a au plus $(2^m - 1)$ monômes et est non nul. De plus, f est factorisable si et seulement si g l'est.

Démonstration : Soit X_i une variable quelconque de f, f_i le coefficient du monôme X_i dans f, et f_0 son coefficient constant. Posons $L = f_i X_i + f_0$. Alors le coefficient constant de $L \times f$, et donc de g, est $f_0^2 \neq 0$ donc g est non nul. Le coefficient de X_i dans g est quant à lui $f_0 f_i + f_i f_0 = 0$. Donc g a au plus $(2^m - 1)$ monômes.

Par définition, g est factorisable si f l'est. De plus, $\textsc{Mult}_0(L \times g) = \textsc{Mult}_0(L^2 \times f) = \textsc{Mult}_0(\alpha^2 f) = \alpha^2 f$ puisque f est multilinéaire. Donc si g est factorisable, f l'est également. □

On traite enfin le cas d'un polynôme incomplet de valuation différente de 1. On travaille cette fois-ci avec l'idéal $\mathcal{I}_1$ et non $\mathcal{I}_0$ comme précédemment.

Lemme 5.23

> Soit f un polynôme multilinéaire incomplet. Alors il existe un monôme X^α tel que $g = \textsc{Mult}_1(X^\alpha f)$ a valuation 1. De plus, f est factorisable si et seulement si g l'est.

Démonstration : Si f est déjà de valuation 1, on peut choisir $\alpha = (0, \ldots, 0)$.

Si f est de valuation strictement supérieure à 1, on note X^β un monôme de degré minimal de f, et on suppose que $\beta_i = 1$ pour un indice i. On pose $X^\alpha = X^\beta / X_i$. En particulier, $\textsc{Mult}_1(X^\alpha X^\beta) = X_i$. De plus, $\textsc{Mult}_1(X^\alpha X^\gamma) = 1$ si et seulement si $\alpha = \gamma$. Puisque $\deg(X^\alpha) < \deg(X^\beta)$, le coefficient de X^α dans f est nul. Ce qui signifie que le coefficient constant de g est nul. Donc g est de valuation 1.

Si f est de valuation 0, on définit X^α comme un monôme de degré

minimal dont le coefficient dans f est nul. Un tel monôme existe puisque f est incomplet. Alors $\textsc{Mult}_1(X^\alpha f)$ n'a pas de coefficient constant. Par minimalité de X^α, tout monôme de degré strictement inférieur a un coefficient non nul dans f. C'est en particulier le cas des monômes de la forme X^α/X_i pour tout i tel que $\alpha_i = 1$. Or $\textsc{Mult}_1(X^\alpha(X^\alpha/X_i)) = X_i$, donc la valuation de f est exactement 1.

Pour conclure, on remarque que si f est factorisable, alors g l'est puisqu'un monôme est un produit de facteurs linéaires. Et la réciproque est également vraie puisque $\textsc{Mult}_1(X^\alpha g) = \textsc{Mult}_1((X^\alpha)^2 f) = f$. □

Ces lemmes justifient entre autres le fait qu'il soit inutile de chercher une représentation ou une factorisation d'un polynôme multilinéaire à coefficients dans $\mathbb{F}$ dans une extension de corps de $\mathbb{F}$. Une preuve algorithmique du corollaire suivant se déduit des algorithmes de la partie 5.4.2. Nous donnons ici une preuve directe du résultat.

Pour la preuve, on utilise le fait suivant. Si un polynôme multilinéaire f est représentable, alors pour toute variable X_i, $\partial f/\partial X_i$ l'est également. En effet, d'après le lemme 5.15, f est représentable par une matrice $\mathcal{M}$ ayant au plus une fois la variable X_i sur la diagonale. Si X_i apparaît comme coefficient d'indice (j, j) de $\mathcal{M}$, le déterminant de la matrice obtenue en supprimant les ligne et colonne j de $\mathcal{M}$ est égal à $\partial f/\partial X_i$.

Corollaire 5.24

> Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$ un polynôme multilinéaire et $\mathbb{G}/\mathbb{F}$ une extension de corps. Si f est représentable par une matrice symétrique à coefficients dans $\mathbb{G} \cup \{X_1, \ldots, X_m\}$ alors il l'est par une matrice symétrique à coefficients dans $\mathbb{F} \cup \{X_1, \ldots, X_m\}$.

Démonstration : On prouve le résultat par induction sur le nombre m de variables. Si $m = 1$, le polynôme f est linéaire et le résultat est clair.

Soit f un polynôme à m variables, $m > 1$, représentable par une matrice à coefficients dans $\mathbb{G} \cup \{X_1, \ldots, X_m\}$. On considère dans un premier temps que le corps de base est $\mathbb{G}$, et les polynômes considérés sont vus comme polynômes à coefficients dans $\mathbb{G}$. Le théorème 5.16 implique l'existence de polynômes linéaires L_1, …, L_k tels que $f = \textsc{Mult}_0(L_1 \times \cdots \times L_k)$. D'après le lemme 5.20, il existe j et une constante $\alpha \in \mathbb{G}$ tels que $\alpha L_j = \mathrm{lin}(f)$. Sans perte de généralité, on peut supposer que $j = 1$. De plus, on peut supposer que $\alpha = 1$ en remplaçant si nécessaire L_1 par αL_1 et l'un des L_j, $j > 1$, par L_j/α. Puisque $g = \textsc{Mult}_0(L_2 \times \cdots L_k)$ vérifie $f = \textsc{Mult}_0(\mathrm{lin}(f) \times g)$, alors d'après le lemme 5.21, $f = \textsc{Mult}_0(\mathrm{lin}(f) \times \frac{1}{\alpha_i}\frac{\partial f}{\partial X_i})$ pour un certain i tel que $\alpha_i X_i$ est un terme non nul de $\mathrm{lin}(f)$. De plus, puisque f est représentable, alors c'est également le cas de $\partial f/\partial X_i$.

Maintenant, $\partial f/\partial X_i$ est un polynôme à coefficients dans $\mathbb{F}$ ayant au plus $(m-1)$ variables. Par induction, puisqu'il est représentable par

une matrice à coefficients dans $\mathbb{G} \cup \{X_1, \ldots, X_m\}$, il est représentable par une matrice dont les constantes proviennent de $\mathbb{F}$. Autrement dit, il existe des polynômes linéaires L_1, …, L_k à coefficients dans $\mathbb{F}$ tels que $\partial f/\partial X_i = \textsc{Mult}_0(L_1 \cdots L_k)$. Ainsi, $f = \textsc{Mult}_0(\mathrm{lin}(f)/\alpha_i \times L_1 \cdots L_k)$ et comme $\mathrm{lin}(f)/\alpha_i$ est à coefficients dans $\mathbb{F}$, le résultat s'en déduit. □

5.4.2 *Test de factorisabilité*

Grâce à la partie précédente, on peut en déduire un algorithme polynomial pour décider si un polynôme multilinéaire donné est factorisable, et donc représentable. La représentation des polynômes est la représentation creuse (ou lacunaire [1]). Un polynôme $f = \sum_\alpha c_\alpha X^\alpha$ où $X^\alpha = X_1^{\alpha_1} \cdots X_m^{\alpha_m}$ est représenté par la liste des couples (c_α, α) où $c_\alpha \in \mathbb{F}$ et $\alpha \in \{0,1\}^m$. Afin d'utiliser les lemmes de la partie précédente, il faut savoir calculer les fonctions $\textsc{Mult}_0$ et $\textsc{Mult}_1$. Plus précisément, il suffit de savoir calculer, pour deux polynômes multilinéaires f et g, le polynôme multilinéaire $\textsc{Mult}_\ell(fg)$, avec $\ell \in \{0,1\}$. Si $f = \sum_\alpha c_\alpha X^\alpha$ et $g = \sum_\beta d_\beta X^\beta$, alors $fg = \sum_{\alpha,\beta} c_\alpha d_\beta X^{\alpha+\beta}$. Alors $\textsc{Mult}_0(fg)$ se calcule en ne gardant que les couples (α, β) tels que $\alpha + \beta \in \{0,1\}^m$, et pour $\textsc{Mult}_1(fg)$, il suffit de remplacer $X^{\alpha+\beta}$ par $X^{\alpha \otimes \beta}$ où $\otimes$ représente l'opération « ou exclusif » appliquée composante par composante.

Le premier algorithme présenté est l'algorithme Preparation (algorithme 5) qui correspond aux lemmes 5.22 et 5.23.

Lemme 5.25

Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$ un polynôme multilinéaire. Alors le polynôme $g = \textsc{Preparation}(f)$ est soit linéaire, soit de valuation 1. De plus, f est factorisable si et seulement si g l'est.

L'algorithme est de complexité polynomiale en le nombre m de variables et le nombre de monômes du polynômes f.

Démonstration : La correction est assurée par les lemmes 5.22 et 5.23. Il faut simplement s'assurer de la terminaison, et de la complexité. Il y a un appel récursif lorsque f est complet, c'est-à-dire qu'il a 2^m monômes. D'après le lemme 5.22, $\textsc{Mult}_0(f \times (f_0 X_i + f_i))$ a alors au plus $2^m - 1$ monômes. Soit c'est encore un polynôme complet, auquel cas le nombre de variables a diminué, soit la condition « f est complet » n'est pas vérifiée et il n'y a plus d'appels récursifs. Ceci montre que le nombre d'appels récursifs est borné par le nombre de variables. □

On décrit maintenant l'algorithme IsFactorizable (algorithme 6) correspondant aux lemmes 5.20 et 5.21.

Théorème 5.26

Soit $f \in \mathbb{F}[X_1, \ldots, X_m]$ un polynôme multilinéaire. Alors l'algorithme

1. Puisque les polynômes sont multilinéaires, ces deux représentations sont équivalentes.

Algorithme 5: PREPARATION(f)

Entrées : Un polynôme multilinéaire f
Sorties : Un polynôme multilinéaire g, factorisable ssi f l'est, de valuation 1 ou linéaire

si f *est linéaire* **alors retourner** f
sinon si f *est complet* **alors**
  $X_i \leftarrow$ une des variables de f
  $f_i \leftarrow$ coefficient de X_i dans f
  $f_0 \leftarrow$ coefficient constant de f
  $f \leftarrow \text{MULT}_0(f \times (f_0 X_i + f_i))$
  retourner PREPARATION(f)
sinon si f *est de valuation* > 1 **alors**
  $X^\alpha \leftarrow$ monôme minimal de f, divisé par une de ses variables
  retourner $\text{MULT}_1(X^\alpha f)$
sinon si f *est de valuation* 0 **alors**
  $X^\alpha \leftarrow$ monôme minimal de coefficient nul dans f
  retourner $\text{MULT}_1(X^\alpha f)$
sinon retourner f

Algorithme 6: ISFACTORIZABLE(f)

Entrées : Un polynôme multilinéaire f
Sorties : Le polynôme f est-il factorisable ?

$f \leftarrow$ PREPARATION(f)
si f *est linéaire* **alors retourner Oui**
sinon
  $\alpha_i X_i \leftarrow$ un terme non nul de $\text{lin}(f)$
  $f_0 \leftarrow \frac{\partial f}{\partial X_i}$
  si $f = \text{MULT}_0(\frac{1}{\alpha_i}\text{lin}(f) \times f_0)$ **alors**
    retourner ISFACTORIZABLE(f_0)
  sinon
    retourner Non

IsFactorizable(f) répond **Oui** si et seulement si f est factorisable.

L'algorithme est de complexité polynomiale en le nombre m de variables et le nombre de monômes de f.

Démonstration : De même que la preuve du lemme précédent, la correction de l'algorithme est assurée par les lemmes 5.20 et 5.21. En effet, combiner ces deux lemmes permet d'affirmer que f est factorisable si et seulement si $f = \textsc{Mult}_0(\frac{1}{\alpha_i}\operatorname{lin}(f) \times \frac{\partial f}{\partial X_i})$. Il reste simplement à vérifier la terminaison de l'algorithme. Celle-ci est assurée par le fait que par définition, $f_0 = \partial f/\partial X_i$ contient moins de variables que f. Ainsi, le nombre d'itérations est borné par le nombre de variables. □

5.4.3 *Algorithme de représentation*

L'algorithme IsFactorizable décide simplement l'existence d'une factorisation, et donc d'une représentation, pour un polynôme multilinéaire donné, mais ne fournit pas directement de factorisation ou de représentation. Une raison pour cela est l'utilisation des deux idéaux $\mathcal{I}_0$ et $\mathcal{I}_1$ de manière alternée. Dans la partie 5.3.3, l'équivalence pour les polynômes multilinéaires entre l'existence d'une représentation déterminantielle symétrique et la possibilité de factoriser *modulo* un idéal $\mathcal{I}(\ell^2)$ a été montrée en produisant un algorithme qui étant donné une factorisation d'un polynôme f construit une représentation de f. Les mêmes idées sont réutilisées ici.

Lemme 5.27

Pour $b \in \{0,1\}$, il existe un algorithme $\textsc{Merge}_b$ qui prend en entrée des représentations symétriques $\mathcal{M}_f$ et $\mathcal{M}_g$ de deux polynômes multilinéaires f et g, et dont la sortie est une représentation symétrique de $\textsc{Mult}_b(f \times g)$. Cet algorithme est de complexité polynomiale en les dimensions de $\mathcal{M}_f$ et $\mathcal{M}_g$.

Démonstration : Cet algorithme est principalement basé sur le lemme 5.15. Construisons une matrice par blocs $\mathcal{N}$ dont un bloc est $\mathcal{M}_f$ et l'autre $\mathcal{M}_g$. Alors $\det(\mathcal{N}) = \det(\mathcal{M}_f)\det(\mathcal{M}_g) = fg$. On peut modifier $\mathcal{N}$ grâce au lemme 5.15 pour faire en sorte que chaque variable apparaisse au plus une fois sur la diagonale. De plus, $\det(\pi_b(\mathcal{N})) = \pi_b(fg)$. On peut appliquer l'algorithme Clean à $\pi_b(\mathcal{N})$ ce qui a pour effet de remplacer tous les éléments non diagonaux par leur valeur absolue, et définir $\mathcal{N}' = \rho_b(\textsc{Clean}(\pi_b(\mathcal{N})))$. Alors $\det(\mathcal{N}')$ est un polynôme multilinéaire tel que $\pi_b(\det(\mathcal{N}')) = \pi_b(fg)$. En d'autres termes, $\det(\mathcal{N}') = \textsc{Mult}_b(fg)$.

Pour implanter l'algorithme $\textsc{Merge}_b$, il suffit de construire $\mathcal{N}$ ayant au plus une fois chaque variable sur la diagonale, puis d'en déduire $\mathcal{N}'$ en remplaçant chaque coefficient non diagonal $p \in \mathbb{F}[X_1, \ldots, X_m]$ par $p(b, \ldots, b)$. La complexité de l'algorithme est clairement polynomiale en les dimensions de $\mathcal{M}_f$ et $\mathcal{M}_g$. □

Théorème 5.28

Il existe un algorithme SymDetRepr qui prend en entrée un polynôme multilinéaire $f \in \mathbb{F}[X_1, \ldots, X_m]$ et retourne une représentation déterminantielle symétrique de f s'il en existe une. Sa complexité est polynomiale en m et le nombre de monômes de f.

Démonstration : L'algorithme SymDetRepr consiste en deux étapes. La première est une modification de l'algorithme IsFactorizable pour qu'il retourne une liste de facteurs à la place de **Oui** lorsque f est factorisable. La deuxième est une construction à partir de ces facteurs d'une représentation de f, en utilisant l'algorithme Merge du lemme 5.27. Dans le cas où f n'est pas factorisable, on peut simplement continuer à répondre **Non**. Une autre solution serait de retourner une liste de facteurs ainsi que la valeur g du polynôme qui ne vérifie pas $g = \text{Mult}_0(g/\alpha_i \times \partial g/\partial X_i)$ pour construire ensuite une matrice symétrique $\mathcal{M}$ telle que $f = \det(\mathcal{M}) \times g$. Ceci permet d'identifier le polynôme g qui est l'obstruction à la représentabilité de f.

Nous expliquons maintenant comment construire la liste de facteurs, puis une représentation à partir de cette liste. Dans les algorithmes Preparation et IsFactorizable, pour tester si f est factorisable on l'écrit sous la forme $f = \text{Mult}_b(L \times g)$ où L est soit linéaire soit un monôme et $b \in \{0, 1\}$, puis on teste récursivement si g est factorisable. On modifie ces algorithmes pour retenir les couples (L, b) à chaque fois qu'une telle opération est effectuée. Dans le détail, on ajoute une variable globale $\mathcal{L}$ qui est une liste de couples dont le premier élément est un polynôme et le deuxième est simplement un bit. On indique maintenant comment Preparation et IsFactorizable modifient cette variable.

Dans Preparation, ligne 6, on ajoute le couple $((f_0 X_i + f_i)/f_i^2, 0)$ à $\mathcal{L}$. En effet, un appel récursif est effectué à cette ligne avec le polynôme $g = \text{Mult}_0(f \times (f_0 X_i + f_i))$. Or $\text{Mult}_0(g \times (f_0 X_i + f_i)/f_i^2) = \text{Mult}_0(f \times (f_0^2 X_i^2 + f_i^2)/f_i^2) = f$. De la même façon, on ajoute à $\mathcal{L}$ le couple $(X^\alpha, 1)$ aux lignes 10 et 13. Enfin, à la ligne 7 de IsFactorizable, on ajoute le couple $(\text{lin}(f)/\alpha_i, 0)$ à $\mathcal{L}$.

Au moment où IsFactorizable répond **Oui**, le polynôme f est linéaire. Au lieu de répondre **Oui**, on retourne la liste $\mathcal{L}$ à laquelle on ajoute le couple $(f, 0)$ (le bit 0 est arbitraire et ne sera pas utilisé). On dispose alors d'une liste de couples $(L_1, b_1), \ldots, (L_k, b_k)$. On définit $f_k = L_k$ et pour i de $(k-1)$ à 1, $f_i = \text{Mult}_{b_i}(L_i \times f_{i+1})$. Alors d'après la construction de $\mathcal{L}$, $f = f_1$. On construit une représentation de f de la manière suivante : pour tout i, on construit une représentation $\mathcal{N}_i$ de L_i à l'aide du théorème 5.4. Puis on définit $\mathcal{M}_k = \mathcal{N}_k$ et pour i de $(k-1)$ à 1, on définit $\mathcal{M}_i = \text{Merge}_{b_i}(\mathcal{N}_i, \mathcal{M}_{i+1})$. Ainsi, si $\det(\mathcal{M}_{i+1}) = f_{i+1}$ et $\det(\mathcal{N}_i) = L_i$, le lemme 5.27 assure que $\det(\mathcal{M}_i) = f_i$. Enfin, on pose $\mathcal{M} = \mathcal{M}_1$, et comme $f = f_1$ on obtient bien $\det(\mathcal{M}) = f$.

La polynomialité de l'algorithme découle de celles de Preparation, IsFactorizable et Merge. □

5.5 Représentations déterminantielles alternées

L'objet de cette courte partie est d'étudier un type de représentation proche de celui étudié dans le chapitre précédent et celui-ci mais qui est indépendant de la caractéristique. Les matrices symétriques correspondent aux formes bilinéaires symétriques. On a vu que dans ce contexte, on obtient des résultats tout à fait différents entre la caractéristique 2 et les autres caractéristiques. Il existe une notion proche des formes bilinéaires symétriques, les formes *alternées*, qui se trouve être indépendante de la caractéristique [2]. Si V est un espace vectoriel sur un corps $\mathbb{K}$ de caractéristique quelconque, une forme bilinéaire alternée est une fonction $\varphi : V \times V \to \mathbb{K}$ telle que $\varphi(v, v) = 0$ pour tout $v \in V$. Les matrices associées aux formes bilinéaires alternées sont les matrices antisymétriques, c'est-à-dire égales à l'opposé de leur transposée. On impose, même en caractéristique 2, que la diagonale soit nulle. On peut alors définir une *représentation déterminantielle alternée* d'un polynôme $f \in \mathbb{K}[X_1, \dots, X_m]$ comme étant une matrice antisymétrique $\mathcal{M}$ à coefficients dans $\mathbb{K} \cup \{X_1, \dots, X_m\}$ telle que $\det(\mathcal{M}) = f$.

L'outil de base est une propriété classique des déterminants de matrices antisymétriques. La preuve de cette proposition est similaire à celle de la proposition 5.1.

Proposition 5.29

Soit $\mathcal{M}$ une matrice antisymétrique de dimension $2n$. Alors il existe un polynôme $\mathrm{Pf}(\mathcal{M})$ en les coefficients de $\mathcal{M}$, appelé le *pfaffien* de $\mathcal{M}$, tel que $\det(\mathcal{M}) = \mathrm{Pf}(\mathcal{M})^2$. De plus, le déterminant d'une matrice antisymétrique dont la dimension est impaire est nul.

On en déduit le résultat suivant.

Théorème 5.30

Soit $f \in \mathbb{K}[X_1, \dots, X_m]$ un polynôme. Alors f admet une représentation déterminantielle alternée si et seulement si f est un carré.

Démonstration : Soit f le déterminant d'une matrice antisymétrique $\mathcal{M}$ à coefficients dans $\mathbb{K} \cup \{X_1, \dots, X_m\}$. Alors f est soit nul, soit le carré du pfaffien de $\mathcal{M}$. Ainsi, f est un carré.

Réciproquement, si $f = g^2$ pour un polynôme g, on peut construire une matrice $\mathcal{N}$ (quelconque) telle que $g = \det(\mathcal{N})$ d'après le théorème 3.11. Alors la matrice

$$\mathcal{M} = \begin{pmatrix} 0 & \mathcal{N} \\ -\mathcal{N}^T & 0 \end{pmatrix}$$

2. Ceci nous a été signalé par Mathieu Florence.

où $\mathcal{N}^T$ est la transposée de $\mathcal{N}$ est bien antisymétrique, et $\det(\mathcal{M}) = \det(\mathcal{N})^2 = f$. □

On peut exprimer ce résultat de manière légèrement différente. Tout polynôme peut être représenté par le pfaffien d'une matrice antisymétrique à coefficients dans $\mathbb{K} \cup \{X_1, \ldots, X_m\}$. D'autre part, il pourrait être intéressant d'étudier des représentations déterminantielles *quasi-alternées*, c'est-à-dire dont la diagonale est non nulle. Une partie des résultats prouvés dans ce chapitre s'étendent de manière directe à ces représentations, mais certains utilisent de manière plus essentielle la caractéristique deux du corps.

Troisième partie

Polynômes de type creux

Chapitre 6

Autour de la τ-conjecture réelle

La τ-conjecture de Mike Shub et Steve Smale relie le nombre de racines entières d'un polynôme au nombre d'opérations élémentaires nécessaires pour l'évaluer. Cette conjecture est centrale en complexité algébrique puisqu'elle implique la séparation des classes $\mathsf{P}_{\mathbb{C}}$ et $\mathsf{NP}_{\mathbb{C}}$ qui sont les équivalents dans le modèle BSS des classes P et NP, mais aussi celle des classes VP^0 et VNP^0 dans le modèle de Valiant. Pascal Koiran a proposé une version affaiblie de cette conjecture, appelée *τ-conjecture réelle*, qui permet également de séparer les classes VP^0 et VNP^0. Cette nouvelle conjecture affirme qu'un certain type de polynômes que l'on peut voir comme une généralisation des polynômes creux a un nombre polynomial de racines réelles. Nous obtenons dans ce chapitre une première borne supérieure sur le nombre de racines réelles des polynômes en question. Nous montrons ensuite comment traduire cette borne supérieure en un algorithme de test d'identité polynomiale d'une part, et en une borne inférieure de complexité pour le permanent. Ces résultats démontrent une nouvelle fois l'étroitesse des liens unissant ces deux problèmes.

Ce chapitre est issu de l'article [46].

6.1 La τ-conjecture réelle

On rappelle qu'un circuit sans constante est un circuit dont les entrées sont soit des variables, soit la constante -1, et que pour un polynôme $f \in \mathbb{Z}[X]$, $\tau(f)$ est la taille du plus petit circuit arithmétique sans constante représentant f. Alors la τ-conjecture de Mike Shub et Steve Smale [115] affirme qu'il existe une constante c telle que le nombre $z(f)$ de racines entières de f vérifie

$$z(f) \leq (1 + \tau(f))^c.$$

On peut noter que cette conjecture est fausse si l'on remplace le nombre de racines entières par le nombre de racines réelles du polynôme. En effet, les polynômes de Tchebychev donnent un contre-exemple. Soit $(T_n)_{n\in\mathbb{N}}$ la famille de polynômes définie par $T_n(\cos\theta) = \cos(n\theta)$ pour tout $n \in \mathbb{N}$ et $\theta \in [-1, 1]$. On peut bien sûr étendre leur définition à l'ensemble des réels. Deux propriétés de ces polynômes nous intéressent ici. Premièrement, T_n est de degré n et possède n racines réelles dans l'intervalle $[-1, 1]$. D'autre part, pour tout couple d'entiers (m, n), $T_{mn} = T_m \circ T_n$ où $\circ$ représente la composition. On peut considérer la sous-famille $(T_{2^n})_{n\in\mathbb{N}}$. Soit $\mathcal{C}_2$ un circuit représentant T_2 avec comme entrées la variable X et la constante -1. Pour représenter $T_4(X) = T_2(T_2(X))$, il suffit d'utiliser deux copies $\mathcal{C}_2^1$ et $\mathcal{C}_2^2$ de $\mathcal{C}_2$. La première copie $\mathcal{C}_2^1$ représente T_2. Dans $\mathcal{C}_2^2$, l'entrée X est remplacée par la porte de sortie de $\mathcal{C}_2^1$. Ainsi, on construit un circuit $\mathcal{C}_4$ représentant T_4. De cette manière, on peut construire pour tout n un circuit $\mathcal{C}_{2^n}$ représentant T_{2^n}. La taille de $\mathcal{C}_{2^n}$ est alors n fois la taille de $\mathcal{C}_2$. Ceci prouve que $\tau(T_{2^n}) = \mathcal{O}(n)$. Ainsi, la famille $(T_{2^n})_{n\in\mathbb{N}}$ est une famille de polynômes de τ-complexité linéaire en n, mais ayant un nombre exponentiel de racines réelles. On remarque cependant que cette famille ne constitue par un contre-exemple à la τ-conjecture puisque les polynômes de Tchebychev n'ont aucune racine entière.

La motivation pour cette conjecture est l'étude des classes de complexité $\mathsf{P}_\mathbb{C}$ et $\mathsf{NP}_\mathbb{C}$ dans le modèle BSS. En effet, Shub et Smale ont prouvé que la τ-conjecture implique la séparation de ces deux classes [115]. Plus récemment, Peter Bürgisser a montré que cette conjecture a également des conséquences dans le modèle de Valiant, et plus précisément sur la complexité du permanent. En effet, la validité de la τ-conjecture implique que $\tau(\textsc{Per}_n)$ n'est pas polynomialement bornée en n [15].

La τ-conjecture est certainement un problème très difficile. Ses implications pour la complexité algébrique, dans les modèles BSS et de Valiant, en sont une bonne preuve. Qi Cheng a également observé qu'une généralisation due à Bürgisser de la τ-conjecture implique des résultats puissants en géométrie arithmétique, tels que le théorème de torsion de Merel [27].

Il est tentant d'affaiblir la conjecture tout en gardant ses implications en complexité algébrique. C'est ce qu'a proposé Pascal Koiran en introduisant sa τ-conjecture réelle, qui implique également que $\tau(\textsc{Per}_n)$ n'est pas polynomialement bornée [76]. L'idée est de restreindre la classe des polynômes considérés en ajoutant des contraintes de structure. Les polynômes considérés sont sous la forme de sommes de produits de polynômes creux. Les polynômes de Tchebychev ne s'écrivent pas de manière concise sous cette forme. Ceci a permis à Koiran de d'étendre la conjecture au nombre racines réelles, et non seulement entières. L'espoir est de pouvoir utiliser certains

outils d'analyse qui ne sont pas applicables au cas des racines entières.

Conjecture 6.1 (*τ-conjecture réelle*)

Soit $f \in \mathbb{R}[X]$ un polynôme non nul de la forme

$$f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}(X) \tag{6.1}$$

où chaque f_{ij} est un polynôme creux à t monômes. Alors le nombre de racines réelles de f est polynomial en k, m et t.

Koiran a montré que la τ-conjecture réelle implique de $\tau(\text{Per}_n)$ n'est pas polynomialement borné. En réalité, il suffit de borner le nombre de racines entières des polynômes de la forme (6.1) pour arriver à la même conclusion. Des résultats plus modestes permettent aussi d'obtenir des bornes inférieures intéressantes. En particulier, si le nombre de racines de polynômes de la forme (6.1) est borné par $q(kmt)$ où q vérifie $q(s) = 2^{s^{o(1)}}$, alors le permanent n'est pas représentable par un circuit de taille polynomiale et de profondeur 4 n'utilisant que des constantes de taille polynomiale. L'intérêt récent pour les bornes inférieures relatives aux circuits de profondeur 4 est due à leurs liens avec les bornes inférieures pour les circuits de profondeur quelconque [2, 73].

Dans ce chapitre, on s'intéresse à un cas particulier de la τ-conjecture réelle, lorsque le nombre de f_{ij} distincts est petit (mais qu'ils peuvent être répétés un grand nombre de fois). Ainsi, on prouve que le résultat suivant.

Théorème 6.2

Soit $f \in \mathbb{R}[X]$ un polynôme non nul de la forme

$$f(X) = \sum_{i=1}^{k} \prod_{j=1}^{m} f_j^{\alpha_{ij}}(X) \tag{6.2}$$

où les f_j ont au plus t monômes. Alors f a au plus $\mathcal{O}(t^{m(2^{k-1}-1)})$ racines réelles.

Une version plus précise de cet énoncé est donnée par le théorème 6.17. La borne est polynomiale lorsque le nombre k de termes de la somme et le nombre m de f_j distincts sont constants. En contrepartie, cette borne est indépendante des exposants α_{ij}.

Cette borne supérieure nous permet d'obtenir une borne inférieure de complexité du permanent pour une classe restreinte de circuits. Ces circuits sont de la forme (6.2) mais où X représente un multiplet de variables. En fixant toujours k et m, nous montrons que tout polynôme de cette forme représentant le permanent doit avoir une taille super-polynomiale. C'est une borne inférieure pour une classe restreinte de circuits de profondeur 4. La porte de sortie à profondeur 4 est d'arité bornée k, et les portes situées à

profondeur 2 doivent ne calculer qu'un nombre constant m de polynômes distincts.

Dans un troisième temps, nous donnons un algorithme de test d'identité polynomiale, toujours pour des polynômes de la même forme. Quand k et m sont fixés, il est possible de tester si un polynôme de la forme (6.2) est identiquement nul en temps polynomial en t et $\max_{ij}(\alpha_{ij})$.

Il faut noter que lorsque les α_{ij} sont bornés par une constante, le nombre total de monômes d'un polynôme de la forme (6.2) est polynomial en t et nos résultats deviennent triviaux. Ainsi, nos résultats sont intéressants pour des exposants α_{ij} de grande taille, et peuvent être vus comme des limites à l'apport des puissances en terme d'expressivité.

6.1.1 *Travaux existants*

L'idée d'obtenir des bornes inférieures pour la complexité arithmétique de certains polynômes en prouvant des bornes supérieures sur leur nombre de racines réelles remonte au moins au résultat de 1976 d'Allan Borodin et Stephen Cook [12]. Ce résultat fut amélioré indépendamment par Jean-Jacques Risler [109] et Dima Grigoriev [48] (voir aussi [18, chapitre 12]). Cependant, l'exemple des polynômes de Tchebychev a longtemps laissé pensé que les bornes inférieures obtenues avec de telles méthodes ne pouvaient être que très modestes. Puisque ces polynômes ont un nombre de racines réelles qui est exponentiel en leur complexité, il semblait illusoire d'obtenir des bornes inférieures super-polynomiales en utilisant ces méthodes. Cependant, les travaux récents de Pascal Koiran ont montré qu'en restreignant la classe des circuits étudiés, de telles bornes inférieures exponentielles étaient atteignables [76]. Ces résultats sont reliés au fait que pour des polynômes de petit degré, les circuits arithmétiques de profondeur 4 ont quasiment la même expressivité que les circuits généraux [2, 73].

L'étude des tests d'identité polynomiale a également une longue histoire. Le lemme de Schwartz-Zippel fournit un algorithme probabiliste polynomial pour ce problème [112, 126, 32]. Dès 1980, Joos Heintz et Claus-Peter Schnorr ont remarqué le rapport entre les algorithmes déterministes de test d'identité polynomiale et les bornes inférieures pour les circuits arithmétiques [52]. Ce n'est pourtant que beaucoup plus récemment que cette question a commencé a être étudiée plus en détail, avec l'important résultat de Valentine Kabanets et Russel Impagliazzo [59]. La littérature récente contient de nombreux algorithmes déterministe de test d'identité polynomiale pour différentes restrictions du modèle de circuit arithmétique — on peut par exemple se référer aux deux survols [1, 111]. Un modèle similaire au nôtre a été étudié récemment par Malte Beecken, Johannes Mittmann et Nitin Saxena [8]. De leur théorème 1, on peut déduire qu'il existe un algorithme déterministe de type boîte-noire pour tester la nullité de polynôme de la forme (6.2) si au lieu de borner k et m comme dans notre algorithme, on borne le degré de trans-

cendance r des polynômes f_j. Comme $r \leq m$, leur résultat est de ce point de vue plus général que le nôtre [1]. Cependant, leur algorithme est de complexité polynomiale en le degré des f_j alors que nous savons traiter des polynômes de degré exponentiel en temps polynomial, et le papier ne contient aucune borne inférieure. Un autre résultat récent de Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi et Nitin Saxena explore également les liens entre test d'identité polynomiale et borne inférieure pour le permanent à l'aide de l'étude de la matrice jacobienne des polynômes considérés, étendant nos résultats à des classes plus générales de polynômes [3]. Enfin, un résultat très récent de Pascal Koiran, Natacha Portier et Sébastien Tavenas continue l'étude de la τ-conjecture réelle en obtenant une nouvelle borne, simplement exponentielle en k, en utilisant un autre outil algébrique appelé le wronskien [78]. Cet outil sera présenté et abondamment utilisé au chapitre 7.

6.1.2 *Notre approche*

On peut voir notre preuve de la borne sur le nombre de racines réelles comme une généralisation de celle de la règle des signes de Descartes. Le résultat de Descartes est en réalité un peu plus précis que celui énoncé ici, et tiens compte des signes des coefficients du polynôme. Cette version est suffisante pour notre propos. Nous appellerons *règle de Descartes* cette version simplifiée qui ne fait pas référence au signe des coefficients.

Proposition 6.3

> Un polynôme $f \in \mathbb{R}[X]$ ayant $t \geq 1$ monômes possède au plus $(t-1)$ racines réelles strictement positives.

On peut borner de la même façon le nombre de racines négatives, en considérant $f(-X)$. En ajoutant la possible racine 0 pour f, on en déduit que f possède au plus $(2t-1)$ racines réelles distinctes.

Démonstration : La preuve de la règle de Descartes peut se faire par induction sur t. Si f n'a qu'un monôme, alors il ne peut pas avoir de racine non nulle. Pour $t > 1$, soit $c_\alpha X^\alpha$ le monôme de plus bas degré de f. On peut diviser f par X^α sans changer ses racines strictement positives, et donc supposer que $\alpha = 0$. Dans ce cas, la dérivée f' de f a $(t-1)$ monômes. Par hypothèse d'induction, f' a au plus $(t-2)$ racines strictement positives. D'après le théorème de Rolle, entre deux racines strictement positives de f se trouve une racine de f'. On peut conclure que f a au plus $(t-2)+1 = t-1$ racines strictement positives. □

Pour un polynôme de la forme (6.2), on remplace une somme de t monômes par une somme de k produits de puissances de polynômes creux.

1. Comme le signalent les auteurs de [8], le résultat est déjà non trivial lorsque m est borné.

Cependant, on utilise une stratégie similaire pour la preuve : on divise par l'un des termes puis on dérive le polynôme. On supprime de cette manière un terme à la somme, mais la différence avec la règle de Descartes est que cela résulte également en une augmentation de la complexité des $(k-1)$ termes restants, et de ce fait en une borne plus large.

Pour donner une idée de la preuve, mais en limitant son côté technique, on traite ici le cas particulier $k = 2$.

Théorème 6.4

> Soit $f = \prod_{j=1}^{m} f_j^{\alpha_{1j}} + \prod_{j=1}^{m} f_j^{\alpha_{2j}}$ où les f_j ont au plus t monômes. Alors f a au plus $2mt^m + 4m(t-1)$ racines réelles distinctes.

Démonstration : Soit $\phi = f / \prod_j f_j^{\alpha_{1j}} = 1 + \prod_j f_j^{\alpha_{2j}-\alpha_{1j}}$. Alors

$$\phi' = \prod_{j=1}^{m} f_j^{\alpha_{2j}-\alpha_{1j}-1} \times \sum_{j=1}^{m} (\alpha_{2j} - \alpha_{1j}) f_j' \prod_{\ell \neq j} f_\ell.$$

Chaque f_j ayant au plus t monômes, alors le polynôme

$$\sum_{j=1}^{m} (\alpha_{2j} - \alpha_{1j}) f_j' \prod_{\ell \neq j} f_\ell$$

en a au plus mt^m. D'après la règle de Descartes, il a donc au plus $2mt^m - 1$ racines réelles distinctes. De plus, une racine ou un pôle de la fraction rationnelle $\prod_j f_j^{\alpha_{2j}-\alpha_{1j}-1}$ est une racine de l'un des f_j. Elle en a donc au plus $1 + 2m(t-1)$. Alors ϕ' a au plus $2mt^m + 2m(t-1) - 1$ racines, puisque la racine nulle est comptée pour ses deux facteurs. En appliquant le théorème de Rolle à ϕ, on en déduit que ϕ a au plus $2mt^m + 2m(t-1)$ racines réelles distinctes. Enfin, une racine de f est une racine de ϕ ou de $\prod_j f_j^{\alpha_{1j}}$. Il y en a donc au plus $2mt^m + 2m(t-1) + 2m(t-1)$. □

De la borne du théorème 6.2 est déduite une borne inférieure sur la complexité du permanent en appliquant la méthode décrite par Koiran [76]. Plus précisément, sous l'hypothèse que le permanent peut se représenter de manière concise sous la forme (6.2), on prouve à l'aide d'un résultat de Bürgisser [15] que c'est aussi le cas du polynôme $\prod_{i=1}^{2^n} (X - i)$. On en tire une contradiction avec notre borne sur le nombre de racines réelles.

Notre troisième résultat est un algorithme de test d'identité polynomiale pour les polynômes de la forme (6.2). Une méthode classique pour obtenir un tel algorithme est l'utilisation d'un *ensemble intersectant* [2]. Un ensemble intersectant pour une classe $\mathcal{F}$ de polynômes est un ensemble H de points tel que pour tout polynôme non nul $f \in \mathcal{F}$, il existe $x \in H$ tel que $f(x) \neq 0$. Un ensemble intersectant fournit donc un certificat de non nullité pour chaque

2. *Hitting set*, en anglais.

polynôme non nul de $\mathcal{F}$. Étant donné un ensemble intersectant pour une famille $\mathcal{F}$, il est facile d'en déduire un algorithme de type *boîte noire* pour tester la nullité des polynômes de $\mathcal{F}$ [3]. D'autre part, une borne $z(\mathcal{F})$ sur le nombre de racines réelles de tout polynôme non nul de $\mathcal{F}$ montre que tout ensemble de taille $(z(\mathcal{F})+1)$ est intersectant pour $\mathcal{F}$. Ainsi, lorsque k et m sont fixés, notre borne fournit un ensemble intersectant de taille polynomiale pour les polynômes de la forme (6.2). Malheureusement, l'algorithme de type boîte noire résultant n'est pas de complexité polynomiale. En effet, il faudrait pour qu'il le soit être capable d'évaluer en temps polynomial la valeur d'un polynôme de la forme (6.2) sur un argument donné. La présence de puissances élevées rend cette évaluation trop coûteuse. Pour parvenir à nos fins, nous adoptons une autre stratégie et transformons en quelque sorte la preuve de la borne supérieure en un algorithme. Ceci impose de connaître la structure du polynôme à chaque étape de l'algorithme, qui est donc de type *boîte blanche*.

6.2 Racines réelles des sommes de produits de polynômes creux

6.2.1 *Définitions*

Nous commençons par définir précisément la classe des *sommes de produits de polynômes creux* avec lesquels nous travaillons. Nous introduisons ensuite notre méthode permettant de réduire le nombre de termes dans de telles sommes afin d'obtenir une borne sur leur nombre de racines réelles. On peut noter des ressemblances avec la preuve du lemme 2 de [87].

Dans la prochaine définition, la notation SPS est formée sur les termes anglais *Sum of Products of Sparse polynomials*, signifiant somme de produits de polynômes creux.

Définition 6.5

On note $\mathsf{SPS}(k,m,t,h)$ la classe des polynômes $f \in \mathbb{R}[X]$ définis par

$$f = \sum_{i=1}^{k} g_i \prod_{j=1}^{m} f_j^{\alpha_{ij}}$$

où

- $g_1, \ldots, g_k$ sont des polynômes ayant au plus h monômes ;
- $f_1, \ldots, f_m$ sont des polynômes non nuls ayant au plus t monômes ;
- $\alpha_{11}, \ldots, \alpha_{km}$ sont des entiers naturels.

On définit $P_i = \prod_{j=1}^{m} f_j^{\alpha_{ij}}$ et $T_i = g_i P_i$ pour tout i.

Enfin, on note $\mathsf{SPS}(k,m,t)$ la sous-classe de $\mathsf{SPS}(k,m,t,h)$ dans laquelle tous les g_i sont égaux à la constante 1.

3. La réciproque est d'ailleurs vraie, tout algorithme de type boîte noire pour une classe $\mathcal{F}$ de polynômes fournit un ensemble intersectant pour cette classe.

Les polynômes de la classe $\mathsf{SPS}(k,m,t)$ sont donc exactement ceux qui ont la forme (6.2), et cette classe est incluse dans $\mathsf{SPS}(k,m,t,1)$. L'idée de la preuve est de construire, étant donné $f \in \mathsf{SPS}(k,m,t,h)$, un polynôme $\Delta f \in \mathsf{SPS}(k-1,m,t,\tilde{h})$ pour un certain entier $\tilde{h}$, de telle sorte que le nombre de racines réelles de f soit au plus égal à celui de Δf. On peut penser à l'opérateur Δ est comme étant un opérateur de dérivation adapté aux polynômes de la classe $\mathsf{SPS}(k,m,t,h)$, bien que ça n'en soit pas formellement un. On rappelle que pour tout i, $P_i = \prod_j f_j^{\alpha_{ij}}$.

Définition 6.6

Soit $f = \sum_i g_i \prod_j f_j^{\alpha_{ij}} \in \mathsf{SPS}(k,m,t,h)$. Alors

$$\Delta f = \begin{cases} g_k^2 P_k \left(\prod_{j=1}^{m} f_j\right) \left(\dfrac{f}{g_k P_k}\right)' & \text{si } g_k \neq 0\,; \\ f & \text{sinon.} \end{cases}$$

Lemme 6.7

Soit $f \in \mathsf{SPS}(k,m,t,h)$. Alors $\Delta f \in \mathsf{SPS}(k-1,m,t,\tilde{h})$ pour un certain $\tilde{h}$.

Plus précisément, il existe pour tout i un polynôme δg_i ayant au plus $\tilde{h}$ monômes tel que

$$\Delta f = \sum_{i=1}^{k-1} \delta g_i \prod_{j=1}^{m} f_j^{\alpha_{ij}}.$$

Démonstration : Le résultat est clair avec $\tilde{h} = h$ et $\delta g_i = g_i$ pour tout i si g_k est nul. Supposons donc que g_k est non nul. On rappelle que $T_i = g_i \prod_j f_j^{\alpha_{ij}}$ pour tout i. Soit

$$\phi = \frac{f}{T_k} = 1 + \frac{1}{T_k} \cdot \sum_{i=1}^{k-1} T_i.$$

Alors

$$\phi' = \frac{1}{T_k^2} \cdot \sum_{i=1}^{k-1} (T_k T_i' - T_k' T_i).$$

De plus, $T_i' = g_i' P_i + g_i P_i'$ et

$$P_i' = \sum_{j=1}^{m} \alpha_{ij} f_j' f_j^{\alpha_{ij}-1} \cdot \prod_{\ell \neq j} f_\ell^{\alpha_{i\ell}} = P_i \cdot \sum_{j=1}^{m} \alpha_{ij} \frac{f_j'}{f_j}.$$

On en déduit que

$$\begin{aligned}\phi' &= \frac{1}{T_k^2}\cdot\sum_{i=1}^{k-1}(g_kP_kg_i'P_i + g_kP_kg_iP_i' - g_k'P_kg_iP_i - g_kP_k'g_iP_i)\\ &= \frac{1}{T_k^2}\cdot\sum_{i=1}^{k-1}\Bigg(g_kg_i'P_kP_i + g_kg_iP_kP_i\sum_j\alpha_{ij}f_j'/f_j\\ &\qquad\qquad - g_k'g_iP_kP_i - g_kg_iP_kP_i\sum_j\alpha_{kj}f_j'/f_j\Bigg)\\ &= \frac{1}{g_kT_k}\cdot\sum_{i=1}^{k-1}P_i\Bigg(g_kg_i' - g_k'g_i + g_kg_i\sum_j(\alpha_{ij}-\alpha_{kj})f_j'/f_j\Bigg).\end{aligned}$$

En multipliant ϕ' par $\pi = \prod_j f_j$, on obtient

$$\pi\phi' = \frac{1}{g_kT_k}\cdot\sum_{i=1}^{k-1}P_i\left(\pi\cdot(g_kg_i' - g_k'g_i) + g_kg_i\sum_j(\alpha_{ij}-\alpha_{kj})f_j'\prod_{\ell\neq j}f_\ell\right).$$

Ainsi $\Delta f = g_kT_k\pi\phi'$ est un polynôme de la classe $\mathsf{SPS}(k-1,m,t,\tilde{h})$ pour un certain $\tilde{h}$. Et

$$\delta g_i = \pi\cdot(g_kg_i' - g_k'g_i) + g_kg_i\sum_j(\alpha_{ij}-\alpha_{kj})f_j'\prod_{\ell\neq j}f_\ell$$

pour tout i. □

Ce lemme motive la définition suivante.

Définition 6.8

Soit $f \in \mathsf{SPS}(k,m,t)$. On définit la suite $(\Delta^n f)_{0\leq n<k}$ par $\Delta^0 f = f$ et $\Delta^{n+1}f = \Delta(\Delta^n f)$. De même, on définit pour tout i la suite $(\delta^n g_i)_{0\leq n\leq k-i}$ par $\delta^0 g_i = g_i$ et $\delta^{n+1}g_i = \delta(\delta^n g_i)$. Enfin pour tout n on note h_n le nombre de monômes du polynôme $\delta^n g_i$. De cette manière, on a pour tout n

$$\Delta^n f = \sum_{i=1}^{k-n}\delta^n g_i\prod_{j=1}^{m}f_j^{\alpha_{ij}} \in \mathsf{SPS}(k-n,m,t,h_n).$$

6.2.2 *Une généralisation de la règle de Descartes*

Le but de cette partie est d'étudier les suites $(\Delta^n f)_n$ et $(h_n)_n$ définies précédemment. Premièrement, on montre que le nombre de racines réelles de f est borné par le nombre de racines réelles de Δf à une constante additive près. Deuxièmement, on calcule une borne supérieure sur h_n qui permet finalement d'obtenir une borne sur le nombre de racines réelles

de tout polynôme de la classe $\mathsf{SPS}(k, m, t)$. Cette borne, exprimée par le théorème 6.13, est polynomiale en t.

Soit $z(f)$ le nombre de racines réelles distinctes d'un polynôme, ou d'une fraction rationnelle, f. On souhaite obtenir une borne de $z(f)$ en fonction de $z(\Delta f)$.

Lemme 6.9

Soit $f \in \mathsf{SPS}(1, m, t, h)$. Si f est non nul, alors

$$z(f) \leq 2h + 2m(t-1) - 1.$$

Démonstration : Par définition, $f = g \cdot \prod_j f_j^{\alpha_j}$. Par la règle de Descartes, le nombre de racines réelles non nulles de g est borné par $2(h-1)$. De même, chaque f_j a au plus $2(t-1)$ racines réelles non nulles. Les racines de f étant celles de g et de chaque f_j, le nombre de racines réelles non nulles de f est borné par $2(h-1) + 2m(t-1)$. Pour obtenir le résultat final, il faut rajouter la possible racine 0. □

Lemme 6.10

Soit $f \in \mathsf{SPS}(k, m, t, h)$. Alors

$$z(f) \leq z(\Delta f) + 4h + 4m(t-1) - 1.$$

Démonstration : Par définition, $f = \sum_i g_i \prod_j f_j^{\alpha_{ij}}$. Si $g_k = 0$, alors $\Delta f = f$ et le résultat est clair. Supposons donc que $g_k \neq 0$.

De même que dans la démonstration du lemme 6.7, on note $\phi = f/T_k$. Puisque $g_k \neq 0$, $\Delta f = g_k T_k \pi \phi'$ où $\pi = \prod_j f_j$. Donc $z(\Delta f)$ est une borne supérieure sur le nombre de racines réelles de ϕ'.

Puisque $f = T_k \phi$, $z(f) \leq z(T_k) + z(\phi)$. De plus, il y a toujours une racine de ϕ' ou du dénominateur T_k entre deux racines consécutives de ϕ. On en déduit que $z(\phi) \leq z(\phi') + z(T_k) + 1$. En combinant ces inégalités, on obtient $z(f) \leq z(\phi') + 2z(T_k) + 1 \leq z(\Delta f) + 2z(T_k) + 1$. Comme $T_k \in \mathsf{SPS}(1, m, t, h)$, $z(T_k) \leq 2h + 2m(t-1) - 1$ d'après le lemme 6.9. Ce qui achève la preuve. □

À l'aide des deux précédents lemmes, on peut exprimer une borne sur le nombre de racines réelles de f en fonction des h_n.

Lemme 6.11

Soit $f \in \mathsf{SPS}(k, m, t, 1)$. Alors

$$z(f) \leq 2h_{k-1} + 4\sum_{i=0}^{k-2} h_i + 2m(2k-1)(t-1) - k.$$

Démonstration : D'après le lemme 6.10, la suite $(z(\Delta^n f))_n$ satisfait la récurrence

$$z(\Delta^n f) \leq z(\Delta^{n+1} f) + 4h_n + 4m(t-1) - 1$$

pour $n < k-1$. En particulier, puisque $f = \Delta^0 f$,

$$z(f) \leq z(\Delta^{k-1} f) + 4\sum_{i=0}^{k-2} h_i + (k-1)(4m(t-1)-1).$$

Or $\Delta^{k-1} f \in \mathsf{SPS}(1, m, t, h_{k-1})$ par définition, donc $z(\Delta^{k-1} f) \leq 2h_{k-1} + 2m(t-1) - 1$. D'où le résultat. □

Pour conclure, il nous suffit donc d'estimer la croissance de la suite $(h_n)_n$.

Lemme 6.12

Pour tout n, $h_n \leq ((m+2)t^m)^{2^n-1}$.

Démonstration : Soit $f \in \mathsf{SPS}(k, m, t, h)$. Alors pour tout i

$$\delta g_i = (g_k g_i' - g_k' g_i)\prod_{j=1}^{m} f_j + g_k g_i \sum_{j=1}^{m} (\alpha_{ij} - \alpha_{kj}) f_j' \prod_{\ell \neq j} f_\ell.$$

Ainsi, δg_i est une somme de $(m+2)$ termes, et chaque terme est un produit de $(m+2)$ polynômes ayant au plus t monômes pour m d'entre eux, et h monômes pour les deux restants. Ainsi, $\tilde{h} \leq (m+2)t^m h^2$.

Comme $h_{n+1} = \tilde{h}_n$, $h_{n+1} \leq (m+2)t^m h_n^2$. Avec $h_0 = 1$, on en déduit que $h_n \leq ((m+2)t^m)^{2^n-1}$. □

Ceci nous permet donc d'obtenir notre première borne sur le nombre de racines réelles des polynômes de la classe $\mathsf{SPS}(k, m, t)$. On note qu'elle est légèrement moins bonne que la borne énoncée au théorème 6.2. Celle-ci est obtenue dans la partie suivante.

Théorème 6.13

Soit

$$f = \sum_{i=1}^{k} \prod_{j=1}^{m} f_j^{\alpha_{ij}} \in \mathsf{SPS}(k, m, t),$$

c'est-à-dire que les f_j ont t monômes et les α_{ij} sont des entiers naturels. Alors il existe une constante universelle C telle que

$$z(f) \leq C \times ((m+2)t^m)^{2^{k-1}-1}.$$

Démonstration : Les lemmes 6.11 et 6.12 montrent que

$$z(f) \leq 2((m+2)t^m)^{2^{k-1}-1} + 4\sum_{i=0}^{k-2} ((m+2)t^m)^{2^i-1} + 2m(2k-1)(t-1) - k.$$

On peut borner la somme $\sum_i((m+2)t^m)^{2^i-1}$ par $(k-1)((m+2)t^m)^{2^{k-2}-1}$. Il est alors clair que le terme $((m+2)t^m)^{2^{k-1}-1}$ domine les deux autres termes dans la borne de $z(f)$. □

6.2.3 *Affinement de l'analyse*

Le but de cette partie est d'affiner la borne supérieure sur h_n, le nombre de monômes des polynômes $\delta^n g_i$, qui a été faite précédemment. Cet affinement permet une version améliorée du théorème 6.13, exprimée par le théorème 6.17. L'idée consiste à tenir compte du support des $\delta^n g_i$ pour améliorer la borne sur leur nombre de monômes.

Pour un polynôme f, son support $S(f)$ est l'ensemble des entiers i tels que X^i apparaît avec un coefficient non nul dans f. Pour un ensemble A d'entiers, on note $A-\mathbf{1}$ l'ensemble $\{i-1 : i \in A\}$. Si A et B sont deux ensembles d'entiers, alors $A+B = \{i+j : i \in A, j \in B\}$, et $n \times A = \{i_1+\cdots+i_n : (i_1,\ldots,i_n) \in A^n\}$. La somme de deux ensembles est une opération commutative, et de plus $A+(B-\mathbf{1}) = (A-\mathbf{1})+B$. Soit f et g deux polynômes, alors il est facile de prouver que leurs supports vérifient $S(f') \subseteq S(f)-\mathbf{1}$, $S(f+g) \subseteq S(f) \cup S(g)$ et $S(fg) \subseteq S(f)+S(g)$. Le prochain lemme concerne une dernière propriété, légèrement moins claire que les précédentes.

Lemme 6.14

Soit A un ensemble d'entiers et $n > 0$. Alors

$$|n \times A| \leq \binom{n+|A|}{n} \leq \left(e \times \left(1+\frac{|A|}{n}\right)\right)^n.$$

Démonstration : Les éléments de $n \times A$ sont des sommes de n éléments, non nécessairement distincts, de A. Si $i = i_1+\cdots+i_n$, on peut imposer que $i_1 \leq i_2 \leq \cdots \leq i_n$. Ainsi, $|n \times A|$ est borné par le nombre de suites croissantes de longueur n de A. Le cardinal de l'ensemble de ces suites est $\binom{n+|A|}{n}$.

Enfin, la deuxième borne provient de la borne $\binom{n}{k} \leq (e\frac{n}{k})^k$, valable pour $0 \leq k \leq n$. □

Pour un polynôme $f \in \mathsf{SPS}(k,m,t)$, on note S l'ensemble $\sum_j S(f_j)-\mathbf{1}$.

Lemme 6.15

Soit $f \in \mathsf{SPS}(k,m,t,h)$. Alors

$$\bigcup_{i=1}^{k-1} S(\delta g_i) \subseteq S + 2 \times \bigcup_{i=1}^{k} S(g_i).$$

Démonstration : Notons $S_g = \bigcup_i S(g_i)$ et $S_\delta = \bigcup_i S(\delta g_i)$. On cherche donc à montrer que $S_\delta \subseteq S + 2 \times S_g$.

Pour tout i,

$$\delta g_i = \pi(g_k g_i' - g_k' g_i) + g_k g_i \sum_{j=1}^{m} (\alpha_{ij} - \alpha_{kj}) f_j' \prod_{\ell \neq j} f_\ell,$$

où $\pi = \prod_j f_j$. Le support de $g_k g_i' - g_k' g_i$ est inclus dans $S(g_k) + S(g_i) - \mathbf{1} \subseteq 2 \times S_g - \mathbf{1}$. Donc le support de $\pi(g_k g_i' - g_k' g_i)$ est inclus dans $S(\pi) + 2 \times S_g - \mathbf{1} = S + 2 \times S_g$.

De plus, pour tout j, le support de $f_j' \prod_{\ell \neq j} f_\ell$ est inclus dans S. Ceci prouve que

$$S\left(g_k g_i \sum_{j=1}^{m} (\alpha_{ij} - \alpha_{kj}) f_j' \prod_{\ell \neq j} f_\ell\right) \subseteq 2 \times S_g + \bigcup_{j=1}^{m} S = 2 \times S_g + S.$$

En conclusion, $S(\delta g_i) \subseteq S + 2 \times S_g$, et donc $S_\delta \subseteq S + 2 \times S_g$. □

Lemme 6.16

Soit $f \in \mathsf{SPS}(k, m, t)$. Alors pour $0 \leq n < k$,

$$\bigcup_{i=1}^{k-n} S(\delta^n g_i) \subseteq (2^n - 1) \times S.$$

Démonstration : On prouve ce résultat par induction sur n. Pour $n = 0$, $\delta^0 g_i = 1$ pour tout i, d'où le résultat. Par définition, $\delta^{n+1} g_i = \delta(\delta^n g_i)$. Le lemme 6.15 permet alors de conclure. □

En particulier, puisque h_n est par définition le nombre maximal de monômes de $\delta^n g_i$, on peut en déduire une borne sur h_n en fonction du cardinal de S. Ceci nous permet de prouver une version améliorée du théorème 6.13. Le théorème suivant est une reformulation un peu plus précise du théorème 6.2.

Théorème 6.17

Soit $f \in \mathsf{SPS}(k, m, t)$. Alors il existe une constante universelle C telle que

$$z(f) \leq C \times \left(e\left(1 + \frac{t^m}{2^{k-1} - 1}\right)\right)^{2^{k-1}-1}.$$

Démonstration : De même que dans la preuve du théorème 6.13, le terme prépondérant dans $z(f)$ est le nombre h_{k-1} de monômes de $\delta^{k-1} g_1$. Comme $f \in \mathsf{SPS}(k, m, t)$, les f_j ont t monômes, et $|S| = \left|\sum_j S(f_j)\right| \leq t^m$.

En appliquant la majoration du lemme 6.14 à la borne du lemme 6.16, on obtient

$$h_{k-1} \leq \left(e \times \left(1 + \frac{t^m}{2^{k-1}-1}\right)\right)^{2^{k-1}-1},$$

d'où le résultat. □

La borne du lemme 6.14 peut être atteinte lorsque les éléments de A sont suffisamment distants les uns des autres. En particulier, si $A = \{a_1, \ldots, a_{|A|}\}$ avec $a_{i+1} > na_i$ pour tout i, alors deux sommes distinctes de n entiers de A ne peuvent pas être égales, et on a alors $|n \times A| = \binom{n+|A|}{n}$.

Pour les polynômes, cela signifie qu'à l'inverse, si les degrés des exposants des f_j ne sont pas régulièrement espacés, la borne peut être améliorée. En particulier, si pour tout j, $S(f_j) = \bigcup_{i=1}^{\ell} S_j^i$ où $\max(S_j^i) - \min(S_j^i) \leq c$ pour tout i, alors le terme t^m peut être remplacé par ℓ^m.

6.3 Borne inférieure pour le permanent

Cette partie est dédiée à l'obtention de bornes inférieures pour le calcul du permanent, à partir des bornes sur le nombre de racines réelles des polynômes de la classe $\mathsf{SPS}(k, m, t)$. C'est une traduction concrète de la méthode décrite par Koiran pour obtenir des bornes inférieures à partir de résultats d'analyse réelle [76].

Pour ce faire, on introduit un analogue des polynômes SPS mais avec plusieurs variables. La classe $\mathsf{mSPS}(k, m)$ est une sous-classe des polynômes *faciles à calculer*, plus précisément $\mathsf{mSPS}(k, m) \subseteq \mathsf{VP}^0_{\mathrm{nb}}$ pour tous k et m fixés. En particulier, les éléments de $\mathsf{mSPS}(k, m)$ sont des familles de polynômes. Leur relation à la classe des polynômes $\mathsf{SPS}(k, m, t)$ est donnée par le fait que pour une famille $(f_n)_n \in \mathsf{mSPS}(k, m)$, il existe un polynôme p tel que $f_n(X^{\alpha_1}, \ldots, X^{\alpha_{p(n)}}) \in \mathsf{SPS}(k, m, p(n))$ pour tout n.

Définition 6.18

Une suite $(f_n)_{n\in\mathbb{N}}$ de polynômes appartient à la classe $\mathsf{mSPS}(k, m)$ s'il existe un polynôme p tel que pour tout n,

(1) f_n dépend d'au plus $p(n)$ variables ;

(2) $f_n = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{jn}^{\alpha_{ij}}$;

(3) pour tout (i, j), $\alpha_{ij} \in \mathbb{N}$ et la taille de α_{ij} est au plus $p(n)$;

(4) pour $1 \leq j \leq m$, $\tau(f_{jn}) \leq p(n)$ et f_{jn} a au plus $p(n)$ monômes.

Pour montrer que $\mathsf{mSPS}(k, m) \subseteq \mathsf{VP}^0_{\mathrm{nb}}$, il suffit de montrer que si $(f_n) \in \mathsf{mSPS}(k, m)$, alors chaque f_n a un circuit sans constante de taille polynomiale en n. Étant donné un circuit de taille polynomiale pour f_{jn}, qui existe par

définition, il suffit de $\lceil \log \alpha_{ij} \rceil$ porte de multiplications pour calculer $f_{jn}^{\alpha_{ij}}$ grâce à l'exponentiation rapide. On peut donc représenter tous les $f_{jn}^{\alpha_{ij}}$ pour $1 \leq i \leq k$ et $1 \leq j \leq m$ par un circuit de taille polynomiale en n. Il suffit alors d'un nombre constant de portes additionnelles pour représenter f_n.

Définition 6.19

Le *polynôme de Pochhammer-Wilkinson* d'ordre 2^n est défini par

$$\mathrm{PW}_n = \prod_{i=1}^{2^n} (X - i).$$

Afin d'obtenir une borne inférieure pour le permanent à partir de notre borne sur les racines réelles des polynômes de $\mathsf{SPS}(k, m, t)$, on se base sur sa VNP-complétude [122] et sur un résultat de Bürgisser concernant le polynôme de Pochhammer-Wilkinson [15].

Proposition 6.20

Si le permanent $\textsc{Per}_n$ admet un circuit sans constante de taille polynomiale en n, alors il existe une famille $(G_n) \in \mathsf{VNP}^0$ telle que

$$\mathrm{PW}_n = G_n(X^{2^0}, X^{2^1}, \ldots, X^{2^n}).$$

Les preuves des théorèmes 1.2(2) et 4.1(2) de [15] contiennent une preuve de cette proposition bien qu'elle ne soit pas explicitement énoncée.

Théorème 6.21

La famille $(\textsc{Per}_n)$ n'appartient à $\mathsf{mSPS}(k, m)$ pour aucune valeur de k et m, c'est-à-dire que le permanent $\textsc{Per}_n$ ne peut pas être représenté sous la forme

$$\sum_{i=1}^{k} \prod_{j=1}^{m} f_{jn}^{\alpha_{ij}} \tag{6.3}$$

où la taille des α_{ij}, le nombre de monômes de f_{jn} et la taille du plus petit circuit sans constante le représentant sont tous bornés par une fonction polynomiale $p(n)$.

Démonstration : Raisonnons par l'absurde : supposons que $(\textsc{Per}_n)$ appartienne à la classe $\mathsf{mSPS}(k, m)$ et soit p le polynôme qui en témoigne. En particulier, $\textsc{Per}_n$ admet un circuit sans constante de taille polynomiale. D'après la proposition 6.20, il existe une famille $(G_n) \in \mathsf{VNP}^0$ telle que

$$\mathrm{PW}_n(X) = G_n(X^{2^0}, X^{2^1}, \ldots, X^{2^n}).$$

Puisque le permanent est VNP-complet, il existe un polynôme q tel que

$$G_n(X_0, \ldots, X_n) = \textsc{Per}_{q(n)}(z_1, \ldots, z_{q(n)^2})$$

où $z_i \in \mathbb{Z} \cup \{X_0, \ldots, X_n\}$ pour tout i. De plus, $(\textsc{Per}_n) \in \mathsf{mSPS}(k, m)$ donc $\textsc{Per}_n$ peut s'écrire sous la forme (6.3) avec le nombre de monômes des f_{jn} borné par $p(n)$. En posant $r = p \circ q$, on en déduit que pour tout n, $\mathrm{PW}_n \in \mathsf{SPS}(k, m, r(n))$.

Le théorème 6.17 affirme que tout polynôme de $\mathsf{SPS}(k, m, r(n))$ possède au plus $\mathcal{O}(r(n)^{m2^k})$ racines réelles distinctes. On arrive donc à une contradiction puisque PW_n possède 2^n racines réelles distinctes, et que pour tous k et m fixés, $r(n)^{m2^k} = o(2^n)$. □

On peut relâcher un peu la condition (4) de la définition 6.18 en supposant l'hypothèse de Riemann généralisée. En effet, si on la remplace par la condition

(4′) pour $1 \leq j \leq m$, f_{jn} est de degré au plus $2^{p(n)}$ et possède au plus $p(n)$ monômes

et qu'on autorise donc les polynômes f_{jn} a avoir des coefficients complexes arbitraires, la preuve du théorème 6.21 reste valide. Deux choses sont à vérifier. La première est que sous l'hypothèse $(\textsc{Per}_n) \in \mathsf{mSPS}(k, m)$, on a bien l'existence de circuit de taille polynomiale pour représenter $\textsc{Per}_n$. Ceci est assuré par la nouvelle borne sur le degré des f_{jn}. La deuxième est l'existence de constantes arbitraires comme coefficients des f_{jn}. Cette difficulté est éliminée à l'aide du corollaire 4.2 de [15]. Ce corollaire nécessite l'hypothèse de Riemann généralisée, et donc notre résultat également. Savoir supprimer le recours à l'hypothèse de Riemann généralisée du résultat de Bürgisser comme de notre borne inférieure serait intéressant. D'autant plus que l'hypothèse est utile pour éliminer des constantes arbitraires alors que notre borne sur le nombre de racines réelles s'applique à des polynômes ayant des constantes arbitraires. Pour se passer de l'hypothèse de Riemann généralisée, il faudrait donc une version « avec constantes » de la proposition 6.20.

6.4 Tests d'identité polynomiale

Comme cela a déjà été souligné, les liens entre bornes inférieures de complexité et tests d'identité polynomiale sont forts. Cette partie donne une nouvelle illustration de ce phénomène. Notre borne sur le nombre de racines réelles des polynômes de la classe $\mathsf{SPS}(k, m, t)$ nous a permis dans un premier temps de prouver une borne inférieure de complexité pour le permanent. Dans cette partie, nous montrons comment utiliser la borne sur les racines réelles pour donner un algorithme en temps polynomial pour tester si un polynôme de $\mathsf{SPS}(k, m, t)$ est le polynôme nul.

On rappelle que pour un polynôme $f = \sum_i g_i P_i \in \mathsf{SPS}(k, m, t)$, on définit la suite $(\Delta^n f)$ par $\Delta^n f = \sum_i \delta^n g_i P_i$.

Lemme 6.22

Soit $f \in \mathsf{SPS}(k, m, t)$. Alors pour tout $\ell < k - 1$, $\Delta^\ell f = 0$ si et seulement si $\Delta^{\ell+1} f = 0$ et $\deg(\Delta^\ell f) < \max_{1 \leq i \leq k-\ell} \deg(\delta^\ell g_i P_i)$.

Démonstration : Si $\delta^\ell g_i$ est le polynôme nul pour tout i, alors $\Delta^\ell f = \Delta^{\ell+1} f = 0$. Sinon, on peut supposer que $\delta^\ell g_1 P_1$ est de degré maximum parmi les $\delta^\ell g_i P_i$ en réordonnant les termes si nécessaire. En particulier, $\delta^\ell g_1$ est non nul.

Soit $T_1 = \delta^\ell g_1 P_\ell$, de telle sorte que $\Delta^{\ell+1} f = g_1 T_1 \pi (\Delta^\ell f / T_1)'$, où $\pi = \prod_j f_j$. Si $\Delta^\ell f = 0$, alors $\Delta^{\ell+1} f = 0$ et comme $T_1 \neq 0$, son degré est positif et $\deg(T_1) > \deg(\Delta^\ell f)$.

Réciproquement, supposons que $\Delta^{\ell+1} f = g_1 T_1 \pi (\Delta^\ell f / T_1)' = 0$ et que $\deg(\Delta^\ell f) < \max_i \deg(\delta^\ell g_i P_i)$. Alors $(\Delta^\ell f / T_1)' = 0$, donc il existe $\lambda \in \mathbb{R}$ tel que $\Delta^\ell f = \lambda T_1$. Puisque par hypothèse $\Delta^\ell f$ et T_1 sont de degrés différents, alors $\lambda = 0$ et $\Delta^\ell f = 0$. □

Pour obtenir un algorithme de test d'identité polynomiale, la suite $(\Delta^n f)$ doit être calculée explicitement. L'algorithme obtenu n'est donc pas de type boîte noire puisqu'il doit avoir accès à la structure interne des polynômes manipulés.

Théorème 6.23

Soit k et m deux entiers, et

$$f = \sum_{i=1}^{k} \prod_{j=1}^{m} f_j^{\alpha_{ij}} \in \mathsf{SPS}(k, m, t),$$

c'est-à-dire que les f_j ont t monômes et les α_{ij} sont des entiers naturels. Alors il existe un algorithme pour tester si f est le polynôme nul dont la complexité est polynomiale en la taille de la représentation creuse des f_j et en la valeur des α_{ij}.

Démonstration : Considérons la suite $(\Delta^n f)$. D'après le lemme 6.22, f est le polynôme nul si et seulement si $\Delta^{k-1} f$ est le polynôme nul et pour tout $\ell < k - 1$, le degré de $\Delta^\ell f$ est strictement inférieur à $\max_i \deg(\delta^\ell g_i P_i)$.

Pour tout i et tout ℓ, on peut calculer la représentation creuse du polynôme $\delta^\ell g_i$ en temps polynomial en la taille des f_j puisque k et m sont fixés. Ensuite, on peut calculer explicitement le monôme de plus haut degré de chaque $\delta^\ell g_i P_i$. En effet, soit a_j le coefficient du monôme de plus haut degré de f_j. Alors le monôme de plus haut degré de P_i a comme coefficient $\prod_j a_j^{\alpha_{ij}}$. Il suffit de multiplier de produit par le coefficient c_i^ℓ du monôme de plus haut degré de $\delta^\ell g_i$. Les entiers manipulés ont une taille

polynomiale en les α_{ij} et ces calculs sont donc de complexité polynomiale en les α_{ij} (et non en leur taille). Le calcul du degré de ce monôme se fait facilement, en temps polynomial en la taille des α_{ij}.

Pour comparer le degré de $\Delta^\ell f$ avec $\max_i \deg(\delta^\ell g_i P_i)$, on considère l'ensemble I des indices i qui maximisent le degré de $\delta^\ell g_i P_i$. Il suffit ensuite tester si la somme $\sum_{i\in I} c_i^\ell \prod_j a_j^{\alpha_{ij}}$ s'annule. Le degré de $\Delta^\ell f$ est strictement inférieur à $\max_i \deg(\delta^\ell g_i P_i)$ si et seulement si cette somme s'annule.

Il ne reste plus qu'à tester si $\Delta^{k-1} f = \delta^{k-1} g_1 P_1$ est le polynôme nul. Or c'est le cas si et seulement si $\delta^{k-1} g_1$ est le polynôme nul. Comme ce polynôme a été calculé explicitement, ce test ne pose pas de difficulté. □

Ceci donne un algorithme polynomial en la valeur des α_{ij} alors qu'il serait plus souhaitable qu'il soit polynomial en leur taille en binaire.

Théorème 6.24

Supposons que l'on dispose d'un oracle pour tester si une somme de la forme

$$\sum_{i\in I} c_i \prod_{j=1}^{m} a_j^{\alpha_{ij}} \tag{6.4}$$

s'annule, où les c_i et les a_j sont des entiers relatifs et les α_{ij} des entiers naturels. Alors on peut tester si un polynôme $f \in \mathsf{SPS}(k, m, t)$ est le polynôme nul en temps polynomial en la taille de la représentation creuse des f_j et en la taille des α_{ij}.

Démonstration : Dans la preuve du théorème 6.23, la seule opération qui est de complexité polynomiale en les α_{ij} plutôt qu'en leur taille est le calcul d'une somme de la forme (6.4). En effet, calculer le degré du monôme de plus haut degré de chaque $\delta^\ell g_i P_i$ se fait en temps polynomial en la taille des α_{ij}. Ceci permet alors de déterminer l'ensemble I avant d'évaluer le coefficient dominant des $\delta^\ell g_i P_i$. Il ne reste plus qu'à tester la nullité de la somme, ce qui est effectué par l'oracle sans avoir besoin de calculer le coefficient dominant des $\delta^\ell g_i P_i$. □

Comme cela a été remarqué dans la preuve du théorème 6.23, calculer explicitement une somme de la forme (6.4) n'est pas possible en temps polynomial lorsque les α_{ij} sont donnés en binaire puisque les calculs intermédiaires (et éventuellement le résultat) font intervenir des entiers de taille exponentielle en la taille des α_{ij}. On pourrait effectuer un test à 0 en évaluant la somme *modulo* des nombres premiers pris aléatoirement. Cependant, cela ne convient pas à notre problème puisqu'on cherche à obtenir un test d'identité polynomiale déterministe. Il est intéressant de noter que l'oracle du théorème 6.24 résout en réalité un test d'identité polynomiale pour des polynômes de la classe $\mathsf{SPS}(k, m, t)$ où les f_j sont des polynômes constants (et donc $t = 1$). Dans le cas des circuits arithmétiques généraux, il est également

connu que le test d'identité polynomiale peut se réduire au cas des circuits dans lesquels aucune variable n'apparaît [4, Proposition 2.2].

Le test d'identité polynomiale du théorème 6.23 peut être étendu facilement au cas de polynômes à plusieurs variables. Soit $(f_n) \in \mathsf{mSPS}(k, m)$, et soit p le polynôme qui en témoigne. On impose de plus que les α_{ij} soient bornés par $p(n)$ puisque l'algorithme du théorème 6.23 est polynomial en les α_{ij}. Notons d_n le degré de f_n pour tout n. Alors on peut appliquer à f_n la substitution classique, souvent attribuée à Kronecker, $X_i \mapsto X^{(d+1)^i}$. Cette substitution a la propriété que f_n est le polynôme nul si et seulement si $\varphi_n(X) = f_n(X^{(d+1)^0}, \ldots, X^{(d+1)^{p(n)}})$ est le polynôme nul. De plus, $\varphi_n \in \mathsf{SPS}(k, m, p(n))$. On peut donc appliquer l'algorithme de test d'identité polynomiale à φ_n pour tester la nullité de f_n. Pour vérifier que l'algorithme ainsi obtenu est bien de complexité polynomiale, il suffit de s'assurer que la représentation de φ_n est bien de taille polynomiale en la taille de f_n. Ceci revient à borner le degré d_n de f_n. Puisque pour tout j, f_{jn} a un circuit sans constante de taille polynomiale $p(n)$, son degré est au plus $2^{p(n)}$. De plus, $\alpha_{ij} \leq p(n)$ pour tous i et j, donc $d_n \leq 2^{mp(n)^2}$. Ainsi, le degré de f_n est simplement exponentiel, donc φ_n s'écrit comme une somme de produits de puissances de polynômes creux dont le degré est simplement exponentiel. En d'autres termes, φ_n admet une représentation de taille polynomiale en celle de f_n.

6.5 Conclusion

Nous avons montré la validité de la τ-conjecture réelle dans un cas particulier, ainsi que la faisabilité pratique de la méthode de Pascal Koiran pour obtenir des bornes inférieures pour la complexité du permanent à partir de bornes sur le nombre de racines réelles des sommes de produits de polynômes creux. Nous avons également donné une nouvelle illustration des liens forts qui unissent la recherche de bornes inférieures en complexité et d'algorithmes déterministes de test d'identité polynomiale.

Comme nous l'avons mentionné précédemment, notre borne a été améliorée par Pascal Koiran, Natacha Portier et Sébastien Tavenas qui ont pu montrer qu'un polynôme de la classe $\mathsf{SPS}(k, m, t)$ possède au plus $2^{\mathcal{O}(k^2 m \log t)}$ racines réelles distinctes [78]. Cette amélioration reste malheureusement assez éloignée de la borne conjecturée. D'autres cas particuliers simples de la conjecture restent ouverts. Dans le cas général, une somme de produits de polynômes creux peut être développée en une somme d'au plus kt^m monômes. D'après la règle de Descartes, un tel polynôme a donc au plus de $2kt^m - 1$ racines. Comme cela est signalé dans [76], même le cas $k = 2$ est ouvert : dans ce cas, le nombre de racines réelles distinctes est-il polynomialement borné ? On peut même simplifier la question et se demander si un polynôme de la forme $f_1 \cdots f_m + 1$ a un nombre de racines réelles polynomial en t et m.

La version la plus basique que l'on puisse imaginer concerne un polynôme de la forme $fg + 1$. Si f et g ont tous deux moins de t monômes, le nombre de racines réelles de $fg + 1$ est-il linéaire ou quadratique en t ?

Comme expliqué précédemment, l'objectif est de borner le nombre de racines entières de polynômes de la forme (6.2). L'idée de la τ-conjecture réelle est que borner le nombre de racines réelles est sans doute plus simple que le nombre de racines entières, et qu'on peut ensuite utiliser le fait que $\mathbb{Z} \subseteq \mathbb{R}$. On peut aussi choisir de plonger $\mathbb{Z}$ dans un autre corps, comme le corps des entiers p-adiques $\mathbb{Q}_p$ pour un certain nombre premier p. Alors borner le nombre de racines des polynômes de la forme (6.2) dans $\mathbb{Q}_p$ est une autre stratégie. Ceci a amené Kaitlyn Phillipson et J. Maurice Rojas à proposer une version *adélique* de la τ-conjecture [105]. L'idée est qu'il suffit pour borner le nombre de racines entières d'un polynôme f de borner son nombre de racine dans un corps $\mathbb{K}$ tel que $\mathbb{Z} \subseteq \mathbb{K}$. Ainsi, la τ-conjecture adélique affirme qu'étant donné un polynôme f de la forme (6.2), il existe un corps $\mathbb{K} \in \{\mathbb{R}, \mathbb{Q}_2, \mathbb{Q}_3, \mathbb{Q}_5, \dots\}$ tel que f a au plus un nombre polynomial de racines distinctes sur le corps $\mathbb{K}$. La τ-conjecture réelle implique la τ-conjecture adélique puisqu'elle dit que le résultat est vrai pour $\mathbb{K} = \mathbb{R}$. Bien qu'étant moins forte que la τ-conjecture réelle, la τ-conjecture adélique a les mêmes implications. Cette nouvelle conjecture ouvre la porte à l'utilisation de nouvelles techniques, en particulier issues du monde p-adique et de la géométrie tropicale.

Chapitre 7

Factorisation des polynômes lacunaires à deux variables

La factorisation des polynômes est une « *success story* » du calcul formel d'après Erich L. Kaltofen [60]. En effet, aussi bien au niveau théorique que pratique, des algorithmes très efficaces existent pour factoriser des polynômes donnés sous forme dense, ou creuse pour les polynômes à plusieurs variables. Lorsque les polynômes sont représentés sous forme lacunaire, l'histoire est beaucoup plus courte. En réalité, on peut résumer cette histoire à quatre articles publiés de 1999 à 2006.

Felipe Cucker, Pascal Koiran et Steve Smale ont montré comment obtenir en temps polynomial les racines entières d'un polynôme univarié à coefficients entiers donné sous forme creuse [30]. Ces racines entières correspondent aux facteurs linéaires dans $\mathbb{Z}$. Assez rapidement, Hendrik W. Lenstra a généralisé leur résultat en montrant que pour tout d fixé, on peut trouver tous les facteurs de degré au plus d d'un polynôme univarié à coefficients dans un corps de nombre quelconque en temps polynomial en la taille de sa représentation creuse [84]. Plus récemment, Erich L. Kaltofen et Pascal Koiran se sont intéressés au cas des polynômes à plusieurs variables. Dans un premier temps, ils ont montré comment obtenir les facteurs linéaires de polynômes à deux variables et à coefficients rationnels en temps polynomial en la représentation lacunaire du polynôme [63]. Ils ont ensuite étendu ce résultat dans trois directions : ils ont en effet obtenu un algorithme polynomial pour trouver les facteurs de degré au plus d d'un polynôme lacunaire à n variables à coefficients dans un corps de nombre quelconque [62].

De cette manière, la question de la factorisation de polynômes lacunaires est également une *success story*, et on pourrait considérer que le problème a été entièrement résolu par le dernier article de Kaltofen et Koiran. Il faut noter en particulier que les algorithmes proposés sont exponentiel en d mais que c'est inévitable puisque le nombre de facteurs de degré au plus d d'un

polynôme lacunaire peut être exponentiel en d [85]. Cependant, plusieurs questions demeurent. Les algorithmes cités précédemment ne fonctionnent que sur des corps de nombres car ils sont basés sur la notion de hauteur algébrique définie sur ces corps. L'utilisation qui est faite de cette notion est d'ailleurs relativement sophistiquée et les algorithmes obtenus reposent sur des résultats tout à fait non triviaux de théorie des nombres. Ceci a poussé Kaltofen et Koiran à poser la question de l'existence de « preuves plus élémentaires » de leurs résultats [63]. Ces preuves plus élémentaires peuvent également résulter en des algorithmes plus élémentaires, qui seraient par exemple plus faciles à implanter. En particulier, l'algorithme le plus général obtenu par Kaltofen et Koiran utilise des constantes issues de la théorie des nombres qui ne sont pas explicitées dans l'article. Afin d'implanter cet algorithme, il faudrait donc commencer par estimer ces constantes, ce qui est une tâche assez délicate. D'autre part, ces constantes pourraient s'avérer être très grandes, ce qui conduirait à un algorithme purement théorique mais inutilisable en pratique. Enfin, l'utilisation de la hauteur algébrique empêche toute utilisation des algorithmes pour des polynômes à coefficients dans d'autres corps, par exemple en caractéristique positive.

Tous les algorithmes cités précédemment ont un autre point commun, au delà de leur utilisation de la notion de hauteur algébrique. Ils sont tous basés sur un théorème de lacune [1]. Ces théorèmes prennent la forme générique suivante. Soit

$$f = \sum_{j=1}^{k} c_j X_1^{\alpha_{j,1}} \cdots X_n^{\alpha_{j,n}}$$

où $\alpha_j = (\alpha_{j,1}, \ldots, \alpha_{j,n}) \in \mathbb{N}^n$ pour tout j et $\alpha_{1,n} \leq \alpha_{2,n} \leq \cdots \leq \alpha_{k,n}$. Supposons qu'il existe k_0 tel que $\alpha_{k_0+1,n} - \alpha_{k_0,n} > \mathrm{gap}(f)$ où $\mathrm{gap}(f)$ est une fonction de f. Alors tout facteur de f est un facteur des deux polynômes

$$\sum_{j=1}^{k_0} c_{\alpha_j} X_1^{\alpha_{j,1}} \cdots X_n^{\alpha_{j,n}} \quad \text{et} \quad \sum_{j=k_0+1}^{k} c_{\alpha_j} X_1^{\alpha_{j,1}} \cdots X_n^{\alpha_{j,n}}.$$

La fonction $\mathrm{gap}(f)$ dépend de la hauteur algébrique du polynôme dans chacun des résultats cités.

Le but de ce chapitre est d'étudier la possibilité d'algorithmes de factorisation des polynômes lacunaires plus élémentaires que ceux actuellement connus. Nous nous concentrons sur l'obtention des facteurs linéaires de polynômes à deux variables, comme dans le premier article de Kaltofen et Koiran [63]. Notre algorithme est en fait le même que celui de Kaltofen et Koiran mais nous prouvons un nouveau théorème de lacune. Celui-ci ne dépend pas des coefficients du polynôme, mais uniquement de la liste des exposants. Il est donc valable quel que soit le corps de caractéristique 0

1. Ceci est à nuancer pour l'article de Cucker, Koiran et Smale qui fournit deux preuves du résultat dont l'une seulement utilise un théorème de lacune [30].

considéré. Ce nouveau théorème permet de donner un algorithme de test d'identité polynomial pour des polynômes de la forme $\sum_j a_j X^{\alpha_j}(1+X)^{\beta_j}$, ainsi qu'un algorithme pour trouver les facteurs linéaires de polynômes lacunaires à deux variables. Comme les différents algorithmes de Kaltofen et Koiran, nos algorithmes reposent sur l'algorithme de Lenstra [84] pour traiter certains cas particuliers qui se ramènent au cas univarié. De ce fait, nos algorithmes ne traitent que des polynômes à coefficients dans un corps de nombres, même si notre théorème de lacune est plus général. L'intérêt de nos algorithmes en comparaison de ceux de Kaltofen et Koiran, outre leur caractère plus élémentaire, est une dépendance moins importante dans la taille des coefficients, et donc une plus grande efficacité pour des polynômes dont les coefficients sont de grande taille. Nous pouvons généraliser notre théorème de lacune, et un cas particulier de cette généralisation nous permet d'étendre notre algorithme à l'obtention de facteurs multilinéaires plutôt que linéaires. Enfin, nous prouvons également une version de notre théorème de lacune pour des corps de caractéristique positive. Nous pouvons en déduire un algorithme polynomial pour détecter les facteurs de la forme $(uX+vY+w)$ où $uvw \neq 0$ d'un polynôme lacunaire à deux variables. Il est intéressant de noter que lorsque u, v ou w est nul, la détection de tels facteurs est un problème NP-difficile [72, 9, 66].

Ce chapitre est issu de l'article [26].

7.1 Valuation et théorème de lacune

Dans cette partie, on considère un corps quelconque de caractéristique 0, noté $\mathbb{K}$. On appelle *valuation* d'un polynôme $f \in \mathbb{K}[X]$, et on note $\mathrm{val}(f)$, l'entier maximal v tel que X^v divise f. Cette partie est essentiellement dédiée à la preuve du théorème suivant.

Théorème 7.1

Soit $f = \sum_{j=1}^{k} a_j X^{\alpha_j}(1+X)^{\beta_j} \in \mathbb{K}[X]$, avec $\alpha_1 \leq \cdots \leq \alpha_k$. Si f n'est pas le polynôme nul, alors sa valuation est au plus $\max_j \left(a_j + \binom{k+1-j}{2}\right)$.

Il est clair que la valuation d'un polynôme f tel que défini dans le théorème est au moins α_1. De plus, si $\alpha_2 > \alpha_1$, il y a un unique monôme de degré α_1. Aucune annulation ne peut donc avoir lieu et la valuation est exactement α_1. La borne telle qu'énoncée est déduite de la borne $\alpha_1 + \binom{k}{2}$, valable si la famille $(X^{\alpha_j}(1+X)^{\beta_j})_{1\leq j\leq k}$ est une famille libre. Cette borne signifie que seuls les $\binom{k}{2}$ monômes de plus petit degré peuvent s'annuler. Si tous les α_j sont égaux à α_1, un résultat de Hajós donne la borne $\alpha_1 + (k-1)$ [50, 100]. Cette borne est exacte comme le montre l'exemple du polynôme $X^{k-1} = \sum_j \binom{k}{j}(-1)^{k-j}(1+X)^j$. Nous montrons que la borne de Hajós n'est plus valable lorsque les α_j ne sont pas constants. En effet, il existe

un polynôme pour lequel la famille $(X^{\alpha_j}(1+X)^{\beta_j})_{1\leq j\leq k}$ est une famille libre, mais dont la valuation est $\alpha_1 + (2k-3)$ (théorème 7.8). Cependant, notre borne supérieure est quadratique en k et l'exemple de plus grande valuation que l'on sache construire est un polynôme de valuation linéaire en k. La borne exacte reste inconnue, et on ne sait pas si elle est linéaire ou quadratique.

Dans une première sous-partie, nous prouvons le théorème 7.1, et en déduisons dans une seconde sous-partie un théorème de lacune.

7.1.1 Preuve du théorème 7.1

La preuve du théorème 7.1 est basée sur le *wronskien* d'une famille de polynôme. Cet objet est un outil classique dans l'étude des équations différentielles, mais il peut également être utile pour donner des bornes sur la valuation de sommes de racines carrées de polynômes [71] ou sur le nombre de racines réelles de polynômes de la classe $\mathsf{SPS}(k,m,t)$ vue au chapitre 6 [78].

Définition 7.2

Soit $f_1,\dots,f_k \in \mathbb{K}[X]$. Le *wronskien* de la famille $(f_1,\dots,f_k)$ est le déterminant de la *matrice wronskienne*

$$\mathrm{wk}(f_1,\dots,f_k) = \det \begin{bmatrix} f_1 & f_2 & \cdots & f_k \\ f_1' & f_2' & \cdots & f_k' \\ \vdots & \vdots & & \vdots \\ f_1^{(k-1)} & f_2^{(k-1)} & \cdots & f_k^{(k-1)} \end{bmatrix}.$$

La propriété essentielle du wronskien est sa relation avec l'indépendance linéaire de la famille $(f_1,\dots,f_k)$. La preuve donnée ici est essentiellement due à Alin Bostan et Philippe Dumas [13].

Proposition 7.3

Soit $\mathbb{K}$ un corps de caractéristique 0 et $f_1,\dots,f_k \in \mathbb{K}[X]$. Le wronskien de f_1, …, f_k est non nul si et seulement si la famille $(f_1,\dots,f_k)$ est libre.

Démonstration : Supposons dans un premier temps que les f_j sont des monômes, c'est-à-dire que $f_j = a_j X^{\alpha_j}$ pour tout j. La famille des f_j est libre si et seulement si les α_j sont tous distincts.

Pour $\ell < k$, $f^{(\ell)} = a_j(\alpha_j)_\ell X^{\alpha_j-\ell}$ où $(m)_n = m(m-1)\cdots(m-n+1)$ est la *factorielle descendante*. Dans la matrice wronskienne, on peut alors factoriser $a_j X^{\alpha_j-k+1}$ dans la colonne j, puis $X^{k-1-\ell}$ dans la ligne ℓ, pour

$0 \leq \ell < k$. En d'autres termes,

$$\mathrm{wk}(f_1, \ldots, f_k) = X^{-\binom{k}{2}} \prod_{j=1}^{k} a_j X^{\alpha_j} \det \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_k \\ \vdots & \vdots & & \vdots \\ (\alpha_1)_{k-1} & (\alpha_2)_{k-1} & \ldots & (\alpha_k)_{k-1} \end{bmatrix}.$$

Le wronskien est donc nul si et seulement si le déterminant de la matrice ci-dessus est nul. Comme le polynôme $(\alpha)_\ell$ est un polynôme unitaire de degré ℓ en α, la matrice peut être transformée par des opérations élémentaires sur les lignes en une matrice de Vandermonde. Son déterminant est donc nul si et seulement si les α_j ne sont pas distincts deux à deux. On a donc bien l'équivalence recherchée quand les f_j sont des monômes.

Considérons maintenant des f_j quelconques. Notons C_1, ..., C_k les colonnes de la matrice wronskienne de $(f_1, \ldots, f_k)$. La matrice obtenue en effectuant l'opération $C_{j_1} \leftarrow C_{j_1} + \lambda C_{j_2}$, c'est-à-dire en ajoutant la colonne C_{j_2} multipliée par λ à la colonne C_{j_1}, est la matrice wronskienne de la famille $(\tilde{f}_j)$ où $\tilde{f}_{j_1} = f_{j_1} + \lambda f_{j_2}$ et $\tilde{f}_j = f_j$ pour $j \neq j_1$. Les familles (f_j) et $(\tilde{f}_j)$ sont simultanément libres ou liées, et leurs wronskiens sont égaux. Si la famille (f_j) est libre, on peut construire à l'aide de telles combinaisons linéaires une famille $(\tilde{f}_j)$ également libre telle que les $\tilde{f}_j$ ont des valuations toutes distinctes. Les wronskiens de ces deux familles sont égaux. Le monôme de plus bas degré de $\mathrm{wk}(\tilde{f}_1, \ldots, \tilde{f}_k)$ est le déterminant de la matrice wronskienne des monômes de plus bas degré de chaque $\tilde{f}_j$. Par le résultat précédent sur les monômes, ce wronskien est non nul. En particulier, le monôme de plus bas degré de $\mathrm{wk}(f_1, \ldots, f_k)$ étant non nul, le polynôme est non nul.

Ceci prouve que si la famille $(f_1, \ldots, f_k)$ est libre, alors son wronskien est non nul. La réciproque est claire. □

La preuve de la proposition précédente fait apparaître la valuation des polynômes f_j. On s'intéresse maintenant à établir un rapport plus précis entre la valuation des polynômes f_j et de $\mathrm{wk}(f_1, \ldots, f_k)$.

Lemme 7.4

Soit $f_1, \ldots, f_k \in \mathbb{K}[X]$. Alors

$$\mathrm{val}(\mathrm{wk}(f_1, \ldots, f_k)) \geq \sum_{j=1}^{k} \mathrm{val}(f_j) - \binom{k}{2}.$$

Démonstration : En utilisant la formule de Leibniz pour le déterminant,

$$\mathrm{wk}(f_1, \ldots, f_k) = \sum_{\sigma \in \mathfrak{S}_k} \epsilon(\sigma) \prod_{j=1}^{k} f_j^{(\sigma(j)-1)},$$

car les ordres de dérivations vont de 0 à $(k-1)$ et non de 1 à k. Or $\mathrm{val}(f_j^{(\ell)}) \geq \mathrm{val}(f_j) - \ell$, donc la valuation de chaque terme de la somme est au moins $\sum_j \mathrm{val}(f_j) - \sum_\ell \ell = \sum_j \mathrm{val}(f_j) - \binom{k}{2}$. Ce qui prouve le lemme. □

On donne maintenant une borne supérieure sur la valuation du wronskien, non plus pour des polynômes quelconques mais pour ceux que l'on souhaite étudier.

Lemme 7.5

Pour $1 \leq j \leq k$, soit $f_j = X^{\alpha_j}(1+X)^{\beta_j}$ tel que $\alpha_j, \beta_j \geq k-1$. Si la famille $(f_1, \ldots, f_k)$ est libre,

$$\mathrm{val}(\mathrm{wk}(f_1, \ldots, f_k)) \leq \sum_{j=1}^{k} \alpha_j.$$

Démonstration : D'après la règle de Leibniz, pour tous j et ℓ

$$f_j^{(\ell)}(X) = \sum_{t=0}^{\ell} \binom{\ell}{t} (\alpha_j)_t (\beta_j)_{\ell-t} X^{\alpha_j - t}(1+X)^{\beta_j - \ell + t}.$$

Dans l'expression précédente, on peut factoriser $X^{\alpha_j-k+1}(1+X)^{\beta_j-k+1}$. Ainsi, ce facteur est commun à tous les coefficients de la colonne j de la matrice wronskienne des f_j. De même, les coefficients de la ligne ℓ ont le facteur $X^{k-1-\ell}(1+X)^{k-1-\ell}$ en commun. On peut donc écrire

$$\mathrm{wk}(f_1, \ldots, f_k) = X^{\sum_j \alpha_j - \binom{k}{2}}(1+X)^{\sum_j \beta_j - \binom{k}{2}} \det(\mathcal{M}) \tag{7.1}$$

où la matrice $\mathcal{M}$ est définie par

$$\mathcal{M}_{j,\ell} = \sum_{t=0}^{\ell} \binom{\ell}{t} (\alpha_j)_t (\beta_j)_{\ell-t} X^{\ell-t}(1+X)^t$$

pour $1 \leq j \leq k$ et $0 \leq \ell < k$. Pour tout j, $\deg(\mathcal{M}_{j,\ell}) = \ell$ donc $\det(\mathcal{M})$ est un polynôme de degré au plus $\binom{k}{2}$. De plus, la famille (f_j) étant libre, le wronskien est non nul et $\det(\mathcal{M}) \neq 0$. Sa valuation ne peut dépasser son degré, donc $\mathrm{val}(\det(\mathcal{M})) \leq \binom{k}{2}$.

D'après la factorisation (7.1), la valuation du wronskien est bornée par $\sum_j \alpha_j - \binom{k}{2} + \binom{k}{2} = \sum_j \alpha_j$. □

On peut combiner les lemmes 7.4 et 7.5 pour obtenir le résultat souhaité.

Démonstration du théorème 7.1 : Soit $f = \sum_j a_j f_j$ où $f_j = X^{\alpha_j}(1+X)^{\beta_j}$ pour $1 \leq j \leq k$. On suppose dans un premier temps que $\alpha_j, \beta_j \geq k-1$ pour tout j et que la famille $(f_1, \ldots, f_k)$ est libre. Par définition, $\mathrm{val}(f_j) = \alpha_j$ pour tout j.

Soit W le wronskien de la famille $(f_1, \dots, f_k)$. En effectuant une combinaison linéaire des colonnes de la matrice wronskienne de la famille (f_j), on peut remplacer f_1 dans la première colonne par f. Le déterminant obtenu est égal à $a_1 W$, et c'est le wronskien de $(f, f_2, \dots, f_k)$. D'après le lemme 7.4,

$$\operatorname{val}(a_1 W) = \operatorname{val}(f) + \sum_{j=2}^{k} \alpha_j - \binom{k}{2}.$$

Puisque $\operatorname{val}(a_1 W) = \operatorname{val}(W)$, on peut combiner cette inégalité et le lemme 7.5 pour obtenir

$$\operatorname{val}(f) \leq \alpha_1 + \binom{k}{2}. \tag{7.2}$$

Il s'agit maintenant de supprimer les hypothèses effectuées. Supposons donc la famille $(f_1, \dots, f_k)$ soit liée. On peut en extraire une base $(f_{j_1}, \dots, f_{j_d})$. Le polynôme f peut se récrire dans cette base sous la forme $f = \sum_\ell b_\ell f_{j_\ell}$. On peut alors appliquer l'équation (7.2) aux f_{j_ℓ} pour obtenir $\operatorname{val}(f) \leq \alpha_{j_1} + \binom{d}{2}$. Puisque $j_d \leq k$, $j_1 + d - 1 \leq k$ et donc $\operatorname{val}(f) \leq \alpha_{j_1} + \binom{k+1-j_1}{2}$. Ne connaissant pas la valeur de j_1, la borne obtenue est

$$\operatorname{val}(f) \leq \max_{1 \leq j \leq k} \left(\alpha_j + \binom{k+1-j}{2} \right). \tag{7.3}$$

On cherche maintenant à supprimer l'hypothèse $\alpha_j, \beta_j \geq k-1$. Pour cela, soit $g = X^{k-1}(1+X)^{k-1} f$. Alors $g = \sum_j a_j X^{\alpha_j + k - 1}(1+X)^{\beta_j + k - 1}$. D'après l'inégalité (7.3), $\operatorname{val}(g) \leq \max_j((\alpha_j + k - 1) + \binom{k+1-j}{2})$. Comme $\operatorname{val}(g) = \operatorname{val}(f) + k - 1$, le résultat s'ensuit. □

On peut aisément étendre ce résultat aux polynômes de la forme

$$\sum_{j=1}^{k} a_j X^{\alpha_j} (uX + v)^{\beta_j}$$

où $uv \neq 0$. Il suffit pour cela d'effectuer le changement de variables $Y = uX/v$ et de réécrire $uX + v = v(Y+1)$. Le polynôme en la variable Y a la même valuation que le polynôme original, et la borne obtenue au théorème 7.1 est valable pour ces polynômes un peu plus généraux.

Les polynômes considérés dans cette partie sont à coefficients dans un corps de caractéristique 0. Si la caractéristique est positive, le théorème 7.1 n'est pas valable. Par exemple en caractéristique 2, le polynôme

$$(1+X)^{2^n} + (1+X)^{2^{n+1}} = X^{2^n}(1+X)$$

fournit un contre-exemple. La caractéristique positive est étudiée avec plus de détails dans la partie 7.4.

7.1.2 Des améliorations possibles ?

La borne du théorème 7.1 est quadratique, alors que dans la situation plus simple du lemme de Hajós dans lequel tous les α_j sont égaux, la borne est linéaire. Il serait intéressant d'affiner nos résultats afin de déterminer une borne plus précise, et en particulier déterminer si la borne optimale est linéaire ou quadratique. Pour cela, on peut remarquer que les bornes énoncées dans les lemmes 7.4 et 7.5 peuvent être atteintes mais *a priori* pas en même temps. Si tous les α_j sont différents, la borne du lemme 7.4 est atteinte, mais celle du lemme 7.5 est remplacée par $\sum_j \alpha_j - \binom{k}{2}$. De la même façon, si les α_j sont tous égaux alors la borne du lemme 7.5 est atteinte, mais c'est la borne du lemme 7.4 qui est remplacée par $\sum_j \mathrm{val}(f_j)$. Dans les deux cas, on peut en déduire une borne meilleure que celle du théorème 7.1. On souhaite donc affiner les bornes de ces deux lemmes dans le cas général pour éventuellement obtenir une amélioration du théorème 7.1. Pour l'instant, on ne sait affiner que le lemme 7.4.

Dans le lemme 7.4, on borne inférieurement la valuation du wronskien de $(f_1, \dots, f_k)$ par le degré du wronskien des monômes de plus bas degré des f_j. Ce wronskien peut s'annuler, auquel cas la valuation du wronskien de $(f_1, \dots, f_k)$ est en réalité plus élevée. C'est par exemple le cas si deux f_j ont la même valuation. Pour prendre en compte cet aspect, on définit une notion de *palier*. Supposons les f_j ordonnés par valuation croissante. Un palier commençant à l'indice j est une suite $(f_j, f_{j+1}, \dots, f_{j+p-1})$ telle que pour tout $t < p$, $\mathrm{val}(f_{j+t}) < \mathrm{val}(f_j) + t$. La *taille du palier* est son nombre d'éléments.

Lemme 7.6

Soit $f_1, \dots, f_k \in \mathbb{K}[X]$. Supposons les f_j partitionnés en d paliers consécutifs, de la forme $(f_{j_i}, \dots, f_{j_i+p_i-1})$ pour $1 \leq i \leq d$. Alors

$$\mathrm{val}(\mathrm{wk}(f_1, \dots, f_k)) \geq \sum_{i=1}^{d} \left(p_i \,\mathrm{val}(f_{j_i}) + \binom{p_i}{2} \right) - \binom{k}{2}.$$

Démonstration : Soit W le wronskien de $(f_1, \dots, f_k)$. En utilisant la multilinéarité du déterminant, on peut écrire W comme une somme d'un nombre exponentiel de wronskiens de monômes. Ces wronskiens sont obtenus en prenant un monôme de chaque f_j. Ainsi, si les f_j ont chacun au plus t monômes, la somme porte sur au plus t^k wronskiens. On cherche à montrer que tous les wronskiens obtenus de cette manière en prenant des monômes de petit degré s'annulent.

Plus précisément, supposons que deux f_j ont la même valuation. Alors les wronskiens obtenus en prenant les deux monômes de plus bas degré de ces f_j s'annulent puisqu'ils contiennent deux monômes de

même degré. En généralisant cette remarque à un palier de taille p, on s'aperçoit que pour qu'un wronskien ne s'annule pas, il faut avoir choisi p monômes de degrés différents dans ce palier. Les degrés minimaux de ces monômes sont donc $\mathrm{val}(f_j)$, $\mathrm{val}(f_j)+1$, …, $\mathrm{val}(f_j)+p-1$ où f_j est le premier polynôme du palier.

Pour obtenir un wronskien non nul, il faut donc avoir choisi des monômes de degrés différents à l'intérieur de chaque palier. Le degré du wronskien est alors au moins

$$\sum_{i=1}^{d}\sum_{t=0}^{p_i-1}(\mathrm{val}(f_{j_i})+t)-\binom{k}{2}=\sum_{i=1}^{d}\left(p_i\,\mathrm{val}(f_{j_i})+\binom{p_i}{2}\right)-\binom{k}{2}.$$

□

On remarque que si chaque f_j est un palier à lui tout seul, la borne obtenue est la même que celle du lemme 7.4. Par contre, à l'inverse, si tous les f_j appartiennent au même palier, la borne est simplement $k\,\mathrm{val}(f_1)$. Ceci nous permet de retrouver le lemme de Hajós.

Corollaire 7.7

Soit $f=\sum_{j=1}^{k}a_jf_j$ où $f_j=X^\alpha(1+X)^{\beta_j}$. Alors

$$\mathrm{val}(f)\leq\alpha+(k-1).$$

Démonstration : C'est la même preuve que celle du théorème 7.1 en remplaçant l'utilisation du lemme 7.4 par le lemme 7.6. En effet, après avoir remplacé f_1 par f dans la matrice wronskienne, on définit les paliers (f) et $(f_2,\dots,f_k)$. On obtient donc $\mathrm{val}(\mathrm{wk}(f,f_2,\dots,f_k))\geq\mathrm{val}(f)+(k-1)\,\mathrm{val}(f_2)+\binom{k-1}{2}-\binom{k}{2}$. Ceci se traduit par $\mathrm{val}(f)\leq\alpha+(k-1)$. □

Ne disposant pas d'affinement de la borne du lemme 7.5, on peut chercher à l'inverse à obtenir une borne inférieure. La borne qu'on obtient ici montre que l'on peut dépasser la borne $\alpha_1+(k-1)$ du lemme de Hajós mais ne permet pas de trancher entre une borne optimale linéaire ou quadratique.

Théorème 7.8

Pour $k\geq 3$, il existe a_1, …, a_k, et une famille libre $(f_1,\dots,f_k)$ où $f_j=X^{\alpha_j}(1+X)^{\beta_j}$ pour $1\leq j\leq k$ tels que $f=\sum_j a_jf_j$ est non nul et $\mathrm{val}(f)=\alpha_1+(2k-3)$.

Démonstration : Le polynôme suivant atteint la borne annoncée :

$$P_k(X)=-1+(1+X)^{2k+3}-\sum_{j=0}^{k}\frac{2k+3}{2j+1}\binom{k+1+j}{k+1-j}X^{2j+1}(1+X)^{k+1-j}.$$

Nous allons montrer que $P_k(X)=X^{2k+3}$. Comme ce polynôme a $(k+3)$ termes et $\alpha_1=0$, on obtient la borne annoncée. Le résultat avec α_1

quelconque s'en déduit en multipliant P_k par X^{α_1}. La liberté de la famille des f_j est assurée par le fait qu'ils ont des degrés deux à deux distincts.

Clairement, P_k est un polynôme unitaire de degré $(2k+3)$. Soit $[X^m]P_k$ le coefficient du monôme de degré m dans P_k. Alors pour $m > 0$,

$$[X^m]P_k = \binom{2k+3}{m} - \sum_{j=1}^{k} \frac{2k+3}{2j+1}\binom{k+1+j}{k+1-j}\binom{k+1-j}{m-2j-1}.$$

Notre but est de prouver que $[X^m]P_k = 0$ pour tout $m < 2k+3$. De manière équivalente, il faut prouver l'égalité

$$\sum_{j=1}^{k} \frac{2k+3}{2j+1}\binom{k+1+j}{k+1-j}\binom{k+1-j}{m-2j-1} = \binom{2k+3}{m}. \tag{7.4}$$

On s'appuie pour cela sur l'algorithme de Herbert Wilf et Doron Zeilberger [104] et sur son implantation dans le paquet Maple `EKHAD` de Doron Zeilberger (cf [104] pour plus de détail sur ce paquet). Le programme assure la correction de l'égalité et fournit un certificat de validité sous forme d'une récurrence à vérifier à la main.

Notons $F(m,j)$ le terme général de la somme dans l'équation (7.4) divisé par $\binom{2k+3}{m}$. Le but est donc de prouver que $\sum_{j=0}^{k} F(m,j) = 1$ pour $m < 2k+3$. Le logiciel `EKHAD` affirme alors que

$$mF(m+1,j) - mF(m,j) = F(m,j+1)R(m,j+1) - F(m,j)R(m,j) \tag{7.5}$$

où

$$R(m,j) = \frac{2j(2j+1)(k+j+2-m)}{(2k+3-m)(2j-m)}.$$

On montre dans un premier temps que la relation de récurrence (7.5) implique bien l'identité (7.4), puis dans un deuxième temps que cette relation de récurrence est bien vérifiée.

Si on somme la relation (7.5) pour $j = 0$ à k, on obtient

$$m\left(\sum_{j=0}^{k} F(m+1,j) - F(m,j)\right) = F(m,k+1)R(m,k+1) - F(m,0)R(m,0).$$

Or $R(m,0) = F(m,k+1) = 0$, donc $S(m) = \sum_j F(m,j)$ est constante. De plus, on vérifie aisément que $S(2k+2) = 1$ puisque le seul terme non nul est celui correspondant à $j = k$. Ainsi, pour tout $m < 2k+3$ [2], $\sum_j F(m,j) = 1$ ce qui prouve l'identité (7.4).

Pour prouver la relation (7.5), on remarque que

$$\frac{F(m+1,j)}{F(m,j)} = \frac{(j+k+2-m)(m+1)}{(m-2j)(2k+3-m)}$$

2. La borne sur m vient du fait que $R(m,j)$ n'est pas définie pour $m = 2k+3$.

et

$$\frac{F(m,j+1)}{F(m,j)} = \frac{(k+2-j)(m-2j-1)(m-2j-2)}{(2j+3)(2j+2)(k+j+3-m)}.$$

Il suffit donc de vérifier que

$$m\frac{(j+k+2-m)(m+1)}{(m-2j)(2k+3-m)} - m \\ = \frac{(k+2-j)(m-2j-1)(m-2j-2)}{(2j+3)(2j+2)(k+j+3-m)}R(m,j+1) - R(m,j).$$

C'est un calcul facile à effectuer quoiqu'un peu fastidieux. □

Grâce à ce théorème, on peut affirmer que la valuation maximale d'un polynôme de la forme $f = \sum_{j=1}^{k} a_j X^{\alpha_j}(1+X)^{\beta_j}$ est comprise entre $(2k-3)$ et $\binom{k}{2}$. On ne connaît pas de bornes plus précises à ce jour.

7.1.3 *Un théorème de lacune*

L'objectif de départ était d'obtenir un nouveau théorème de lacune. Il est une conséquence naturelle de la borne sur la valuation exprimée par le théorème 7.1.

Théorème 7.9

Soit $f = \sum_{j=1}^{k} a_j X^{\alpha_j}(uX+v)^{\beta_j}$ où $uv \neq 0$ et $\alpha_j \leq \alpha_{j+1}$ pour $1 \leq j < k$. Supposons qu'il existe ℓ tel que

$$\alpha_{\ell+1} > \max_{1\leq j\leq \ell}\left(\alpha_j + \binom{\ell+1-j}{2}\right). \tag{7.6}$$

Alors f est le polynôme nul si et seulement si

$$g = \sum_{j=1}^{\ell} a_j X^{\alpha_j}(uX+v)^{\beta_j} \qquad \text{et} \qquad h = \sum_{j=\ell+1}^{k} a_j X^{\alpha_j}(uX+v)^{\beta_j}$$

sont tous deux des polynômes nuls.

En particulier, le plus petit indice ℓ vérifiant l'inégalité (7.6) est le plus petit indice tel que

$$\alpha_{\ell+1} > \alpha_1 + \binom{\ell}{2}.$$

Démonstration : Supposons que g n'est pas le polynôme nul. D'après le théorème 7.1, sa valuation est au plus $\max_j(\alpha_j + \binom{\ell+1-j}{2})$. Puisque $\alpha_j \geq \alpha_{\ell+1}$ pour tout $j > \ell$, la valuation de h est au moins $\alpha_{\ell+1} > \max_j(\alpha_j + \binom{\ell+1-j}{2})$. Le monôme de plus bas degré de g ne peut donc pas être annulé

par un monôme de h. En d'autres termes, $f = g + h$ n'est pas le polynôme nul.

Pour prouver la deuxième partie de l'énoncé, on considère le plus petit indice ℓ vérifiant l'inégalité (7.6). Il est par conséquent clair que $\alpha_{\ell+1} > \alpha_1 + \binom{\ell}{2}$. De plus, $\alpha_{j+1} \leq \max_{i \leq j}(\alpha_i + \binom{j+1-i}{2})$ pour tout $j < \ell$. Par induction sur j, on peut en déduire que $\alpha_{j+1} \leq \alpha_1 + \binom{j}{2}$ pour tout $j < \ell$. C'est évident pour $j = 0$. Soit donc $j < \ell$ et supposons que pour tout $i < j$, $\alpha_{i+1} \leq \alpha_1 + \binom{i}{2}$. Alors $\alpha_{i+1} + \binom{j-i}{2} \leq \alpha_1 + \binom{i}{2} + \binom{j-i}{2}$. De plus, $\binom{i}{2} + \binom{j-i}{2} \leq \binom{j}{2}$ pour tout $i < j$. Ainsi, $\alpha_{j+1} \leq \max_{i \leq j}(\alpha_1 + \binom{j}{2}) = \alpha_1 + \binom{j}{2}$. □

7.2 Algorithmes

On s'intéresse maintenant aux algorithmes que l'on peut déduire de la borne sur la valuation. Ces algorithmes ne sont pas à proprement parler nouveaux. Ce sont ceux proposés par Erich L. Kaltofen et Pascal Koiran [63] dans lesquels on remplace leur théorème de lacune par le nôtre. Il en résulte une différence de complexité, qui est analysée à la fin de la partie. Deux algorithmes sont présentés. Le premier est un test d'identité polynomiale pour des polynômes de la forme

$$f = \sum_{j=1}^{k} a_j X^{\alpha_j} (uX + v)^{\beta_j}. \tag{7.7}$$

Le deuxième calcule les facteurs linéaires d'un polynôme lacunaire à deux variables. Ces deux algorithmes sont déterministes et fonctionnent en temps polynomial.

La taille d'un polynôme de la forme (7.7) est

$$\text{taille}(f) = \text{taille}(u) + \text{taille}(v) + \sum_{j=1}^{k} (\text{taille}(a_j) + \log(1 + \alpha_j) + \log(1 + \beta_j)).$$

Cette taille dépend donc de la taille des coefficients qui varie selon le corps sur lequel on travaille.

Les algorithmes faisant appel à l'algorithme de Lenstra de factorisation des polynômes creux à une variable [84], ils ne sont utilisables que pour des polynômes à coefficients dans un corps de nombres. On considère donc un corps de nombres $\mathbb{K}$ que l'on représente sous la forme d'un quotient $\mathbb{Q}[\xi]/\langle\varphi\rangle$ où $\varphi \in \mathbb{Z}[\xi]$ est un polynôme unitaire irréductible. Les éléments de $\mathbb{K}$ sont donnés sous la forme de vecteurs dans la base $(1, \xi, \xi^2, \ldots, \xi^{\deg(\varphi)-1})$. Un élément $e \in \mathbb{K}$ est donc un multiplet $(e_0, \ldots, e_{\deg(\varphi)-1})$ où pour tout t, $e_t = n_t/d_t$ avec $n_t, d_t \in \mathbb{Z}$. La taille de e est définie par

$$\text{taille}(e) = \sum_{t=0}^{\deg(\varphi)-1} \log(1 + n_t) + \log(1 + d_t).$$

La taille d'un élément, et par suite d'un polynôme, est une approximation du nombre de bits nécessaires pour écrire l'élément ou le polynôme. Bien entendu, il faudrait pour avoir une approximation plus précise utiliser des parties entières pour que la taille soit un entier, mais le but ici est de donner une définition aussi simple que possible.

Théorème 7.10

> Il existe un algorithme déterministe polynomial pour décider si un polynôme de la forme (7.7) est le polynôme nul.

Démonstration : On peut supposer sans perte de généralité que $\alpha_{j+1} \geq \alpha_j$ pour tout $j < k$, et que $\alpha_1 = 0$. Si α_1 est non nul, X^{α_1} divise f et on peut appliquer notre algorithme à $f(X)/X^{\alpha_1}$, c'est-à-dire au polynôme obtenu en remplaçant chaque α_j par $(\alpha_j - \alpha_1)$.

Supposons en premier lieu que $u = 0$. Alors f est une somme de monômes et il suffit de tester la nullité de chaque coefficient. Cependant, les α_j n'étant pas distincts, un coefficient de f est de la forme $\sum_j a_j v^{\beta_j}$. En utilisant l'algorithme de Lenstra pour trouver les facteurs linéaires du polynôme $\sum_j a_j X^{\beta_j}$ [84], on peut vérifier si v est une racine et en déduire la nullité ou non de du coefficient de f. Pour réduire le temps de calcul, on peut extraire de l'algorithme la partie appropriée, à savoir un algorithme qui teste si un élément $v \in \mathbb{K}$ donné est une racine. Le cas $v = 0$ est similaire.

On peut maintenant supposer que $uv \neq 0$. En appliquant récursivement notre théorème de lacune (théorème 7.9) au polynôme f, on peut le découper sous la forme $f = g_1 + \cdots + g_s$ de telle sorte que f est le polynôme nul si et seulement si chacun des polynômes g_t, $1 \leq t \leq s$, est nul. Formellement, on définit $I_1, \ldots, I_s$ comme étant l'unique partition de $\{1, \ldots, k\}$ en intervalles définie par la procédure inductive suivante. On commence avec $I_1 = \{1\}$. Pour $1 \leq j < k$, on suppose que $\{1, \ldots, j\}$ a été partitionné en intervalles $I_1, \ldots, I_t$ et on note i_t l'élément minimal de I_t. Alors on ajoute l'élément $(j+1)$ à I_t si $\alpha_{j+1} \leq \alpha_{i_t} + \binom{j-i_t+1}{2}$, et on crée un nouvel intervalle $I_{t+1} = \{j+1\}$ sinon. Les polynômes $g_1, \ldots, g_s$ sont définis par $g_t = \sum_{j \in I_t} a_j X^{\alpha_j}(1+X)^{\beta_j}$ pour $1 \leq t \leq s$. On vérifie aisément que les polynômes g_t vérifient les conditions du théorème de lacune. Plus précisément, f est nul si et seulement si g_1 et $g_2 + \cdots + g_s$ sont tous deux nuls. Et récursivement, on obtient la propriété souhaitée, c'est-à-dire que f est nul si et seulement si chaque g_t est nul. Il ne nous reste donc plus qu'à savoir tester si chaque g_t est nul. De même que pour f, on peut diviser chaque g_t par X^{i_t}.

Pour prouver que l'on peut effectuer ces tests en temps polynomial, considérons un polynôme g de la forme (7.7) vérifiant $\alpha_1 = 0$ et $\alpha_{j+1} \leq \binom{j}{2}$

pour tout j. On effectue le changement de variable $Y = uX + v$. Alors

$$g(Y) = \sum_{j=1}^{k} a_j u^{-\alpha_j} (Y - v)^{\alpha_j} Y^{\beta_j}$$

est le polynôme nul si et seulement si $g(X)$ également. On développe entièrement le polynôme $g(Y)$ comme une somme de puissances de Y :

$$g(Y) = \sum_{j=1}^{k} \sum_{\ell=0}^{\alpha_j} a_j u^{-\alpha_j} \binom{\alpha_j}{\ell} (-v)^{\ell} Y^{\alpha_j + \beta_j - \ell}.$$

Il y a au plus $k\binom{k-1}{2} = \mathcal{O}(k^3)$ monômes, et il suffit de tester la nullité de chaque coefficient. Un coefficient est de la forme

$$\sum_{j} \binom{\alpha_j}{\ell_j} a_j u^{-\alpha_j} (-v)^{\ell_j}$$

où la somme porte sur au plus k indices. Puisque $\ell_j, \alpha_j \leq \binom{k-1}{2}$ pour tout j, les termes de cette somme sont de taille polynomiale. On peut donc calculer explicitement la somme en temps polynomial pour tester sa nullité.

Pour conclure, le partitionnement de l'ensemble $\{1, \ldots, k\}$ en intervalles s'effectue en temps polynomial. Il reste ensuite au plus k polynômes de petit degré à tester, et ceci s'effectue en temps polynomial. On peut donc tester la nullité de f en temps déterministe polynomial. □

Le test d'identité polynomiale est sans doute l'application la plus directe du théorème de lacune. On étudie maintenant comment utiliser ce théorème pour trouver les facteurs linéaires d'un polynôme lacunaire à deux variables.

Théorème 7.11

Soit

$$f(X, Y) = \sum_{j=1}^{k} a_j X^{\alpha_j} Y^{\beta_j} \in \mathbb{K}[X, Y].$$

Il existe un algorithme déterministe polynomial pour trouver les facteurs linéaires de f, avec leur multiplicité.

Démonstration : Les facteurs linéaires de f sont de deux types : soit $(Y - v)$ ou $(X - v)$, soit $(Y - uX - v)$ avec $u \neq 0$.

Pour trouver les facteurs du premier type, disons de la forme $(Y - v)$, on voit f comme un polynôme de $\mathbb{K}[Y][X]$. On regroupe donc les termes de f qui ont le même exposant α_j, et on cherche les facteurs linéaires du polynôme à une variable Y qui est le coefficient de X^{α_j} dans f. Pour trouver ces facteurs, on fait appel à l'algorithme de Lenstra [84].

Le polynôme $(Y - uX - v)$ divise f si et seulement si $f(X, uX + v)$ est le polynôme nul. Si $v = 0$, $f(X, uX) = \sum_j a_j u^{\beta_j} X^{\alpha_j + \beta_j}$. Trouver les facteurs de f de la forme $(Y - uX)$ revient donc à trouver les valeurs $u \in \mathbb{K}$ telles que tous les coefficients de $f(X, uX)$ s'annulent. Or un coefficient de ce polynôme est de la forme $\sum_j a_j u_j^{\beta_j}$. Il suffit donc de trouver les facteurs linéaires d'un petit nombre (au plus k) de polynômes creux à une variable. À nouveau, cette tâche est confiée à l'algorithme de Lenstra [84].

On s'intéresse maintenant aux facteurs de la forme $(Y - uX - v)$ avec $uv \neq 0$. Le polynôme $f(X, uX + v)$ est de la forme (7.7). En lui appliquant le théorème de lacune, on peut l'écrire sous la forme

$$f(X, uX + v) = \sum_{t=1}^{s} X^{\alpha_{i_t}} g_t(X, uX + v)$$

où chaque $g_t(X, uX + v)$ est de la forme (7.7) et vérifie $\alpha_1 = 0$ et $\alpha_k \leq \binom{k-1}{2}$, de telle sorte que f est le polynôme nul si et seulement si chaque $g_t(X, uX + v)$ également. Mais $g_t(X, uX + v)$ est le polynôme nul si et seulement si $(Y - uX - v)$ divise $g_t(X, Y)$. On peut à nouveau appliquer le même découpage de chaque g_t en inversant maintenant les rôles de X et Y. C'est-à-dire qu'on écrit

$$g_t = \sum_{\ell=1}^{s_t} Y^{\beta_{i_\ell}} h_{t,\ell}$$

où $h_{t,\ell}(X, uX + v)$ est de la forme (7.7) et vérifie $\alpha_1 = \beta_1 = 0$ et $\alpha_k, \beta_k \leq \binom{k-1}{2}$, de telle sorte que $(Y - uX - v)$ est un facteur de g_t si et seulement si c'en est un de chaque $h_{t,\ell}$.

On a donc construit un ensemble de polynômes $h_{t,\ell}$, $1 \leq t \leq s$, $1 \leq \ell \leq s_t$, tels que les facteurs linéaires de f sont les facteurs linéaires communs à tous les $h_{t,\ell}$. De plus, il y a au plus k polynômes $h_{t,\ell}$, et ils sont de degré au plus $2\binom{k-1}{2} = k^2 - 3k + 2$. Pour trouver les facteurs linéaires de ces $h_{t,\ell}$, on utilise l'un des nombreux algorithmes de factorisation de polynômes dense à deux variables qui apparaissent dans la littérature, de [64] et [83] pour les plus anciens à [37] et [82] pour les plus récents. Plusieurs stratégies peuvent alors être utilisées pour trouver les facteurs linéaires de f. Soit on cherche les facteurs linéaires de chaque $h_{t,\ell}$ et on ne garde que ceux qui sont communs, soit on cherche les facteurs linéaires de l'un d'entre eux puis on teste lesquels sont facteurs des autres $h_{t,\ell}$ avec notre algorithme de test d'identité polynomiale, soit enfin on calcule le plus grand diviseur commun (pgcd) des $h_{t,\ell}$ pour ensuite chercher ses facteurs linéaires.

Pour obtenir les multiplicités des facteurs, on utilise la même technique que Lenstra [84]. Tout d'abord, l'algorithme de Lenstra fournit

les multiplicités des facteurs de la forme $(X - v)$, $(Y - v)$ et $(Y - uX)$. Ensuite, on considère les *dérivées partielles généralisées*. Pour $f \in \mathbb{K}[X, Y]$ de la même forme que dans l'énoncé, on définit

$$D_X^{[1]}(f) = \frac{\partial}{\partial X}\left(\frac{f}{X^{\min_j(\alpha_j)}}\right), \quad D_X^{[t]}(f) = D_X^{[1]}\left(D_X^{[t-1]}(f)\right).$$

Alors pour tout t, $D_X^{[t]}(f)$ est de la même forme que f mais avec moins de termes. De plus, si $(Y - uX - v)$ est un facteur de multiplicité μ, alors c'est un facteur des $D_X^{[t]}(f)$ pour $1 \leq t \leq \mu - 1$. Il suffit donc de chercher les facteurs linéaires des $D_X^{[t]}(f)$ pour $t \leq k$ pour obtenir la multiplicité des facteurs linéaires de f. En d'autres termes, il suffit d'appliquer l'algorithme décrit précédemment à k polynômes ou moins.

De même que pour l'algorithme de test d'identité polynomiale, le découpage de f en polynômes $h_{t,\ell}$ se fait en temps polynomial. Ensuite, on applique un algorithme de factorisation dense de complexité polynomiale à des polynômes de degré $\mathcal{O}(k^2)$. L'algorithme fonctionne donc globalement en temps polynomial. □

Une mesure pertinente pour estimer un peu plus précisément la complexité de nos algorithmes est la valeur du *saut* dans le théorème de lacune. En effet, cette valeur se traduit en le degré maximal des polynômes auxquels on applique un algorithme de factorisation dense par exemple. Dans notre théorème de lacune, cette valeur est un $\mathcal{O}(k^2)$. En comparaison, Erich Kaltofen et Pascal Koiran avaient une valeur de $\mathcal{O}(k \log k + k \log h_f)$ où h_f est la hauteur algébrique du polynôme f [63]. En particulier, si les coefficients a_j de f sont des entiers, $h_f = \max_j |a_j|$. La dépendance en k est donc meilleure, mais il y a une dépendance en la taille des coefficients que nous n'avons pas. Notre algorithme est donc plus rapide pour des polynômes ayant des grands coefficients. Une meilleure borne pour la valuation maximale d'un polynôme de la forme (7.7) que celle exprimée par le théorème 7.1 permettrait de se rapprocher de la borne de Kaltofen et Koiran, voire de l'améliorer. Pour finir la comparaison, il faut noter que notre valeur de saut dans le théorème de lacune est particulièrement simple à calculer. Pour l'algorithme de Koiran et Kaltofen, il faut estimer la hauteur algébrique du polynôme, ce qui n'est pas aussi direct lorsque le corps considéré n'est pas celui des rationnels. Cette partie ne ralentit pas de manière significative leur algorithme, mais le rend moins facile à implanter.

7.3 Généralisations

7.3.1 *Une borne générale sur la valuation*

Dans cette partie, on généralise le théorème 7.1 à une somme de produits de puissances de polynômes. La classe considérée ressemble à la classe

SPS(k, m, t) du chapitre 6, mais la différence est que l'on borne le degré des polynômes et non leur nombre de monômes.

Théorème 7.12

Soit $(\alpha_{ij}) \in \mathbb{N}^{m\times k}$ et

$$f = \sum_{j=1}^{k} a_j \prod_{i=1}^{m} f_i^{\alpha_{ij}} \in \mathbb{K}[X]$$

où f_i est de degré d_i pour tout i. Soit $\xi \in \mathbb{K}$ et soit μ_i la multiplicité de ξ comme racine de f_i.

Si f est non nul, la multiplicité $\mu_f(\xi)$ de ξ comme racine de f vérifie

$$\mu_f(\xi) \leq \max_{1\leq j\leq k} \sum_{i=1}^{m} \left(\mu_i \alpha_{ij} + (d_i - \mu_i) \binom{k+1-j}{2} \right).$$

Démonstration : Pour tout j, notons $P_j = \prod_{i=1}^{m} f_i^{\alpha_{ij}}$. De même que dans la preuve du théorème 7.1, on peut supposer la famille $(P_1, \ldots, P_k)$ libre et les α_{ij} supérieurs à $(k-1)$.

D'après la règle de Leibniz,

$$P_j^{(\ell)} = \sum_{t_1+\cdots+t_m=\ell} \binom{\ell}{t_1, \ldots, t_m} \prod_{i=1}^{m} (f_i^{\alpha_{ij}})^{(t_i)} \tag{7.8}$$

où $\binom{\ell}{t_1,\ldots,t_m}$ est le coefficient multinomial. Considérons maintenant une dérivée de la forme $(f^\alpha)^{(t)}$. C'est une somme de termes qui contiennent tous le facteur $f^{\alpha-t}$, le cas le pire ayant lieu lorsque t copies différentes de f ont chacune été dérivées une fois. Dans l'équation (7.8), chaque t_i est borné par ℓ donc il existe un polynôme $Q_{\ell,j}$ tel que

$$P_j^{(\ell)} = Q_{\ell,j} \prod_{i=1}^{m} f_i^{\alpha_{ij}-\ell}.$$

Le degré de $P_j^{(\ell)}$ étant $\sum_i d_i\alpha_{ij} - \ell$, le degré de $Q_{\ell,j}$ est

$$\sum_{i=1}^{m} d_i\alpha_{ij} - \ell - \sum_{i=1}^{m} (d_i\alpha_{ij} - d_i\ell) = \ell \sum_{i=1}^{m} (d_i - 1). \tag{7.9}$$

Soit W le wronskien de la famille (P_j). On peut écrire $f_i^{\alpha_{ij}-\ell}$ sous la forme $f_i^{\alpha_{ij}-k+1} f_i^{k-1-\ell}$. Cela signifie que $\prod_i f_i^{\alpha_{ij}-k+1}$ est un facteur commun des coefficients de la colonne j, et $\prod_i f_i^{k+1-\ell}$ un facteur commun des

coefficients de la ligne ℓ. Ainsi

$$W = \prod_{i=1}^{m} f_i^{\sum_j \alpha_{ij} - \binom{k}{2}} \det(Q)$$

où Q est la matrice dont le coefficient d'indice (ℓ, j) est $Q_{\ell,j}$ pour $0 \leq \ell \leq k-1$ et $1 \leq j \leq k$. D'après l'équation (7.9), $\det(Q)$ est un polynôme de degré au plus $\binom{k}{2} \sum_i (d_i - 1)$.

La multiplicité de ξ comme racine du wronskien W, notée $\mu_W(\xi)$, est bornée par sa multiplicité en tant que racine de $\prod_i f_i^{\sum_j \alpha_{ij} - \binom{k}{2}}$ additionnée au degré de $\det(Q)$. On en déduit que

$$\begin{aligned} \mu_W(\xi) &\leq \sum_{i=1}^{m} \mu_i \left(\sum_{j=1}^{k} \alpha_{ij} - \binom{k}{2} \right) + \binom{k}{2} \sum_{i=1}^{m} (d_i - 1) \\ &= \sum_{i=1}^{m} \left(\mu_i \sum_{j=1}^{k} \alpha_{ij} - (d_i - \mu_i) \binom{k}{2} \right) - \binom{k}{2}. \end{aligned} \tag{7.10}$$

On généralise maintenant le lemme 7.4 aux multiplicités de racines quelconques (la valuation étant la multiplicité de la racine nulle) sous la forme $\mu_W(\xi) \geq \sum_j \mu_{P_j}(\xi) - \binom{k}{2}$ où $\mu_{P_j}(\xi)$ est la multiplicité de P_j et vaut $\sum_i \mu_i \alpha_{ij}$. En utilisant comme dans la preuve du théorème 7.1 des opérations élémentaires sur les colonnes, on peut remplacer P_1 par f dans la matrice wronskienne en multipliant le wronskien par une constante. On en déduit que

$$\mu_W(\xi) \geq \mu_f(\xi) + \sum_{j=2}^{k} \mu_{P_j}(\xi) - \binom{k}{2}. \tag{7.11}$$

En combinant les équations (7.10) et (7.11), on conclut que

$$\begin{aligned} \mu_f(\xi) &\leq \mu_W(\xi) - \sum_{j=2}^{k} \mu_{P_j}(\xi) + \binom{k}{2} \\ &\leq \sum_{i=1}^{m} \left(\mu_i \sum_{j=1}^{k} \alpha_{ij} + (d_i - \mu_i) \binom{k}{2} \right) - \binom{k}{2} - \sum_{j=2}^{k} \sum_{i=1}^{m} \mu_i \alpha_{ij} + \binom{k}{2} \\ &\leq \sum_{i=1}^{m} \left(\mu_i \alpha_{i1} + (d_i - \mu_i) \binom{k}{2} \right). \end{aligned}$$

La suppression des deux hypothèses de départ est effectuée de la même manière que dans la preuve du théorème 7.1. □

L'ordre des P_j dans l'énoncé du théorème est arbitraire. Pour minimiser la borne obtenue, il faut trouver l'ordre optimal. Celui-ci consiste à ordonner les P_j selon la valeur de $\sum_i \mu_i \alpha_{ij}$. En effet, soit

$$s_j = \sum_{i=1}^{m} \mu_i \alpha_{ij} + \sum_{i=1}^{m} (d_i - \mu_i) \binom{k+1-j}{2}.$$

Le théorème 7.12 affirme que $\mu_f(\xi) \leq \max_j(s_j)$. Le terme $\sum_i (d_i - \mu_i)\binom{k+1-j}{2}$ est une fonction décroissante de j. Pour minimiser $\max_j(s_j)$, il faut donc que le terme $\sum_i \mu_i \alpha_{ij}$ soit une fonction croissante de j. On remarque d'ailleurs que cet ordre est cohérent avec celui utilisé pour le théorème 7.1.

Si on considère le cas $m = 2$ avec $f_1 = X$ et $f_2 = (1 + X)$ et qu'on regarde la multiplicité de la racine nulle, les théorèmes 7.1 et 7.12 donnent exactement la même borne. De même, les affinements de la partie 7.1.2 peuvent également être utilisés ici.

Enfin, il est possible de généraliser encore ce théorème dans deux directions. D'une part on peut considérer non plus la multiplicité d'une racine mais la multiplicité d'un facteur irréductible de f. Le résultat reste le même et la preuve est en fait inchangée. D'autre part, on peut ne plus supposer que les α_{ij} sont des entiers. On peut autoriser des rationnels, ou même des réels, et s'intéresser à une racine ξ telle que chaque f est développable en série entière au voisinage de ξ. On obtient à nouveau le même résultat. Ainsi, il généralise le résultat de Neeraj Kayal et Chandan Saha sur la valuation des sommes de la forme $\sum_j a_j g_j \sqrt{f_j}$ où $f_j(0) \neq 0$ [71].

7.3.2 *Généralisations des algorithmes*

En utilisant le théorème 7.12, et sa généralisation esquissée précédemment aux facteurs irréductibles de f, on peut généraliser le test d'identité polynomial du théorème 7.10. Étant donné

$$f = \sum_{j=1}^{k} a_j \prod_{i=1}^{m} f_i^{\alpha_{ij}}$$

où f_i est de degré d_i pour tout i, on peut commencer par factoriser chaque f_i. Supposons donc que les f_i sont des polynômes irréductibles. On s'intéresse alors à la multiplicité de chaque f_i comme facteur de f. Si f est non nul, la multiplicité est bornée grâce au théorème 7.12. Si les α_{ij} croissent rapidement, on peut appliquer le même argument que pour le théorème 7.10 pour couper f en plusieurs polynômes qui sont nuls si et seulement si f l'est. En appliquant cette méthode avec tous les f_i consécutivement, on se ramène à des tests d'identité polynomiale de polynômes de petit degré, ce qui est fait en calculant leurs coefficients explicitement.

On va voir maintenant une autre généralisation du théorème 7.10. Le test d'identité est moins général que celui esquissé ci-avant, mais sa preuve se

base directement sur le théorème 7.1 (et non sa généralisation) et utilise une astuce très simple.

Théorème 7.13

Il existe un algorithme déterministe polynomial pour tester si un polynôme de la forme $\sum_{j=1}^{k} a_j X^{\alpha_j}(uX^d + v)^{\beta_j}$ est le polynôme nul.

Démonstration : Soit f un polynôme de la même forme que dans l'énoncé. Pour tout j, on considère la division euclidienne de α_j par d : $\alpha_j = dq_j + r_j$, avec $r_j < d$. On peut alors récrire

$$f(X) = \sum_{j=1}^{k} a_j X^{r_j}(X^d)^{\beta_j}(uX^d + v)^{\beta_j}.$$

On peut regrouper les termes de la somme par valeur de reste r_j, c'est-à-dire qu'on définit pour $0 \leq i < d$,

$$f_i(Y) = \sum_{\substack{1 \leq j \leq k \\ r_j = i}} a_j Y^{q_j}(uY + v)^{\beta_j}.$$

Alors $f(X) = \sum_{i=0}^{d-1} X^i f_i(X^d)$. Un monôme X^α de $X^i P_i(X^d)$ vérifie $\alpha \equiv i \mod d$. Donc f est le polynôme nul si et seulement si chacun des f_i l'est.

Pour tester la nullité des f_i, il suffit de faire appel à l'algorithme du théorème 7.10. On remarque que qu'indépendamment de la valeur de d, le nombre de f_i est toujours borné par k. □

On utilise maintenant une version très simplifiée du théorème 7.12 pour obtenir de nouveaux facteurs des polynômes lacunaires à deux variables. On montre en particulier comment obtenir les facteurs multilinéaires, c'est-à-dire linéaires vis-à-vis de chaque variable mais de degré total potentiellement 2.

Théorème 7.14

Soit $f = \sum_{j=1}^{k} a_j X^{\alpha_j} Y^{\beta_j}$. Il existe un algorithme déterministe polynomial pour trouver tous les facteurs multilinéaires de f.

Démonstration : La preuve repose la remarque suivante : $XY - (uX - vY + w)$ est un facteur de f si et seulement si $f(X, \frac{uX+w}{X+v})$ est la fraction rationnelle identiquement nulle. On suppose pour simplifier que $uvw \neq 0$, les autres cas se traitant à part comme dans la preuve du théorème 7.11. Soit

$$g(X) = (X+v)^{\max_j \beta_j} f\left(X, \frac{uX+w}{X+v}\right) = \sum_{j=1}^{k} a_j X^{\alpha_j}(uX+w)^{\beta_j}(X+v)^{\gamma_j}$$

où $\gamma_j = \max_\ell(\beta_\ell) - \beta_j$ pour tout j. Alors g est le polynôme nul si et seulement si $f(X, \frac{uX+w}{X+v})$ est nulle. Autrement dit, pour trouver les facteurs

multilinéaires de f, il suffit de trouver les valeurs de u, v et w telles que g est le polynôme nul. On peut appliquer le théorème 7.12 à g avec trois polynômes de degré 1 et la racine nulle pour déduire que si g est non nul, alors sa valuation est au plus $\max_j(\alpha_j + 2\binom{k+1-j}{2})$.

Le reste de la démonstration est identique à celle du théorème 7.11. Cette borne sur la valuation permet de partitionner l'ensemble $\{1, \ldots, k\}$ en intervalles I_1, ..., I_s de telle sorte que g est nul si et seulement si

$$g_t(X) = \sum_{j \in I_t} a_j X^{\alpha_j} (uX + w)^{\beta_j} (X + v)^{\gamma_j}$$

est nul pour tout t et vérifie $\max_{j \in I_t}(\alpha_j) - \min_{j \in I_t}(\alpha_j) \leq \binom{k-1}{2}$. De plus, $g_t(X)$ est nul si et seulement si $XY - (uX - vY + w)$ est facteur de

$$f_t(X, Y) = \sum_{j \in I_t} a_j X^{\alpha_j} Y^{\beta_j}.$$

On peut alors appliquer le même argument aux f_t en échangeant les rôles de X et Y. *In fine*, on a exprimé f sous la forme

$$f(X, Y) = X^{\alpha_{j_1}} Y^{\beta_{j_1}} f_1(X, Y) + \cdots + X^{\alpha_{j_s}} Y^{\beta_{j_s}} f_s(X, Y)$$

où les f_t sont de degré au plus $4\binom{k-1}{2}$ et sont tels que $XY - (uX - vY + w)$ est facteur de f si et seulement s'il l'est de tous les f_t.

Pour conclure, on fait appel aux mêmes algorithmes de factorisation dense que pour le théorème 7.11. □

7.4 Caractéristique positive

On l'a vu précédemment, le théorème 7.1 est faux en caractéristique positive. En caractéristique 2 par exemple, le polynôme $(1 + X)^{2^n} + (1 + X)^{2^{n+1}}$ n'a que deux termes mais sa valuation est 2^n. On peut obtenir un résultat similaire en caractéristique p quelconque en considérant le polynôme $\sum_{i=1}^{p} (1 + X)^{p^{n+i}}$ qui est de valuation p^n. Ceci montre que la valuation ne peut pas être bornée par une fonction du nombre de termes en toute généralité.

Cependant, les exposants choisis dans les exemples précédents dépendent de la caractéristique. Plus précisément, ils sont plus grand que la caractéristique. Nous allons montrer qu'en grande caractéristique, le théorème 7.1 reste vrai. Ceci contraste avec le résultat de Koiran et Kaltofen [63] qui n'est valable qu'en caractéristique 0 à cause de son utilisation de la notion de hauteur algébrique.

Tout ce dont on a besoin pour que le théorème 7.1 reste vrai est de pouvoir supposer que le wronskien d'une certaine famille de polynômes ne s'annule pas. La proposition 7.3 qui était utilisée à cette fin n'est vraie qu'en caractéristique nulle, mais une variante existe pour la caractéristique

positive. On dit qu'une famille de polynômes $(f_1, \ldots, f_k) \in \mathbb{K}[X]^k$ est *libre sur* $\mathbb{K}[X^p]$ si pour tout $p_1, \ldots, p_k \in \mathbb{K}[X^p]$, $p_1 f_1 + \cdots + p_k f_k = 0$ implique $p_1 = \cdots = p_k = 0$.

Proposition 7.15

Soit $\mathbb{K}$ un corps de caractéristique p et $f_1, \ldots, f_k \in \mathbb{K}[X]$. Alors la famille $(f_1, \ldots, f_k)$ est libre sur $\mathbb{K}[X^p]$ si et seulement si son wronskien est non nul.

Nous ne démontrons pas cette proposition car nous allons en fait n'utiliser qu'un cas particulier plus simple de ce résultat. Une preuve peut être trouvée dans l'ouvrage d'Irving Kaplansky [67].

Corollaire 7.16

Soit $\mathbb{K}$ un corps de caractéristique p et $f_1, \ldots, f_k \in \mathbb{K}[X]$, de degrés inférieurs à $(p-1)$. Alors la famille $(f_1, \ldots, f_k)$ est libre (sur $\mathbb{K}$) si et seulement si son wronskien ne s'annule pas.

Il est facile de voir que le corollaire se déduit de la proposition. La famille $(f_1, \ldots, f_k)$ est libre sur $\mathbb{K}[X^p]$ si et seulement si pour tous $p_1, \ldots, p_k \in \mathbb{K}[X^p]$, $p_1 f_1 + \cdots + p_k f_k = 0 \implies p_1 = \cdots = p_k = 0$. Mais les f_j étant de degré au plus $(p-1)$, on obtient le résultat souhaité. Nous donnons maintenant une preuve directe du corollaire.

Démonstration : On suit la preuve faite en caractéristique 0. Si la famille est liée, le wronskien s'annule. Sinon, on peut échelonner les valuations des f_j par opérations élémentaires sur les colonnes. Tous les f_j ont alors des valuations distinctes, inférieures à $(p-1)$. On s'intéresse au wronskien des monômes de plus bas degré des f_j. Si le monôme de plus bas degré de f_j est $a_j x^{\alpha_j}$, on peut mettre $a_j x^{\alpha_j - k + 1}$ en facteur dans la colonne j, et donc en facteur du wronskien. Alors pour $0 \leq \ell < k$, $x^{k-1-\ell}$ est facteur des coefficients de la ligne ℓ. En effectuant ces factorisations sur les colonnes et les lignes de la matrice wronskienne, on obtient une matrice semblable à la matrice de Vandermonde de terme général α_j^ℓ avec $\alpha_j \in \{0, \ldots, p-1\}$ pour tout j, et les α_j tous distincts. Ce déterminant est donc non nul, et le wronskien des monômes de plus bas degré des f_j est non nul. Ceci prouve que le wronskien de la famille $(f_1, \ldots, f_k)$ est non nul. □

On peut alors en déduire un équivalent du théorème 7.1 en caractéristique positive.

Théorème 7.17

Soit $\mathbb{K}$ un corps de caractéristique p et

$$f = \sum_{j=1}^{k} a_j X^{\alpha_j} (1+X)^{\beta_j} \in \mathbb{K}[X]$$

où $\alpha_1 \leq \cdots \leq \alpha_k$. Si f est de degré au plus $(p-1)$, alors sa valuation est bornée par $\max_j(\alpha_j + \binom{k+1-j}{2})$.

Démonstration : On note en premier lieu que si f est de degré fini, alors ce n'est pas le polynôme nul. Notons $f_j = X^{\alpha_j}(1+X)^{\beta_j}$ pour tout j. La preuve du théorème 7.1 procède en deux étapes. On montre qu'on peut supposer que le wronskien des f_j ne s'annule pas, puis on en déduit la borne sur la valuation. On peut ici effectuer les deux mêmes étapes. La première est une conséquence du corollaire 7.16, tandis que la deuxième est parfaitement identique au cas de la caractéristique 0. □

Ce résultat nous permet d'étendre les algorithmes de la partie 7.2 au cas de la caractéristique positive dans une certaine mesure. Les problèmes algorithmiques concernant les polynômes lacunaires en caractéristique positive sont souvent difficiles [39, 70, 72, 63, 9, 66]. En particulier, Jigguo Bi, Qi Cheng et J. Maurice Rojas ont récemment montré que le calcul des racines dans $\mathbb{F}_p$ des polynômes creux à une variable et à coefficients dans $\mathbb{F}_p$ est un problème NP-difficile sous réduction probabiliste [9]. Nous donnons ici des résultats positifs sous la forme d'algorithmes de complexité polynomiale qui contrastent avec les résultats cités précédemment. Nous allons voir que nos résultats sont en certain sens *les plus généraux possibles* en raison de la NP-difficulté du problème évoqué précédemment.

Pour les algorithmes qui suivent, le corps considéré est le corps fini à p^s éléments $\mathbb{F}_{p^s}$, où p est un nombre premier et s un entier naturel non nul. Ce corps est représenté, par analogie au cas de la caractéristique 0, sous la forme $\mathbb{F}_p[\xi]/\langle\varphi\rangle$ où φ est un polynôme irréductible unitaire de degré s à coefficients dans $\mathbb{F}_p$. Le corps $\mathbb{F}_{p^s}$ est vu comme un espace vectoriel de dimension s sur $\mathbb{F}_p$ et ses éléments sont représentés par des multiplets de taille s.

Théorème 7.18

Il existe un algorithme déterministe polynomial pour tester si un polynôme de la forme

$$f = \sum_{j=1}^{k} a_j X^{\alpha_j}(1+X)^{\beta_j} \in \mathbb{F}_{p^s}[X]$$

de degré au plus $(p-1)$ est le polynôme nul.

Démonstration : À nouveau, il n'y a que peu de changements à effectuer par rapport au théorème 7.10 qui est la version en caractéristique 0 de ce théorème. L'algorithme est quasiment le même. Si $uv \neq 0$, la seule différence consiste à remplacer le théorème 7.1 par son équivalent en caractéristique positive qui est le théorème 7.17. Quand $u = 0$ ou $v = 0$, l'algorithme en caractéristique 0 fait appel à l'algorithme de Lenstra [84]. Il n'existe pas d'équivalent de cet algorithme en caractéristique positive,

mais il se trouve que le problème devient beaucoup plus simple.

En effet, si $u = 0$ par exemple, l'algorithme de Lenstra est utilisé pour tester si une somme de la forme $\sum_j a_j v^{\beta_j}$ est nulle. On ne peut pas calculer explicitement cette somme en caractéristique 0 car elle fait intervenir des éléments de trop grande taille. En caractéristique positive, un tel calcul explicite est possible en utilisant l'exponentiation rapide. La somme peut alors être calculée de manière explicite en temps polynomial en $\sum_j \log(\beta_j)$, c'est-à-dire en temps polynomial en la taille de l'entrée. □

On s'intéresse maintenant au calcul des facteurs linéaires de polynômes lacunaires à deux variables. Selon la forme des facteurs recherchés, soit le problème se résout en temps polynomial (probabiliste), soit il est NP-difficile.

Théorème 7.19

Soit

$$f(X, Y) = \sum_{j=1}^{k} a_j X^{\alpha_j} Y^{\beta_j} \in \mathbb{F}_{p^s}[X, Y]$$

de degré au plus $(p-1)$. Il existe un algorithme probabiliste polynomial pour trouver les facteurs de f de la forme $(uX + vY + w)$ avec $uvw \neq 0$.

Par contre, trouver les facteurs linéaires de f de la forme $(uX + w)$ est NP-difficile, de même que trouver les facteurs de la forme $(vY + w)$ ou de la forme $(uX + vY)$.

Démonstration : La deuxième partie du théorème est une conséquence de la NP-difficulté de trouver les racines des polynômes creux à une variable et à coefficients dans $\mathbb{F}_{p^s}$ [72, 9, 66]. En effet, considérons un polynôme creux g à une variable X et à coefficients dans $\mathbb{F}_{p^s}$. On peut définir $f(X, Y) = g(X)$. Alors f est bien de la même forme que dans l'énoncé avec $\beta_j = 0$ pour tout j. Or les facteurs de f de la forme $(X - x_0)$ sont les racines x_0 de g. Le problème est donc NP-difficile. De même, trouver les facteurs de la forme $(vY + w)$ est NP-difficile. Enfin, pour le dernier type de facteurs, on associe à g le polynôme homogène associé $f(X, Y) = Y^{\deg(g)} g(X/Y)$. Alors $f(x_0 Y, Y) = Y^{\deg(f)} f(x_0, 1) = Y^{\deg(f)} g(x_0)$, donc $(X - x_0 Y)$ est facteur de f si et seulement si x_0 est une racine de g. Ce qui prouve la NP-difficulté du problème.

Pour les facteurs de la forme $(uX + vY + w)$ avec $uvw \neq 0$, l'algorithme est le même qu'en caractéristique 0 (qui n'utilise pas l'algorithme de Lenstra [84] si $uvw \neq 0$). La seule différence réside dans l'algorithme de factorisation dense utilisé. En caractéristique 0, il existe des algorithmes déterministes polynomiaux pour effectuer cette tâche, alors qu'en caractéristique positive, les seuls algorithmes connus sont probabilistes. Ceci signifie que notre algorithme est également probabiliste. □

Bibliographie

[1] M. Agrawal et R. Saptharishi, « Classifying Polynomials and Identity Testing », *Current Trends in Science*, 2009, p. 149–162 (cf. p. 106).

[2] M. Agrawal et V. Vinay, « Arithmetic circuits : A chasm at depth four », *Proc. FOCS*, 2008, p. 67–75 (cf. p. 105–106).
doi : 10.1109/FOCS.2008.32.

[3] M. Agrawal, C. Saha, R. Saptharishi et N. Saxena, « Jacobian hits circuits : hitting-sets, lower bounds for depth-D occur-k formulas & depth-3 transcendence degree-k circuits », *Proc. STOC*, 2012, p. 599–614 (cf. p. 107).
doi : 10.1145/2213977.2214033.
arXiv : 1111.0582.

[4] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen et P. B. Miltersen, « On the Complexity of Numerical Analysis », *SIAM J. Comput.* 38(5), 2009, p. 1987–2006 (cf. p. 121).
doi : 10.1137/070697926.
eccc : TR05-037.

[5] S. Arora et B. Barak, *Computational Complexity : A Modern Approach*, 1re éd., Camb. U. Press, 2009, isbn : 978-0521424264 (cf. p. 11, 27, 36).

[6] J. Balcázar, « The complexity of searching implicit graphs », *Artificial Intelligence* 86(1), 1996, p. 171–188 (cf. p. 34).
doi : 10.1016/0004-3702(96)00014-8.

[7] J. Balcázar, A. Lozano et J. Torán, « The complexity of Algorithmic Problems on Succinct Instances », *Computer Science : Research and Applications*, 1992, p. 351–377 (cf. p. 34).

[8] M. Beecken, J. Mittmann et N. Saxena, « Algebraic Independence and Blackbox Identity Testing », *Information and Computation*, 2012, à paraître (cf. p. 106–107).
doi : 10.1016/j.ic.2012.10.004.
arXiv : 1102.2789.

[9] J. Bi, Q. Cheng et J. M. Rojas, *Sub-Linear Root Detection, and New Hardness Results, for Sparse Polynomials Over Finite Fields*, manuscrit, 2012 (cf. p. 125, 145–146).
arXiv : 1204.1113.

[10] L. Blum, M. Shub et S. Smale, « On a theory of computation and complexity over the real numbers : NP-completeness, recursive functions and universal machines », *Bull. Am. Math. Soc.* 21(1), 1989, p. 1–46 (cf. p. 11).

[11] L. Blum, F. Cucker, M. Shub et S. Smale, *Complexity and Real Computation*, Springer, 1997, isbn : 978-0387982817 (cf. p. 11).

[12] A. Borodin et S. Cook, « On the number additions to compute specific polynomials », *SIAM J. Comput.* 5(1), 1976, p. 146–157 (cf. p. 106).
doi : 10.1137/0205013.

[13] A. Bostan et P. Dumas, « Wronskians and linear independence », *Am. Math. Mon.* 117(8), 2010, p. 722–727 (cf. p. 126).
doi : 10.4169/000298910X515785.

[14] P. Brändén, « Obstructions to determinantal representability », *Adv. Math.* 226(2), 2011, p. 1202 –1212 (cf. p. 64).
doi : 10.1016/j.aim.2010.08.003.
arXiv : 1004.1382.

[15] P. Bürgisser, « On defining integers and proving arithmetic circuit lower bounds », *Comput. Complex.* 18(1), 2009, p. 81–103 (cf. p. 104, 108, 117–118).
doi : 10.1007/s00037-009-0260-x.
eccc : TR06-113.

[16] P. Bürgisser, F. Cucker et P. De Naurois, « The complexity of semilinear problems in succinct representation », *Comput. Complex.* 15(3), 2006, p. 197–235 (cf. p. 34).
doi : 10.1007/s00037-006-0213-6.

[17] P. Bürgisser, *Completeness and Reduction in Algebraic Complexity Theory*, Algorithms Comput. Math. Springer, 2000, isbn : 978-3540667520 (cf. p. 13).

[18] P. Bürgisser, M. Clausen et M. A. Shokrollahi, *Algebraic Complexity Theory*, t. 315, Grundlehren Math. Wiss. Springer, 1997, isbn : 978-3540605829 (cf. p. 106).

[19] L. Busé et C. D'Andrea, « On the irreducibility of multivariate subresultants », *CR Math.* 338(4), 2004, p. 287–290 (cf. p. 20, 33).
doi : 10.1016/j.crma.2003.12.019.
arXiv : math/0309374.

[20] J.-Y. Cai, X. Chen et D. Li, « Quadratic Lower Bound for Permanent Vs. Determinant in any Characteristic », *Comput. Complex.* 19(1), 2010, p. 37–56 (cf. p. 60–61).
DOI : 10.1007/s00037-009-0284-2.

[21] J. F. Canny, « Some algebraic and geometric computations in PSPACE », *Proc. STOC*, 1988, p. 460–469 (cf. p. 19).
DOI : 10.1145/62212.62257.

[22] J. F. Canny, *The complexity of robot motion planning*, t. 1987, ACM Doctoral Dissertation Award, MIT Press, 1988, ISBN : 978-0262031363 (cf. p. 19–21, 33–34).

[23] J. F. Canny, E. Kaltofen et L. Yagati, « Solving systems of nonlinear polynomial equations faster », *Proc. ISSAC*, 1989, p. 121–128 (cf. p. 19).
DOI : 10.1145/74540.74556.

[24] J. F. Canny et J. H. Reif, « New Lower Bound Techniques for Robot Motion Planning Problems », *Proc. FOCS*, 1987, p. 49–60 (cf. p. 19).
DOI : 10.1109/SFCS.1987.42.

[25] E. Cattani et A. Dickenstein, « Introduction to residues and resultants », *Solving polynomial equations*, 2005, p. 1–61 (cf. p. 19–20, 33).
DOI : 10.1007/3-540-27357-3_1.

[26] A. Chattopadhyay, B. Grenet, P. Koiran, N. Portier et Y. Strozecki, *Factoring bivariate lacunary polynomials without heights*, manuscrit, 2012 (cf. p. 125).
arXiv : 1206.4224 [cs.CC].

[27] Q. Cheng, « Straight-line programs and torsion points on elliptic curves », *Comput. Complex.* 12(3-4), 2003, p. 150–161 (cf. p. 104).
DOI : 10.1007/s00037-003-0180-0.

[28] D. Coppersmith et S. Winograd, « Matrix multiplication via arithmetic progressions », *J. Symb. Comput.* 9(3), 1990, p. 251–280 (cf. p. 60).
DOI : 10.1016/S0747-7171(08)80013-2.

[29] L. Csanky, « Fast Parallel Matrix Inversion Algorithms », *SIAM J. Comput.* 5(4), 1976, p. 618–623 (cf. p. 37).
DOI : 10.1137/0205040.

[30] F. Cucker, P. Koiran et S. Smale, « A polynomial time algorithm for Diophantine equations in one variable », *J. Symb. Comput.* 27(1), 1999, p. 21–30 (cf. p. 123–124).
DOI : 10.1006/jsco.1998.0242.

[31] C. D'Andrea et A. Dickenstein, « Explicit formulas for the multivariate resultant », *J. Pure Appl. Algebra* 164(1-2), 2001, p. 59–86 (cf. p. 20, 33).
DOI : 10.1016/S0022-4049(00)00145-6.
arXiv : math/0007036.

[32] R. DeMillo et R. Lipton, « A probabilistic remark on algebraic program testing », *Inf. Process. Lett.* 7(4), 1978, p. 193–195 (cf. p. 28, 106).
DOI : 10.1016/0020-0190(78)90067-4.

[33] A. Dixon, « The eliminant of three quantics in two independent variables », *P. Lond. Math. Soc.* 6, 1908, p. 468–478 (cf. p. 19).
DOI : 10.1112/plms/s2-7.1.49.

[34] I. Emiris et B. Mourrain, « Matrices in elimination theory », *J. Symb. Comput.* 28(1-2), 1999, p. 3–43 (cf. p. 19).
DOI : 10.1006/jsco.1998.0266.

[35] J. Feigenbaum, S. Kannan, M. Y. Vardi et M. Viswanathan, « The Complexity of Problems on Graphs Represented as OBDDs », *Chicago J. Theor. Comput. Sci.* 1999(5), 1999 (cf. p. 34).

[36] H. Galperin et A. Wigderson, « Succinct representations of graphs », *Inform. Control* 56(3), 1983, p. 183–198 (cf. p. 34).
DOI : 10.1016/S0019-9958(83)80004-7.

[37] S. Gao, « Factoring multivariate polynomials via partial differential equations », *Math. Comput.* 72(242), 2003, p. 801–822 (cf. p. 137).
DOI : 10.1090/S0025-5718-02-01428-X.

[38] M. R. Garey et D. S. Johnson, *Computers and Intractability : A Guide to the Theory of* NP*-Completeness*, Ser. Books Math. Sci. W. H. Freeman, 1979, ISBN : 978-0716710455 (cf. p. 23).

[39] J. von zur Gathen, M. Karpinski et I. Shparlinski, « Counting curves and their projections », *Comput. Complex.* 6(1), 1996, p. 64–99 (cf. p. 21, 26, 145).
DOI : 10.1007/BF01202042.

[40] D. Glynn, « The permanent of a square matrix », *European J. Combin.* 31(7), 2010, p. 1887–1891 (cf. p. 60).
DOI : 10.1016/j.ejc.2010.01.010.

[41] B. Grenet, *An Upper Bound for the Permanent versus Determinant Problem*, manuscrit (soumis), 2012 (cf. p. 42).
URL : http://perso.ens-lyon.fr/bruno.grenet/publis/PermVsDet.pdf.

[42] B. Grenet, E. L. Kaltofen, P. Koiran et N. Portier, « Symmetric Determinantal Representation of Formulas and Weakly Skew Circuits », *Randomization, Relaxation, and Complexity in Polynomial Equation Solving*, Contemp. Math. 556, 2011, p. 61–96 (cf. p. 42, 64, 77).
DOI : 10.1090/conm/556.
arXiv : 1007.3804.

[43] B. Grenet, E. L. Kaltofen, P. Koiran et N. Portier, « Symmetric Determinantal Representation of Weakly-Skew Circuits », *Proc. STACS*, 2011, p. 543–554 (cf. p. 64).
DOI : 10.4230/LIPIcs.STACS.2011.543.

[44] B. Grenet, P. Koiran et N. Portier, « On the complexity of the multivariate resultant », *J. Complex.* 2012, à paraître (cf. p. 22).
doi : 10.1016/j.jco.2012.10.001.
arXiv : 1210.1451 [cs.CC].

[45] B. Grenet, P. Koiran et N. Portier, « The Multivariate Resultant is NP-hard in Any Characteristic », *Proc. MFCS*, 2010, p. 477–488 (cf. p. 22).
doi : 10.1007/978-3-642-15155-2_42.
arXiv : 0912.2607.

[46] B. Grenet, P. Koiran, N. Portier et Y. Strozecki, « The Limited Power of Powering : Polynomial Identity Testing and a Depth-four Lower Bound for the Permanent », *Proc. FSTTCS*, 2011, p. 127–139 (cf. p. 103).
doi : 10.4230/LIPIcs.FSTTCS.2011.127.
arXiv : 1107.1434 [cs.CC].

[47] B. Grenet, T. Monteil et S. Thomassé, *Symmetric determinantal representations in characteristic 2*, manuscrit (soumis), 2012 (cf. p. 74, 77).
arXiv : 1210.5879 [cs.CC].

[48] D. Y. Grigor'ev, « Lower bounds in algebraic complexity », *Notes of Scientific Seminars of LOMI* 118, 1982, en russe, traduction anglaise : [49], p. 25–82 (cf. p. 106).

[49] D. Y. Grigor'ev, « Lower Bounds in Algebraic Computational Complexity », *J. Soviet. Math.* 29, 1985, p. 1388–1425.

[50] G. Hajós, « [Solution to problem 41] (in Hungarian) », *Mat. Lapok* 4, 1953, p. 40–41 (cf. p. 125).

[51] J. Heintz et J. Morgenstern, « On the intrinsic complexity of elimination theory », *J. Complex.* 9, 1993, p. 471–498 (cf. p. 21–23).
doi : 10.1006/jcom.1993.1031.

[52] J. Heintz et C.-P. Schnorr, « Testing polynomials which are easy to compute », *Logic and Algorithmic (an International Symposium held in honour of Ernst Specker)*, t. 30, Monograph. Enseign. Math. 1982, p. 237–254 (cf. p. 106).

[53] J. W. Helton, S. A. McCullough et V. Vinnikov, « Noncommutative convexity arises from linear matrix inequalities », *J. Funct. Anal.* 240(1), 2006, p. 105–191 (cf. p. 64, 68).
doi : 10.1016/j.jfa.2006.03.018.

[54] J. Helton et V. Vinnikov, « Linear matrix inequality representation of sets », *Comm. Pure Appl. Math.* 60(5), 2006, p. 654–674 (cf. p. 64).
doi : 10.1002/cpa.20155.
arXiv : math.OC/0306180.

[55] P. Hrubeš et A. Yehudayoff, « Arithmetic Complexity in Ring Extensions », *Th. Comput.* 7(1), 2011, p. 119–129 (cf. p. 68).
DOI : 10.4086/toc.2011.v007a008.

[56] D. Ierardi, « Quantifier elimination in the theory of an algebraically-closed field », *Proc. STOC*, 1989, p. 138–147 (cf. p. 19).
DOI : 10.1145/73007.73020.

[57] M. Jansen, « Lower Bounds for Syntactically Multilinear Algebraic Branching Programs », *Proc. MFCS*, 2008, p. 407–418 (cf. p. 46).
DOI : 10.1007/978-3-540-85238-4_33.

[58] G. Jeronimo et J. Sabia, « Computing multihomogeneous resultants using straight-line programs », *J. Symb. Comput.* 42(1-2), 2007, p. 218–235 (cf. p. 20, 33).
DOI : 10.1016/j.jsc.2006.03.006.

[59] V. Kabanets et R. Impagliazzo, « Derandomizing polynomial identity tests means proving circuit lower bounds », *Comput. Complex.* 13(1), 2004, p. 1–46 (cf. p. 106).
DOI : 10.1007/s00037-004-0182-6.
ECCC : TR02-055.

[60] E. Kaltofen, « Polynomial factorization : a success story », *Proc. ISSAC*, 2003, p. 3–4 (cf. p. 123).
DOI : 10.1145/860854.860857.

[61] E. Kaltofen et P. Koiran, « Expressing a fraction of two determinants as a determinant », *Proc. ISSAC*, 2008, p. 141–146 (cf. p. 20, 33, 46).
DOI : 10.1145/1390768.1390790.

[62] E. Kaltofen et P. Koiran, « Finding small degree factors of multivariate supersparse (lacunary) polynomials over algebraic number fields », *Proc. ISSAC*, 2006, p. 162–168 (cf. p. 123).
DOI : 10.1145/1145768.1145798.

[63] E. Kaltofen et P. Koiran, « On the complexity of factoring bivariate supersparse (lacunary) polynomials », *Proc. ISSAC*, 2005, p. 208–215 (cf. p. 21, 26, 47, 123–124, 134, 138, 143–145).
DOI : 10.1145/1073884.1073914.

[64] E. Kaltofen, « Polynomial-Time Reductions from Multivariate to Bi- and Univariate Integral Polynomial Factorization », *SIAM J. Comput.* 14(2), 1985, p. 469–489 (cf. p. 137).
DOI : 10.1137/0214035.

[65] E. Kaltofen et G. Villard, « On the Complexity of Computing Determinants », *Comput. Complex.* 13(3-4), 2005, p. 91–130 (cf. p. 60).
DOI : 10.1007/s00037-004-0185-3.

[66] E. L. Kaltofen et G. Lecerf, « Factorization of Multivariate Polynomials », *Handbook of Finite Fields*, Disc. Math. Appl. à paraître, 2013 (cf. p. 125, 145–146).

[67] I. Kaplansky, *An introduction to differential algebra*, Actualités scientifiques et industrielles, Hermann, 1976, isbn : 978-2705612511 (cf. p. 144).

[68] D. Kapur et T. Saxena, « Comparison of Various Multivariate Resultant Formulations », *Proc. ISSAC*, 1995, p. 187–194 (cf. p. 20, 33).
doi : 10.1145/220346.220370.

[69] D. Kapur, T. Saxena et L. Yang, « Algebraic and geometric reasoning using Dixon resultants », *Proc. ISSAC*, 1994, p. 99–107 (cf. p. 19).
doi : 10.1145/190347.190372.

[70] M. Karpinski et I. Shparlinski, « On the Computational Hardness of Testing Square-Freeness of Sparse Polynomials », *Proc. AAECC*, 1999, p. 492–497 (cf. p. 145).
doi : 10.1007/3-540-46796-3_47.
eccc : TR99-027.

[71] N. Kayal et C. Saha, « On the Sum of Square Roots of Polynomials and Related Problems », *Proc. CCC*, 2011, p. 292–299 (cf. p. 126, 141).
doi : 10.1109/CCC.2011.19.
eccc : TR10-189.

[72] A. Kipnis et A. Shamir, « Cryptanalysis of the HFE public key cryptosystem by relinearization », *Proc. CRYPTO*, 1999, p. 19–30 (cf. p. 125, 145–146).
doi : 10.1007/3-540-48405-1_2.

[73] P. Koiran, « Arithmetic circuits : the chasm at depth four gets wider », *Theor. Comput. Sci.* 448, 2012, p. 56–65 (cf. p. 105–106).
doi : 10.1016/j.tcs.2012.03.041.
arXiv : 1006.4700.

[74] P. Koiran, « Circuits versus trees in algebraic complexity », *Proc. STACS*, 2000, p. 35–54 (cf. p. 28).
doi : 10.1007/3-540-46541-3_3.

[75] P. Koiran, « Hilbert's Nullstellensatz is in the polynomial hierarchy », *J. Complex.* 12(4), 1996, p. 273–286 (cf. p. 23, 26).
doi : 10.1006/jcom.1996.0019.

[76] P. Koiran, « Shallow circuits with high-powered inputs », *Proc. ICS*, 2011 (cf. p. 104–108, 116, 121).
arXiv : 1004.4960.

[77] P. Koiran et S. Perifel, « VPSPACE and a Transfer Theorem over the Reals », *Comput. Complex.* 18(4), 2009, p. 551–575 (cf. p. 33).
DOI : 10.1007/s00037-009-0269-1.
arXiv : cs/0610009.

[78] P. Koiran, N. Portier et S. Tavenas, *A Wronskian approach to the real τ-conjecture*, manuscrit, 2012 (cf. p. 107, 121, 126).
arXiv : 1205.1015.

[79] S. Lang, *Algebra*, 3e éd., Springer, 2002, ISBN : 978-0387953854 (cf. p. 20, 33).

[80] P. D. Lax, « Differential equations, difference equations and matrix theory », *Comm. Pure Appl. Math.* 11(2), 1958, p. 175–194 (cf. p. 64).
DOI : 10.1002/cpa.3160110203.

[81] D. Lazard, « Résolution des systèmes d'équations algébriques », *Theor. Comput. Sci.* 15(1), 1981, p. 77 –110 (cf. p. 19).
DOI : 10.1016/0304-3975(81)90064-5.

[82] G. Lecerf, « Improved dense multivariate polynomial factorization algorithms », *J. Symb. Comput.* 42(4), 2007, p. 477–494 (cf. p. 137).
DOI : 10.1016/j.jsc.2007.01.003.

[83] A. K. Lenstra, « Factoring Multivariate Polynomials over Algebraic Number Fields », *SIAM J. Comput.* 16(3), 1987, p. 591–598 (cf. p. 137).
DOI : 10.1137/0216040.

[84] H. Lenstra Jr, « Finding small degree factors of lacunary polynomials », *Number theory in progress*, 1999, p. 267–276 (cf. p. 123–125, 134–137, 145–146).

[85] H. Lenstra Jr, « On the factorization of lacunary polynomials », *Number theory in progress*, 1999, p. 277–291 (cf. p. 124).

[86] A. Lewis, P. Parrilo et M. Ramana, « The Lax conjecture is true », *Proc. Am. Math. Soc.* 133(9), 2005, p. 2495–2500 (cf. p. 64).
DOI : 10.1090/S0002-9939-05-07752-X.
arXiv : math.OC/0304104.

[87] T. Li, J. Rojas et X. Wang, « Counting real connected components of trinomial curve intersections and m-nomial hypersurfaces », *Discrete Comput. Geom.* 30(3), 2003, p. 379–414 (cf. p. 109).
DOI : 10.1007/s00454-003-2834-8.
arXiv : math/0212178.

[88] H. Liu et K. Regan, « Improved construction for universality of determinant and permanent », *Inf. Process. Lett.* 100(6), 2006, p. 233–237 (cf. p. 41, 44, 53).
DOI : 10.1016/j.ipl.2006.05.017.

[89] S. Lovett, « Computing Polynomials with Few Multiplications », *Th. Comput.* 7(1), 2011, p. 185–188 (cf. p. 68).
doi : 10.4086/toc.2011.v007a013.
eccc : TR11-094.

[90] A. Lozano et J. Balcázar, « The complexity of graph problems for succinctly represented graphs », *Proc. WG*, 1989, p. 277 (cf. p. 34–36).
doi : 10.1007/3-540-52292-1_20.

[91] F. S. Macaulay, *The Algebraic Theory of Modular Systems*, Camb. Math. Lib. Camb. U. Press, 1916, isbn : 978-0521455626 (cf. p. 34).

[92] F. Macaulay, « Some Formulæ in Elimination », *P. Lond. Math. Soc.* 1(1), 1902, p. 3 (cf. p. 19).
doi : 10.1112/plms/s1-35.1.3.

[93] M. Mahajan et V Vinay, « Determinant : Combinatorics, algorithms, and complexity », *Chicago J. Theor. Comput. Sci.* 5(1997), 1997, p. 730–738 (cf. p. 54–57).
eccc : TR97-036.

[94] M. Mahajan, P. R. Subramanya et V Vinay, « The combinatorial approach yields an NC algorithm for computing Pfaffians », *Discrete Appl. Math.* 143(1–3), 2004, p. 1–16 (cf. p. 54).
doi : 10.1016/j.dam.2003.12.001.

[95] M. Mahajan et V Vinay, « Determinant : Old Algorithms, New Insights », *SIAM J. Discrete Math.* 12(4), 1999, p. 474–490 (cf. p. 54).
doi : 10.1137/S0895480198338827.
eccc : TR98-012.

[96] G. Malod, « Polynômes et coefficients », thèse de doct., Université Claude Bernard Lyon, 2003 (cf. p. 13–15, 41, 48).

[97] G. Malod et N. Portier, « Characterizing Valiant's algebraic complexity classes », *J. Complexity* 24(1), 2008, p. 16–38 (cf. p. 15, 41, 48).
doi : 10.1016/j.jco.2006.09.006.

[98] G. Malod, « Succinct Algebraic Branching Programs Characterizing Non-uniform Complexity Classes », *Proc. FCT*, 2011, p. 205–216 (cf. p. 33).
doi : 10.1007/978-3-642-22953-4_18.

[99] T. Mignon et N. Ressayre, « A quadratic bound for the determinant and permanent problem », *Int. Math. Res. Notices* 2004(79), 2004, p. 4241 (cf. p. 60).
doi : 10.1155/S1073792804142566.

[100] H. Montgomery et A. Schinzel, « Some arithmetic properties of polynomials in several variables », *Transcendence Theory : Advances and Applications*, 1977, chap. 13, p. 195–203 (cf. p. 125).

[101] T. Netzer, D. Plaumann et A. Thom, « Determinantal representations and the Hermite matrix », *Mich. Math. J.* 2012, à paraître (cf. p. 64).
arXiv : 1108.4380.

[102] T. Netzer et A. Thom, « Polynomials with and without determinantal representations », *Linear Algebra Appl.* 437(7), 2012, p. 1579–1595 (cf. p. 64).
DOI : 10.1016/j.laa.2012.04.043.
arXiv : 1008.1931.

[103] C. Papadimitriou et M. Yannakakis, « A note on succinct representations of graphs », *Inform. Control* 71(3), 1986, p. 181–185 (cf. p. 34).
DOI : 10.1016/S0019-9958(86)80009-2.

[104] M. Petkovšek, H. S. Wilf et D. Zeilberger, *A=B*, AK Peters, 1996, ISBN : 978-1568810638 (cf. p. 132).

[105] K. Phillipson et J. M. Rojas, « Fewnomial systems with many roots, and an adelic τ conjecture », *Proc. of Bellairs Workshop on tropical and non-Archimedean geometry*, CRM Monograph, à paraître, 2012 (cf. p. 122).
arXiv : 1011.4128.

[106] D. A. Plaisted, « New NP-hard and NP-complete polynomial and integer divisibility problems », *Theor. Comput. Sci.* 31(1-2), 1984, p. 125–138 (cf. p. 21, 24).
DOI : 10.1016/0304-3975(84)90130-0.

[107] R. Quarez, « Symmetric determinantal representation of polynomials », *Linear Algebra Appl.* 436(9), 2012, p. 3642–3660 (cf. p. 64, 68–70).
DOI : 10.1016/j.laa.2012.01.004.
HAL : hal-00275615.

[108] J. Renegar, « On the Worst-Case Arithmetic Complexity of Approximating Zeros of Systems of Polynomials », *SIAM J. Comput.* 18, 1989, p. 350 (cf. p. 19).
DOI : 10.1137/0218024.

[109] J.-J. Risler, « Additive complexity and zeros of real polynomials », *SIAM J. Comput.* 14(1), 1985, p. 178–183 (cf. p. 106).
DOI : 10.1137/0214014.

[110] H. J. Ryser, *Combinatorial Mathematics*, t. 14, Carus Math. Monogr. Math. Assoc. Am., 1963, ISBN : 978-0883850145 (cf. p. 60).

[111] N. Saxena, « Progress on Polynomial Identity Testing », *Bull. EATCS* 99, 2009, p. 49–79 (cf. p. 106).
ECCC : TR09-101.

[112] J. T. Schwartz, « Fast probabilistic algorithms for verification of polynomials identities », *J. Ass. Comput. Mach.* 27, 1980, p. 701–717 (cf. p. 28, 106).
DOI : 10.1145/322217.322225.

[113] A. Seidenberg, « A new decision method for elementary algebra », *Ann. Math.* 60(2), 1954, p. 365–374 (cf. p. 19).

[114] V. Shoup, « New algorithms for finding irreducible polynomials over finite fields », *Math. Comput.* 54(189), 1990, p. 435–447 (cf. p. 28, 31).
DOI : 10.1090/S0025-5718-1990-0993933-0.

[115] M. Shub et S. Smale, « On the intractability of Hilbert's nullstellensatz and an algebraic version of "P = NP" », *Duke Math. J.* 81(1), 1995, p. 47–54 (cf. p. 103–104).
DOI : 10.1215/S0012-7094-95-08105-8.

[116] A. Sommese et C. Wampler, *The Numerical Solution of Systems of Polynomials, Arising in Engineering And Science*, World Scientific, 2005, ISBN : 978-9812561848 (cf. p. 28).

[117] W. Stein et al., *Sage Mathematics Software (Version 4.5.3)*, http://www.sagemath.org, 2010 (cf. p. 74).

[118] A. Stothers, « On the complexity of matrix multiplication », thèse de doct., The University of Edinburgh, 2010 (cf. p. 60).

[119] B. Sturmfels, « Sparse Elimination Theory », *Proc. Comput. Algebr. Geom. Commut. Algebra*, 1991 (cf. p. 19).

[120] S. Toda, « Classes of arithmetic circuits capturing the complexity of computing the determinant », *IEICE T. Inf. Syst.* 75(1), 1992, p. 116–124 (cf. p. 15, 41).

[121] J. Torán, « Succinct representations of counting problems », *Proc. AAECC*, 1989, p. 415–426 (cf. p. 34).
DOI : 10.1007/3-540-51083-4_77.

[122] L. Valiant, « Completeness classes in algebra », *Proc. STOC*, 1979, p. 249–261 (cf. p. 13–15, 41, 44, 117).
DOI : 10.1145/800135.804419.

[123] B. L. van der Waerden, *Modern Algebra*, trad. par F. Blum, 2e éd., F. Ungar Publishing Co., New York, 1949 (cf. p. 19–20, 33).

[124] K. Wagner, « The complexity of combinatorial problems with succinct input representation », *Acta Inform.* 23(3), 1986, p. 325–356 (cf. p. 34).
DOI : 10.1007/BF00289117.

[125] V. Williams, « Multiplying matrices faster than Coppersmith-Winograd », *Proc. STOC*, 2012, p. 887–898 (cf. p. 60).
DOI : 10.1145/2213977.2214056.

[126] R. Zippel, « Probabilistic algorithms for sparse polynomials », *Proc. EUROSAM*, 1979, p. 216–226 (cf. p. 28, 106).
DOI : 10.1007/3-540-09519-5_73.