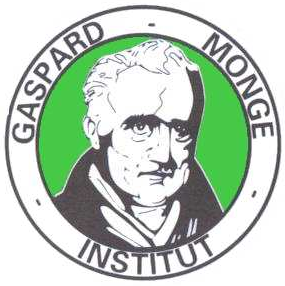

Université de Marne-la-Vallée

Habilitation à diriger des recherches

Spécialité : Informatique

présentée par

Frédérique Bassino

sur le sujet

Automates, Énumération et Algorithmes

Soutenue le 6 Décembre 2005 devant le jury composé de :

Jean Berstel (IGM, Université de Marne-la-Vallée)
Mireille Bousquet-Mélou (LaBRI)
Philippe Flajolet (INRIA-Roquencourt), *Rapporteur*
Christiane Frougny (LIAFA, Université Paris VIII), *Rapporteur*
Dominique Perrin (IGM, Université de Marne-la-Vallée), *Directeur*
Antonio Restivo (Università di Palermo), *Rapporteur*

Remerciements

Je tiens à remercier :

– Dominique Perrin pour tout ce qu'il m'a appris depuis que j'ai commencé mon DEA à l'Université Paris VII en 1992,
– Philippe Flajolet pour sa disponibilité, ses précieux conseils dispensés au bord d'une piscine en Crète, et surtout de m'avoir fait l'honneur d'être rapporteur,
– Christiane Frougny d'avoir accepté de remplir cette fonction, que la langue française n'accorde pas au féminin,
– Antonio Restivo d'avere accettato di scrivere un rapporto sul mio lavoro, di avere affrontato il rigore dell'inverno pariggino per assistere alla discussione di questa tesi e di avermi sempre accolta molto calorosamente nel suo laboratorio di Palermo,
– Jean Berstel, qui déjà membre de mon jury de thèse, a de nouveau accepté de faire partie de celui de mon habilitation, pour ses questions stimulantes et ses encouragements,
– Mireille Bousquet-Mélou, qui délaissant la combinatoire du baton, m'a fait le plaisir de particper au jury,
– mes co-auteurs, Shigeki Akiyama, Marie-Pierre Béal, Christiane Frougny, Dominique Perrin, Helmut Prodinger, Gadiel Seroussi, Alfredo Viola, et plus particulièrement Julien Clément et Cyril Nicaud qui ont la malchance de me supporter quotidiennement,
– Maxime Crochemore de m'avoir chaleureusement accueillie à mon arrivée à l'Institut Gaspard-Monge, qu'il dirigeait jusque récemment,
– Gilles Roussel pour avoir eu la délicatesse de me demander chaque jour que compte le mois de septembre, si j'avais fini d'écrire le présent document,
– Line Fonfrède pour son aide et sa grande disponibilité, pour sa contribution à la réussite des Journées Montoises et de l'école jeunes chercheurs en algorithmique qui se sont à l'Institut Gaspard-Monge,
– Patrice Hérault pour son assistance technique en temps réel,
– Toutes celles et ceux, que je ne citerais pas de peur d'en oublier, qui contribuent à faire régner la bonne humeur au sein de l'Institut Gaspard-Monge,
– La communauté ALEA pour son accueil chaleureux et ses très enrichissantes réunions annuelles.

Table des matières

1 Introduction **3**
- 1.1 Mots finis et infinis . . . 4
- 1.2 Automates finis . . . 6
- 1.3 Dynamique symbolique . . . 7
- 1.4 Théorie de Perron-Frobenius . . . 9
- 1.5 Séries génératrices . . . 10

2 Codes préfixes **13**
- 2.1 Théorème de Kraft-McMillan . . . 13
  - 2.1.1 Inégalité de Kraft . . . 13
  - 2.1.2 Une version régulière du théorème de Kraft-McMillan . . . 14
  - 2.1.3 Séries génératrices de langages préfixiels réguliers . . . 17
- 2.2 Codage de Huffman . . . 18
  - 2.2.1 Sources finies . . . 19
  - 2.2.2 Sources infinies . . . 19

3 Séquences lexicographiques **25**
- 3.1 Représentation des nombres en base réelle . . . 25
  - 3.1.1 Le β-shift . . . 26
  - 3.1.2 β-réseaux et ensembles de Meyer . . . 29
- 3.2 Mots de Lyndon . . . 32
  - 3.2.1 Mots de Lyndon ayant un facteur droit donné . . . 35
  - 3.2.2 Étude en moyenne de la factorisation standard . . . 36

4 Énumération et génération aléatoire d'automates **41**
- 4.1 Énumération . . . 41
  - 4.1.1 Formule d'énumération exacte . . . 42
  - 4.1.2 Estimation asymptotique . . . 43
- 4.2 Génération aléatoire . . . 45

Chapitre 1

Introduction

Ce document est une présentation des travaux que j'ai réalisés depuis mon doctorat qui portait sur les séries régulières et les distributions de longueurs de codes.

Ces travaux s'inscrivent dans le cadre général de la théorie des automates, de la combinatoire des mots, de la combinatoire énumérative et de l'algorithmique. Ils ont en commun de traiter des automates et des langages réguliers, de problèmes d'énumération et de présenter des résultats constructifs, souvent explicitement sous forme d'algorithmes. Les domaines dont sont issus les problèmes abordés sont assez variés.

Ce texte est composé de trois parties consacrées aux codes préfixes, à certaines séquences lexicographiques et à l'énumération d'automates.

La première partie porte sur des problèmes de codage sans perte d'information, et plus particulièrement sur les codes préfixes qui permettent un décodage instantané. On y présente une version régulière du théorème de Kraft-McMillan, obtenue en collaboration avec M.-P. Béal et D. Perrin. On expose ensuite des constructions de codes préfixes pour des sources infinies selon un procédé généralisant l'algorithme de Huffman. Ces dernières sont le fruit d'un travail réalisé avec J. Clément, G. Seroussi et A. Viola.

Le chapitre suivant traite de suites de symboles finies ou infinies définies sur un alphabet totalement ordonné et caractérisées par des conditions lexicographiques. On y examine des problèmes issus de la numération dans une base réelle strictement supérieure à 1, les résultats obtenus sont d'une part la caractérisation de certains systèmes dynamiques de type fini, d'autre part la construction effective d'ensembles finis liés à l'addition de nombres analogues aux nombres entiers en base entière. Ce dernier résultat, lié au problème de la modélisation des quasicristaux, a été obtenu en collaboration avec S. Akiyama et C. Frougny. On étudie ensuite un autre type de suites définies par des conditions lexicographiques, les mots de Lyndon, qui servent à la construction de bases pour le monoïde libre, le groupe libre ou les

algèbres de Lie libres. Leur factorisation standard joue un rôle essentiel dans la plupart des algorithmes dans lesquels ils sont utilisés. Nous avons, avec J. Clément et C. Nicaud, étudié cette opération et caractérisé les mots de Lyndon ayant un facteur standard droit fixé. Nous proposons également une modélisation des mots de Lyndon permettant d'obtenir des résultats en moyenne.

Le dernier chapitre a pour objet les automates déterministes et accessibles. Ces automates sont assez proches des automates minimaux associés aux langages réguliers. Avec C. Nicaud, nous nous sommes intéressés à l'énumération de tels objets, ainsi qu'à des méthodes pour les engendrer aléatoirement de manière équiprobable. Ces résultats sont liés à l'étude de propriétés en moyenne des langages réguliers.

Ce mémoire ne comprend pas une description exhaustive de l'état de l'art dans les différents domaines abordés, mais devrait permettre de situer les résultats que j'ai obtenus seule ou en collaboration.

Le présent chapitre d'introduction est destiné à un lecteur qui ne serait pas familier avec les automates finis, la dynamique symbolique ou les séries génératrices. On y trouvera des définitions classiques d'objets utilisés dans la suite, quelques résultats fondamentaux sont également rappelés.

1.1 Mots finis et infinis

Soit A un ensemble de symboles, les lettres, que l'on appelle alphabet (sauf mention contraire, on supposera que cet ensemble est fini). Un mot fini w est une suite finie d'éléments de A notée $w_0 w_1 \cdots w_{n-1}$, sa longueur $|w|$ est le nombre n de lettres qui composent le mot w. L'ensemble de tous les mots finis sur l'alphabet A, noté A^*, muni de l'opération de concaténation est un *monoïde libre* dont l'élément neutre est le mot vide ε. On note A^+ le *semigroupe libre* formé par l'ensemble des mots non vides sur l'alphabet A.

L'ensemble des suites infinies à droite de lettres de A est noté $A^{\mathbb{N}}$. Un mot infini w est dit *ultimement périodique* s'il s'écrit sous la forme $w = pz^\omega$, où p et z sont des mots finis et $z^\omega = zzz\cdots$.

Un *bloc* ou *facteur* f d'un mot w est un mot fini non vide qui apparaît dans le mot. Si $w = ps$ où p est un mot non vide, le mot p est un *préfixe* de w ; si, de plus, p est différent de w alors p est un préfixe *propre* de w. De manière analogue, si s est un mot non vide, alors s est un suffixe de w ; si, de plus, le mot s est différent de w alors s est suffixe *propre* de w.

On rappelle qu'un ordre *lexicographique* sur le semigroupe libre A^+ est donné par l'extension d'un ordre total, $<_{lex}$, sur l'alphabet A à l'ensemble des mots de la manière suivante : pour tous mots finis non vides u et v, $u <_{lex} v$ si et seulement

si u est un préfixe propre de v ou les deux mots s'écrivent

$$u = ras, \quad v = rbt \qquad \text{avec } a, b \in A,\ r, s, t \in A^* \text{ et } a <_{lex} b.$$

On définit ainsi un ordre total sur A^+, noté simplement $<$ en l'absence d'ambiguïté.

Cet ordre vérifie les deux propriétés suivantes :

(i) Pour tout mot w de A^*, $u < v$ si et seulement si $wu < wv$.

(ii) Soient $u, v \in A^*$ deux mots tels que $u < v$. Si u n'est pas un préfixe de v alors pour tous $u', v' \in A^*$, on a $uu' < vv'$.

Sur le vaste domaine que constitue la combinatoire des mots, le lecteur intéressé pourra consulter [106, 107, 108].

Les *langages* désignent des ensembles de mots. Une classe de langages structurellement simples est constituée par les *langages réguliers* qui peuvent être définis par des opérations rationnelles ou par des automates finis (voir Section 1.2). Plus précisément, l'union étant entendue au sens ensembliste et en définissant le produit de deux langages L_1 et L_2 comme l'ensemble des concaténations des mots de L_1 et de L_2 et l'étoile (de Kleene) L^* d'un langage L comme l'union des concaténations finies des mots de L, un langage sur l'alphabet A est régulier s'il peut être obtenu à partir des langages finis de A^* par un nombre fini d'unions, de produits et d'étoiles. Ces langages peuvent être décrits par des expressions rationnelles à partir des symboles de l'alphabet et des opérateurs $\cdot, +, *$ correspondant respectivement au produit, à l'union et à l'étoile sur les langages.

On s'intéressera dans la suite aux langages particuliers que sont les codes et, spécialement, aux codes préfixes. On rappelle qu'un *code* sur A^* est un ensemble de mots non vides tel que tout mot de w de A^* se décompose au plus d'une manière en un produit d'éléments du code. Un ensemble de mots est *préfixe* si aucun de ses éléments n'est le préfixe d'un autre. Un tel ensemble est un code appelé un *code préfixe*.

Les codes préfixes sur A^* sont en bijection avec les langages préfixiels (*i.e*, qui contiennent tous les préfixes de leurs éléments) non vides de A^*. En effet, si C est un code préfixe sur A^*, alors $P = A^* \setminus CA^*$ est un langage préfixiel non vide. Réciproquement, si P est un langage préfixiel non vide, $C = PA \setminus P$ est un code préfixe.

Les codes préfixes et les langages préfixiels admettent des représentation simples sous forme d'arbres. Si l'alphabet totalement ordonné A est de taille k, A^* peut être représenté par un arbre k-aire dont les nœuds sont étiquetés par les mots de A^* de telle sorte que le parcours en profondeur de tout sous-arbre corresponde à un parcours dans l'ordre lexicographique sur les mots de A qui apparaissent dans les nœuds du sous-arbre.

À tout langage de A^*, on associe le sous-arbre obtenu en conservant tous les chemins menant de la racine de l'arbre associé à A^* aux nœuds étiquetés par les

mots du langage. Ce sous-arbre est appelé la *représentation littérale* du langage. Dans ces condition, un langage C est un code préfixe si et seulement si, dans la représentation littérale de C, les mots de C sont les étiquettes des feuilles. Le langage préfixiel associé à C est alors l'ensemble des étiquettes des nœuds internes de la représentation littérale de C. Inversement, un langage est préfixiel si et seulement si toutes les étiquettes des nœuds de sa représentation littérale sont des mots du langage. Le code préfixe associé à un ensemble préfixiel P est alors l'ensemble des étiquettes des nœuds qui permettent de compléter la représentation littérale de P de telle sorte que les feuilles deviennent des nœuds internes. Quand le langage est régulier, sa repésentation littérale n'a qu'un nombre fini de sous-arbres non isomorphes, elle est *régulière*.

Pour une référence générale sur les codes, on reportera à [34].

1.2 Automates finis

Les *langages réguliers* sont aussi les langages acceptés par un automate fini. Un *automate fini* sur l'alphabet A est un multigraphe orienté dont les transitions sont étiquetées par une lettre de l'alphabet A. On appellera par la suite *graphe* un multigraphe orienté. Un automate est représenté, selon le contexte, par un quadruplet (I, Q, E, F) ou simplement par un couple (Q, E), où Q est l'ensemble des états de l'automate, E l'ensemble des transitions, I l'ensemble des états initiaux et F celui des états finaux. On notera (p, a, p') la transition de l'état p à l'état p' étiquetée par la lettre a. Un automate est fini lorsque son nombre d'états et son nombre de transitions sont finis. Les chemins d'un automate sont les suites de transitions consécutives. L'ensemble des étiquettes de ces chemins forme le langage *reconnu* ou *accepté* par l'automate.

Un automate est dit *non-ambigu* si deux chemins qui partent d'un même état, arrivent sur un même état et ont même étiquette, sont égaux. Un automate est *déterministe* si, à partir d'un état donné, il a au plus une transition avec une étiquette donnée. Il est dit *déterministe complet* pour un certain alphabet si de chaque état part exactement une transition d'étiquette donnée dans l'alphabet. Il est dit *accessible* (ou *initialement connexe*), si tout état peut être atteint par un chemin issu d'un état initial. L'automate *minimal* d'un langage rationnel est l'automate déterministe complet ayant le moins d'états qui reconnaît ce langage. Un automate est dit *local*, ou *défini*, s'il existe deux entiers positifs ou nuls m et a (m pour mémoire et a pour anticipation) tels que tous les chemins de longueur $(m + a)$ de même étiquette arrivent dans le même état de l'automate. Il s'agit d'une propriété de confluence des chemins à un instant donné. Lorsque l'automate est déterministe local, la propriété est satisfaite avec une anticipation nulle. Les automates déterministes locaux ont été introduits en [124] où ils étaient appelés

automates définis.

Enfin, les *transducteurs* sont des automates finis étiquetés sur $A^* \times B^*$, où A et B sont deux alphabets. On considère dans la suite des transducteurs étiquetés sur $A \times B$. Chaque transition a ainsi une lettre comme étiquette d'entrée et une lettre comme étiquette de sortie. En masquant les sorties, respectivement les entrées, on obtient un automate fini.

Pour plus de résultats sur les automates, on pourra se reporter à [56, 83, 133].

1.3 Dynamique symbolique

Pour une introduction à la dynamique symbolique, on pourra consulter [101, 24].

On associe à une suite infinie $(a_n)_{n\in\mathbb{N}}$ sa suite décalée définie par $\sigma((a_n)_{n\in\mathbb{N}}) = (a_{n+1})_{n\in\mathbb{N}}$. Il s'agit ici d'un décalage du mot vers la gauche. Les parties fermées de $A^{\mathbb{N}}$ invariantes par décalage sont appelées *sous-systèmes* en dynamique symbolique. La topologie dont est muni $A^{\mathbb{N}}$ est celle induite par exemple par la distance d qui vaut 0 pour deux suites identiques et est définie comme suit pour deux suites distinctes $a = (a_n)_{n\in\mathbb{N}}$ et $b = (b_n)_{n\in\mathbb{N}}$:

$$d(a, b) = 2^{-l} \text{ où } l = \min\{i \mid a_i \neq b_i\}.$$

Les *facteurs autorisés* d'un sous-système sont les blocs finis pouvant apparaître comme facteurs d'un mot infini du système. Les *facteurs interdits* sont ceux qui ne sont pas autorisés, c'est-à-dire ceux qui n'apparaîtront jamais comme facteurs d'un mot infini du système. On distinguera aussi les *mots minimaux interdits* qui sont des blocs interdits dont tous les facteurs propres sont autorisés.

La *capacité* de Shannon, appelée encore *entropie topologique*, d'un sous-système S de $A^{\mathbb{N}}$, permet de mesurer le taux de croissance des facteurs autorisés, elle est définie par :

$$\mathrm{Cap}(S) = \limsup_{n\to\infty} \frac{1}{n} \log \mathrm{Card}(S_n \cap A^n),$$

où S_n désigne l'ensemble des facteurs autorisés de longueur n du système. Les logarithmes sont en général pris en base 2 et le nombre de blocs autorisés de longueur n est ainsi bien approché par $2^{n\,\mathrm{Cap}(S)}$ pour n assez grand.

Classes de systèmes dynamiques

Une classe assez générale de systèmes dynamiques liés aux automates est celle des systèmes codés [39]. Ces systèmes, reconnus par des automates infinis dénombrables [89], sont ceux pour lesquels il existe un code C tel que l'ensemble des mots du système soit C^ω.

Une sous-classe des systèmes codés est constituée par les ensembles de suites infinies acceptés par un automate fini et appelés *systèmes sofiques*. Ils ont été introduits par Weiss [143]. Il est à noter qu'il n'est pas précisé d'ensemble d'états initiaux ou terminaux dans cette définition. Tous les chemins infinis sont acceptants. Les systèmes sofiques peuvent être reconnus par des automates déterministes, c'est-à-dire des automates tels que chaque état admet au plus une transition sortante d'étiquette donnée. Ce sont des sous-systèmes au sens défini précédemment. Les systèmes qui peuvent être reconnus par un automate dont le graphe est fortement connexe sont appelés *systèmes irréductibles*. Ils sont reconnus par un automate déterministe minimal, encore appelé *couverture de Fisher* du système. Le lien avec l'automate minimal déterministe qui accepte un langage de mots fini avec un unique état initial et un ensemble d'états terminaux est étroit (voir [25], [24], et [27]). Une représentation minimale unique n'existe cependant plus lorsque le système n'est pas irréductible.

Les systèmes qui peuvent être reconnus par des automates locaux sont appelés *systèmes de type fini*. Ils sont caractérisés par le fait que leur ensemble de mots interdits minimaux est un ensemble fini.

Éclatements d'états

L'opération dite d'éclatement d'états, entrant ou sortant, est une transformation fondamentale en dynamique symbolique. Sur un automate fini, l'éclatement entrant consiste en une partition des transitions entrant dans un état et une duplication des transitions qui en sortent. On peut définir de façon symétrique la notion d'éclatement sortant. L'opération inverse d'un éclatement est appelée une *fusion* ou une *amalgamation*.

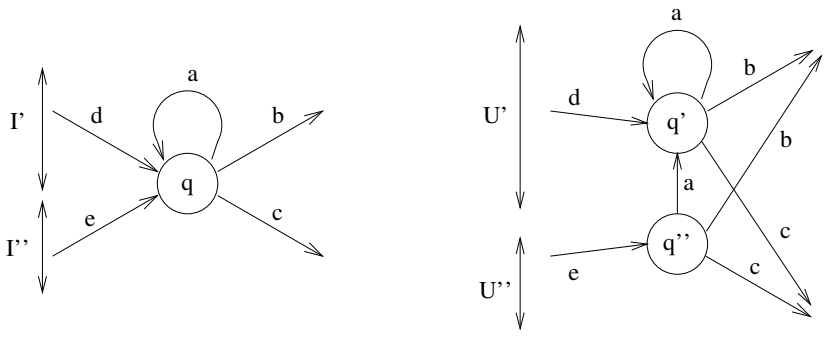

FIG. 1.1 – Un éclatement entrant

Plus précisément, soient $\mathcal{A} = (Q, E)$ un automate, q un des états de l'automate et I l'ensemble des transitions entrant en q. Soit $I = I' + I''$ une partition de I.

L'opération d'*éclatement entrant* relativement à la partition (I', I'') transforme l'automate $\mathcal{A}$ en un automate $\mathcal{B} = (Q', E')$, où $Q' = (Q - \{q\}) \cup \{q'\} \cup \{q''\}$, est obtenu en "éclatant" l'état q en deux états q', q'', et où les transitions définies par :

- Les transitions non incidentes à q sont inchangées.
- Les transitions qui étaient entrantes en q dans $\mathcal{A}$ sont partagées selon la partition (I', I''). Les ensembles U' des transitions entrant en q' et U'' des transitions entrant en q'' sont respectivement $U' = \{(p, x, q') \mid (p, x, q) \in I'\}$ et $U'' = \{(p, x, q'') \mid (p, x, q) \in I''\}$.
- Les états q' et q'' ont les "mêmes transitions" sortantes dupliquées de celles de q. Si O désigne l'ensemble des transitions sortant de q dans $\mathcal{A}$, les ensembles O' des transitions sortant de q' et O'' des transitions sortant de q'' sont alors respectivement $O' = \{(q', x, p) \mid (q, x, p) \in O\}$ et $O'' = \{(q'', x, p) \mid (q, x, p) \in O\}$.

Un éclatement entrant transforme un automate déterministe en un automate déterministe qui reconnaît le même système sofique. Bien que l'automate obtenu par éclatement d'état semble différent de l'automate initial, les systèmes dynamiques définis par les étiquettes des chemins dans chacun des automates sont conjugués, *i.e* il existe un application locale inversible qui transforme le premier système en le second. De plus, les systèmes dynamiques définis par les étiquettes des chemins dans deux automates sont conjugués si et seulement si on peut passer d'un automate à l'autre par une suite d'éclatements et d'amalgamations entrants ou sortants. Ce résultat est dû à Williams [144]. La décidabilité de la conjugaison entre deux systèmes, même de type fini, reste un problème ouvert.

1.4 Théorie de Perron-Frobenius

La théorie de Perron-Frobenius sur les matrices positives (*i.e.*, les matrices à coefficients réels positifs ou nuls) intervient en théorie des automates. On l'applique aux matrices d'adjacence des graphes des automates finis. Ces matrices sont indexées sur les lignes et les colonnes par les états de l'automate. Si p et q sont deux états de l'automate, le coefficient d'indice p, q de la matrice d'adjacence est le nombre de transitions de l'automate allant de p vers q. Les matrices d'adjacence irréductibles sont celles qui correspondent aux automates dont le graphe est fortement connexe.

Le théorème de Perron-Frobenius [73] pour les matrices irréductibles à coefficients positifs ou nuls énonce que ces matrices admettent une plus grande valeur propre réelle, appelée *rayon spectral*, qui est une valeur propre simple. L'espace propre associé est de dimension 1 et la direction propre admet un vecteur propre à coefficients strictement positifs. Lorsque la matrice possède plusieurs composantes irréductibles, son rayon spectral est la valeur maximale des rayons spectraux des

différentes composantes fortement connexes. Parmi les matrices irréductibles, les *matrices primitives*, ou *apériodiques*, sont celles qui n'admettent pas de valeur propre de module égal au rayon spectral autre que le rayon spectral lui-même. Les matrices apériodiques ont comme propriété remarquable le fait que la direction propre est attractive pour les vecteurs en ce sens que si un vecteur $\mathbf{v}$ n'est pas dans l'espace supplémentaire à la direction propre de la matrice M dans la décomposition de Jordan, la direction de la suite de vecteurs $(\mathbf{v}M^n)_{n\in\mathbb{N}}$ converge vers la direction propre. Différentes preuves du théorème de Perron-Frobenius ainsi que de nombreuses applications sont données en [111].

On peut montrer que la capacité d'un système sofique est calculable à partir d'un automate non-ambigu qui le reconnaît. Cette capacité est en effet égale à $\log(\lambda)$, où λ est le rayon spectral de la matrice d'adjacence de cet automate. Ce résultat est dû à Shannon [140].

1.5 Séries génératrices

Les séries génératrices ou fonctions génératrices jouent un rôle essentiel dans l'énumération d'objets combinatoires simples (mots, arbres, partitions ...). Elles sont définies pour une classe $\mathcal{O}$ d'objets combinatoires sur laquelle est définie une taille :

$$\begin{array}{rl} \text{taille} : & \mathcal{O} \to \mathbb{N} \\ & O \to |O|. \end{array}$$

Si, pour tout entier n, le nombre s_n d'objets de taille n dans $\mathcal{O}$ est fini, la *série génératrice (ordinaire)* des objets de $\mathcal{O}$, énumérés selon leur taille, est alors la série formelle

$$s(z) = \sum_{n\geq 0} s_n z^n.$$

Beaucoup de problèmes d'énumération peuvent être résolus en transformant les opérations de base sur les ensembles en opérations algébriques sur les séries formelles (voir [61, Chapitre I]).

En particulier, la série s_X est la série génératrice d'un langage X de mots finis si s_n est le nombre de mots de longueur n du langage.

Les opérations rationnelles sur les langages s'expriment aisément en terme de séries génératrices. Ainsi, pour tous langages X et Y de A^*,

- Si $X \cap Y = \emptyset$, alors $s_{X\cup Y} = s_X + s_Y$.
- Si XY est un langage non-ambigu, alors $s_{XY} = s_X s_Y$.
- Si X est un code, alors $s_{X^*}(X) = \frac{1}{1-s_X(z)}$.

Pour un état de l'art sur les séries génératrices de langages réguliers, on pourra se reporter à [126, 16] et sur celles des langages algébriques à [41].

L'utilisation des séries génératrices permet par des techniques issues de la combinatoire analytique d'obtenir des résultats d'énumération asymptotique. Pour un panorama des méthodes et des résultats connus, on consultera [61].

Séries régulières

On utilisera, dans la suite, le terme *régulière* dans un contexte où une terminologie plus précise est souvent utilisée. En particulier, on appellera ici série régulière l'objet appelé, selon la terminologie d'Eilenberg, série $\mathbb{N}$-rationnelle (voir [56, 134, 35]).

Suivant le point de vue de Schützenberger, les langages peuvent être vus comme des séries en plusieurs variables non-commutatives. La série génératrice d'un langage est alors l'image de la série non-commutative par un homomorphisme. Les séries K-rationnelles, où K est un demi-anneau, en une ou plusieurs variables ont été étudiées comme extensions des langages rationnels ou réguliers [35, 134].

Une série $s = \sum_{n\geq 0} s_n z^n$ à coefficients entiers positifs est dite *régulière*, ou $\mathbb{N}$-*rationnelle*, si et si seulement si il existe un graphe fini G et deux ensembles de sommets du graphe I et T tels que, pour tout entier positif n, s_n est le nombre de chemins du graphe de longueur n partant d'un état de I et arrivant sur un état de T. On dit alors que l'automate ou la représentation (I, G, T) reconnaît la série.

Les séries régulières admettent des représentations dites *normalisées*, c'est-à-dire des représentations (I, G, T) ayant un unique état initial sans transition entrant dans cet état, et un unique état final sans transition sortant de cet état.

La définition suivante donne une description matricielle équivalente des séries régulières. Les séries régulières sont celles pour lesquelles il existe une matrice à coefficients entiers entiers positifs ou nuls M dont les lignes et les colonnes sont indicées dans un ensemble fini Q, un vecteur ligne $\mathbf{i}$ de taille $\mathrm{Card}(Q)$ et un vecteur colonne $\mathbf{t}$ de taille $\mathrm{Card}(Q)$ tels que, pour tout entier positif n, $s_n = \mathbf{i}M^n\mathbf{t}$. Lorsque les coefficients de $\mathbf{i}$, M, et $\mathbf{t}$ sont dans $\mathbb{Z}$, la série est dite $\mathbb{Z}$-*rationnelle* et $(\mathbf{i}, M, \mathbf{t})$ est une représentation linéaire sur $\mathbb{Z}$ de la série. La représentation est *réduite* si M a une taille minimale parmi toutes les représentations linéaires de la série sur le même anneau.

Tout série régulière est rationnelle, mais la réciproque n'est pas vraie. On dira qu'une suite $\mathbb{Z}$-rationnelle admet une *racine dominante* si elle admet une représentation linéaire réduite dont la matrice admet une unique valeur propre réelle positive ou nulle λ telle que $\lambda > |\mu|$ pour tout autre valeur propre μ. Lorsque, de plus, $\lambda > 0$ et λ est valeur propre simple, alors la matrice et la série sont dites *spectralement Perron*. Une série s est un *emboîtement* de séries s'il existe un entier

p tel que

$$s(z) = \sum_{i=0}^{p-1} z^i s_i(z^p),$$

où $(s_i)_{0 \leq i \leq p-1}$ sont elles-mêmes des séries. D'après le théorème de Soittola (voir [137, 87], [35, p. 83] ainsi que [125]), une série à coefficients positifs est régulière si et seulement si elle est l'emboîtement de séries rationnelles ayant une racine dominante. Ce résultat montre la décidabilité de la régularité d'une série rationnelle [134]. De plus, lorsque la réponse est positive, il existe un algorithme permettant de calculer une représentation de la série.

Chapitre 2

Codes préfixes

Ce chapitre porte sur les codes préfixes, qui ont l'avantage de permettre une procédure de décodage instantanée et de réaliser des taux de compression moyens optimaux dans les processus de codage utilisant des codes de longueur variable.

Plus précisément, on abordera le problème de la construction de codes préfixes réguliers dont la distribution de longueurs est donnée et de codes préfixes optimaux pour des sources émettant des symboles à valeurs dans un alphabet dénombrable selon une distribution de probabilités connue.

La première question est résolue par des techniques issues de la théorie des automates et de la dynamique symbolique qui nous ont permis, avec M.-P. Béal et D. Perrin, d'établir une version régulière du théorème de Kraft-McMillan.

On présente ensuite un état de l'art sur la généralisation du codage de Huffman à un ensemble dénombrable de symboles, ainsi que des résultats récemment obtenus en collaboration avec J. Clément, G. Seroussi et A. Viola sur le codage, par blocs, de symboles géométriquement distribués. Les méthodes utilisées proviennent de la théorie de l'information et de la combinatoire des mots.

2.1 Théorème de Kraft-McMillan

2.1.1 Inégalité de Kraft

La série génératrice $s = \sum_{n\geq 0} s_n z^n$ d'un code préfixe sur un alphabet à k lettres (ou des feuilles d'un arbre k-aire régulier) est la série dont le coefficient s_n d'ordre n est égal au nombre de mots de longueur n du code (ou au nombre de feuilles de hauteur n de l'arbre). Elle satisfait l'*inégalité de Kraft* pour l'entier positif k : $s(1/k) \leq 1$. Le nombre $s(1/k)$ peut alors être interprété comme la probabilité qu'un mot suffisamment long ait un préfixe dans le code.

La condition de Kraft pour une série est une condition d'entropie. En effet, si

C est un code préfixe, l'entropie de C^* est inférieure ou égale (resp. égale) à $\log(k)$ si et seulement si $s(1/k) \leq 1$ (resp. $s(1/k) = 1$).

D'après le théorème de Kraft-McMillan (voir [10, p.35] ou [47, p.82] par exemple), pour toute série s qui satisfait l'inégalité de Kraft pour l'entier k, il existe un code préfixe sur un alphabet à k lettres dont s est la série génératrice. Ce résultat peut être prouvé par induction. En effet, si un code préfixe C, formé de mots de longueur au plus $(n-1)$ et ayant comme distribution de longueurs $(s_1, s_2, \ldots, s_{n-1})$ sur l'alphabet $A = \{0, 1, \ldots, (k-1)\}$, a déjà été construit, alors, comme $\sum_{i=1}^{n} s_i k^{-i} \leq 1$, on a

$$\sum_{i=1}^{n} s_i k^{n-i} \leq k^n.$$

Autrement dit, on peut choisir s_n mots de longueur n sur l'alphabet A qui n'ont pas de préfixe dans C. Pour que la description de la construction soit complète, il reste à spécifier le choix, fait à chaque étape, parmi les mots de longueur n qui n'ont pas de préfixe dans C. Une possibilité consiste à préférer les mots les plus grands dans l'ordre lexicographique.

Dans le cas $s(1/k) = 1$ d'égalité dans l'inégalité de Kraft, le code préfixe C est *complet* (*i.e.*, tout mot sur l'alphabet A a un préfixe dans C ou est préfixe d'un mot de C). La réciproque n'est en générale pas vraie (voir [56, p.231] ou [102]). Par exemple, sur l'alphabet $\{0, 1\}$, le code préfixe

$$C = \left(\cup_{i \geq 2} \left(\prod_{n=2}^{i} \{0,1\}^n \setminus \{0^n\} \right) 0^{i+1} \right) \cup \{00\},$$

qui est complet, a pour série génératrice

$$s(z) = z^2 + \sum_{i \geq 2} \prod_{n=2}^{i} (2^n - 1) z^{i(i+3)/2}$$

et $s(1/2) < 1/2$ n'atteint pas l'égalité dans l'inégalité de Kraft. Néanmoins, la série génératrice s d'un code complet régulier satisfait $s(1/k) = 1$ [34, p.102].

2.1.2 Une version régulière du théorème de Kraft-McMillan

Nous nous sommes intéressés, avec M.-P. Béal et D. Perrin, à la caractérisation des séries génératrices des codes préfixes réguliers sur un alphabet à k lettres et avons obtenu le résultat suivant.

Théorème 1 *[15] Une série est la série génératrice d'un code préfixe régulier sur un alphabet à k lettres si et seulement si*

– *elle est régulière*
– *et satisfait l'inégalité de Kraft pour l'entier k : $s(1/k) \leq 1$.*

Autrement dit, les deux conditions nécessaires pour la série d'être la série génératrice d'un code régulier et d'un code préfixe sur un alphabet à k lettres sont indépendantes en ce sens que leur conjonction est suffisante pour garantir que la série est la série génératrice d'un code préfixe régulier sur un alphabet à k lettres. De plus, pour obtenir la série génératrice d'un code préfixe régulier complet sur un alphabet à k lettres, la série doit satisfaire l'égalité de Kraft.

La preuve fournit un algorithme pour obtenir un code préfixe régulier à partir d'une série satisfaisant les conditions précédentes.

Deux méthodes simples de construction viennent à l'esprit. La première consiste à appliquer directement la preuve du théorème de Kraft-McMillan. Mais le code ainsi obtenu n'est pas nécessairement régulier. Par exemple, si $s(z) = z^2/(1-2z^2)$, alors $s(1/2) = 1/2$, le code préfixe correspondant

$$C = \bigcup_{n \geq 0} 10^n 1\{0,1\}^n.$$

n'est pas cependant pas régulier.

La seconde technique s'appuie sur le fait que la série est régulière. Si s est une série régulière telle que $s_0 = 0$ et si (i, G, t) est une représentation normalisée de s, en étiquetant chaque transition partant d'un même état par une lettre différente, le langage reconnu par l'automate ainsi défini est un code préfixe régulier dont la série génératrice est s. Le problème ici est que le nombre de symboles utilisés pour étiqueter les transitions peut être supérieur à k comme le montre l'exemple suivant.

Soit s la série régulière reconnue par l'automate de gauche de la Figure 2.1 où $i = 1$ et $t = 4$. La série s s'écrit $s(z) = 3z^2/(1 - z^2)$ et satisfait l'égalité de Kraft pour $k = 2 : s(1/2) = 1$. Le sommet 2 est néanmoins d'arité 4 et la construction précédente conduit à un code préfixe sur un alphabet à 4 lettres. Une solution, dans ce cas, est représentée par l'automate de droite de la Figure 2.1, qui reconnaît la série s et, en étiquetant les transitions, le code préfixe régulier $(11)^*(00 + 01 + 10)$.

Dans sa version naïve, cette construction ne permet pas nécessairement d'obtenir un code sur un alphabet à k lettres, mais la solution est un raffinement de cette idée.

L'algorithme, que nous avons conçu, pour obtenir un code préfixe régulier sur un alphabet à k lettres est basé sur la construction d'un automate appelé *automate des multi-ensembles* et généralise l'algorithme de déterminisation d'un automate fini en ajoutant des multiplicités dans les états.

Cette transformation conserve la reconnaissabilité : la série reconnue par l'automate ainsi obtenu est la même que celle reconnue par l'automate initial. En

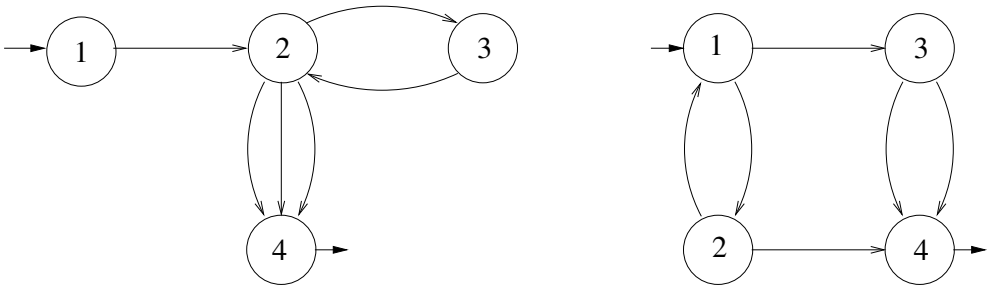

FIG. 2.1 – Automates qui reconnaissent $s(z) = 3z^2/(1-z^2)$

revanche, elle n'est pas dynamique, au sens où les automates construits par cette méthode définissent des systèmes dynamiques qui ne sont pas conjugués. Elle utilise néanmoins la théorie de Perron-Frobenius au moment de la construction de l'automate des multi-ensembles. Plus précisément, on calcule un *vecteur propre approché* associé à la valeur k pour la matrice d'une représentation de l'étoile de la série. Un tel vecteur propre pour une valeur k et une matrice M est, ici, un vecteur $\mathbf{v}$ à coefficients entiers positifs tel que $M\mathbf{v} \leq k\mathbf{v}$. Le vecteur propre approché ayant la plus petite composante maximale peut être calculé par un algorithme dû à Franaszek [101] dont la complexité en temps est exponentielle dans le pire des cas. Ce vecteur propre approché sert de guide pour garantir l'équilibrage des transitions de l'automate, c'est-à-dire son caractère k-aire, tout comme un vecteur propre approché similaire sert de guide dans l'algorithme d'éclatements d'états de Adler *et al.* [3] utilisé pour construire des codeurs pour canaux contraints (par exemple, dans lesquels les symboles 1 sont séparés par un nombre contraint supérieurement et inférieurement de 0). Notre preuve utilise également une variante du lemme des tiroirs [6, p.125], qui intervient également en [3], pour la construction des états de l'automate des multi-ensembles.

Béal et Perrin ont, depuis l'obtention de ce résultat, introduit dans [26] une notion plus générale d'induction entre représentations linéaires. Une représentation linéaire $(\mathbf{j}, N, \mathbf{x})$ est dite *induite à gauche* par la représentation $(\mathbf{i}, M, \mathbf{t})$ s'il existe une matrice U telle que

$$\begin{aligned} NU &= UM, \\ \mathbf{j}U &= \mathbf{i}, \\ \mathbf{x} &= U\mathbf{t}. \end{aligned}$$

La représentation $(\mathbf{i}, M, \mathbf{t})$ est dite *induite à droite* par $(\mathbf{j}, N, \mathbf{x})$.

Dans ce cadre, la construction de l'automate des multi-ensembles est une induction gauche, et la représentation linéaire finale de la série est obtenue par une induction droite. D'une manière plus générale, lorsque deux représentations

linéaires reconnaissent la même série, on peut passer d'une représentation à l'autre par une chaîne d'inductions gauches ou droites [26]. Du point de vue de la dynamique symbolique, l'induction gauche est une transformation plus faible qu'une conjugaison [101, 88] entre ces matrices.

Une construction différente, mais dynamique, que nous avons, avec M.-P. Béal et D. Perrin, décrite en [12, 14] redonne la caractérisation des séries génératrices des codes préfixes réguliers sur un alphabet à k lettres dans le cas où la série satisfait l'inégalité de Kraft stricte. Le cas d'égalité est cependant inaccessible par cette méthode et il est plus généralement inaccessible par des méthodes dynamiques.

2.1.3 Séries génératrices de langages préfixiels réguliers

On ne peut obtenir un résultat similaire pour les séries génératrices des langages préfixiels réguliers sur un alphabet à k lettres (ou des noeuds internes d'un arbre k-aire régulier). En effet, une série t est la série génératrice d'un langage préfixiel régulier [134, p.104] si et seulement si la série t est régulière, $t_0 = 1$ et t_n/t_{n-1} est uniformément borné sur $\mathbb{N}^*$. De plus, la série t est série génératrice d'un langage préfixiel sur un alphabet à k lettres si et seulement si $t_0 = 1$ et, pour tout entier n strictement positif, $t_n \leq kt_{n-1}$. Mais la conjonction de ces deux conditions nécessaires ne suffit pas à garantir que la série t sera la série génératrice d'un langage préfixiel régulier sur un alphabet à k lettres, comme le montrera l'exemple mentionné ci-après.

Nous avons, avec M.-P. Béal et D. Perrin, obtenu une caractérisation des séries génératrices des langages préfixiels réguliers sur un alphabet à k lettres en ajoutant aux conditions précédentes une condition calculable qui permet de se ramener au théorème obtenu pour les séries génératrices des codes préfixes.

Théorème 2 *[15] Soient $t = \sum_{n\geq 0} t_n z^n$ une série régulière et k un entier strictement positif. La série t est la série génératrice d'un langage préfixiel régulier sur un alphabet à k lettres si et seulement si*

(i) *son rayon de convergence est soit strictement supérieur à $1/k$, soit égal à $1/k$ et $1/k$ est alors un pôle simple de la série t,*

(ii) *et la série $s(z) = t(z)(kz - 1) + 1$ est régulière.*

La condition (ii) du Théorème 2 implique la positivité des coefficients de la série s et, par suite, que, pour tout entier n strictement positif, $t_n \leq kt_{n-1}$ et $t_0 \leq 1$.

Il existe des séries régulières t satisfaisant, pour tout $n \geq 1$, $t_n \leq kt_{n-1}$ et dont le rayon de convergence est strictement supérieur à $1/k$, mais telles que la série $s(z) = t(z)(kz - 1) + 1$ ne soit pas régulière. L'exemple qui suit est construit à

partir d'une série rationnelle à coefficients positifs qui n'est pas régulière [35, p.95]. Soit

$$r_n = b^{2n} cos^2(n\theta)$$

avec $cos(\theta) = \frac{a}{b}$ où les entiers a, b sont tels que $b \neq 2a$ et $0 < a < b$. La série r est rationnelle, à coefficients positifs et n'est pas régulière, ses pôles étant $\frac{1}{b^2}$, $\frac{1}{b^2}e^{2i\theta}$ et $\frac{1}{b^2}e^{-2i\theta}$. On suppose que $b^2 < k$ et on définit alors la série t de la manière suivante :

$$\text{pour tout } h \geq 0, \quad t_{2h} = k^h \quad \text{et} \quad t_{2h+1} = k^h + r_h.$$

D'après le théorème de Soittola (voir Section 1.5), la série t est régulière puisque elle est l'emboîtement de séries rationnelles ayant un pôle dominant. Son rayon de convergence est $\frac{1}{\sqrt{k}} > \frac{1}{k}$. Par suite, la série t satisfait la première condition du Théorème 2. Soit s la série définie par $s(z) = t(z)(kz - 1) + 1$. Comme, pour tout entier strictement positif p, $s_{2p} = kr_{p-1} + 1$, la série s n'est pas régulière.

En utilisant les inductions entre représentations linéaires mentionnées précédemment, Béal et Perrin [26] ont caractérisé, plus généralement, les séries génératrices des langages réguliers sur un alphabet à k lettres : ce sont les séries $s = \sum_{n\geq 0} s_n z^n$ telles que s et $\sum_{n\geq 0}(k^n - s_n)z^n$ soient régulières, la deuxième série est, en fait, la série génératrice du complémentaire du langage dont s est la série génératrice.

2.2 Codage de Huffman

On considère maintenant une source qui engendre une suite de variables aléatoires à valeurs dans un alphabet $\mathcal{S}$ indépendantes et identiquement distribuées pour une distribution discrète connue $(p_i)_{i\in\mathcal{S}}$. L'*entropie* de la source est alors $H = -\sum_{i\in\mathcal{S}} p_i \log(p_i)$.

Le problème auquel on s'intéresse est le suivant : trouver un code, parmi tous les codes de longueur variable, dont la longueur moyenne des mots est minimale. La longueur moyenne d'un code correspond au nombre moyen de bits nécessaires pour coder un symbole de la source. Cette minimisation permet de compresser [139, 145] sans perte d'information les suites de symboles issue de la source en codant par les mots les plus courts du code les symboles les plus probables et par les mots les plus longs les symboles les moins probables. Elle est aussi utilisée dans la conception d'algorithmes de recherche [4, 5], elle correspond alors à la minimisation du nombre de tests qui servent à identifier en moyenne des objets à partir d'un ensemble donné en utilisant moins de tests pour les objets les plus probables et plus de tests pour les objets les moins probables.

Quand la source est d'entropie finie, si $(l_i)_{i\in\mathcal{S}}$ est la distribution de longueurs

d'un code optimal alors :

$$H \leq \sum_{i \in \mathcal{S}} p_i l_i < H + 1.$$

La différence $\sum_{i \in \mathcal{S}} p_i l_i - H$ est appelée la *redondance* (par symbole) du code.

Pour plus de détails sur ces notions, on pourra consulter [10, 47].

2.2.1 Sources finies

Quand l'alphabet $\mathcal{S}$ des symboles émis par la source est fini, l'algorithme de Huffman [84] permet d'obtenir un code préfixe optimal. Après avoir associé à chaque symbole i de l'alphabet $\mathcal{S}$ un nœud de probabilité p_i, cet algorithme construit, de bas en haut, un arbre en fusionnant, à chaque étape, en une paire de nœuds frères les deux symboles les moins probables et en les remplaçant par leur père dont la probabilité est égale à la somme de celles de ses fils. La longueur moyenne des mots de tout code, ayant comme distribution de longueurs la suite des hauteurs des feuilles de l'arbre ainsi obtenu, est minimale parmi toutes celles des codes de longueur variable. En particulier, les *codes de Huffman* obtenus par étiquetage des chemins de la racine aux feuilles dans l'arbre de Huffman sont optimaux.

Gallager [71] a caractérisé les arbres de Huffman : un arbre est un arbre de Huffman si et seulement si ses nœuds peuvent être énumérés en ordre décroissant selon leurs probabilités de telle sorte que tous les nœuds frères soient adjacents dans cette liste. Cette propriété est utilisée pour vérifier si un arbre donné peut ou non être obtenu par l'algorithme de Huffman pour une distribution de probabilités donnée.

Pour un survol sur le codage de Huffman et une mine de références bibliographiques, on pourra se reporter à [2].

2.2.2 Sources infinies

On s'intéresse, dans la suite, au cas où la source émet des symboles à valeurs dans un alphabet dénombrable. On fait également l'hypothèse naturelle suivante : la distribution de probabilités sur les symboles de la source est décroissante (ou décroissante à partir d'un certain rang). On mentionnera dans la suite certaines situations où ce type de sources intervient.

Le but est de construire un code (préfixe) qui minimise parmi tous les codes de longueur variable, la longueur moyenne des mots du code, la distribution de probabilité des symboles de la source étant donnée. Comme l'algorithme du Huffman opère en fusionnant successivement les symboles les moins probables de l'alphabet des symboles émis par la source, il ne peut pas être appliqué directement à un alphabet infini.

Tous les résultats présentés ici utilisent la notion suivante de *convergence* d'une suite de codes : on dira qu'une suite $(C_j)_{j\geq 1}$ de codes converge vers un code C si, les mots des codes étant ordonnés par longueur croissante selon l'ordre lexicographique, pour tout $n \geq 1$, le n-ième mot de C_j est ultimement constant, quand j tend vers l'infini, et égal au n-ième mot de C.

Dans [102], Linder, Tarokh et Zeger ont montré que pour une source, émettant des symboles à valeurs dans $\mathbb{N}$, d'entropie finie, un code optimal peut être obtenu comme limite d'une sous-suite des codes de Huffman $(C_j)_{j\geq 1}$ pour les versions tronquées des variables aléatoires de la source (les variables aléatoires sont alors à valeurs dans $[\![0, j]\!]$ avec la distribution de probabilités $(p_i / \sum_{i=0}^{j} p_i)_{0\leq i\leq j}$). Ils montrent également que tout code optimal atteint l'égalité dans l'inégalité de Kraft. Les preuves ne permettent cependant pas la construction effective d'un code optimal.

La méthode, utilisée initialement par Gallager et Van Voorhis [72], qui sert à construire les codes optimaux évoqués ici peut être décrite de la manière suivante.

- Définir une suite d'alphabets réduits $(\mathcal{S}_j)_{j=-1}^{\infty}$ dans lesquels les symboles strictement plus grands que j sont partitionnés en un nombre fini de parts. Chaque part correspond à un ensemble non vide de symboles et est appelée *symbole virtuel*, sa probabilité est égale à la somme des probabilités des symboles qu'elle contient.
- Vérifier que la suite d'alphabets réduits $(\mathcal{S}_j)_{j=-1}^{\infty}$ est compatible avec le processus de construction de bas en haut d'un arbre de Huffman. Autrement dit, après un nombre fini de fusions dans l'algorithme de Huffman appliqué à l'alphabet réduit $\mathcal{S}_j$, on obtient l'alphabet réduit $\mathcal{S}_{j'}$ où $j' < j$. Une manière simple de vérifier que le processus de transformation correspond à l'algorithme de Huffman est d'utiliser la caractérisation des arbres de Huffman établie par Gallager [71] (voir Section 2.2.1).
- Appliquer l'algorithme de Huffman à l'alphabet réduit $\mathcal{S}_{-1}$ qui ne contient plus de symbole de l'alphabet initial, mais uniquement des symboles virtuels.
- Utiliser un argument de convergence pour s'assurer que la suite de codes de Huffman finis ainsi obtenue converge vers un code infini C.

Le code infini ainsi obtenu a une distribution de longueurs qui minimise la longueur moyenne parmi toutes celles des codes de longueur variable. La partie difficile de la construction est celle qui consiste à "deviner" la structure des alphabets réduits.

Codes de Golomb

Les sources dont les variables aléatoires sont à valeurs dans $\mathbb{N}$ avec la distribution de probabilités géométrique

$$p_i = (1-q)q^i \quad i \geq 0$$

pour un réel q, $0 < q < 1$, modélisent, par exemple, les sources qui émettent indépendamment les symboles 0 et 1, avec les probabilités respectives p et $1-p$, p_i est alors la probabilité que le mot 0^i1 soit émis par la source. Une stratégie de codage consiste dans ce cas à coder l'ensemble de suites de symboles $\{0^i1 \mid i \in \mathbb{N}\}$ émis par la source, plutôt que simplement $\{0, 1\}$. Golomb [76] a exhibé des codes préfixes optimaux quand $q^\ell = 1/2$ pour un entier positif ℓ quelconque. Gallager et Van Vooris [72] ont montré que les codes ainsi obtenus sont en fait optimaux pour toute valeur de q. Plus précisément, soit ℓ l'entier strictement positif tel que

$$q^\ell + q^{\ell+1} \leq 1 < q^\ell + q^{\ell-1}. \tag{2.1}$$

Un codage optimal simple consiste alors à représenter tout entier i comme la concaténation :

- du codage binaire de $(i \mod \ell)$ sur $\lfloor \log \ell \rfloor$ bits si $(i \mod \ell) < 2^{\lceil \log \ell \rceil} - \ell$, sur $\lceil \log \ell \rceil$ sinon,
- et du codage unaire de $\lfloor i/\ell \rfloor$, c'est-à-dire $0^{\lfloor i/\ell \rfloor}1$.

Ce code est obtenu comme limite de codes de Huffman. Comme ce code est la concaténation de deux codes préfixes, on peut inverser l'ordre des deux codes. Le codage de tout entier i est alors le codage unaire de $\lfloor i/\ell \rfloor$ suivi du codage de $(i \mod \ell)$ sur le nombre de bits adéquats. Ce code optimal, appelé *code de Golomb d'ordre* ℓ, ne peut être directement obtenu à partir de codes de Huffman. Dans le contexte du codage des plages de 0, ce code a l'avantage de ne pas nécessiter la lecture de la totalité de la plage de 0 pour commencer le codage ; l'encodeur produit un 0 pour chaque suite de ℓ symboles 0 lus, quand un symbole 1 apparaît, il produit un symbole 1 terminant ainsi le codage unaire, puis le codage correspondant à la position du 1 dans la suite ℓ symboles en cours de lecture. Les codes de Golomb pour les valeurs de q qui sont une puissance (négative) de 2 sont appelés les *codes de Rice* [130] et sont utilisés, par exemple, dans la librairie FLAC de compression de données audio sans perte d'information.

On peut noter que, pour tout entier strictement positif ℓ, le code de Golomb d'ordre ℓ contient exactement ℓ mots de longueur n dès que $n > \lceil \log \ell \rceil$ et que, de manière générale, les codes de Golomb sont des codes réguliers.

Par des méthodes combinatoires, Golin [75] a, par ailleurs, montré que la distribution de longueurs des codes optimaux pour les sources dont les variables aléatoires suivent une loi de probabilités géométrique est unique sauf quand $q^\ell + q^{\ell+1} = 1$ pour un entier positif ℓ, dans ce dernier cas il existe un ensemble infini de codes préfixes optimaux ayant des distributions de longueurs distinctes.

L'extension des résultats de Gallager et Van Voorhis [72] a permis

– à Humblet [85] de montrer qu'un code optimal pour les sources émettant des symboles à valeurs dans $\mathbb{N}$ avec une distribution de Poisson de paramètre λ,

$\lambda > 0$,

$$p_i = e^{-\lambda}\frac{\lambda^i}{i!} \quad i \geq 0,$$

pouvait être obtenu en utilisant, à partir d'une longueur qui dépend du paramètre λ, les mots du code de Golomb d'ordre 1 complété d'un ensemble fini de mots de longueurs inférieures,

– à Abrahams [1] d'établir que, pour une source émettant des symboles à valeurs dans $\mathbb{N}$ avec une distribution de probabilités $(p_i)_{i\geq 0}$ satisfaisant pour, tout $j \geq J$,

$$\sum_{i\geq 1} p_{j+1+i\ell} \leq p_j < \sum_{i\geq 1} p_{j-1+i\ell},$$

pour un entier strictement positif ℓ, l'ensemble des mots de longueur supérieure à J du code de Golomb d'ordre ℓ complété d'un ensemble fini de mots de longueurs inférieures constituait un code optimal.

Sources régulières

Une généralisation assez naturelle des sources émettant des symboles dont les valeurs suivent une loi de probabilités géométrique consiste à considérer des sources émettant des symboles dont les valeurs suivent une loi de probabilités donnée par une série $p(z) = \sum_{i\geq 0} p_i z^i$ $\mathbb{R}^+$-rationnelle.

Plus précisément, on suppose que la suite $(p_i)_{i\geq 0}$ des coefficients de la série p est une suite à termes réels positifs, décroissante et telle $\sum_{i\geq 0} p_i = 1$. On suppose également la série $\mathbb{R}^+$-rationnelle. D'après le théorème de Soittola, la série $p(z) = \sum_{i\geq 0} p_i z^i$ est alors l'emboîtement de r séries rationnelles $(s_i)_{0\leq i\leq r-1}$ ayant une racine dominante.

Nous avons montré [18], avec J. Clément, que, sous ces hypothèses, les r séries $(s_i)_{0\leq i\leq r-1}$ ont alors la même racine dominante $1/q$ avec le même ordre de multiplicité $m+1$ et que

$$p_i \sim c_{i \bmod r} i^m q^{i/r}$$

où les coefficients $(c_i)_{0\leq i\leq r-1}$ satisfont

$$c_0 \geq c_1 q^{1/r} \cdots \geq c_{r-1} q^{(r-1)/r} \geq c_0 q > 0.$$

Ainsi la série $p(z) = (2+z)/(4-z^2)$ est l'emboîtement des séries $s_0(z) = 2/(4-z)$ et $s_1(z) = 1/(4-z)$ qui ont comme racine dominante simple $1/q = 4$. De plus, $c_0 = 1/2$ et $c_1 = 1/4$ vérifient $c_0 \geq c_1\sqrt{q} \geq c_0\, q$.

Quand la série p a un unique pôle $1/q$ qui, de plus, est simple, elle définit la distribution de probabilités géométrique de paramètre q.

Si la série p a un unique pôle $1/q$ qui est multiple, et si ℓ est l'entier strictement positif tel que $q^\ell + q^{\ell+1} \leq 1 < q^\ell + q^{\ell-1}$, alors un code optimal [18] peut être obtenu

en utilisant à partir d'une certaine longueur, qui dépend de la multiplicité du pôle $1/q$ dans p, les mots du code de Golomb d'ordre ℓ et en complétant par un ensemble fini de mots de longueurs inférieures.

Quand la série p est l'emboîtement de deux séries ayant un unique pôle simple $1/q$, en posant, pour tout entier i, $p'_i = p_{2i}$ et, pour tout entier i strictement positif, $p'_{-i} = p_{2i-1}$, on obtient la distribution de probabilités

$$p_i = \frac{1-q}{q^{1-d}+q^d}\, q^{|i+d|}, \quad i \in \mathbb{Z},$$

où $0 \leq d \leq 1/2$, connue en théorie de l'information sous le nom *distribution géométrique bilatérale décentrée* [142]. Merhav, Seroussi et Weinberger [114] ont construit pour les sources qui émettent des symboles à valeurs dans $\mathbb{Z}$ selon une telle distribution de probabilités des codes optimaux qui sont des variantes non-intuitives des codes de Golomb. De nouveau, les codes obtenus ont un nombre uniformément borné de mots de même longueur et sont réguliers. Ces modèles de sources servent à la prédiction d'erreurs en compression d'image. Les codes optimaux, pour des valeurs de q qui sont une puissance (négative) de 2, sont utilisés dans l'algorithme LOCO-I (Lossless Image Compression Algorithm) qui est à la base de JPEG-LS [142].

Ce sont à ma connaissance les seuls résultats connus concernant les sources dont les variables aléatoires suivent une loi de probabilités donnée par une série $\mathbb{R}^+$-rationnelle, la construction, dans le cas général, de codes optimaux pour de telles sources reste un problème ouvert.

Sources géométriques et codage par blocs

Nous sommes intéressés, avec J. Clément, G. Seroussi et A. Viola, au problème du codage des mots de longueur fixée émis par une source.

En effet, si une source d'entropie H émet des symboles à valeurs dans un alphabet $\mathcal{S}$, l'alphabet des blocs de longueur n de symboles de la source est alors $\mathcal{S}^n$. En notant $(p'_i)_{i\in\mathcal{S}^n}$ la distribution de probabilités sur les éléments de $\mathcal{S}^n$, si $(l'_i)_{i\in\mathcal{S}^n}$ est la distribution de longueurs d'un code optimal pour $\mathcal{S}^n$, on obtient

$$H \leq \frac{1}{n}\sum_{i\in\mathcal{S}^n} p'_i l'_i < H + \frac{1}{n}.$$

Autrement dit, la redondance par symbole du codage par blocs de longueur n des symboles émis par la source est au plus égale à $1/n$ bit au lieu de la redondance maximale de 1 bit par symbole du code obtenu en codant chaque symbole de la source. Un tel codage améliore donc le taux de compression moyen des données issues de la source considérée.

Les résultats [22] que nous avons obtenus concernent les sources pour lesquelles les variables aléatoires sont à valeurs dans $\mathbb{N}$ et suivent une loi de probabilités géométrique :

$$p_i = (1-q)q^i, \quad i \in \mathbb{N}$$

pour un réel q, $0 < q < 1$.

Plus précisément, on souhaite coder les mots de longueur 2 émis par la source. L'alphabet peut alors être vu comme l'ensemble des couples d'entiers positifs (i, j) dont la probabilité d'émission est alors $p_i p_j = (1-q)^2 q^{i+j}$.

De manière équivalente, on cherche un codage optimal pour le multi-ensemble

$$\{0, 1, 1, 2, 2, 2, \ldots, \underbrace{f, \ldots, f}_{\substack{f+1 \\ \text{fois}}}, \ldots\}.$$

où l'entier positif f a $f+1$ occurrences sachant que le couple (i, j) est bijectivement associé à la i-ème occurrence du symbole $f = i + j$ dans le multi-ensemble. La distribution de probabilités d'une variable aléatoire sur ce multi-ensemble est donnée par $p_f = (1-q)^2 q^f$.

Dans [22], nous présentons des constructions de codes préfixes optimaux, à partir de codes de Huffman finis, quand q est une puissance (négative) de 2 et quand $q^\ell = 1/2$ pour un entier ℓ positif. Ces dernières suivent essentiellement le schéma déjà utilisé par Gallager et Van Voorhis et décrit au début de cette Section.

Quand $q = 1/2$, un code préfixe optimal peut être obtenu comme produit de deux codes de Golomb d'ordre 1. Plus précisément, le codage défini par la concaténation, pour tout couple (i, j), des mots du code de Golomb d'ordre 1 servant à coder respectivement i et j est optimal.

Pour les autres valeurs du paramètres q, les codes optimaux que nous définissons sont totalement différents des codes de Golomb. Ils sont encore réguliers, mais le nombre de mots de même longueur n'est plus uniformément borné ; il est, cette fois, majoré par une fonction linéaire de la longueur. De plus, il existe, dans ce cas, une infinité de codes préfixes optimaux ayant des distributions de longueurs distinctes.

Quand $q = 2^{-\ell}$ et que ℓ tend vers l'infini, contrairement au cas du codage par symbole, nous obtenons une suite infinie de codes, qui converge vers un code limite.

Nous donnons également, dans chaque cas, des algorithmes de codage et décodage efficaces dérivés de ces constructions.

La généralisation de nos constructions à toutes les valeurs du paramètre q ainsi que la construction de codes pour des blocs de symboles de longueur supérieure à 2 restent à faire.

Chapitre 3

Séquences lexicographiques

Les séquences de symboles abordées dans ce chapitre, quoique de nature très différentes, ont en commun d'être définies sur un alphabet totalement ordonné et d'être caractérisées par des conditions lexicographiques. Il m'a semblé intéressant de rapprocher l'étude de ces objets, qui nécessite dans les deux cas d'appréhender des propriétés combinatoires liées à la notion d'ordre.

Plus précisément, ce chapitre traite de la β-numération, système de numération qui généralise la numération positionnelle en base entière à une base réelle β strictement supérieure à 1, et de la factorisation standard des mots de Lyndon.

Les techniques utilisées sont, dans les deux cas, en partie issues de la combinatoire des mots. L'étude de la β-numération s'appuie également sur des résultats de théorie des automates et de théorie des nombres. L'étude de la factorisation des mots de Lyndon nécessite l'étude du comportement asymptotique des coefficients de séries génératrices par des méthodes issues de la combinatoire analytique.

3.1 Représentation des nombres en base réelle

Un système de numération de position peut être défini par la donnée d'une base ou d'une suite de nombres et d'un ensemble de digits. Pour une introduction générale sur le sujet et un survol des résultats connus, on pourra se reporter à [107, Chapter 7]. La β-numération généralise la numération en base entière à une base réelle β strictement supérieure à 1. Quand la base β n'est pas un entier, un nombre peut avoir plusieurs représentations, le système de numération associé est redondant [123]. On rend unique le choix de la représentation d'un nombre dans une telle base par l'ajout d'une condition lexicographique.

Plus précisément, soit $\beta > 1$ un nombre réel strictement supérieur à 1, le β-*développement* $\langle x \rangle_\beta$ d'un nombre réel positif x est la plus grande, pour l'ordre lexicographique, de ses représentations en base β.

Cette représentation est obtenue grâce à l'algorithme glouton suivant [128]. Soient $k \in \mathbb{Z}$ tel que $\beta^k \leq x < \beta^{k+1}$, $x_k = \lfloor x/\beta^k \rfloor$ et $r_k = \{x/\beta^k\}$, où $\{.\}$ représente la partie fractionnaire d'un nombre. Pour $i < k$, on note $x_i = \lfloor \beta r_{i+1} \rfloor$ et $r_i = \{\beta r_{i+1}\}$. Alors, le β-développement de x est

$$\langle x \rangle_\beta = x_k x_{k-1} \cdots x_1 x_0 \cdot x_{-1} x_{-2} \cdots$$

et le suffixe $x_{-1}x_{-2}\cdots$ en est la *partie β-fractionnaire*.

Quand β est un entier, le β-développement d'un nombre est sa représentation usuelle en base entière. Dans le cas contraire, les digits x_i sont les éléments de l'alphabet $A_\beta = \{0, \ldots, \lfloor \beta \rfloor\}$. Le β-développement de 1 en base β est 1.

3.1.1 Le β-shift

À l'origine, la notion de β-développements a été introduite par Rényi [128] qui les a définis à partir des orbites de la transformation croissante par morceaux de l'intervalle unité :

$$T_\beta : x \to \beta x \mod 1.$$

Tout nombre x de l'intervalle $[0, 1]$ est ainsi représenté par $d_\beta(x) = (d_i)_{i \geq 1}$, où $d_i = \lfloor \beta T_\beta^{i-1}(x) \rfloor$. Quand x est un nombre de l'intervalle $[0, 1[$, cette définition est équivalente à la précédente. En revanche, on obtient comme représentation de 1, non plus 1, mais la plus grande, pour l'ordre lexicographique, de ses représentations comme somme de puissances négatives de la base. On appellera, dans la suite, $d_\beta(1)$ la *β-représentation* de 1.

La clôture topologique $\mathcal{S}_\beta$ de l'ensemble des suites infinies qui sont les β-développements des nombres de l'intervalle $[0, 1[$, munie du décalage à gauche σ, forme un système dynamique symbolique (voir Section 1.3) appelé le *β-shift*.

Les éléments du β-shift sont entièrement caractérisés, d'un point de vue combinatoire, par la β-représentation de 1. En effet, une suite s d'entiers naturels appartient au β-shift si et seulement si, pour tout $p \geq 1$, la suite décalée $\sigma^p(s)$ est inférieure, dans l'ordre lexicographique, au développement impropre $d^*_\beta(1)$ [1] de $d_\beta(1)$; la suite s est le β-développement d'un réel de l'intervalle $[0, 1[$ si et seulement les précédentes inégalités sont strictes [123].

Ainsi, quand $\beta = \frac{1+\sqrt{5}}{2}$, comme $d_\beta(1) = 11$ et $d^*_\beta(1) = (10)^\omega$, les éléments du β-shift sont les mots qui ne contiennent pas le facteur 11 et $(10)^\omega$ n'est pas le β-développement d'un nombre de l'intervalle $[0, 1[$. Quand $\beta = \frac{3+\sqrt{5}}{2}$, alors $d_\beta(1) = 21^\omega = d^*_\beta(1)$ et le β-shift est composé de l'ensemble des mots qui n'ont pas de facteur dans l'ensemble 21^*2.

[1] $d^*_\beta(1) = d_\beta(1)$ si $d_\beta(1)$ est infini et $d^*_\beta(1) = (d_1 \ldots d_{m-1}(d_m - 1))^\omega$ si $d_\beta(1) = d_1 \ldots d_m$.

Un problème naturel est de chercher à classifier [38] les β-shifts en fonction de leurs propriétés dynamiques. Les β-shifts sont des *systèmes codés* [40], *i.e*, pour tout réel β strictement supérieur à 1, il existe un code C_β tel que $\mathcal{S}_\beta = C_\beta^\omega$. Un tel code C_β peut être défini comme l'ensemble des mots finis strictement inférieurs à $d_\beta(1)$ et dont le plus long préfixe propre est préfixe de $d_\beta(1)$. De plus, l'entropie topologique d'un β-shift est $\log \beta$ [128, 123].

β-shifts sofiques

À partir de la définition du code C_β associé au β-shift, on obtient que le β-shift est sofique si et seulement si la β-représentation de 1 est ultimement périodique [37] ; β est alors appelé un *nombre de Parry* (ou β-nombre).

La caractérisation des classes de β-shifts est profondément liée aux propriétés algébriques de β. On rappelle qu'un entier algébrique est un *nombre de Perron* s'il est strictement plus grand que le module de chacun de ses conjugués algébriques ; c'est un *nombre de Pisot* si tous ses conjugués sont de module strictement inférieur à 1 et un *nombre de Salem* si tous ses conjugués sont de module inférieur à 1, l'un au moins d'entre eux étant de module égal à 1.

Si le β-shift est sofique, alors β est un nombre de Perron [52, 100]. En effet, la matrice d'adjacence de l'automate fini qui reconnaît l'ensemble des facteurs finis du β-shift est alors une matrice primitive intégrale (voir Section 1.4) dont le rayon spectral, β, est donc un nombre de Perron.

Pour les nombres de Perron de degré 2, la divisibilité du polynôme caractéristique de cette matrice par le polynôme minimal de β, implique que si le β-shift est sofique alors β est un nombre de Pisot. Inversement, Bertrand [37] et Schmidt [135] ont montré indépendamment que tous les nombres de Pisot sont des nombres de Parry. Ce résultat découle en fait de l'ultime périodicité de $d_\beta(x)$ pour tout réel $x \in [0,1] \cap \mathbb{Q}(\beta)$, quand β est un nombre de Pisot [135].

Concernant les nombres de Salem, on sait seulement que si β est un nombre de Salem quartique (de degré 4), la β-représentation de 1 est ultimement périodique [42]. Selon Boyd [44], tous les nombres de Salem de degré 6 seraient des nombres de Parry, mais, pour tout degré supérieur, il existerait des nombres de Salem qui ne sont pas des nombres de Parry. Ces conjectures sont fondées sur l'étude d'un modèle probabiliste heuristique qui prédit que les orbites de 1 sous l'action de T_β peuvent être arbitrairement grandes quand β est un nombre de Salem de degré 6 et qu'une proportion positive de nombres de Salem de degré fixé supérieur à 8 ne sont pas des nombres de Parry.

D'une manière plus générale, le domaine des conjugués algébriques des nombres de Parry a été étudié indépendamment par Solomyak [138] et par Flatto, Lagarias et Poonen [65]. Ils ont montré en particulier que si le β-développement de 1 est ultimement périodique alors les conjugués algébriques de β sont de module stric-

tement inférieur au nombre d'or $(1+\sqrt{5})/2$. Il était déjà connu [123] que β ne pouvait avoir de conjugué algébrique réel supérieur à 1.

La caractérisation complète des nombres de Perron dont le β-shift associé est sofique reste un problème ouvert.

β-shifts de type fini

Une sous-classe des β-shifts sofiques est également caractérisée en termes simples par la β-représentation de 1 : celle des β-shifts de type fini. En effet, le β-shift est de type fini si et seulement si la β-représentation de 1 est finie [86] ; β est alors un *nombre de Parry simple* (ou β-nombre simple).

Solomyak [138] a prouvé que la clôture topologique des conjugués d'un nombre de Parry et celle des conjugués des nombres de Parry simples étaient égales. Néanmoins, il existe une importante différence entre ces deux ensembles de conjugués algébriques : si β est un nombre de Parry simple alors β n'a pas de conjugué algébrique réel positif. En effet, si β un nombre de Parry simple dont la β-représentation de 1 est $d_\beta(1) = d_1 \dots d_n$, alors β est racine du polynôme $P = X^n - \sum_{i=1}^{n} d_i X^{n-i}$ qui n'a qu'une seule racine positive.

De plus, la matrice d'adjacence de l'automate des facteurs finis du β-shift est, dans ce cas, la matrice compagnon du polynôme P. Une construction due à Handelman [80] permet d'obtenir à partir d'un nombre de Perron β qui n'a pas de conjugué algébrique réel positif une matrice compagnon primitive intégrale dont le rayon spectral est β. Mais l'algorithme basé sur des arguments algébriques ne permet pas de contrôler le comportement combinatoire des coefficients de la matrice ainsi associée à β.

Cette condition sur les conjugués algébriques réels positifs d'un nombre de Parry simple permet néanmoins

- de montrer facilement que les nombres de Salem, qui sont racines de polynômes réciproques, ne sont pas des nombres de Parry simples
- et de caractériser les nombres de Parry simples de degré 2 : ce sont exactement les nombres de Pisot quadratiques qui n'ont pas de conjugué réel positif [69].

J'ai entièrement caractérisé les nombres de Parry simples et cubiques [17]. Pour obtenir ce résultat, j'ai montré qu'il suffisait d'étudier les nombres de Pisot cubiques. En effet,

Théorème 3 *[17] Si β est un nombre de Parry simple et cubique alors β est nombre de Pisot.*

Ce résultat, obtenu par des arguments algébriques, n'est plus vrai pour les nombres de Parry simples de degré strictement supérieur à 3. Par exemple, la

racine positive du polynôme $X^4 - 3X^3 - 2X^2 - 3$ est un nombre de Parry simple et quartique qui n'est pas un nombre de Pisot.

Il reste alors à identifier parmi les nombres de Pisot de degré 3 ceux pour lesquels la β-représentation de 1 est finie.

Théorème 4 *[17] Soient β un nombre de Pisot cubique et $M_\beta(x) = X^3 - aX^2 - bX - c$ son polynôme minimal. Alors β est nombre de Parry simple si et seulement si β satisfait une des conditions suivantes :*

(i) $b \geq 0$ *et* $c > 0$,

(ii) $-a < b < 0$ *et* $b + c \geq 0$,

(iii) $b \leq -a$ *et* $b(k-1)+c(k-2) \leq (k-2)-(k-1)a$, *où* k *l'entier de* $[\![2, a-2]\!]$ *tel que* $(a-2)/k \leq a+b+c-1 < (a-2)/(k-1)$.

Ce résultat est la conséquence d'une classification complète des β-shifts définis par des nombres de Pisot cubiques [17]. À partir de la caractérisation des coefficients des polynômes minimaux des nombres de Pisot cubiques due à Akiyama [7], j'ai calculé toutes les β-représentations de 1 en fonction des valeurs prises par les coefficients de ces polynômes.

Cette construction met en évidence un phénomène inattendu : quand β est un nombre de Pisot cubique, la longueur de la β-représentation de 1 peut être arbitrairement grande. Ainsi, pour tout entier k supérieur à 2, si β est la racine réelle du polynôme irréductible $X^3 - (k+2)X^2 + 2kX - k$, alors $d_\beta(1)$ est de longueur $2k+2$. Quand $k = 2$, on obtient $d_\beta(1) = 221002$; quand $k = 3$, on a $d_\beta(1) = 31310203$.

Suivant le même procédé, Gijni [74] a calculé la β-représentation de 1 pour tous les nombres de Pisot quartiques unitaires. La limite de cette méthode vient d'une augmentation très rapide, en fonction du degré algébrique de la base, du nombre de cas à étudier.

La caractérisation des nombres de Parry simples reste un problème essentiellement ouvert. Une autre approche du problème consisterait à déterminer les nombres de Perron pour lesquels la β-représentation de 1 est d'une longueur finie fixée.

3.1.2 β-réseaux et ensembles de Meyer

L'ensemble $\mathbb{Z}_\beta$ des *β-entiers* est l'ensemble des nombres réels dont le β-développement de la valeur absolue a une partie β-fractionnaire nulle. En notant $\mathbb{Z}_\beta^+$ l'ensemble des β-entiers positifs, on a $\mathbb{Z}_\beta = \mathbb{Z}_\beta^+ \cup \mathbb{Z}_\beta^-$ et $\mathbb{Z}_\beta^- = -\mathbb{Z}_\beta^+$. Par construction, l'ensemble des β-entiers est symétrique et stable pour la multiplication par β. Quand β est un entier, l'ensemble des β-entiers est $\mathbb{Z}$. À la différence de la

numération en base entière, si β n'est pas un entier, l'ensemble des β-entiers n'est stable ni pour l'addition, ni pour la multiplication.

L'ensemble des β-développements des éléments de $\mathbb{Z}_\beta^+$ est le langage des facteurs finis du β-shift. En effet, quand x est un nombre réel strictement supérieur à 1, il existe un entier positif k tel que $\beta^k \leq x < \beta^{k+1}$. Par suite, comme $0 < x/\beta^{k+1} < 1$, le β-développement de x est l'image par décalage du β-développement d'un nombre de l'intervalle $[0, 1[$. Quand β est un nombre de Parry, l'ensemble des β-développements des éléments de $\mathbb{Z}_\beta^+$ et, par symétrie, celui des β-développements de Z_β^- sont donc reconnaissables par un automate fini. Et, quand β est un nombre de Pisot, l'addition des β-entiers est calculable par un transducteur fini [67].

Modélisation des quasicristaux

Dans un cristal, les positions des atomes peuvent être modélisées par les points d'un réseau et sont invariantes par une symétrie d'ordre 2, 3, 4 ou 6. La découverte dans les années 80 d'un alliage dont le diagramme de diffraction présentait une symétrie d'ordre 5 et un ordre apériodique à longue portée a stimulé la recherche de modèles mathématiques permettant de rendre compte d'une telle structure, appelée maintenant *quasicristal* [98]. Différentes idéalisations de ces solides apériodiques ont été proposées ; tel est le cas, en particulier, des ensembles coupe et projection [95, 136], des ensembles de Delaunay [96, 50], de certains pavages apériodiques [70, 94, 109] ou des quasiréseaux construits sur les β-entiers [11, 68, 57]. Ces derniers, appelés aussi *β-réseaux*, sont les ensembles de la forme $\Lambda = \sum_{i=1}^{d} \mathbb{Z}_\beta \mathbf{e_i}$ où $(\mathbf{e_i})_{1 \leq i \leq d}$ est une base $\mathbb{R}^d$. Ils ont été construits, par analogie avec le rôle joué par les réseaux pour les cristaux, sur les β-entiers qui semblent, d'un point de vue expérimental, de bon candidats pour décrire les coordonnées en dimension 1, 2 ou 3 des atomes dans les quasicristaux [58]. Dans les cas réellement observés, les valeurs pertinentes de β sont les nombres de Pisot $(1+\sqrt{5})/2$, $1+\sqrt{2}$ et $2+\sqrt{3}$.

Ces différentes approches de nature géométrique sont liées à un modèle très riche étudié par Meyer [115, 116], avant la découverte des quasicristaux ; Meyer utilisait d'ailleurs le terme de quasicristal pour décrire une généralisation des structures mathématiques modélisant les structures cristallines. Les objets qu'il a définis, maintenant appelés *ensembles de Meyer*, sont les sous-ensembles Λ de $\mathbb{R}^d$ uniformément discrets (*i.e.*, il existe un réel $r > 0$ tel que toute boule ouverte de rayon r contient au plus un point de l'ensemble) et relativement denses (*i.e.*, il existe un réel $R > 0$ tel que toute boule ouverte de rayon R contient au moins un point de l'ensemble) dont les distances entre les points sont contraintes de la manière suivante : $\Lambda - \Lambda \subset \Lambda + F$ où F est ensemble fini. Sur la pertinence des ensembles de Meyer comme modèles des positions des atomes d'un quasicristal et sur leurs liens avec les autres modèles cités, on pourra se porter à [117, 118, 119, 97].

Lagarias [97] a montré que les ensembles de Meyer pouvaient être caractérisés

par leurs propriétés topologiques : ce sont les sous-ensembles Λ de $\mathbb{R}^d$ tels que Λ et $\Lambda - \Lambda$ soient uniformément discrets et relativement denses. Meyer [115] a prouvé que si Λ est un ensemble de Meyer et $\beta > 1$ est un nombre réel tel que $\beta\Lambda \subset \Lambda$ alors β est un nombre de Pisot ou de Salem. Réciproquement, pour tout entier strictement positif d et tout nombre de Pisot ou de Salem β, il existe un ensemble de Meyer $\Lambda \subset \mathbb{R}^d$ tel que $\beta\Lambda \subset \Lambda$.

Quand β est un nombre de Pisot, l'ensemble $\mathbb{Z}_\beta$ des β-entiers et, par suite, tout β-réseau sont des ensembles de Meyer [45, 9]. Les preuves de ce résultat utilisent la caractérisation topologique des ensembles de Meyer due à Lagarias et permettent seulement de prouver l'existence d'un ensemble fini F tel que $\mathbb{Z}_\beta - \mathbb{Z}_\beta \subset \mathbb{Z}_\beta + F$.

Additions de β-entiers

Nous nous sommes intéressés [9], avec S. Akiyama et C. Frougny, à la construction d'un ensemble F tel que $\mathbb{Z}_\beta - \mathbb{Z}_\beta \subset \mathbb{Z}_\beta + F$ qui soit de taille minimale, quand β est un nombre de Pisot.

Avec des méthodes géométriques, Lagarias [97] a donné une technique générale permettant d'obtenir, pour tout ensemble de Meyer Λ, un ensemble fini F tel que $\Lambda - \Lambda \subset \Lambda + F$. Cependant aucune méthode pour minimiser la taille d'un tel ensemble n'est connue.

Une autre approche du problème consiste à montrer que l'ensemble des parties β-fractionnaires des β-développements de la somme de deux β-entiers est fini. Un tel ensemble satisfait alors la définition de l'ensemble F. Frougny et Solomiak [69] ont montré que, quand β est un nombre de Pisot, la longueur de la partie β-fractionnaire de la somme de deux β-entiers, quand elle est finie, est uniformément bornée. De plus, si β est un nombre de Pisot quadratique, ces parties β-fractionnaires sont toujours finies [69] et une majoration de leurs longueurs est calculée dans [79]. Quand β est un nombre de Pisot quadratique unitaire, les β-développements des sommes de deux β-entiers sont entièrement caractérisés dans [45], leurs parties β-fractionnaires sont alors dans $\{0, \pm 1/\beta, \pm 1/\beta\}$. Récemment, Bernat [28] a étudié les parties β-fractionnaires des β-développements de la somme de deux β-entiers pour certains nombres de Pisot cubiques, en particulier le nombre de Tribonacci (*i.e.*, la racine positive de $X^3 - X^2 - X - 1$), et pour les nombres de Perron [29].

Ainsi, l'étude de l'addition des β-entiers permet d'obtenir des ensembles finis F pour $\mathbb{Z}_\beta$ quand β est un nombre de Pisot quadratique [79] ; ces ensembles ne sont en général pas de taille minimale, sauf quand β est unitaire [45].

En utilisant des techniques issues de la théorie des automates, nous avons, avec S. Akiyama et C. Frougny, établi le résultat suivant.

Théorème 5 *[9] Pour tout nombre de Pisot β, un ensemble F de taille minimi-*

nale tel que $\mathbb{Z}_\beta - \mathbb{Z}_\beta \subset \mathbb{Z}_\beta + F$ peut être calculé par un algorithme qui est exponentiel en temps et en espace.

Cet algorithme consiste essentiellement en la construction d'un transducteur qui transforme une représentation d'un élément de l'ensemble de Meyer $\mathbb{Z}_\beta - \mathbb{Z}_\beta$ en sa représentation sous la forme $\mathbb{Z}_\beta + F$ d'un β-entier et d'une partie "fractionnaire" appartenant à F.

Dans un premier temps, des arguments algébriques permettent de montrer que la partie β-fractionnaire du β-développement de deux β-entiers ne prend qu'un nombre fini de valeurs et de définir un sous-ensemble fini F' de $\mathbb{Z}[\beta] \cap]-1, 1[$ tel que $\mathbb{Z}_\beta - \mathbb{Z}_\beta \subset \mathbb{Z}_\beta + F'$.

Ensuite, les éléments de $\mathbb{Z}_\beta - \mathbb{Z}_\beta$ sont représentés comme l'addition bit à bit des β-développements de deux éléments de $\mathbb{Z}_\beta$. Quand β est un nombre de Parry, le langage ainsi obtenu est reconnaissable par un automate fini.

Des produits cartésiens et l'intersection d'automates permettent alors de mettre en correspondance chaque élément x de $\mathbb{Z}_\beta - \mathbb{Z}_\beta$ représenté par l'addition formelle de deux β-développements avec tous les couples (y, f) formés du β-développement y d'un β-entier et d'un élément f de F', tels que la valeur représentée par x et la somme des valeurs de y et de f soient égales. Les éléments de l'ensemble F' sont les valuations des états finaux des automates utilisés dans cette construction.

La dernière étape consiste à minimiser la taille de l'ensemble F' de telle sorte que la correspondance décrite soit une application. Cette étape requiert la déterminisation d'un automate, ce qui explique la complexité exponentielle en espace de l'algorithme, ainsi qu'une recherche sur toutes les parties de F' ce qui conduit à une complexité exponentielle en temps de l'algorithme.

3.2 Mots de Lyndon

La deuxième partie de ce chapitre a pour objet l'étude d'un autre type de suites définies par des conditions lexicographiques : les mots de Lyndon.

Ces mots ont été introduits par Lyndon [110] sous le nom de "suites lexicographiques standard" dans le but de définir une base pour les algèbres de Lie libres. Ils peuvent également être utilisés pour construire des bases du monoïde libre et du groupe libre.

Un mot, défini sur un alphabet totalement ordonné A, est un *mot de Lyndon* s'il est strictement plus petit, dans l'ordre lexicographique, que tous ses conjugués (*i.e.*, tous les mots obtenus par une permutation circulaire des lettres). En d'autres termes, un mot de Lyndon est un mot primitif (*i.e.*, il ne s'écrit pas comme puissance entière d'un autre mot) qui est minimal pour l'ordre lexicographique dans sa classe de conjugaison.

L'ensemble des mots de Lyndon de longueur n est noté $\mathcal{L}_n$ et $\mathcal{L} = \cup_n \mathcal{L}_n$. Par exemple, sur l'alphabet $A = \{a, b \mid a < b\}$, les mots de Lyndon jusqu'à la longueur 5 sont

$$\begin{aligned}\mathcal{L} = \{&a, b, ab, aab, abb, aaab, aabb, abbb,\\ &aaaab, aaabb, aabab, aabbb, ababb, abbbb, \dots\}\end{aligned}$$

Le nombre $\mathrm{Card}(\mathcal{L}_n)$ de mots de Lyndon de longueur n sur l'alphabet A est, selon la *formule de Witt* [106],

$$\mathrm{Card}(\mathcal{L}_n) = \frac{1}{n} \sum_{d|n} \mu(d)\, \mathrm{Card}(A)^{n/d},$$

où μ est la fonction de Moebius définie sur $\mathbb{N} \setminus \{0\}$ par $\mu(1) = 1$, $\mu(n) = (-1)^i$ si n est le produit de i nombres premiers distincts et $\mu(n) = 0$ dans le cas contraire.

Quand $\mathrm{Card}(A) = k$, on obtient l'estimation suivante

$$\mathrm{Card}(\mathcal{L}_n) = \frac{k^n}{n} \left(1 + O\left(2^{-k/2}\right)\right).$$

Ainsi un mot aléatoire de longueur n a une probabilité proche de $1/n$ d'être un mot de Lyndon. On peut néanmoins engendrer aléatoirement, pour la distribution uniforme, un mot de Lyndon par un algorithme avec rejet dont la complexité moyenne en temps est linéaire [21].

Il existe également un algorithme [66] qui engendre, en temps moyen constant [33], tous les mots de Lyndon, dans l'ordre lexicographique, jusqu'à une longueur fixée. Dans MuPad-`Combinat`, tous les mots de Lyndon de longueur donnée sont engendrés, en temps amorti constant, grâce à un algorithme générique dédié à la génération de cycles non étiquetés dû à Martinez et Molinero [113]. Fredricksen et Maiorana [66] ont mis en évidence la relation suivante entre mots de Lyndon et mots de de Bruijn (*i.e.*, les mots circulaires de longueur $\mathrm{Card}(A)^n$ dont tout mot de longueur n est facteur). Pour tout entier $n \geq 1$, la concaténation en ordre croissant des mots de Lyndon dont la longueur divise n est le plus petit mot de de Bruijn pour l'ordre lexicographique. Cette caractérisation fournit un algorithme, linéaire en temps, pour calculer un mot de de Bruijn. Les liens entre mots de Lyndon et mots de Bruijn ainsi que des algorithmes permettant de les engendrer sont présentés par Knuth dans [91].

Une des propriétés importantes de l'ensemble des mots de Lyndon est que tout mot du monoïde libre A^* se décompose de manière unique en un produit décroissant de mots de Lyndon. Le lecteur intéressé par l'origine historique de ce résultat dont la paternité est généralement attribuée à Lyndon, trouvera des éléments d'information dans [32]. D'un point de vue algorithmique, cette factorisation peut être

calculée en temps linéaire [66, 55]. D'autres propriétés de cette factorisation sur les mots sont obtenues grâce à sa relation avec la notion de factorisation sur les polynômes.

Comme le laisse supposer la formule de Witt, il existe une bijection [77] entre les polynômes irréductibles de degré n à coefficients dans $\mathbb{F}_k$ (où k est la puissance d'un nombre premier) et les mots de Lyndon de longueur n sur un alphabet de taille k. Cette bijection met en fait en relation tout polynôme irréductible de degré n à coefficients dans $\mathbb{F}_k$ avec une classe de conjugaison de mots primitifs. Elle est définie de la manière suivante.

Soit α une racine primitive (k^n-1)-ème de l'unité dans $\mathbb{F}_k$, alors un polynôme P de degré n est irréductible dans $\mathbb{F}_k[X]$ si et seulement si il se factorise dans $\mathbb{F}_{k^n}[X]$ sous la forme

$$P = \prod_{i=0}^{n-1} \left(X - \alpha^{mk^i}\right)$$

où tous les α^{mk^i}, pour $0 \leq i \leq n-1$, sont deux à deux distincts. Comme α est une racine primitive (k^n-1)-ème de l'unité, cette dernière condition est équivalente au fait que tous les $mk^i \mod (k^n-1)$, pour $0 \leq i \leq n-1$, sont distincts. Or, pour tout $x \in [\![\, 0, k^n-1 \,]\!]$, la multiplication de x par k modulo (k^n-1) se traduit par une permutation circulaire de la représentation de x en base k : si

$$x_{n-1} \dots x_0$$

est la représentation en base k de x, alors

$$x_{n-2} \dots x_0 x_{n-1}$$

est la représentation de $xk \mod (k^n-1)$ en base k. Par suite, l'ensemble des représentations des $mk^i \mod (k^n-1)$, pour $0 \leq i \leq n-1$, est l'ensemble des conjugués de la représentation en base k de l'entier m. Les $mk^i \mod (k^n-1)$, pour $0 \leq i \leq n-1$, sont donc deux à deux distincts si et seulement si l'écriture en base k de m est un mot primitif.

Grâce à l'existence de cette bijection, de nombreux résultats concernant la factorisation d'un mot aléatoire de longueur n en un produit décroissant de mots de Lyndon sont connus :

- le nombre moyen de facteurs est proche de $\log n$ avec une forte probabilité [60],
- la longueur moyenne du plus long facteur est proche de gn où $g = 0.62432$ est la constante de Golomb [60],
- la longueur moyenne du plus court facteur est établie dans [122],
- la distribution du nombre de facteurs distincts de longueur m est approximativement une loi de Poisson de paramètre $1/m$ [60].

Les mots de Lyndon sont également caractérisés par leurs suffixes : ce sont les mots non vides qui sont strictement plus petits, pour l'ordre lexicographique, que tous leurs suffixes propres. Cette propriété permet de définir la *factorisation standard* d'un mot de Lyndon w qui n'est pas une lettre : si v est le plus petit, pour l'ordre lexicographique, suffixe propre de $w = uv$ alors les mots u et v sont des mots de Lyndon et sont respectivement appelés *facteurs gauche* et *droit* de la factorisation standard de w.

Par exemple, sur l'alphabet $A = \{a, b \mid a < b\}$, on obtient les factorisations standard suivantes :

$$aaabaab = aaab \cdot aab, aaababb = a \cdot aababb, aabaabb = aab \cdot aabb.$$

La factorisation standard d'un mot w peut être effectuée par un algorithme linéaire en temps [21]. Il suffit pour cela de factoriser le suffixe de longueur $(n-1)$ de w en un produit décroissant de mots de Lyndon, opération dont la complexité est linéaire en temps [66, 55], le dernier facteur obtenu est alors le facteur droit de la factorisation standard de w.

La factorisation standard des mots de Lyndon est un élément clé de la construction de bases des algèbres de Lie libres ou du groupe libre. Plus précisément, les mots de Lyndon permettent de construire des commutateurs par un processus dichotomique utilisant la notion de factorisation standard. Ainsi le mot de Lyndon *aababb* s'écrit en itérant le processus de factorisation standard $[a[[ab][[ab]b]]]$. Ces commutateurs peuvent être interprétés soit comme éléments du groupe libre [46] avec $[xy] = xyx^{-1}y^{-1}$, soit comme éléments de l'algèbre de Lie libre [106, 129, 132] avec $[xy] = xy - yx$. Dans les deux cas, les mots de Lyndon sont utilisés pour construire une base. La complexité en moyenne des algorithmes calculant ces bases est essentiellement déterminée par le nombre maximal, en moyenne, d'appels récursifs à l'opération de factorisation standard et nécessite une connaissance fine du résultat de cette factorisation.

3.2.1 Mots de Lyndon ayant un facteur droit donné

Alors que l'ensemble des mots de Lyndon n'est pas un langage algébrique [31], nous avons montré, avec J. Clément et C. Nicaud [21], que l'ensemble des mots de Lyndon ayant un facteur droit fixé dans leur factorisation standard est un langage régulier dont on peut explicitement calculer la série génératrice. Les techniques utilisées pour établir ces résultats relèvent essentiellement de la combinatoire des mots.

Plus précisément, soit $A = \{a_1 < \cdots < a_k = \gamma\}$ où γ représente le plus grand symbole de l'alphabet ordonné A. Soit w un mot de $A^* \setminus \{\gamma\}^*$, le *successeur* $S(w)$ de $w = u\alpha\gamma^i$, où α est un symbole de $A \setminus \{\gamma\}$ et $i \geq 0$, est défini par $S(w) = u\beta$ où

β est le symbole qui suit immédiatement α dans A pour l'ordre lexicographique. Pour tout mot de Lyndon v, on définit le langage

$$\mathcal{X}_\gamma = \{\gamma\} \quad \text{et} \quad \mathcal{X}_v = \{v, S(v), S^2(v), \ldots, S^{p-1}(v) = \gamma\} \quad \text{si } v \neq \gamma.$$

Le langage $\mathcal{X}_v$ est un code préfixe et, par construction, le mot v est le plus petit élément de $\mathcal{X}_v A^*$ pour l'ordre lexicographique.

Ainsi, quand $A = \{a, b \mid a < b\}$ et $v = aabab$, on a $\mathcal{X}_{aabab} = \{aabab, aabb, ab, b\}$, et, quand $A = \{a, b, c \mid a < b < c\}$ et $v = abb$, on obtient $\mathcal{X}_{abb} = \{abb, abc, ac, b, c\}$.

En notant, pour toute lettre $\alpha \in A$, $A_{\leq\alpha}$ l'ensemble $\{a \in A \mid a \leq \alpha\}$ des lettres de l'alphabet A qui sont plus petites que α pour l'ordre lexicographique, on peut caractériser les mots ayant un facteur droit donné dans leur factorisation standard de la manière suivante.

Théorème 6 *[21] Soit v un mot de Lyndon dont la première lettre est α et $u \in A^+$ un mot non vide. Alors uv est un mot de Lyndon ayant $u \cdot v$ comme factorisation standard si et seulement si $u \in (A_{\leq\alpha}\mathcal{X}_v^*) \setminus \mathcal{X}_v^+$. L'ensemble $\mathcal{F}_v$ des mots de Lyndon ayant le mot v comme facteur standard droit est donc un langage régulier.*

De plus, la série génératrice $F_v(z) = \sum_{x \in \mathcal{F}_v} z^{|x|}$ de $\mathcal{F}_v$ est régulière et peut être explicitement calculée.

Théorème 7 *[21] Soit v un mot de Lyndon sur un alphabet à k lettres. La série génératrice de l'ensemble $\mathcal{F}_v$ des mots de Lyndon ayant le mot v comme facteur standard droit s'écrit sous la forme*

$$F_v(z) = z^{|v|}\left(1 + \frac{kz - 1}{1 - X_v(z)}\right),$$

où $X_v(z)$ la série génératrice de $\mathcal{X}_v$.

Bien que la série $F_v(z)$ soit régulière, il est difficile de faire une étude en moyenne du comportement de ses coefficients lorsque le facteur standard droit v décrit l'ensemble des mots de Lyndon. Le problème vient, en partie, du fait que l'ensemble des mots de Lyndon n'est pas algébrique.

3.2.2 Étude en moyenne de la factorisation standard

Pour étudier la longueur moyenne des éléments de la factorisation standard des mots de Lyndon, nous avons été amenés à adopter un autre point de vue. Nous avons cherché à modéliser l'ensemble des mots de Lyndon d'une manière suffisamment simple pour pouvoir mener à bien une étude en moyenne et d'une

manière suffisamment fine pour que la différence entre le modèle et l'ensemble original puisse être négligée. La construction de ce modèle et surtout sa comparaison avec l'ensemble des mots de Lyndon se basent sur une étude fine des propriétés des mots de Lyndon utilisant des techniques probabilistes [59] et des résultats issus de combinatoire analytique [61].

Par souci de clarté, on se restreint au cas d'un *alphabet à deux lettres* $A = \{a, b \mid a < b\}$.

Modélisation des mots de Lyndon de longueur n

Dans un premier temps, on partitionne l'ensemble $\mathcal{L}_n$ des mots de Lyndon de longueur n en deux sous-ensembles dont la taille proche est de $2^{n-1}/n$:

- l'ensemble $a\mathcal{L}_{n-1}$ des mots de Lyndon de longueur $n-1$ précédés de la lettre a
- et son complémentaire $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$ dans $\mathcal{L}_n$.

Les mots de $a\mathcal{L}_{n-1}$ ont tous la lettre a pour facteur gauche et leur suffixe de longueur $n-1$ pour facteur droit de leur factorisation standard. Une description plus fine de cet ensemble peut être obtenue grâce à la modélisation de $\mathcal{L}_{n-1}$.

Pour étudier l'ensemble $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$, l'idée est de se servir des plus longues suites de a qui apparaissent dans un mot de Lyndon pour le décrire. À cette fin, on définit une nouvelle décomposition des mots de $\mathcal{L} \setminus a\mathcal{L}$.

Soit w un mot de l'ensemble $\mathcal{L} \setminus a\mathcal{L}$. On note p la longueur de la plus longue plage de a dans w et on définit l'entier P comme étant égal à $p-1$ quand w ne contient qu'une seule plage de a de longueur maximale et comme étant égal à p dans le cas contraire. La *décomposition selon les plages maximales* de w est définie par

$$w = f_1 \dots f_m,$$

où $\mathcal{X}_P = \{a^i b \mid 0 \leq i \leq P-1\}$, $f_1 \in a^p b \mathcal{X}_P^*$ et, pour $2 \leq i \leq m$, $f_i \in a^P b \mathcal{X}_P^*$.

La factorisation standard du mot w peut alors être décrite par cette décomposition selon les plages maximales. En effet, il existe un indice $j \in [\![2, m]\!]$ tel que la factorisation standard de w soit

$$\prod_{i=1}^{j-1} f_i \cdot \prod_{i=j}^{m} f_i.$$

Par exemple, le mot de Lyndon *aababab* a pour factorisation standard $aabab \cdot ab$ et pour décomposition $aab \cdot ab \cdot ab$; le mot *aababaabbaabbb* a pour factorisation standard $aabab \cdot aabbaabbb$ et pour décomposition $aabab \cdot aabb \cdot aabbb$.

L'étude de cette décomposition permet d'affiner la description de la structure des mots de $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$.

Théorème 8 *[21] La décomposition selon les plages maximales d'un mot de Lyndon de $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$ satisfait, avec la probabilité $1 + o(1)$, les trois propriétés suivantes :*

1. *la plus longue plage de a est de longueur $p \in [\log_2 n - \log_2 \log_2 n - 1, 2\log_2 n[$,*
2. *le nombre m de facteurs est inférieur à $2\log_2 n$,*
3. *ces m facteurs ont des préfixes de longueurs comprises entre $2p-1$ et $3p-3$ qui forment un code préfixe.*

La notion de plage de a est très proche de celle de "success run" en probabilité. L'estimation du maximum de leurs longueurs fait appel à la méthode, dite de "bootstrapping", utilisée par Knuth [90] pour étudier le temps moyen de propagation d'une retenue au cours d'une addition. De manière générale, les techniques utilisées pour établir ce résultat relèvent de la combinatoire analytique [61]. On étudie essentiellement des fonctions génératrices, univariées et multivariées, construites de manière à capter la notion de cycle primitif [63] associée aux mots de Lyndon. L'analyse asymptotique des coefficients de ces fonctions est obtenue par l'étude leurs singularités [48].

Longueur moyenne des facteurs standard

À l'aide du modèle qui vient d'être décrit, nous avons établi, avec C. Nicaud et J. Clément, le résultat suivant

Théorème 9 *[21] Pour la distribution uniforme sur les mots de Lyndon de longueur n, la longueur moyenne du facteur droit de la factorisation standard est asymptotiquement égale à $3n/4$.*

La Figure 3.1 compare le résultat expérimental d'un calcul de la longueur moyenne du facteur standard droit des mots de Lyndon à sa valeur théorique représentée par la droite de pente 3/4.

Plus précisément, tout mot de $a\mathcal{L}_{n-1}$ a son suffixe de longueur $(n-1)$ comme facteur droit de sa factorisation standard. Comme

$$\mathrm{Card}\left(\mathcal{L}_{n-1}\right) = \frac{\mathrm{Card}\left(\mathcal{L}_n\right)}{2}\left(1 + o(1)\right),$$

la contribution de l'ensemble $a\mathcal{L}_{n-1}$ à la valeur moyenne de la longueur du facteur standard droit est

$$\frac{n}{2}\left(1 + o\left(1\right)\right).$$

Pour calculer la longueur moyenne du facteur des mots de $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$, on utilise des arguments issus de la combinatoire des mots. L'idée est de construire une transformation φ, qui soit une involution sur presque tout l'ensemble $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$,

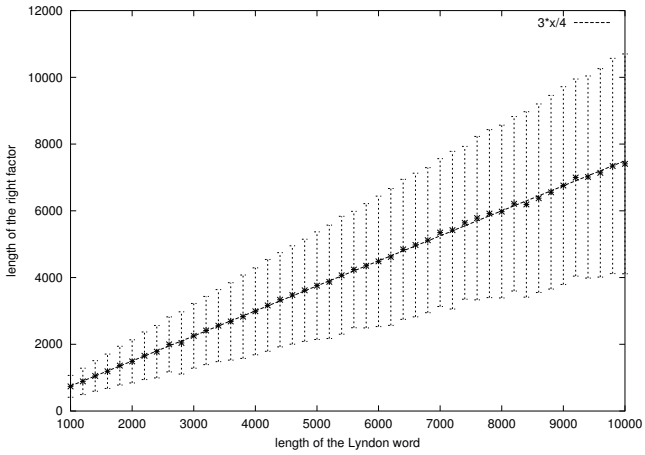

FIG. 3.1 – Longueur moyenne du facteur standard droit de mots de Lyndon aléatoires dont la longueur varie de 1000 à 10000. Chaque point est calculé à partir de 1000 mots. Les barres verticales représentent l'écart-type.

et telle que la somme des longueurs des facteurs standard droits de w et $\varphi(w)$ soit presque égale à la longueur de w. Le mot $\varphi(w)$ est obtenu à partir de w en échangeant des suffixes particuliers des facteurs de la factorisation standard de w de telle sorte φ préserve globalement les plus longues plages de a ainsi que les préfixes qui permettent d'ordonner les facteurs de la décomposition selon les plus longues plages ; en particulier, les préfixes associés aux facteurs gauche et droit de la factorisation standard sont conservés par $\varphi(w)$.

Ainsi, pour la distribution uniforme sur l'ensemble $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$, la longueur moyenne du facteur droit est

$$\frac{n}{2}(1 + o(1)).$$

Comme, de nouveau,

$$\mathrm{Card}\left(\mathcal{L}_n \setminus a\mathcal{L}_{n-1}\right) = \frac{\mathrm{Card}\left(\mathcal{L}_n\right)}{2}\left(1 + o(1)\right),$$

la contribution de l'ensemble $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$ à la valeur moyenne de la longueur du facteur standard droit est

$$\frac{n}{4}\left(1 + o\left(1\right)\right).$$

La Figure 3.2 laisse supposer une propriété d'équirépartition de la longueur du facteur standard droit sur l'ensemble $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$. La loi limite, corroborant cette constatation, de ce paramètre sur un alphabet à k lettres a été établie dans un article récent [112].

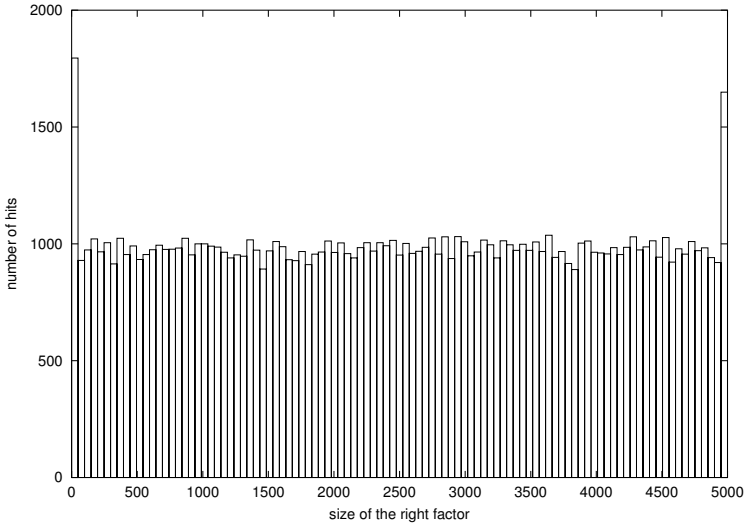

FIG. 3.2 – Distribution de la longueur du facteur standard droit sur $\mathcal{L}_n \setminus a\mathcal{L}_{n-1}$ calculée à partir de $100,000$ mots de Lyndon aléatoires de longueur $5,000$.

Chapitre 4

Énumération et génération aléatoire d'automates

Ce chapitre porte sur l'énumération et la génération aléatoire d'automates déterministes complets et accessibles à n états sur un alphabet fini. Les résultats présentés sont le fruit d'une collaboration avec C. Nicaud.

Dans un premier temps, nous estimons asymptotiquement le nombre de tels automates, en utilisant des transformations combinatoires et des arguments d'analyse asymptotique. Nous donnons ensuite deux méthodes pour engendrer aléatoirement ces automates et analysons la complexité des algorithmes dont nous nous servons.

4.1 Énumération

On rappelle que deux automates finis déterministes et complets (voir Section 1.2) sont isomorphes s'il ne diffèrent que par l'étiquette de leurs états. Nous nous sommes intéressés, avec C. Nicaud, au problème de l'énumération des automates non isomorphes déterministes complets et accessibles à n états sur un alphabet fini. On note, pour tout entier n strictement positif, $\mathcal{A}_n$ l'ensemble de ces automates et k la taille de l'alphabet.

L'énumération des automates finis selon divers critères (avec ou sans état initial [92], non isomorphes [82], à une permutation des étiquettes des transitions près [82], ayant un graphe sous-jacent fortement connexe [104, 92, 131, 93], acycliques [105]...) est un problème étudié depuis 1959 [141]. Compter les automates finis apparaît comme le problème 19 dans la liste d'Harary de problèmes non résolus en énumération de graphes [81]. La plupart des travaux effectués dans ce domaine portent sur d'autres automates, que ceux considérés ici, et sont essentiellement différents d'un point de vue combinatoire. On pourra se reporter à [53] pour une bibliographie plus fournie sur le sujet.

Plusieurs auteurs ont néanmoins étudié les automates accessibles [103, 92, 131] ; en particulier, Korshunov [92, 93] a donné un équivalent du nombre d'automates de l'ensemble $\mathcal{A}_n$. L'énoncé de ce résultat, sur lequel nous reviendrons dans la suite, est cependant difficile à formuler et à prouver.

4.1.1 Formule d'énumération exacte

En utilisant des constructions combinatoires, Nicaud [120] a obtenu une formule d'énumération exacte des automates de $\mathcal{A}_n$, dans le cas d'un alphabet de taille 2, qui a été généralisée à un alphabet fini quelconque dans [49]. Ce résultat est établi par la transformation de ce problème d'énumération d'automates en une sommation de produits d'entiers.

Plus précisément, on suppose l'alphabet totalement ordonné. La première étape pour compter les automates non isomorphes déterministes complets et accessibles de $\mathcal{A}_n$ consiste à se ramener à l'énumération de structures de transitions $\mathcal{D}_n$ définies de la manière suivante. Ces structures sont des automates déterministes, complets et accessibles à n états sans états finaux distingués et dont chaque état q est étiqueté par le plus petit mot, dans l'ordre lexicographique, qui étiquette un chemin simple (qui ne passe pas deux fois par un même état) de l'état initial à l'état q. Deux telles structures de transitions ne peuvent être isomorphes. De plus, à chaque structure de transitions à n états correspondent, selon le choix de l'ensemble d'états finaux, 2^n automates de $\mathcal{A}_n$. On obtient ainsi, pour tout entier n strictement positif,

$$|\mathcal{A}_n| = 2^n |\mathcal{D}_n|.$$

On associe, ensuite, à chaque structure de transitions de $\mathcal{D}_n$, par un parcours en profondeur à partir de l'état initial et selon l'ordre lexicographique des étiquettes des transitions, une suite de $(k-1)n+1$ entiers positifs. En fait, l'algorithme de parcours en profondeur identifie, au fur et à mesure, selon l'ordre dans lequel il traite les transitions, un arbre couvrant et produit pour chacune des $(k-1)n+1$ transitions qui ne font pas partie de l'arbre couvrant l'entier correspondant au nombre d'états appartenant à la partie de l'arbre couvrant déjà construit. La suite $(x_1, \cdots, x_{(k-1)n+1})$ d'entiers ainsi construite vérifie alors les propriétés suivantes :

(i) elle est croissante,

(ii) $x_{(k-1)n+1} = n$,

(iii) et, pour tout $i \in [\![1, (k-1)n]\!]$, $\lceil \frac{i}{k-1} \rceil \leq x_i \leq n$.

La Figure 4.1 illustre la transformation d'une structure de transitions en une suite d'entiers, les transitions en gras correspondent à l'arbre couvrant et la suite produite dans cet exemple est $(2, 4, 4, 5, 5, 6, 6)$.

On note, pour tout $m \geq 1$, $\|(x_1, \ldots, x_m)\| = \prod_{i=1}^{m} x_i$.

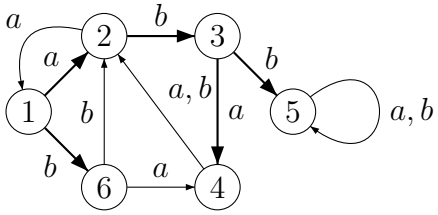

FIG. 4.1 – Une structure de transitions où 1 est l'état initial

Inversement, pour tout élément x de $[\![1, n]\!]^{(k-1)n+1}$ satisfaisant les conditions (i), (ii) et (iii), il existe exactement $\|x\|$ structures de transitions qui sont transformées en cette séquence. Cette propriété permet d'engendrer aléatoirement, pour la distribution uniforme sur $\mathcal{D}_n$, en temps linéaire [23], une structure de transitions de taille n à partir de telles suites d'entiers.

En relation avec ces suites d'entiers, on introduit l'ensemble

$$\mathcal{F}_n = \{(x_1, \ldots, x_{(k-1)n}) \in [\![1, n]\!]^{(k-1)n} \mid \text{ pour tout } i,\ x_i \geq \lceil \frac{i}{k-1} \rceil \text{ et } x_i \geq x_{i-1}\},$$

qui peut être vu comme un ensemble de chemins du réseau carré ; en particulier, quand k vaut 2, il s'agit des chemins de Dyck de longueur n.

Les transformations décrites précédemment permettent d'obtenir des formules d'énumération exactes [120, 49]

- pour les structures de transitions : $\mathcal{D}_n = n f_n$, où $f_n = \sum_{F \in \mathcal{F}_n} \|F\|$,
- et pour les automates : $\mathcal{A}_n = n\, 2^n f_n$.

4.1.2 Estimation asymptotique

Pour estimer asymptotiquement le nombre $|\mathcal{A}_n|$ d'automates déterministes, complets et accessibles à n états, il reste à étudier le comportement de f_n quand n tend vers l'infini. Cette estimation s'exprime en terme de nombres de Stirling de deuxième espèce.

On rappelle que, pour tous entiers positifs m et n, le *nombre de Stirling de deuxième espèce*, noté $\{{n \atop m}\}$, est le nombre de partitions d'un ensemble à n éléments en m parts (non vides). Ces nombres peuvent être calculés par la relation de récurrence suivante

$$\forall n, m > 0, \quad \{{n \atop m}\} = m\{{n-1 \atop m}\} + \{{n-1 \atop m-1}\}.$$

sachant que, par convention $\{{0 \atop 0}\} = 1$, et pour tout $n \geq 1$, on a $\{{n \atop 0}\} = 0$.

Théorème 10 *[23] Pour tout entier n strictement positif, $f_n = \Theta\left(\{{kn \atop n}\}\right)$, où $\{{kn \atop n}\}$ est le nombre de partitions d'un ensemble à kn éléments en n parts.*

Ce résultat est établi par le calcul de bornes supérieure et inférieure pour le nombre f_n. La majoration est obtenue en relaxant la contrainte qui impose aux éléments de $\mathcal{F}_n$ d'être au dessus de la droite de pente $1/(k-1)$. Plus précisément, on définit l'ensemble

$$\mathcal{S}_n = \{(x_1, \ldots, x_{(k-1)n}) \in [\![\, 1, n \,]\!]^{(k-1)n} \mid \text{si } i < j \text{ alors } x_i \leq x_j\}.$$

Alors, pour tout entier n strictement positif, en notant $s_n = \sum_{x \in \mathcal{S}_n} \|x\|$, on a $s_n = \left\{ {kn \atop n} \right\}$. Cette égalité peut être prouvée en vérifiant que les deux suites sont définies par les mêmes relations de récurrence et ont les mêmes premiers termes ou en construisant un bijection entre les éléments de $\mathcal{S}_n$ et les partitions d'un ensemble à kn éléments en n parts (voir [30] et [61, p.59]). Comme $\mathcal{F}_n$ est un sous-ensemble de $\mathcal{S}_n$, on obtient $f_n \leq s_n$ et $f_n \leq \left\{ {kn \atop n} \right\}$.

Le calcul d'une borne inférieure asymptotique pour f_n est plus technique. Il consiste en une majoration de la contribution à la valeur de s_n des suites d'entiers $(x_1, \ldots, x_{(k-1)n})$ qui ne satisfont pas la contrainte : pour tout i, $x_i \geq \lfloor i/(k-1) \rfloor$.

L'ordre de grandeur de f_n donné par le Théorème 10 permet d'estimer le nombre d'automates de $\mathcal{A}_n$.

Théorème 11 *[23] Le nombre d'automates déterministes complets et accessibles à n états sur un alphabet à k lettres est* $\Theta\Big(n\, 2^n \left\{ {kn \atop n} \right\}\Big)$.

De plus, le développement asymptotique des nombres de Stirling de deuxième espèce $\left\{ {kn \atop n} \right\}$ peut être calculé par la méthode du col [78], en particulier

$$\left\{ {kn \atop n} \right\} \sim \alpha_k \beta_k^n n^{(k-1)n-1/2}$$

où α_k et β_k sont deux constantes positives qui dépendent de k.

Enfin, en utilisant les nombres de Stirling de deuxième espèce, le résultat de Korshunov [92, 93] peut être reformulé en des termes plus simples.

Théorème 12 *[92, 93, 23] Le nombre d'automates déterministes complets et accessibles à n états sur un alphabet à k lettres est asymptotiquement égal à*

$$C_k\, n\, 2^n \left\{ {kn \atop n} \right\} \quad \textit{où} \quad C_k = \frac{1 + \sum_{r=1}^{\infty} \frac{1}{r} \binom{kr}{r-1} \left(e^{k-1}\beta_k\right)^{-r}}{1 + \sum_{r=1}^{\infty} \binom{kr}{r} \left(e^{k-1}\beta_k\right)^{-r}}.$$

Dans le tableau ci-dessous, on compare pour des alphabets de taille $k = 2, 3$ et 4 les valeurs du rapport $\frac{|\mathcal{A}_n|}{2^n n \left\{ {kn \atop n} \right\}}$ pour $n = 100$, 200 et 300 avec celle de

$$C_k = \lim_{n \to +\infty} \frac{|\mathcal{A}_n|}{2^n n \left\{ {kn \atop n} \right\}}.$$

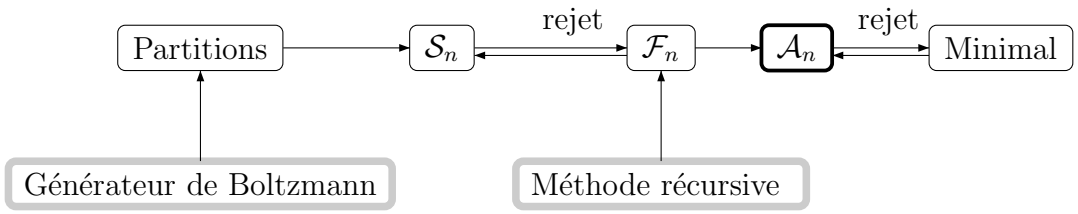

FIG. 4.2 – Schéma récapitulatif des procédés de génération aléatoire uniforme

Les nombres f_n sont calculés à partir de leur décomposition récursive dont il est fait mention dans la prochaine section. La constante C_k tend très vite vers 1, quand k tend vers $+\infty$. Par exemple, $C_{26} \asymp 0.99999999987$.

k	100	200	300	400	C_k
2	0.74490782	0.74497737	0.74498956	0.74499374	0.74499902
3	0.87341820	0.87342408	0.87342509	0.87342543	0.87342586
4	0.93931196	0.93931392	0.93931428	0.93931440	0.93931456

4.2 Génération aléatoire

On présente maintenant, dans leurs grandes lignes, deux méthodes permettant d'engendrer aléatoirement pour la distribution uniforme sur $\mathcal{A}_n$, des automates déterministes complets et accessibles à n états sur un alphabet à k lettres.

Le schéma de la Figure 4.2 récapitule les différentes étapes de la génération aléatoire qui est explicitée dans la suite. À chacune d'elles, les objets (partitions, suites d'entiers de $\mathcal{S}_n$ ou de $\mathcal{F}_n$ et automates de $\mathcal{A}_n$) sont engendrés aléatoirement de manière équiprobable. La reconstruction d'un automate de $\mathcal{A}_n$ à partir d'une suite d'entiers de $\mathcal{F}_n$ est effectuée en temps linéaire [23].

Génération des suites d'entiers de $\mathcal{F}_n$

Les suites d'entiers de $\mathcal{F}_n$ peuvent être engendrés de manière récursive [120, 49]. Cette méthode a été introduite par Nijenhuis et Wilf [121] et systématisée par Flajolet, Zimmermann et Van Custem [64]. On utilise ici le fait que tout élément $(x_1, \ldots, x_{(k-1)n})$

– se décompose, quand $x_{(k-1)n}$ est égal à n, en une suite de $(k-1)n-1$ entiers, satisfaisant des conditions analogues à celles imposées à la suite initiale, concaténée avec n

– est, dans le cas contraire, une suite d'entiers inférieurs à $n-1$, satisfaisant les mêmes contraintes que les éléments de $\mathcal{F}_n$.

Cette propriété permet d'engendrer aléatoirement de droite à gauche les entiers d'une suite $(x_1, \ldots, x_{(k-1)n})$ de $\mathcal{F}_n$. Après un important précalcul dont le résultat est stocké, chaque tirage aléatoire d'un élément de $\mathcal{F}_n$ est réalisé en temps linéaire, à condition d'utiliser des nombres à virgule flottante [51]. Cette méthode nécessite le calcul préalable d'un tableau d'entiers $t_{i,j}$ correspondant aux nombres de suites de j entiers inférieurs i satisfaisant les conditions requises. Le tableau occupe un espace en $O(n^2)$ et requiert un temps de calcul en $O(n^2)$.

La deuxième méthode [23] repose sur les générateurs de Boltzmann introduits par Duchon, Flajolet, Louchard et Schaeffer [54]. Elle permet d'engendrer aléatoirement un élément de $\mathcal{F}_n$ en un temps en $O(n^{3/2})$ et ne requiert aucun précalcul.

L'idée est d'engendrer, dans un premier temps, les partitions d'un ensemble à kn éléments en n parts. À cette fin, on tire aléatoirement, selon une loi de Poisson de paramètre adéquatement choisi, n ensembles non vides. On obtient ainsi une partition en n parts dont la taille est en moyenne kn. En se servant d'un algorithme avec rejet, on engendre des partitions de taille exactement kn avec une complexité en temps en $O(n^{3/2})$. La transformation d'une partition d'un ensemble à kn éléments en n parts en une suite d'entiers de $\mathcal{S}_n$ est ensuite réalisée en temps linéaire grâce à une bijection due à Bernardi [30]. Finalement, comme la majorité des éléments de $\mathcal{S}_n$ sont aussi des éléments de $\mathcal{F}_n$, les suites d'entiers de $\mathcal{F}_n$ sont obtenues à partir de celles de $\mathcal{S}_n$ grâce à un algorithme avec rejet.

Automates minimaux

Les automates de $\mathcal{A}_n$ ne sont pas tous minimaux, mais tous les automates minimaux à n états appartiennent à l'ensemble $\mathcal{A}_n$. De plus, expérimentalement, une proportion constante d'éléments de $\mathcal{A}_n$ semblent minimaux [120, 49], ce qui justifie en pratique l'utilisation d'un algorithme avec rejet. L'efficacité d'un tel algorithme n'est cependant pas prouvée.

L'estimation du nombre d'automates minimaux est nécessaire pour établir ce résultat. D'autre part, comme à chaque langage régulier est associé de manière unique un automate minimal, la connaissance du nombres d'automates minimaux permettrait de compter le nombre de langages réguliers dont l'automate minimal est de taille donné (voir [53]). Enfin, en interprétant la complexité en espace d'un langage régulier comme le nombre d'états de son automate minimal, elle serait une première étape vers des résultats en moyenne sur les langages réguliers comme, par exemple, l'estimation de la taille de l'intersection de deux langages.

Bibliographie

[1] J. Abrahams, Huffman-type codes for infinite source distributions, *J. of the Franklin Instit.*, 331B(3) (1994) 265–271.

[2] J. Abrahams, Code and parse trees for lossless source encoding, *Commun. Inform. Syst.*, 1(2)(2001) 113–146.

[3] R. L. Adler, D. Coppersmith, M. Hassner, Algorithms for sliding block codes, *IEEE Trans. Inform. Theory*, IT-29 (1983), 5–22.

[4] R. Ahlswede, I. Wegener, *Search problems*, Wiley, 1987.

[5] M. Aigner, *Combinatorial search*, B. G. Teubernew, Stuttgart, Wiley, NY, 1988.

[6] M. Aigner, G. M. Ziegler, *Proofs from the book*, Springer-Verlag, 1998.

[7] S. Akiyama, Cubic Pisot units with finite beta expansions, In F.Halter-Koch and R.F. Tichy, Eds., *Algebraic Number Theory and Diophantine Analysis*, pages 11–26, de Gruyter, 2000.

[8] S. Akiyama, F. Bassino and C. Frougny, Automata for arithmetic Meyer sets, In *LATIN'04*, volume 2976 in Lect. Notes Comput. Sci., p. 252–261. Springer, 2004.

[9] S. Akiyama, F. Bassino and C. Frougny, Arithmetic Meyer sets and finite automata, *Inform. and Comput.*, 201 (2005), 199–215.

[10] R. B. Ash, *Information theory*, new ed., Dover, 1990.

[11] D. Barache, B. Champagne, J.-P. Gazeau, Pisot-cyclotomic quasilattices and their symmetry semigroups, in : *Quasicrystals and discrete geometry*, J. Patera, Ed., Fields Institute Monogr., Amer. Math. Soc. (1998) 15–66.

[12] F. Bassino, M.-P. Béal, D. Perrin, Enumerative sequences of leaves in rational trees, in *ICALP'97*, vol. 1256 in *Lect. Notes Comput. Sci.*, Springer-Verlag, 1997, 76–86.

[13] F. Bassino, M.-P. Béal, D. Perrin, Super-state automata and rational trees, in *LATIN'98*, C. L. Lucchesi and A. V. Moura, eds., vol. 1380 in *Lect. Notes Comput. Sci.*, Springer-Verlag, 1998, 42–52.

[14] F. Bassino, M.-P. Béal, D. Perrin, Enumerative sequences of leaves and nodes in rational trees, *Theoret. Comput. Sci.*, (1999), 41–60.

[15] F. Bassino, M.-P. Béal, D. Perrin, A finite state version of the Kraft-McMillan theorem, *SIAM J. Comput.*, 30 (2000), 1211–1230.

[16] F. Bassino, M.-P. Béal, D. Perrin, Length distributions and regular sequences, In *Codes, Systems and Graphical Models*, J. Rosenthal and B. Marcus, Eds., Volume 123 in the series IMA Volumes in Mathematics and its Applications, Springer-Verlag, p. 415–438, 2001.

[17] F. Bassino, Beta-expansions for cubic Pisot numbers, In *LATIN'02*, S. Rajsbaum, Ed., volume 2286 in Lect. Notes Comput. Sci., p. 141–152. Springer, 2002.

[18] F. Bassino, J. Clément, Optimal codes for regular sources, *en préparation.*

[19] F. Bassino, J. Clément, C. Nicaud. The average lengths of the factors of the standard factorization of Lyndon words. In *DLT'02*, M. Ito and M. Toyama, Eds., volume 2450 in *Lect. Notes Comput. Sci.*, Springer (2003) , 307-318.

[20] F. Bassino, J. Clément, C. Nicaud. Lyndon words with a fixed standard right factor. In *SODA'04*, ACM-SIAM (2004), 646-647.

[21] F. Bassino, J. Clément, C. Nicaud. The standard factorization of Lyndon words : an average point of view. *Discrete Mathematics*, 290 (2005), 1–25.

[22] F. Bassino, J. Clément, G. Seroussi, A. Viola, Optimal prefix codes for families of two-dimensional geometric distributions, nov. 2005, preprint.

[23] F. Bassino, C. Nicaud, Enumeration and random generation of accessible automata, nov. 2005, *preprint.*

[24] M.-P. Béal, *Codage Symbolique*, Masson, 1993.

[25] M.-P. Béal, D. Perrin, Symbolic dynamics and finite automata, in *Handbook of Formal Languages*, G. Rosenberg and A. Salomaa, Eds., vol. 2, Springer-Verlag, 1997.

[26] M.-P. Béal, D. Perrin, On the generating sequences of regular languages on k symbols, *Journal of the ACM (JACM)*, 50 :6 (2003), 955-980.

[27] D. Beauquier, Minimal automaton for a factorial transitive rational language, *Theoret. Comput. Sci.*, 67 (1989), 65–73.

[28] J. Bernat, Computation of $L_\oplus$ for several cubic Pisot numbers, *J. Autom. Lang. Comb.*, à paraître.

[29] J. Bernat, Arithmetics in β-numeration, preprint available at http ://iml.univ-mrs.fr/ bernat/engautol.ps.

[30] O. Bernardi, A note on Stirling numbers, *preprint.*

[31] J. Berstel, L. Boasson, The set of Lyndon words is not context-free, *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* 63 (1997) 139–140.

[32] J. Berstel, D. Perrin, The beginning of combinatorics on words, *European Journal of Combinatorics*, à paraître.

[33] J. Berstel, M. Pocchiola, Average cost of Duval's algorithm for generating Lyndon words, *Theoret. Comput. Sci.*, 132 (1994), 415–425.

[34] J. Berstel, D. Perrin, *Theory of codes*, Academic Press, 1985.

[35] J. Berstel, Ch. Reutenauer, *Rational series and their languages*, Springer-Verlag, 1988.

[36] A. Bertrand, Développements en base de Pisot et répartition modulo 1, *C. R. Acad. Sci. Paris*, 285 (1977), 419–421.

[37] A. Bertrand-Mathis, Développement en base θ, répartition modulo 1 de la suite $(x\theta^n)_{n\geq 0}$, langages codés et θ-shift, *Bull. Soc. Math. France*, 114 (1986), 271–323.

[38] F. Blanchard, β-expansions and symbolic dynamics, *Theor. Comput. Sci.*, 65 (1989), 131–141.

[39] F. Blanchard, G. Hansel, Languages and subshifts, in *Automata on Infinite Words*, M. Nivat and D. Perrin, Eds., vol. 192 of *Lect. Notes Comput. Sci.*, p. 138–146, Springer, 1985.

[40] F. Blanchard, G. Hansel Systèmes codés, *Theor. Comput. Sci.*, 44 (1986), 17–49.

[41] M. Bousquet-Mélou, Algebraic generating functions in enumerative combinatorics and context-Free languages, *STACS 2005*, vol. 3404 in *Lect. Notes Comput. Sci.* (2005), 18–35.

[42] D. W. Boyd, Salem numbers of degree four have periodic expansions, In *Number theory*, pages 57–64, de Gruyter, 1989.

[43] D. W. Boyd, On beta expansions for Pisot numbers, *Mathematics of Computation*, 65 :214 (1996), 841–860.

[44] D. W. Boyd, On the beta expansion for Salem numbers of degree 6, *Mathematics of Computation*, 65 :214 (1996), 861–875.

[45] Č. Burdík, C. Frougny, J.-P. Gazeau, R. Krejcar, Beta-integers as natural counting systems for quasicrystals, *J. Phys. A, Math. Gen.*, **31** (1998) 6449–6472.

[46] K. Chen, R. Fox, R. Lyndon, Free differential calculus IV : The quotient groups of the lower central series, *Ann. Math.*, 58 (1958), 81–95.

[47] T. M. Cover, J. A. Thomas, *Information theory*, John Wiley & Sons, 1991.

[48] N. G. de Bruijn, *Asymptotic method in analysis*, North Holland, 1961.

[49] J.-M. Champarnaud, T. Paranthoën, Random generation of DFAs, *Theoret. Comput. Sci.*, 330 (2005), 221–235.

[50] B. N. Delone, Neue darstellung der geometrischen kristallographie, *Zeit. Kristallographie*, 84 (1932), 109–149.

[51] A. Denise, P. Zimmermann, Uniform random generation of decomposable structures using floating-point arithmetics, *Theoret. Comput. Sci.*, 218 (1999), 233–248.

[52] M. Denker, C. Grillenberger, K. Sigmund *Ergodic theory on comact spaces* volume 527 in Lect. Notes Math., Springer, 1976.

[53] M. Domaratzki, D. Kisman, J. Shallit, On the number of distinct languages accepted by finite automata with n states, *J. Autom. Lang. Comb.*, no. 4 (2002), 469–486.

[54] P. Duchon, P. Flajolet, G. Louchard, G. Schaeffer, Boltzmann Samplers for the Random Generation of Combinatorial Structures, *Combinatorics, Probability, and Computing*, Special issue on Analysis of Algorithms, 13 (2004), 577–625.

[55] J.-P. Duval, Factorizing words over an ordered alphabet, *J. Algorithms*, 4 (1983), 363–381.

[56] S. Eilenberg, *Automata, languages and machines*, Vol. A, Academic Press, 1974.

[57] A. Elkharrat, C. Frougny, J.P. Gazeau, J.-L. Verger-Gaugry, Symmetry groups for beta-lattices, *Theoret. Comput. Sci.*, 319 (2004), 281-305.

[58] V. Elser, Indexing problems in quasicrystal diffraction, *Phys. Rev.*, B32 (1985), 4892–4898.

[59] W. Feller, *An introduction to probability theory and its applications*, 3rd Edition, Vol. 1, Wiley, 1968.

[60] P. Flajolet, X. Gourdon, D. Panario, The complete analysis of a polynomial factorization algorithm over finite fields, *J. Algorithms*, 40 (2001), 37–81.

[61] P. Flajolet, R. Sedgewick, *Analytic combinatorics*, Book in preparation, (672p.+x. Version of August 16, 2005 is available at http ://www.algo.inria.fr/flajolet/publist.html).

[62] P. Flajolet, R. Sedgewick, *An introduction to the analysis of algorithms*, Addison-Wesley Publishing Company, 1996.

[63] P. Flajolet, M. Soria, The cycle construction, *SIAM J. Disc. Math.*, 4 (1991), 58–60.

[64] P. Flajolet, P. Zimmermann, B. Van Cutsem, A calculus of random generation of labelled combinatorial structures, *Theoret. Comput. Sci.*, 132 (1994), no. 1-2, 1–35.

[65] L. Flatto, J. Lagarias, B. Poonen, The zeta function of the beta transformation, *Ergod. Th. & Dynam. Sys.*, 14 (1994), 237–266.

[66] H. Fredricksen, J. Maiorana, Necklaces of beads in k colors and k-ary de Bruijn sequences, *Discrete Math.*, 23 :3 (1978), 207–210.

[67] C. Frougny, Representation of numbers and finite automata. *Math. Systems Theory* **25** (1992) 37–60.

[68] C. Frougny, J.-P. Gazeau, R. Krejcar, Additive and multiplicative properties of point sets based on beta-integers, *Theoret. Comput. Sci.* **303** (2003) 491–516.

[69] C. Frougny, B. Solomyak, Finite β-expansions, *Ergod. Th. & Dynam. Sys.*, 12 (1992), 713–723.

[70] F. Gähler, J. Rhyner, Equivalence of the generalized grid and projection methods for the construction of quasiperiodic tilings, *J. Phys.*, A 19 (1986), 267–277.

[71] R. G. Gallager, Variations on a theme by Huffman, *IEEE Trans. Inform. Theory*, IT-24 (1978) 668–674.

[72] R. G. Gallager, D. C. Van Voorhis, Optimal source codes for geometrically distributed integer alphabets, *IEEE Trans.Inform. Theory*, (March 1975) 228–230.

[73] F. R. Gantmatcher, *Matrix theory, volume II*, Chelsea Publishing Company, 1960.

[74] N. Gjini, β-expansion of 1 for quartic Pisot units, *Periodica Mathematica Hungarica*, 47 (2003), 73–87.

[75] M. J. Golin, A combinatorial approach to Golomb forests, *Theoret. Comput. Sci.*, 263 (2001) 283–304.

[76] S. W. Golomb, Run length encodings, *IEEE Trans. Inform. Theory*, IT-12 (1966) 399–401.

[77] S. Golomb, Irreducible polynomials, synchronizing codes, primitive necklaces and cyclotomic algebra, in : *Proc. Conf Combinatorial Math. and Its Appl.*, Univ. of North Carolina Press, Chapel Hill (1969), 358–370.

[78] I. Good, An asymptotic formula for the differences of the powers at zero, *Ann. Math. Statist.*, 32 (1961), 249–256.

[79] L. S. Guimond, Z. Masáková, E. Pelantová, Arithmetics on beta-expansions, *Acta Arithmetica* **112** (2004), 23–40.

[80] D.E. Handelman. Spectral radii of primitive integral companion matrices and log-concave polynomials. In Peter Walters, editor, *Symbolic Dynamic and its Applications*, volume 135 of *Contempory Mathemetics*, pages 231–238. 1992.

[81] F. Harary, Unsolved problems in the enumeration of graphs, *Magyar Tud. Akad. Math. Kutató Int. Közl.*, 5 (1960), 63–95.

[82] M. A. Harrison, A census of finite automata, *Canadian Journal of Mathematics*, 17 (1965), 100–113.

[83] J. E. Hopcroft, J. Ullman, *Introduction to automata theory, languages, and computation*, Addison-Wesley, N. Reading, MA, 1980.

[84] D. A. Huffman, A method for the construction of minimum-redundancy codes, *Proceedings of the IRE*, 40 (1952) 1098–1101.

[85] P. A. Humblet, Optimal source coding for a class of integer alphabets, *IEEE Trans. Inform. Theory*, IT-24 (1978), 110-112.

[86] S. Ito, Y. Takahashi, Markov subshifts and realization of β-expansions, *J. Math. Soc. Japan*, 26 (1994), 33–55.

[87] T. Katayama, M. Okamoto, H. Enomoto, Characterization of the structure-generating functions of regular sets and DOL growth functions. *Inform. and Control*, 36 (1978), 85–101.

[88] K. H. Kim, F. W. Roush, J. B. Wagoner, The shift equivalence problem, *The Mathematical Intelligencer*, 21 (1999), 18–29.

[89] B. P. Kitchens, *Symbolic dynamics : one-sided, two-sided and countable state Markov shifts*, Springer-Verlag, 1997.

[90] D. Knuth, The average time for carry propagation, *Indagationes Mathematicae* 40 (1978) 238–242.

[91] D. Knuth, *The art of computer programming, fascicle 2, generating all tuples and permutations*, Addison Wesley, 2005.

[92] D. Korshunov, Enumeration of finite automata, *Problemy Kibernetiki*, 34 (1978), 5–82, In Russian.

[93] A. D. Korshunov, On the number of non-isomorphic strongly connected finite automata, *Journal of Information Processing and Cybernetics*, 9 (1986), 459–462.

[94] P. Kramer, Nonperiodic central space fillings with isocahedral symmetry using copies of seven elementary cells, *Acta Crysta.*, **A 38** (1984), 257–264.

[95] P. Kramer, N. Negri, On periodic and nonperiodic space fillings of $\mathbb{E}^n$ obtained by projection, *Acta Crysta.*, **A 40** (1984), 580–587.

[96] J. C. Lagarias, Geometric models for quasicrystals : I. Delone sets of finite type, *Discrete Comput. Geom.* **21** (1999) 161–191.

[97] J. C. Lagarias, Meyer's concept of quasicrystal and quasiregular sets, *Commun. Math. Phys.* **179** (1996) 365–376.

[98] D. Levine, P. J. Steinhardt, Quasicrystals : a new class of ordered structures, *Phys. rev. lett.*, 53 (1984), 2477–2480.

[99] D. A. Lind, Entropies and factorizations of topological Markov shifts, *Bull. Amer. Math. Soc.*, (1983), 219–222.

[100] D. A. Lind, The entropies of topological Markov shifts and their related class of algebraic integers, *Ergod. Th. & Dynam. Sys.*, (1984), 283–300.

[101] D. A. Lind, B. H. Marcus, *An introduction to symbolic dynamics and coding*, Cambridge, 1995.

[102] T. Linder, V. Tarokh, K. Zeger, Existence of optimal prefix codes for infinite source alphabets, *IEEE Trans. Inform. Theory*, 43(6) (1997) 2026–2028.

[103] V. Liskovets, The number of connected initial automata, *Kibernetika*, 5 (1969), 16–19, In Russian.

[104] V.A. Liskovets, Enumeration of non-isomorphic strongly connected automata, *Vesci Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk* 3 (1971), 26-30, in Russian.

[105] V.A. Liskovets, Exact enumeration of acyclic automata, in *FPSAC'03*, available at http ://www.i3s.unice.fr/fpsac/FPSAC03/ARTICLES/5.pdf.

[106] M. Lothaire, *Combinatorics on words*, Vol. 17 of Encyclopedia of Mathematics and its Applications, Addison-Wesley, 1983.

[107] M. Lothaire, *Algebraic combinatorics on words*, Vol. 90 of Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2002.

[108] M. Lothaire, *Applied combinatorics on words*, Vol. 105 of Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2005.

[109] W. F. Lunnon, P. A. Pleasants, Quasicrystallographic tilings, *J. Math. Pures and Appl.*, 66 (1987), 217–263.

[110] R. Lyndon, On Burnside problem I, *Trans. American Math. Soc.*, 77 (1954), 202–215.

[111] C. R. MacCluer, The many proofs and applications of Perron's theorem, *SIAM Rev.*, 42 (2000), 487–498.

[112] R. Marchand, E. Zohoorian Azad, *Limit law of the length of the standard right factor of a Lyndon word*, arXiv :math.PR/0407016v1.

[113] C. Martinez, X. Molinero, An efficient generic algorithm for generation of unlabelled cycles, in *Mathematics and Computer Science III*, Trends in Mathematics, Birkhäuser, 2004.

[114] N. Merhav, G. Seroussi, M. J. Weinberger, Optimal Prefix Codes for Sources with Two-Sided Geometric Distributions, *IEEE Trans. Inform. Theory*, 46(1) (2000) 229–236.

[115] Y. Meyer, *Nombres de Pisot, nombres de Salem et analyse harmonique*, Lecture Notes in Math. **117**, Springer-Verlag (1970).

[116] Y. Meyer, *Algebraic numbers and harmonic analysis*, North-Holland (1972).

[117] Y. Meyer, Quasicrystals, diophantine approximation and algebraic numbers, in *Beyond Quasicrystals*, F. Axel and D. Gratias, Eds., Les Éditions de Physique, Springer-Verlag, (1995) 1–16.

[118] R. V. Moody, Meyer sets and their duals, in *The Mathematics of Long-Range Aperiodic Order*, R. V. Moody, Ed., NATO ASI Series C 489, Kluwer (1997) 403–441.

[119] R. V. Moody, Model Sets : a survey, in *From Quasicrystals to More Complex Systems*, F. Axel, F. Denoyer and J.-P. Gazeau, Eds, EDP Sciences and Springer Verlag, (2000) 145–166.

[120] C. Nicaud, *Étude du comportement en moyenne des automates finis et des langages rationnels*, Ph.D. thesis, Université Paris 7, 2000.

[121] A. Nijenhuis, H. S. Wilf, *Combinatorial Algorithms*, 2nd ed., Academic Press, 1978.

[122] D. Panario, B. Richmond, Smallest components in decomposable structures : exp-log class, *Algorithmica*, 29 (2001), 205–226.

[123] W. Parry, On the beta expansions of real numbers, *Acta Math. Acad. Sci. Hung.*, 11 (1960), 401–416.

[124] M. Perles, M. O. Rabin, E. Shamir, The theory of definite automata, *IEEE Trans. Electr. Comp.*, EC-12 (1963), 233–243.

[125] D. Perrin, On positive matrices, *Theoret. Comput. Sci.*, (1992), 357–366.

[126] D. Perrin, Enumerative combinatorics on words , in *Algebraic Combinatorics and Computer Science*, H. Crapo and G.-C. Rota, Eds., Springer Verlag, 2001, 391-430.

[127] C.E. Radke, Enumeration of strongly connected sequential machines, *Inform. Control*, 8 (1965), 377-389.

[128] A. Rényi, Representations for real numbers and their ergodic properties, *Acta Math. Acad. Sci. Hung.*, 8 (1957), 477–493.

[129] C. Reutenauer, *Free Lie algebras*, Oxford University Press, 1993.

[130] R. F. Rice, Some practical universal noiseless coding techniques - Parts I-III, Jet Propulsion Laboratory, Pasaneda, Tech. Rep. *JPL-79-22, JPL-83-17, JPL-91-3* Mar. 1979, Mar. 1983 and Nov. 1991.

[131] R. Robinson, Counting strongly connected finite automata, In *Graph theory with Applications to Algorithms and Computer Science*, Y. Alavi et al., Eds., p. 671–685, Wiley, 1985.

[132] F. Ruskey, J. Sawada, Generating Lyndon brackets : a basis for the n-th homogeneous component of the free Lie algebra, *J. Algorithms*, 46 (2003), 21–26.

[133] J. Sakarovitch, *Eléments de théorie des automates*, Vuibert, 2003. English translation : *Elements of automata theory*, Cambridge University Press, to appear.

[134] A. Salomaa, M. Soittola, *Automata theoretic properties of formal power series*, Springer-Verlag, 1978.

[135] K. Schmidt, On periodic expansions of Pisot numbers and Salem numbers, *Bull. London Math. Soc.*, 12 (1980), 269–278.

[136] M. Senechal, *Quasicrystal and geometry*, Cambridge University Press, 1995.

[137] M. Soittola, Positive rational sequences. *Theoret.Comput.Sci*, 2 (1976),317–322.

[138] B. Solomyak, Conjugates of beta-numbers and the zero-free domain for a class of analytic functions, *Proc. London Math. Soc.*, 68 :3 (1994), 477–498.

[139] J. A. Storer, Ed., *Image and text compression*, Kluwer, 1992.

[140] C. E. Shannon, The mathematical theory of communications, *Bell. Sys. Tech. J.*, 27 (1948), 379–423.

[141] V. Vyssotsky, A counting problem for finite automata, *Tech. report, Bell Telephone Laboratories*, May 1959.

[142] M. J. Weinberger, G. Seroussi, G. Sapiro, The LOCO-I Lossless Image Compression Algorithm : principles and standardization into JPEG-LS, Hewlett-Packard Labs., Palo Alto, Hewlett Packard Tech. Report *HPL-98-193*, 1998.

[143] B. Weiss, Subshifts of finite type and sofic systems, *Monats. für Math.*, 77 (1973), 462–474.

[144] F. Williams, Classification of subshifts of finite type, *Annals of Math., 98 (1973)*, 120–153. Errata ibid. **99** :380–381, 1974.

[145] I. H. Witten, A. Moffat, T. C. Bell, *Managing Gigabytes*, 2nd ed., Morgan Academic, 1999.