

Numéro d'ordre : 147

UNIVERSITÉ BORDEAUX I  
ÉCOLE DOCTORALE MATHÉMATIQUES ET  
INFORMATIQUE

Laboratoire Bordelais de Recherche en Informatique

RAPPORT SCIENTIFIQUE

présenté par

GUY MELANÇON

pour obtenir

L'HABILITATION À DIRIGER DES RECHERCHES

Thèmes : VISUALISATION DE GRAPHES ET COMBINATOIRE DES  
MOTS

Soutenance le 7 décembre 1999

Président	Christophe Reutenauer	Professeur à l'Université de Strasbourg
Rapporteurs	Didier Arquès	Professeur à l'Université de Marne-la-Vallée
	Jean Berstel	Professeur à l'Université de Marne-la-Vallée
	Dave D. Duce	Directeur de recherche, Rutherford Appleton Lab., Royaume-Uni
Examineurs	Maylis Delest	Professeur à l'Université Bordeaux I
	Ivan Herman	Directeur de recherche, CWI Amsterdam, Pays-Bas
	André Raspaud	Professeur à l'Université Bordeaux I



# Table des matières

<b>Remerciements</b>	<b>5</b>
<b>Prologue</b>	<b>9</b>
<b>I Visualisation de graphes</b>	<b>13</b>
1.1 Introduction . . . . .	15
1.1.1 Visualisation d'information . . . . .	15
1.1.2 Visualisation d'information et graphes . . . . .	15
1.1.3 Graph Drawing . . . . .	16
1.1.4 Partitionnement des données et réduction de la complexité visuelle . . . . .	19
1.2 Métriques combinatoires et visualisation de graphes — le cas des arbres . . . . .	21
1.2.1 Premiers exemples de métriques . . . . .	23
1.2.2 Définition d'attributs graphiques . . . . .	25
1.2.3 Bénéfices pour la navigation . . . . .	27
1.2.4 Choix de la métrique . . . . .	27
1.2.5 Distribution statistique et réduction de la complexité vi- suelle . . . . .	30
1.2.6 Vues schématiques . . . . .	32
1.3 Extension au cas des graphes orientés acycliques . . . . .	35
1.3.1 Attributs graphiques et vues schématiques . . . . .	37
1.3.2 Métriques spécifiques aux graphes orientés acycliques . . . . .	37
1.3.3 Combinaison de métriques . . . . .	38
1.3.4 Bénéfices pour la navigation . . . . .	39
1.4 Conclusions et perspectives . . . . .	40
<b>II Combinatoire des mots</b>	<b>43</b>
2.1 Introduction . . . . .	45
2.2 Factorisation de mots . . . . .	46
2.2.1 Mots de Lyndon . . . . .	47
2.2.2 Théorème de factorisation . . . . .	48
2.2.3 Factorisation de Lyndon de mots infinis . . . . .	48
2.2.4 Régularités inévitables . . . . .	50
2.2.5 Cas des mots sturmiens . . . . .	50

2.2.6	Factorisation de Lyndon et facteurs singuliers . . . . .	52
2.2.7	Facteurs des mots sturmiens et mots de Christoffel . . . . .	53
2.3	Mots de de Lyndon équilibrés . . . . .	54
2.3.1	Factorisation standard des mots de Lyndon . . . . .	54
2.3.2	Le cas $A = \{a, b\}$ . . . . .	56
2.3.3	Mots de Lyndon équilibrés à plus de deux lettres . . . . .	57
2.3.4	Mots de Lyndon équilibrés et compositions . . . . .	61
2.3.5	Arbre de Stern-Brocot pour les compositions . . . . .	62
2.3.6	Approximation de points dans le plan . . . . .	64
2.3.7	Mots de Lyndon équilibrés infinis . . . . .	67
2.4	Conclusion et perspectives . . . . .	67
	<b>Bibliographie</b>	<b>69</b>
	<b>Curriculum vitae</b>	<b>77</b>
	<b>Publications</b>	<b>78</b>

# Remerciements

Voilà les mots les plus difficiles à poser sur le papier. La rédaction de ce mémoire étant terminée, je ne peux m'empêcher de penser combien mon parcours fut rempli de surprises. Ma première rencontre avec Jean Berstel eut lieu lors d'une descente en canot le long de la rivière rouge en compagnie de joyeux lurons rassemblés par Pierre Leroux. C'est ce même Pierre Leroux qui m'avait encouragé à lire les écrits de Didier Arquès sur les cartes planaires dans le cadre d'un projet estival du CRSNG, pour les mettre à la sauce « Théorie des espèces » bien appréciée à Montréal. J'étais à mille lieux de penser que ces compagnons bordelais qui avaient investi la conférence de combinatoire de Montréal de 1985 pour l'agrémenter de bijections deviendraient des collègues qui m'accueilleraient dans leur grande maison. Je leur dois d'avoir trouvé dans leur pays mon adorable épouse et de nombreux amis. Qui aurait pu prédire que l'inépuisable énergie de Maylis m'ouvrirait les portes de la Venise du Nord ? Je suis ému que ce mémoire me donne l'occasion de rassembler ces gens rencontrés tout au long de ces années.

Je commencerai par remercier Christophe Reutenauer. La présidence du jury lui revenait naturellement. Il m'a guidé tout au long de mon parcours et le fait encore chaque fois que je vais quérir son avis. Je le remercie de m'avoir poussé à creuser les idées sur la combinatoire des mots de Lyndon équilibrés, que je présente en fin de seconde partie. Son coup de fil me proposant de le rejoindre pour un court séjour à San Diego en 1988, pour poursuivre l'étude des factorisations de Hall–Viennot et des bases des algèbres de Lie libres, a été à l'origine d'une longue chaîne d'événements à laquelle la soutenance de ce mémoire vient s'ajouter. Le temps passé à ses côtés à l'UQAM m'a donné bien plus que les seules connaissances en combinatoire. Je reconnais désormais de nombreux airs d'opéras et peux attester de la difficulté de siffler correctement les aigus de « L'air de la reine de la nuit » de la flûte enchantée. Cher Christophe, je te remercie d'avoir traversé la diagonale est-ouest française pour être ici aujourd'hui et je tiens à souligner combien j'apprécie que tu te sois déplacé pour présider le jury.

Je suis flatté que Jean Berstel ait accepté de rapporter mon mémoire. J'avais eu l'occasion de lui exposer brièvement mes travaux sur la factorisation de Lyndon des mots sturmiens lors d'un court passage à l'institut Gaspard Monge et au LIAFA. Ses réactions m'avaient alors lancé dans une direction très profitable. J'ose espérer que la lecture du paragraphe sur les mots de Lyndon équilibrés lui plaira. Son travail avec Aldo de Luca sur les mots sturmiens, de Christoffel et de Lyndon a été pour moi une source d'inspiration. Je suis aussi heureux qu'il juge mon travail sur la visualisation de graphes. J'ai souvent été spectateur

de conversations où j'ai pu constaté que de nombreux sujets en informatique, théorique ou appliquée, ont peu de secrets pour lui.

Je suis honoré que Didier Arquès ait accepté d'être rapporteur. Je n'étais qu'étudiant de D.E.A. à Montréal lorsqu'il venait présenter ses travaux sur les cartes combinatoires à l'UQAM, peu de temps avant la conférence de combinatoire de 1985 qui allait ensuite donner naissance au colloque SFCA. Il est aujourd'hui à la tête d'une équipe de recherche en graphisme. Je suis heureux qu'il ait employé son double talent de combinatoriste et de graphiste à lire mon mémoire.

Dave Duce me fait le plus grand plaisir en acceptant de rapporter mon mémoire. La recherche en visualisation de graphes exige un large éventail de compétences, et Dave possède la qualité remarquable d'en maîtriser un grand nombre. Je suis heureux qu'il pose sur mon travail en visualisation son regard d'expert.

La soutenance de ce mémoire me permet d'associer André Raspaud à mon activité de chercheur. Nous avons eu souvent l'occasion au LaBRI de discuter théorie des graphes et combinatoire (pour ne pas mentionner nos longues dissertations oenologiques . . .) et ses réponses à mes questions posées depuis Amsterdam m'ont souvent aidé à y voir plus clair.

Maylis Delest retrouvera dans mon mémoire des écrits qui lui sont familiers. Je suis heureux que cette première rencontre à Bordeaux à mon arrivée nous ait conduits jusqu'à une collaboration scientifique. Cher Maylis, je me sens privilégié d'avoir si souvent été associé à tes initiatives. Je te remercie de ton ouverture d'esprit et de ta curiosité pour le monde de la visualisation de graphes qui m'ont permis d'effectuer ce séjour à Amsterdam chez Ivan Herman.

Je dois aussi à *van Gogh* d'avoir rencontré Ivan Herman lors de ses nombreuses visites au LaBRI en 1997 et 1998. Je me souviens du sourire d'Ivan lorsque je lui demandais si le poste qu'il cherchait à pourvoir au CWI pouvait accueillir, non pas un post-doc, mais un chercheur plus mûr, cependant très envieux de tremper dans le monde de la visualisation. Cher Ivan, ton savoir étendu et ton sens critique me permettent chaque jour de prendre du recul et d'apprécier mon activité en visualisation de graphes.

La rédaction de ce mémoire ne se compare pas à la rédaction d'une thèse, mais a tout de même demandé sa part d'efforts (ou est-ce par ce que je ne garde qu'un souvenir vague de la rédaction de ma thèse, ou encore parce que « c'est le métier qui rentre » . . . ?). Ces efforts, je n'aurais pu les faire sans le soutien de ma « petite tribu ». Merci ma Corinne adorée pour les biberons, les bains et les jeux dont tu t'es chargée au cours de cet été consacré au travail ; merci pour les petits cafés avec ou sans sucre qui étaient autant d'encouragements à poursuivre notre projet commun. Merci ma douce Margot pour ton aide à « faire des petits mots sur l'ordinateur » et pour tes innombrables « Pourquoi ? » qui venaient aérer les longues heures passées devant l'écran. Merci Victor pour ces délicieux sourires qui me sortaient de mes pensées dès que tu m'apercevais au bas des escaliers de la maison de Dordogne. Tu me ramenaes *illico* auprès de vous pour continuer les vacances. Merci Jeanine, merci Jean-Charles pour cette chambre d'étudiant où j'allais me réfugier pour trouver les bons mots à coucher sur le papier. Ce château de Dordogne gardera peut-être en mémoire quelques idées que j'ai dû échappées lorsque je réfléchissais à haute voix. Merci pour votre soutien, sans vous je n'en serais pas là !

Je veux aussi profiter de cette occasion pour remercier toutes ces personnes

avec qui j'ai partagé les années qui se sont écoulées depuis mon arrivée en Europe. Je pense d'abord à Xavier qui avait concocté ce projet franco-canadien qui m'a amené dans sa grande maison bordelaise. J'ai trouvé auprès de Robert Cori une oreille attentive pour mes questions et remarques dès mon arrivée à Bordeaux, puis au long des années où nous avons été voisins de couloirs. Bétréma rejoignait souvent nos discussions, ajoutant aux remarques éclairées de Robert des commentaires d'une pertinence toujours cinglante comme lui seul sait le faire. Je pense aussi à Serge Dulucq et Srecko Brlek qui ont souvent séjourné dans mon bureau à une époque où je commençais à m'intéresser à la factorisation des mots infinis. Je dois à Serge d'avoir fait quelques remarques clés sur le mot de Fibonacci qui m'ont ensuite lancé sur la voie gagnante. Cher Serge, je ne voudrais pas manquer de souligner l'aide et le soutien que toi et Nadine m'avez prodigués lors de mon arrivée sur le sol bordelais. J'en garderai toujours le meilleur souvenir. Celui qui le premier a donné un volet interdisciplinaire à mon arrivée à Bordeaux est Jean-Guy Penaud. Je lui dois d'avoir appliqué la méthode bijective à l'ensemble des boulons obtenus en démontant (et remontant) le moteur d'une Peugeot 104. Jean-Marc Fédou, maintenant devenu Niçois, fait partie des collègues qui m'ont ouvert les bras dès mon arrivée. C'est Jean-Marc qui m'a fait découvrir les Pyrénées et les rivières où nous sommes allés taquiner la truite, pour ne pas mentionner les nombreuses heures à discuter combinatoire ou visualisation. J'ai connu le collègue de bureau parfait : il a un second bureau ailleurs sur le campus, ce qui vous laisse libre de réfléchir à haute voix sans paraître idiot. Il est toujours curieux de vos commentaires et jamais lassé de jeter un coup d'oeil à votre écran. Il s'appelle Jean-Claude Lalanne. Il parle un peu fort Jean-Claude lorsqu'il passe un coup de fil ? Mais à qui dois-je d'avoir perdu mon accent québécois au profit de l'accent bordelais, sinon à lui ? Merci aussi à André Arnold qui m'avait écouté lorsque j'envisageais un détachement au CWI et pour les nombreuses signatures qui m'ont été nécessaires lorsqu'il dirigeait le laboratoire. Merci à Yves Métivier pour ses conseils et pour son aide dans l'acheminement des dossiers pour cette habilitation. Je ne voudrais pas oublier mes collègues combinatoristes Mireille, Isabelle, Sacha, Philippe, Gilles, Oliviers (Guibert et Baudon) et Cyril avec qui j'ai eu l'occasion d'échanger idées et points de vue. Et tous les autres membres du LaBRI avec qui j'ai pu partager enseignements, activités de recherche et potins de couloirs. Je terminerai par un remerciement tout spécial pour mon collègue Scott Marshall ici à Amsterdam. Cher Scott, merci de corriger mon anglais, mais surtout merci de m'aider à élargir mon horizon en partageant tous les jours avec moi ton riche savoir informatique.

J'ai une dernière pensée pour Frank Kafka. Je croyais avoir balayé tout l'éventail des formulaires administratifs depuis mon arrivée en France, et je m'étais laissé convaincre du caractère « peu ordonné » de l'administration française. Il n'en est rien, ce mal doit avoir atteint toute la planète et transforme le détachement administratif en un défi de taille. Fonctionnaire français, mais canadien, détaché au ministère des affaires étrangères mais salarié de l'état néerlandais est une bonne combinaison pour provoquer un nombre impressionnant d'exceptions dans les algorithmes des administrations. On aura qu'à dire que je l'ai bien cherché. Mais à mon avis si le patron du mammoth souhaite vraiment favoriser la mobilité des chercheurs, il a du pain sur la planche.

*Amsterdam, novembre 1999*



# Prologue

Ce mémoire comporte deux parties qui correspondent aux deux volets de mon activité de recherche. Je dois ma formation en combinatoire énumérative à mes racines montréalaises et à mon adoption par l'équipe bordelaise en 1991. Je dois à Christophe Reutenauer de m'avoir formé à la combinatoire des mots. Dès 1988, dans le cadre d'un séminaire qu'il dirigeait à l'université du Québec à Montréal, je faisais connaissance avec les mots de Lyndon ; en 1989, j'avais l'occasion de rencontrer celui qui leur a donné naissance. Je garde précieusement une dédicace de Roger Lyndon, auteur de la préface du premier ouvrage de M. Lothaire [Lot83], et la montre à tous ceux que je veux rendre envieux. Depuis cette première rencontre avec les mots de Lyndon, j'ai travaillé en combinatoire des mots où je ne cesse de les revoir et d'en découvrir les innombrables facettes. En 1991 j'ai fait la connaissance de Maylis Delest qui n'allait pas tarder à me décrire les activités de son groupe autour du développement du logiciel *CalICo* pour la visualisation et la manipulation de structures combinatoires (probablement seulement quelques heures après avoir posé le pied en sol bordelais, où je me retrouvais à tanguer sur le bassin et à manger des huîtres pêchées « à la sauvette » autour de l'île aux oiseaux). Mon goût pour l'expérimentation et le calcul symbolique, qui s'était développé au travers du calcul de nombreuses bases des algèbres de Lie libres, allait ici se retrouver en terrain connu. C'est aussi Maylis qui m'a incité à participer aux activités en visualisation de graphes qui démarraient avec l'équipe d'Amsterdam il y a près de deux ans. Le détachement qu'on m'accordait en 1998 m'a permis de me concentrer sur ce volet de mes activités. Le travail accompli avec l'équipe d'Amsterdam a renforcé mon activité dans ce domaine, et dans tous ses aspects.

Lors de la rédaction du mémoire, j'ai décidé de prendre le titre « Mémoire d'habilitation à diriger des recherches » au pied de la lettre et d'écrire un document qui puisse convaincre le lecteur que les thèmes de recherche et problèmes que je décris offrent le potentiel que je leur accorde.

La première partie fait le point sur l'ensemble des résultats en « visualisation de graphes » que nous avons obtenus au cours des deux dernières années, depuis la création du projet « Information Visualization » au CWI. Cette partie ne contient pas de théorème, ni de preuve, comme le mathématicien a l'habitude de trouver dans ses lectures. Le projet est par nature pragmatique et consiste à mettre au point des techniques de visualisation pour les graphes apparaissant dans divers domaines. Le projet est né, d'une certaine manière, de la demande d'une société d'Amsterdam souhaitant visualiser les structures de données internes générées par le compilateur qu'elle mettait alors au point. La visualisation de graphes apparaît aussi naturellement, par exemple, lorsqu'on

fouille une arborescence de fichiers ou un site web. Les réseaux, téléphoniques ou autres, sont encore d'autres exemples de graphes qu'il faut parfois visualiser. Les graphes sont souvent de grande taille et leur visualisation ne peut reposer que sur leur « dessin » (au sens où on l'entend dans le domaine du « Graph Drawing » [BETT99], par exemple ; voir section 1.1.3). Une partie des chercheurs en « visualisation d'information » se penche plus particulièrement sur la visualisation de graphes et plusieurs auteurs ont proposé des alternatives aux dessins classiques pour les graphes. Notre approche consiste à exploiter la panoplie de statistiques, aussi appelées métriques, sur les graphes et de baser sur celles-ci le calcul d'attributs graphiques pour la représentation des graphes. Les métriques ne sont pas inconnues de la communauté de la visualisation d'information [CMS99], mais ne semblent pas avoir été pleinement utilisées pour enrichir la représentation des éléments à visualiser. C'est notre sentiment que cette voie de recherche est très prometteuse et doit être poursuivie.

Ce type de recherche ne peut être mené en solitaire et exige un travail d'équipe de tous les jours. Je n'utilise pas ici le « nous » académique, mais le nous qui désigne les personnes qu'il faut réunir pour voir se réaliser tous les aspects qui entrent en jeu dans ce type de projet. En particulier, et c'est là une différence essentielle avec le travail de recherche théorique, il est nécessaire d'expérimenter pour confirmer nos résultats. Ainsi, une partie importante de notre travail de recherche doit être consacrée au développement d'outils d'expérimentation. Ce travail fait lui aussi apparaître des problèmes algorithmiques sur les graphes [MH98] [BDM99], ou des problèmes de génie logiciel propres à la programmation avec les graphes et à la visualisation. Nos derniers efforts dans cette direction aboutiront à un ensemble de classes Java qui pourront être utilisées comme plate-forme de développement d'un système de visualisation de graphes<sup>1</sup>. Nous avons aussi été en mesure de faire un état des lieux sur la visualisation de graphes [HMM99a], dont certains éléments sont repris ici.

Je me suis limité dans ce mémoire à décrire l'utilisation que nous avons proposée des métriques sur les graphes et à illustrer les résultats obtenus à l'aide de nombreuses figures. Je reprends essentiellement le contenu des articles [HDM98], [HMM<sup>+</sup>99b] et [MHD98] ; je ne décrirai pas le logiciel utilisé pour produire les figures [HMRD99], et ne commenterai pas notre travail de conception et de programmation plus récent. Nous avons d'abord étudié le cas des arbres pour lesquels existe un vaste ensemble de métriques combinatoires. Nos résultats s'étendent au cas des graphes orientés acycliques, qui sont une généralisation naturelle des arbres, mais sont essentiellement calqués sur le cas des arbres. Une vraie extension à une classe de graphes plus vaste reste à faire et le mémoire cherche en partie à convaincre du potentiel de l'approche développée jusqu'à maintenant pour ce cas plus général.

Mon activité en combinatoire des mots remonte aux travaux de ma thèse. Je me concentre dans le mémoire sur l'ensemble des résultats sur les mots infinis que j'ai rassemblés ces dernières années. La factorisation de Lyndon tient dans cette seconde partie du mémoire un rôle important. Mon premier travail en combinatoire des mots [MR89] utilisait les mots de Lyndon pour montrer combinatoirement certaines identités dans les algèbres de Lie libres et autres algèbres voisines. J'ai eu l'occasion par la suite d'apporter quelques contributions en combinatoire algébrique [Mel93], [MR96], [DDKM94] mais je ne les commenterai

---

<sup>1</sup>Voir le site [www.cwi.nl/InfoVisu](http://www.cwi.nl/InfoVisu).

pas ici. Tous ces travaux montrent l'importance de la notion de factorisation en combinatoire des mots, comme l'avait reconnu Schützenberger [Sch59]. Dans ce chapitre de la combinatoire des mots, les mots de Lyndon tiennent un rôle princier. Ils sont de presque toutes les batailles et aident souvent à les gagner.

Les résultats que je présente sont tous nés de la factorisation de Lyndon des mots infinis. J'y montre comment le problème du calcul de la factorisation de Lyndon des mots infinis peut mener à mieux en cerner certains aspects. Le cas important des mots sturmiens tient une place centrale dans cette seconde partie et le calcul de leur factorisation de Lyndon a été à la source de plusieurs questions les concernant. Les résultats de la dernière section sont nouveaux et encore non publiés. J'ai voulu profiter de l'occasion que m'offrait la rédaction de ce mémoire pour les annoncer et confirmer la pertinence des questions soulevées par le calcul de la factorisation de Lyndon des mots sturmiens. Comme c'est souvent le cas dans ce domaine de la combinatoire des mots, les énoncés et preuves sont techniques. Les preuves peuvent rarement être ramenées à des dessins, ou à des manipulations de « viennotique » et je les ai souvent omises. J'ai préféré faire apparaître le cheminement qui m'a conduit dans cette direction et, en restant concis, présenter mes travaux comme un ensemble cohérent de résultats.

Aussi, j'ai résolument limité la discussion sur chacun des deux thèmes pour que le mémoire soit de taille raisonnable et facile à lire. La partie sur la visualisation est par nature moins technique et reprend en détail certains des articles qui sont à sa source. La seconde partie contient un minimum de notations pour permettre d'apprécier les résultats et leurs énoncés. Il faudra se rapporter aux articles cités pour des preuves et une discussion plus complètes.



Première partie

Visualisation de graphes



## 1.1 Introduction

### 1.1.1 Visualisation d'information

La visualisation d'information pourrait être définie comme la visualisation et la navigation de données abstraites. Cette définition englobe l'ensemble des travaux qu'on retrouve dans le domaine, et qui remontent à environ une dizaine d'années [CMS99]. On connaît mieux le domaine de la visualisation scientifique, même si son nom peut laisser croire qu'il s'agit là d'une sous-discipline de la visualisation d'information. La visualisation scientifique s'est développée plus tôt, répondant à des besoins exprimés principalement par des chercheurs. La visualisation d'information s'est développée de manière indépendante, plus près des disciplines d'ingénierie, menant à des applications plus faciles d'accès. Leurs méthodes et buts communs les ont récemment rapprochés, comme l'illustre par exemple le Symposium « Data Visualization » co-sponsorisé par la société *IEEE* et l'association *Eurographics*. Il est toutefois utile de mentionner certaines différences entre les deux disciplines, dans le but de mieux cerner ce qui revient en propre à la visualisation d'information. La visualisation scientifique se penche habituellement sur des structures et des modèles mathématiques, souvent continus. La visualisation d'information s'occupe de structures abstraites comme les structures de données de programmes, structures hypermédias ou organigrammes. La différence fondamentale tient aux objets étudiés : les modèles ou structures mathématiques ont une géométrie associée que le processus de visualisation doit inclure. S'il s'agit de chercher à confirmer une conjecture en construisant une surface possédant certaines propriétés [HP97], la visualisation doit se faire dans l'espace où le problème est défini. Dans le cas de la visualisation d'information, la géométrie doit souvent être inventée. Bien que la visualisation se fasse à partir d'une figure dans le plan, la disposition des données a un impact sur son analyse et c'est ce qu'il faut anticiper. Les aspects concernant les interactions homme-machine rapprochent certainement les deux disciplines. On retrouve d'ailleurs bon nombre de résultats en visualisation dans les actes de la conférence annuelle *ACM SIGCHI*. La raison est qu'en visualisation scientifique les utilisateurs sont souvent des experts (non nécessairement informaticiens), alors qu'en visualisation d'information ils peuvent être de niveaux d'expertise variés. Le type de matériel utilisé pour la visualisation différencie aussi les disciplines. Il n'est pas rare d'utiliser des environnements très spécialisés (comme les CAVE) en visualisation scientifique. La visualisation d'information se limite souvent aux moyens informatiques plus communs.

### 1.1.2 Visualisation d'information et graphes

Une question naturelle se pose lorsqu'il s'agit de visualiser des données : « *Les données viennent-elles équipées de relations ?* ». Dans le cas où la réponse est « *Non* », la représentation des données est à inventer. C'est le cas pour des données statistiques, par exemple, qui sont souvent visualisées comme des nuages de points. Dans le cas où la réponse est « *Oui* », les données à visualiser sont portées par un graphe, les sommets étant les éléments de données et les arêtes (ou arcs), les relations entre ceux-ci. C'est dans ce cadre que tout le reste de la discussion va se situer. Les applications de la visualisation de graphes sont vastes et justifient qu'elle constitue en soi une sous-discipline de la visualisation

d'information. Un espace de fichiers est une structure hiérarchique commune pour tout informaticien. Qui n'a pas eu à naviguer dans une telle structure à la recherche d'un fichier, pour finalement se demander « *Où suis-je dans la structure maintenant ?* », « *Comment dois-je procéder pour retrouver le fichier que je cherche ?* ». D'autres graphes familiers sont, par exemple, les organigrammes ou les classifications taxonomiques d'espèces. Les sites web ou les historiques de navigation sont un autre exemple. La biologie et la chimie utilisent aussi souvent des arbres généalogiques et des schémas moléculaires ou génétiques. Certaines applications sont plus près de l'informatique : systèmes orientés objets (bases d'objets), structures de données (générees lors de la compilation, par exemple), systèmes temps-réels (systèmes de transition, réseaux de Pétri), diagrammes de flux de données, diagrammes entités-relations (UML et bases de données), réseaux sémantiques et de représentations de connaissances, diagrammes PERT (gestion de projet), diagrammes VLSI (circuits logiques) et systèmes de gestion de documents constituent une longue liste d'applications potentielles.

### 1.1.3 Graph Drawing

Pour visualiser un graphe, il faut certainement commencer par le dessiner et on peut se demander en quoi la visualisation de graphes se différencie du problème d'en trouver un dessin (une représentation dans le plan, par exemple, où les sommets sont des points et les arcs, des courbes). Tous les algorithmes de positionnement et les algorithmes de recherche de propriétés (le graphe est-il planaire, est-il acyclique?) constituent un arsenal à disposition pour le développement d'une stratégie de visualisation d'un graphe. Mais les aspects concernant la navigation ou l'exploration d'un graphe ne sont habituellement pas pris en compte par les algorithmes de dessin. La visualisation de graphes se déroule souvent de manière interactive et nécessite une réaction en temps réel du système de visualisation, qui doit alors reposer sur des algorithmes de faible complexité. La navigation du graphe nécessite elle-même le développement de techniques propres et indépendantes du dessin du graphe. Le zoom ou le recadrage du dessin du graphe sont certainement les outils les plus basiques pour la navigation. Les techniques « Focus + Contexte » consistent à offrir un niveau de détail plus grand pour une partie du graphe, tout en maintenant une vue de la globalité de la structure. L'effet de distorsion « Fish-eye » [SB92] est certainement la technique de ce type la plus connue. Certains travaux récents qui s'intéressent à l'utilisation de la géométrie hyperbolique obtiennent des effets similaires [LRP95, Mun97]. Une autre approche consiste à développer des indices visuels pour aider l'utilisateur à développer des repères et à structurer son exploration [HDM98, HMM<sup>+</sup>99b]. C'est dans ce dernier chapitre que se situent les contributions qui seront décrites dans ce mémoire.

Nous allons dans un premier temps décrire comment visualisation de graphes et dessin de graphes (« Graph Drawing ») se comparent et diffèrent. Les deux disciplines ont de manière évidente beaucoup à tirer l'une de l'autre. On retrouve d'ailleurs nombre d'articles concernant la visualisation de graphes dans les dernières éditions des actes de la conférence « Graph Drawing ». Le problème du dessin d'un graphe se formule simplement : *Etant donné un ensemble de sommets et d'arêtes (ou d'arcs), calculer les positions des sommets et les courbes à dessiner pour représenter le graphe.* La bibliographie commentée de Battista



graphe [BMK95, DC98]. Les études d'utilisabilité sont encore peu fréquentes en visualisation d'information et on peut s'attendre à ce qu'elles prennent de l'importance dans le futur et contribuent à cerner les problèmes importants de la discipline.

Le problème de dessiner un graphe a introduit un certain nombre de sous-problèmes pour lesquels des solutions ont été proposées. Ainsi s'est développé un ensemble d'algorithmes qui ne sont pas des algorithmes de dessin en soi, mais qui s'occupent d'étapes de calculs préliminaires au calcul des positions des éléments du graphe. Les cadres du côté gauche du diagramme de la figure 1.1 en évoquent quelques-uns : partitionnement des sommets par couches, extraction d'un sous-graphe planaire, etc. Certains des problèmes posés par ces règles esthétiques correspondent à des problèmes de grande complexité. Par exemple, la minimisation des croisements des arêtes, même dans le cas des graphes bipartis, est un problème *NP-hard* [GJ83, EW94]. Cet état de fait a motivé le développement de nombreuses heuristiques [ES90, BETT99]. Le cadre « Crossing Minimization » de la figure 1.1 contient une liste des heuristiques les plus utilisées pour la minisation des croisements d'arêtes et apparaît comme une étape de pré-calcul pour le dessin d'un graphe (quelconque) pour lequel cet objectif est recherché.

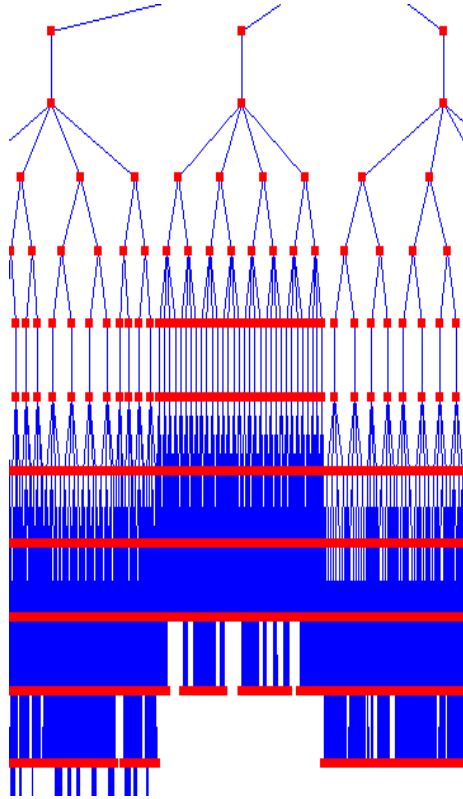


FIG. 1.2: Vue partielle d'un arbre de 30 000 sommets

En visualisation de graphes, le problème de la taille des graphes à visualiser est majeur. La plupart des algorithmes de dessin de graphes doivent être écartés comme seule approche pour visualiser ces graphes, essentiellement à cause de

leur trop grande complexité. Mais la complexité des algorithmes de dessin n'est pas le seul obstacle à contourner lors de la visualisation d'un graphe de grande taille. Un algorithme peut très bien être inefficace, pour un graphe de milliers de sommets, à produire une vue qui permette d'interagir avec celui-ci. Le dessin de l'arbre de la figure 1.2 a été obtenu par l'algorithme de Reingold et Tilford, qui est l'un des algorithmes de dessin d'arbres classiques les plus connus. Sa complexité linéaire et son aptitude à produire des dessins voisins pour des arbres qui diffèrent peu en font un outil central pour la visualisation d'arbres. Cependant, l'arbre de la figure 1.1.3 (dont on ne donne qu'une vue partielle) contient près de 30 000 sommets et il est clair que les feuilles de l'arbre sont trop rapprochées pour qu'un utilisateur puisse questionner les données sous-jacentes à celles-ci sans avoir au préalable à naviguer, en zoomant ou en appliquant des filtres (mise en relief des feuilles ayant certaines propriétés, par exemple). C'est précisément ici que se différencie les deux disciplines. La visualisation de graphes doit se préoccuper de concevoir des outils de navigation pour surmonter les problèmes posés par la taille des graphes, ou de manière générale, par la complexité de l'information sous-jacente au graphe. Des approches nouvelles qui s'éloignent de la perspective traditionnelle du « Graph Drawing » ont déjà été proposées pour la visualisation de grands graphes. NicheWorks [Wil97] permet de visualiser des hiérarchies de plusieurs dizaines de milliers de sommets. Des sommets appartenant à une composante à forte connectivité sont alors placés à proximité, et les arêtes sont tout simplement omises pour alléger la représentation (figure 1.3). H3Viewer [Mun98] mise sur la géométrie hyperbolique pour visualiser des hiérarchies de grande taille; l'argument principal ici est que si le nombre de sommets dans un arbre croît de manière exponentielle, l'espace disponible pour le dessiner ne croît que de manière linéaire, dans le monde euclidien. La géométrie hyperbolique permet alors de profiter d'une croissance en espace qui suit celle de l'arbre. L'utilisation de la troisième dimension a aussi été proposée, l'espoir étant que cette dimension additionnelle donne littéralement « plus d'espace » (cf le survey [HMM99a] qui discute des techniques de visualisation de graphes en 3D).

#### 1.1.4 Partitionnement des données et réduction de la complexité visuelle

Une voie semble prometteuse pour la visualisation de graphe de grande taille. Nous l'avons déjà mentionné, encore peu de travaux ont étudié les aspects cognitifs de la visualisation des graphes. Il est encore difficile de déterminer la taille optimale du graphe, quand il s'agit de permettre à un utilisateur de comprendre et analyser les informations qui s'y trouvent. Mais on peut penser que ce nombre doit être petit, même si une vision globale de la structure peut lui permettre d'élaborer sa stratégie d'exploration. Dans ce sens, il est certainement important d'envisager de réduire la taille des graphes à visualiser. Cette approche est connue sous le nom de « clustering » [Eve74, Mir96] (ou encore « clumping », « grouping », ou « classification »); nous utiliserons le substantif plus français « partitionnement » bien que l'on puisse envisager de découper les données en sous-ensembles non-disjoints. Le partitionnement des éléments d'un graphe présente deux variantes essentielles. On peut le baser purement sur la structure du graphe, ou alors utiliser une information contextuelle — on parle

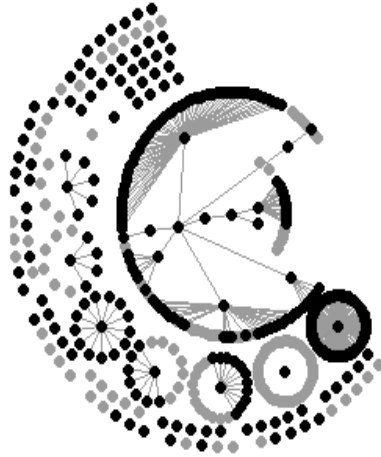


FIG. 1.3: Dessin de grands graphes – allègement de la représentation par suppression des arcs

alors de partitionnement sémantique. Les techniques de partitionnement, structurel ou sémantique, sont nombreuses. Les problèmes posés en termes de partitionnement de données apparaissent naturellement dans la discipline du « data mining », par exemple, mais aussi dans des disciplines variées et non nécessairement voisines. Le récent survey de Alpert [AK95] donne une liste complète des techniques existantes (telles qu'on les utilise dans le domaine de la conception des circuits logiques). Bon nombre de techniques cherchent à calculer une partition des sommets du graphe qui satisfait une contrainte d'optimisation. C'est le cas de la plupart des algorithmes que Alpert place dans la catégorie « Combinatorial Formulations » [AK95, Sect. 5]. Chacun des algorithmes, bien que générique, laisse voir le champ d'applications qui lui a donné naissance. C'est le cas, par exemple, de l'algorithme Min-Delay de Lawler *et al.* [LLT69] qui cherche à minimiser les délais dans un réseau digital (digital network). D'autres techniques reposent sur le calcul de flots maximum (ou minimum), et il faut alors traduire le problème de partitionnement en définissant le flot approprié. D'autres algorithmes basent le partitionnement sur un premier positionnement des sommets du graphe; le partitionnement se fait alors selon les coordonnées assignées aux sommets [DGK98].

Les techniques de partitionnement ont été utilisées de façons diverses en visualisation d'information, mais n'ont pas encore été explorées de manière exhaustive. Le système Narcissus [HDWB95] permet d'obtenir et de visualiser le partitionnement des éléments d'un graphe en animant un algorithme de positionnement basé sur un modèle de forces (Force-Directed placement) : les sommets sont assimilés à des corps sur lesquels agissent des forces déterminées par les arêtes du graphe. Un travail récent [He 99] partitionne les éléments d'un graphe modélisant un réseau et introduit de nouveaux sommets autour desquels ils placent les sommets regroupés lors du partitionnement. Ils ne dessinent alors des

arêtes qu'entre ces nouveaux sommets, réduisant ainsi la complexité visuelle du dessin. D'autres auteurs définissent la forme que peuvent prendre les méta-sommets, ou définissent des icônes ou des textures à utiliser pour les représenter (cf, par exemple, [FS98]). C'est notre sentiment que l'utilisation des techniques de partitionnement en visualisation doit permettre de définir de nouvelles techniques de navigation des graphes. Nous indiquerons comment les résultats qui seront présentés dans les sections à venir peuvent être vus comme une contribution dans cette voie.

Nous allons d'abord définir ce que nous entendons par « métrique » associée aux sommets (ou aux arêtes ou arcs) d'un graphe. Les combinatoristes parlent plus volontiers d'une « statistique », mais nous nous en tiendrons à la terminologie acceptée en visualisation d'information<sup>1</sup>. Nous indiquerons comment une métrique peut être utilisée pour induire des attributs graphiques dans la représentation d'un graphe. Les métriques utilisées peuvent dépendre du type de graphes à visualiser. Nous nous concentrerons d'abord sur le cas, important, des arbres, puis sur celui des graphes orientés acycliques qui sont une généralisation naturelle des arbres. Nous indiquerons ensuite comment ces mêmes mesures permettent de calculer ce que nous appelons le squelette d'un graphe. Les résultats que nous présentons sont le fruit de recherches et d'expérimentations qui se sont déroulées sur les deux dernières années et qui ont déjà fait l'objet de publications [HDM98, HMM<sup>+</sup>99b, MHD98, HMRD99]. Les idées sous-jacentes sont souvent simples. Cependant, les efforts qui ont été nécessaires à leur mise-en-oeuvre doivent être mentionnés. Le logiciel utilisé pour produire les images présentées dans le texte a permis d'implémenter et de valider les techniques décrites dans ce chapitre. Bien que nos résultats ne portent pas sur le dessin de graphes en tant que tel, il nous a aussi été nécessaire d'implémenter des versions adaptées d'algorithmes de dessin connus.

Chacun des paragraphes du texte suit la même logique et présente une idée de l'ensemble de notre travail. Après une brève description des idées mises en avant, nous présentons des figures pour les illustrer. Le chapitre se termine par un commentaire critique concernant les valeurs des idées présentées. Nous concluons en indiquant les généralisations et directions de recherche qu'elles suggèrent.

## 1.2 Métriques combinatoires et visualisation de graphes — le cas des arbres

Schneiderman [Sch96] suggère que lors de l'exploration, l'utilisateur développe d'abord une image de la globalité de la structure, pour ensuite appliquer une séquence de filtres ou de transformations avant d'arriver finalement aux détails portés par la structure. La nécessité de préserver cette image mentale que développe l'utilisateur de la structure a été évoquée dans de nombreux travaux [Nor95, MELS95, HDM98]. Ce constat confirme la nécessité de proposer à l'utilisateur des indices visuels pour l'aider à former cette image et à la préserver au cours de son exploration. La figure XX montre deux images du

---

<sup>1</sup>Cette appellation n'a donc rien en commun avec le sens qu'on lui prête habituellement en mathématiques.

même arbre. L'image du bas est enrichie d'indices visuels et indique l'importance relative d'un sous-arbre en attribuant à la couleur de ses arêtes une intensité et une épaisseur variable. L'intensité attribuée à un sommet (ou une arête ou un arc) dépend d'une valeur numérique associée à chaque sommet de l'arbre.

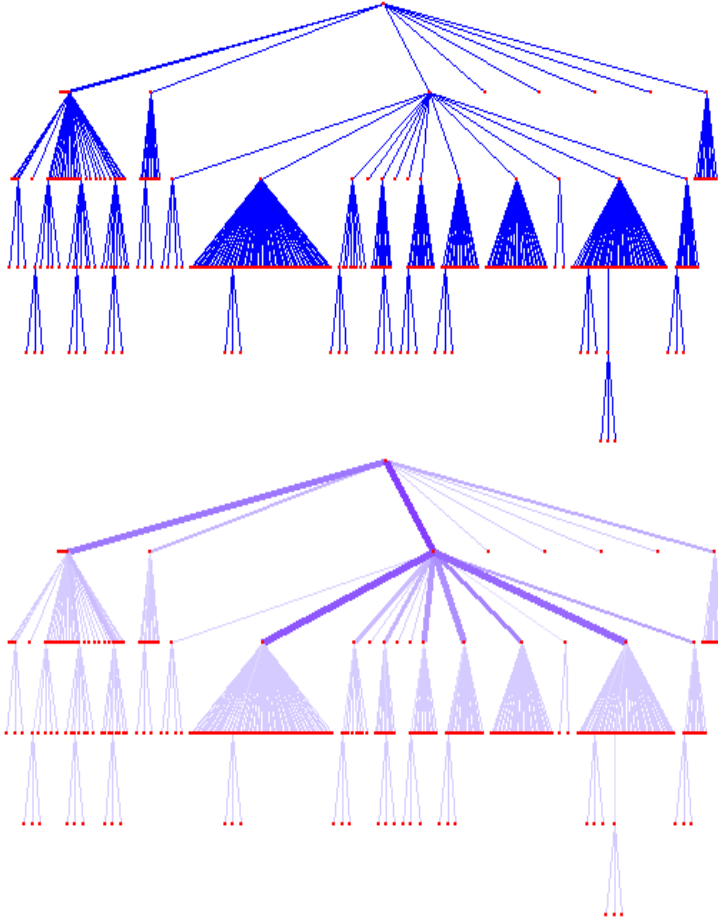


FIG. 1.4: Dessin d'un arbre. Sans attributs (haut) et avec attributs graphiques (bas).

Une *métrie* est une application associant à chaque sommet d'un graphe une valeur numérique. Nous parlerons aussi de la *métrie d'un sommet* pour désigner la valeur qui lui est associée. Dans certains cas, un sommet du graphe peut avoir un sous-graphe associé et sa valeur peut alors être interprétée comme une mesure associée au sous-graphe. La méthode que nous proposons suppose donnée une métrie pour le graphe à visualiser. Des attributs graphiques du dessin du graphe sont ensuite calculés, basés sur l'ensemble des valeurs des éléments du graphes pour la métrie utilisée.

Cette idée, formulée de manière aussi générale, est d'une décevante simplicité. Elle peut cependant être délicate à mettre en oeuvre. Le choix de la

métrique à calculer est déterminant. Il faut évidemment veiller à ce qu'elle ait un sens pour l'utilisateur. Le but recherché doit être de concevoir une métrique qui reflète le *degré d'intérêt* que l'utilisateur peut avoir pour un sommet ou pour son voisinage dans le graphe. De plus, les valeurs calculées ne se soumettent parfois pas à une utilisation directe. Il faut alors composer la métrique avec une autre application afin de pouvoir travailler dans un intervalle adéquat. On peut penser aux triplets de nombres flottants définissant souvent la couleur dans les applications ; ces valeurs sont habituellement situées dans l'intervalle  $[0, 1]$ . Cette contrainte impose donc que les valeurs métriques des sommets du graphes soient envoyés dans l'intervalle  $[0, 1]$ . Plusieurs applications peuvent alors être utilisées et toutes sont susceptibles de produire des effets différents.

La combinatoire énumérative regorge de nombre définis sur les graphes pour en mesurer les propriétés. Le *degré d'un sommet* (degré entrant ou degré sortant dans le cas d'un graphe orienté) est certainement l'exemple le plus élémentaire. La distance à un sommet fixe du graphe est un autre exemple. Dans le cas où le graphe est un arbre et où le sommet choisi est la racine, la valeur qu'on obtient est appelée la *profondeur du sommet dans l'arbre*. La *profondeur de l'arbre* est, elle, égale à la valeur maximale atteinte par ses sommets et peut se définir de manière récursive (cf Eq. (1.3)). Botafogo *et al.* [BRS92], dans un travail se penchant sur la conception de documents hypertextes, mentionnent ces deux exemples et beaucoup d'autres basés sur le calcul des distances dans un graphe. L'étude des réseaux en psychologie ou en sociologie utilise des métriques spécifiques qui permettent de quantifier d'autres aspects des graphes étudiés dans ces disciplines [BKW99].

### 1.2.1 Premiers exemples de métriques

L'approche décrite dans ce paragraphe a déjà été exposée dans un cadre légèrement plus restreint [HDM98].

La métrique profondeur que nous avons définie plus haut n'est qu'une des nombreuses métriques définies pour les arbres (on parle aussi parfois de la *hauteur* d'un sommet dans l'arbre). Nous avons aussi mentionné que dans un arbre, tout sommet induit un sous-arbre, celui des sommets qui sont ses descendants. Nous identifierons souvent le sommet et son sous-arbre et parlerons sans distinction de la métrique du sommet ou de la métrique de son sous-arbre associé. La *largeur* d'un sous-arbre donne le nombre de feuilles qu'il contient. Le *nombre de Strahler* d'un arbre binaire se définit récursivement et exige que nous introduisions quelques notations. Un arbre binaire est identifié à un triplet  $B = (r, B_1, B_2)$  où  $r$  est sa racine et  $B_1$  et  $B_2$  sont respectivement ses sous-arbres gauche et droit. On désignera par  $s(B)$  le nombre de Strahler d'un arbre binaire. Il se définit par :

$$s(B) = \begin{cases} 1 & \text{si } B \text{ est réduit à un sommet} \\ s + 1 & \text{si } s = s(B_1) = s(B_2) \\ \max(s(B_1), s(B_2)) & \text{si } s(B_1) \neq s(B_2) \end{cases} \quad (1.1)$$

La figure 1.5 illustre le calcul du nombre de Strahler pour tous les sommets d'un arbre binaire. Ces nombres avaient été introduits par Horton [Hor45], repris plus tard par Strahler [Str52], pour étudier la morphologie de bassins

hydrologiques modélisés par des arbres binaires. Il a depuis été utilisé en infographie pour produire des images réalistes d'arbres [VEJA89, DK96], et en biologie moléculaire [VC85]. Il apparaît donc comme une mesure quantitative de la structure d'un arbre.

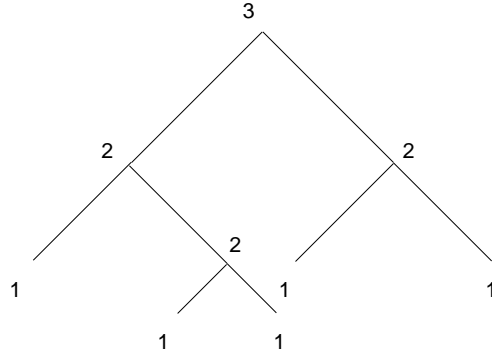


FIG. 1.5: Nombre de Strahler des arbres binaires.

Les arbres que nous allons considérer seront d'arités quelconques et il nous faut définir pour ceux-ci une métrique qui généralise le nombre de Strahler pour les arbres binaires. Plusieurs généralisations sont possibles. Introduisons d'abord les notations nécessaires. Nous identifierons un arbre d'arité quelconque avec un  $n$ -uplet  $(r, T_1, \dots, T_k)$  où  $T_1, \dots, T_k$  sont eux-mêmes des arbres d'arités quelconques. Dans [HDM98], nous avons proposés une extension du nombre de Strahler en posant :

$$s(T) = \begin{cases} 1 & \text{si } T \text{ est réduit à un sommet} \\ s + k - 1 & \text{si } s = s(T_1) = \dots = s(T_k) \\ \max(s(T_1), \dots, s(T_k)) + 1 & \text{sinon} \\ k - 2 & \end{cases} \quad (1.2)$$

On voit facilement que cette métrique coïncide avec le nombre de Strahler défini en (1.1) lorsqu'appliqué aux arbres binaires (cf figure 1.6). D'autres généralisations sont possibles. On sait que le nombre de Strahler d'un arbre binaire calcule le nombre de registres nécessaires à l'évaluation d'une expression arithmétique ayant cet arbre pour structure sous-jacente (on n'utilise pas l'associativité des opérations). On peut généraliser le nombre de Strahler de manière à préserver cette propriété [DF99], et le définir pour qu'il donne précisément cette valeur. Nous ne donnerons pas ici d'expression explicite pour son calcul.

On peut aussi donner des définitions précises pour la profondeur  $d(T)$  et la largeur  $w(T)$  d'un arbre  $T$  quelconque. Elles sont :

$$d(T) = \begin{cases} 0 & \text{si } T \text{ est réduit à un sommet} \\ \max(d(T_1), \dots, d(T_k)) + 1 & \text{sinon} \end{cases} \quad (1.3)$$

et

$$w(T) = \begin{cases} 1 & \text{si } T \text{ est réduit à un sommet} \\ w(T_1) + \dots + w(T_k) & \text{sinon} \end{cases} \quad (1.4)$$

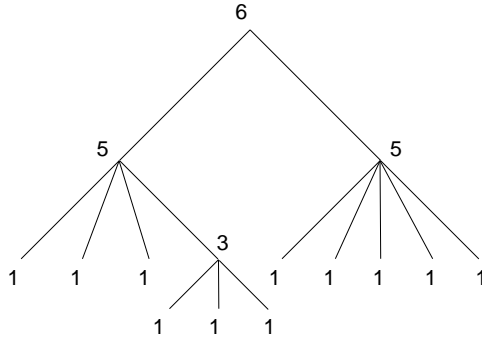


FIG. 1.6: Nombre de Strahler des arbres d'arité quelconque.

La figure 1. illustre les métriques largeur et profondeur pour un arbre quelconque. Il existe encore d'autres exemples de métriques. Le nombre de sommets d'un arbre est aussi une métrique pour laquelle on peut donner une formulation similaire aux équations (1.2), (1.3) et (1.4). Une autre métrique possible est la longueur de cheminement (la somme des longueurs des chemins d'un sommet à toutes ses feuilles). Nous ne donnerons pas une liste exhaustive des métriques sur les arbres. Nous aurons plus loin l'occasion de définir d'autres métriques (cf section 1.3).

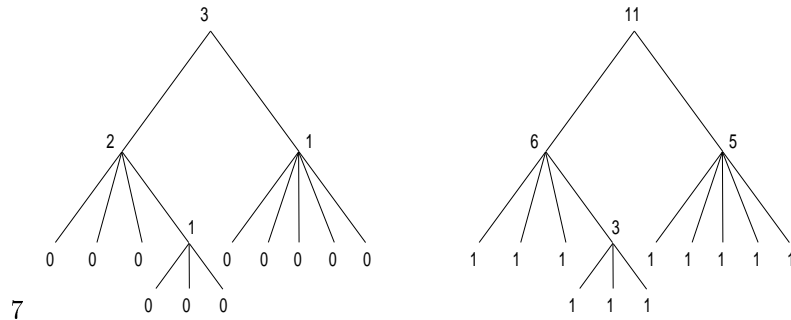


FIG. 1.7: Métriques profondeur (gauche) et largeur (droite).

### 1.2.2 Définition d'attributs graphiques

Toutes les métriques sur les arbres définies plus haut sont croissantes lorsque l'on parcourt un chemin allant d'un sommet vers la racine. La largeur et le nombre de Strahler sont des métriques croissantes au sens large, la profondeur l'est au sens strict. Il est raisonnable de penser que c'est là une caractéristique de la plupart des métriques « naturelles » définies sur les arbres. Il est aussi raisonnable d'affirmer que les éléments plus près de la racine de l'arbre sont plus importants dans la structure. Cette interprétation est confirmée par les travaux de Horton et Strahler et est certainement plausible dans le cas où l'arbre représente une arborescence de fichiers, et pour les autres métriques que nous

avons définies. Il est donc logique de chercher à faire ressortir les éléments de l'arbre associés aux valeurs les plus élevées.

La figure 1.4 en page 22 illustre l'effet obtenu lorsqu'on utilise le nombre de Strahler (1.2) pour définir la couleur et l'épaisseur des arêtes de l'arbre. L'arbre est dessiné avec l'algorithme classique de Reingold-Tilford [RT81]. Il apparaît sans attribut graphique en haut (toutes les arêtes ont même couleur et même épaisseur), et avec attributs graphiques au bas de la figure. Cet indice visuel permet de localiser les sommets correspondant aux sous-arbres les plus importants au sens de la métrique de Strahler.

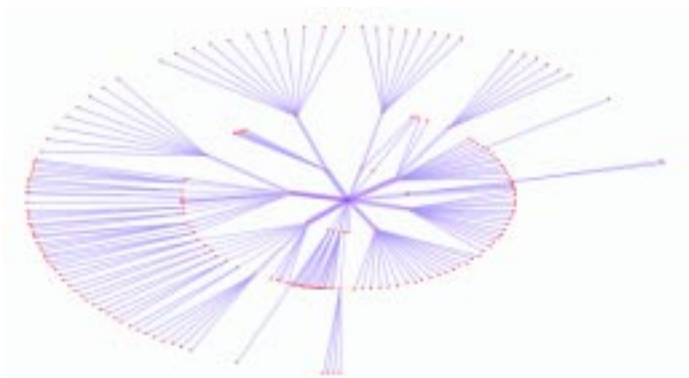


FIG. 1.8: Vue enrichie d'un arbre; positionnement radial et métrique « largeur ».

Le calcul des attributs graphiques, dans ce cas précis, se fait en deux étapes. On choisit d'abord des épaisseurs minimale et maximale  $m$ ,  $M$  pour les arêtes et l'intervalle des valeurs métriques est mis en bijection avec l'intervalle  $[m, M]$ . Dans le cas de la figure 1.4, l'épaisseur d'une arête  $(v, v')$  orientée *vers* la racine est égale au nombre de l'intervalle  $[m, M]$  associé au sommet  $v$ . On dessine alors pour chaque arête un rectangle. Pour obtenir un effet plus continu on peut calculer pour chaque arête un quadrilatère en utilisant les deux valeurs associées à  $v$  et  $v'$ . L'effet obtenu est similaire à un réseau de cours d'eau, les segments les plus larges transportant une somme d'information plus importante. On peut ensuite choisir une couleur et faire varier la saturation de la couleur en fonction des valeurs métriques des sommets.

La figure 1.8 illustre le même arbre, mais dessiné avec l'algorithme de positionnement radial [Ead92]. C'est là un point important. Les métriques que nous avons introduites plus haut sont « structurelles », dans le sens où elles sont définies pour l'arbre lui-même, sans rapport avec la représentation qu'on en donne. Ainsi, notre technique n'est pas tributaire de l'algorithme de positionnement utilisé, mais peut *a priori* être appliquée à toutes représentations des arbres. Par exemple, la figure 1.9 illustre le résultat obtenu lorsque l'algorithme de positionnement place les sous-arbres dans des « ballons » [MH98].

On peut bien entendu imaginer d'autres variantes pour enrichir les représentations des arbres. On peut, par exemple, induire un dégradé de couleur le long d'une arête pour adoucir la variation de couleur le long d'un chemin de la racine vers une feuille. (Nous ne proposons pas de figure ici à cause de la qualité trop pauvre des images à l'impression). On pourrait de la même manière chercher à

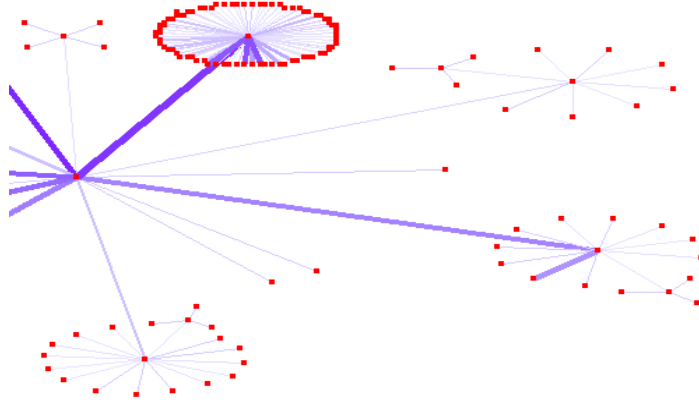


FIG. 1.9: Vue (partielle) enrichie d'un arbre; positionnement « ballon » et métrique de Strahler.

obtenir des effets de volume, de relief et/ou d'ombre en dessinant les graphes en 3D.

### 1.2.3 Bénéfices pour la navigation

L'objectif recherché ici est de concevoir, en se basant sur les valeurs métriques des éléments de l'arbre, des indices visuels qui donnent une information quantitative sur la complexité de l'arbre. Il faudrait, pour affirmer sans aucun doute le bénéfice acquis pour la navigation de l'arbre, pouvoir effectuer une étude auprès d'utilisateurs. Les exemples étudiés semblent toutefois indiquer que les attributs graphiques aident l'utilisateur dans son exploration de l'arbre. Nous allons définir plus loin d'autres types d'attributs qui viendront renforcer cette affirmation. La figure 1.9 plus haut montre une partie d'un arbre après zoom et recadrage. On peut déjà, à partir de cet exemple, constater le bénéfice obtenu pour la navigation. Il semble impossible, sans les attributs graphiques, d'affirmer par exemple où se trouve la racine de l'arbre ou de trouver le chemin à suivre pour aller dans la direction de sous-arbres de relative importance, mais invisibles à l'écran. Dans ce cas, les veines de couleur aident l'utilisateur dans son choix. Elles indiquent clairement que pour retrouver la racine il faut suivre la veine de couleur qui remonte vers le nord-ouest. On peut aussi observer que, bien qu'on ne les voit pas, des sous-arbres relativement importants se trouvent dans la direction sud-ouest.

### 1.2.4 Choix de la métrique

Les métriques (1.2), (1.3) et (1.4) que nous avons présentées jusqu'ici n'ont pas de caractère universel. Leur qualité tient dans leur capacité à quantifier l'importance d'un sous-arbre d'un sommet dans le réseau sous-jacent à l'arbre étudié. Elles n'ont de valeur que dans le cas où la mesure fournie est pertinente

par rapport au contexte. Ce qu'il faut en retenir est plutôt le schéma de calcul qui est suivi pour attribuer une valeur aux sommets de l'arbre. Il serait futile de chercher à caractériser les domaines pour lesquels une métrique donnée est pertinente, ou de chercher à dresser un catalogue des variantes d'une métrique pour tous les domaines d'applications possibles.

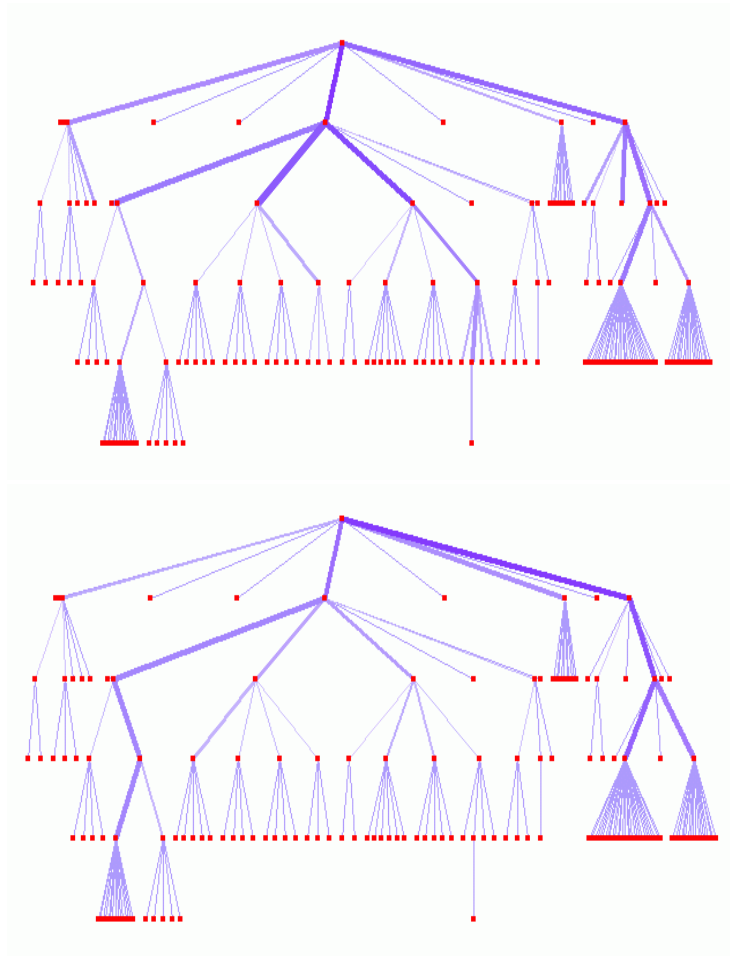


FIG. 1.10: Attributs visuels pour les versions avec (haut) et sans poids (bas) pour la métrique de Strahler.

Mais étant donné une application, l'implémenteur voudra certainement adapter une métrique à son contexte. On peut imaginer que des poids (valeurs numériques) soient attachés aux sommets de l'arbre. Par exemple, dans le cas où l'arbre représente une arborescence de fichiers, chaque feuille pourrait avoir un poids reflétant la taille du fichier correspondant. On peut alors utiliser une version « pondérée » de la métrique afin de répercuter ce poids tout au long du calcul en remontant vers la racine. De manière plus générale, on peut admettre pour tout sommet, donc pour tout sous-arbre  $T$ , un poids  $\omega(T)$  qui peut être nul. Par exemple, la version pondérée de la métrique de Strahler se définirait comme

suit :

$$s(T) = \begin{cases} 1 + \omega(f) & \text{si } T \text{ est réduit à un feuille } f \\ s + k - 1 + \omega(T) & \text{si } s = s(T_1) = \dots = s(T_k) \\ \max(s(T_1), \dots, s(T_k)) & + \text{ sinon} \\ k - 2 + \omega(T) \end{cases} \quad (1.5)$$

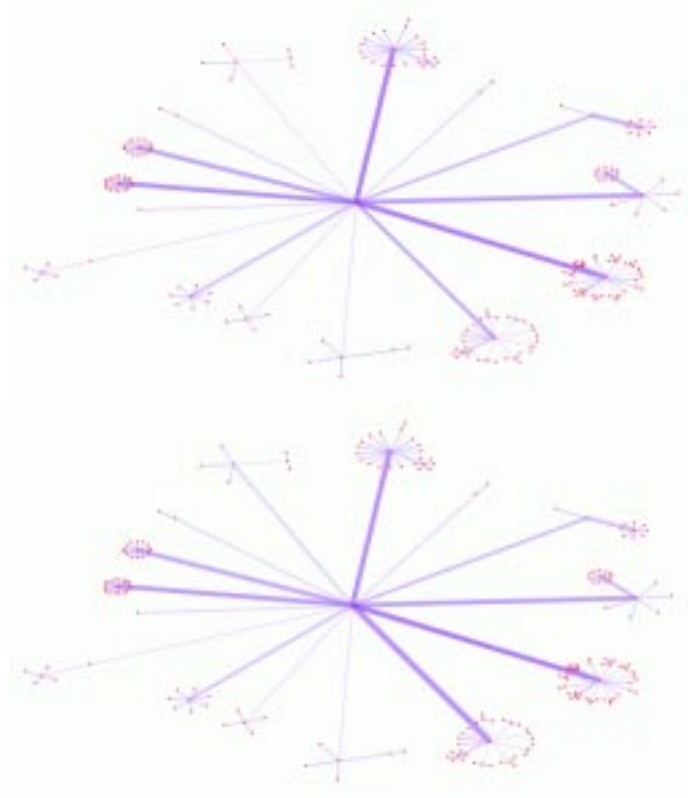


FIG. 1.11: Les métriques *Strahler* (haut) et *largeur* (bas) appliquée au même arbre induisent des attributs visuels sensiblement différents.

Noter que la valeur obtenue n'est pas égale au nombre de Strahler de l'arbre auquel on doit ajouter le poids de sa racine. La figure 1.10 illustre le même arbre, avec attributs graphiques sans poids (vue du bas) et avec poids (vue du haut). Seulement certains sommets ont un poids non-nul. Comme on s'y attend, les sous-arbres avec poids ressortent de manière plus évidente. Nous discuterons plus loin (section 1.3.3) d'autres façons d'adapter au contexte les métriques utilisées.

L'introduction d'une dépendance par rapport au contexte ne libère pas le concepteur du choix de la métrique à implémenter pour refléter le degré d'intérêt de l'utilisateur pour certaines composantes de l'arbre. Des éléments différents

du même arbre seront mis en évidence selon la métrique choisie. La figure 1.11 montre une vue partielle d'un arbre dont la représentation est enrichie à l'aide de deux métriques différentes. On observe que le sous-arbre placé au coin inférieur droit est mis en évidence avec la métrique de Strahler (figure du haut) alors qu'il ne l'est pas (ou moins) pour la métrique « largeur » (figure du bas). Il en va de même pour l'arête qui fuit vers le coin supérieur gauche et pour le sous-arbre du coin supérieur droit. On ne peut formuler aucune règle permettant de décider de manière absolue de la pertinence d'une métrique. L'application devrait pouvoir déterminer la ou les métriques qui lui sont applicables et utiles. En fin de ligne, il revient peut-être à l'utilisateur de faire ce choix.

### 1.2.5 Distribution statistique et réduction de la complexité visuelle

Nous avons déjà évoqué le problème d'espace lors de la visualisation de graphe de grande taille. On peut rencontrer un problème similaire, sans que le graphe à visualiser ne soit trop grand. Un arbre qui est anormalement profond et effilé donnera à l'écran une image difficile à lire qui occupera une zone très peu dense d'information. En effet, pour afficher l'arbre il faudra utiliser une zone rectangulaire de hauteur considérable. Et si on veut l'inclure dans une fenêtre aux proportions équilibrées, il faudra accorder au rectangle une base qui sera plusieurs fois égale à la largeur effective de l'arbre. Le même genre de problème peut se poser avec des arbres qui sont anormalement peu profonds et larges. Ce constat demande qu'on précise ce qu'on entend par « effilé », « profond », « peu profond » ou « large ».

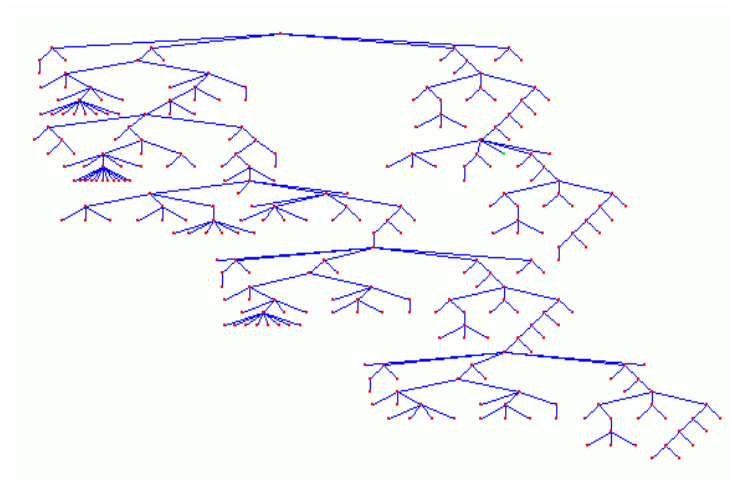


FIG. 1.12: L'arbre original auquel on applique la procédure automatique pour fermer les sous-arbres « anormaux »

La connaissance de la distribution statistique d'une métrique, ou à tout le moins de certaines informations comme la moyenne (pour l'ensemble des arbres d'une taille donnée), permet de préciser les notions comme « trop large » ou « peu profond ». Dans le cas où il est possible de prédire si un arbre présente un

profil normal ou anormal, on peut visiter les sommets de l'arbre et « fermer » les sous-arbres anormaux, c'est-à-dire les supprimer pour ne conserver que leur racine. Ce sommet devient alors une feuille d'un nouvel arbre et on peut l'afficher de manière à le mettre en évidence et indiquer à l'utilisateur qu'il s'agit de la racine d'un sous-arbre qui est masqué. L'utilisateur pourra retrouver le sous-arbre masqué, à condition que l'application l'ait mémorisée. Le but recherché est de pouvoir présenter à l'utilisateur un arbre dont le dessin est équilibré et plus facile à lire et à analyser.

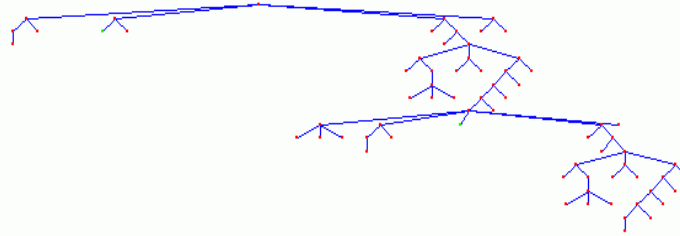


FIG. 1.13: L'arbre obtenu en fermant les sous-arbres « anormaux ». Les sommets ont gardé leur position originale (et sont marqués en vert).

Déterminer les propriétés statistiques des métriques peut être, en général, un problème combinatoire difficile. Le cas des arbres a heureusement déjà fait l'objet de plusieurs travaux. Un article récent de Drmota [Drm97] présente une approche générale qui permet de répondre à notre question, dans le cas où la série génératrice de la métrique satisfait certaines conditions. Nous n'entrerons pas ici dans les détails. Précisons seulement que sa méthode permet de montrer facilement que la largeur, sur l'ensemble des arbres (à  $n$  sommets), est une statistique qui suit une loi qui tend vers une distribution normale (lorsque  $n \rightarrow \infty$ ). Ce résultat avait déjà été établi par Kreweras et Mozskowski [KM86]. Drmota précise que d'autres paramètres (sur les arbres) admettent une limite qui approche une distribution normale.

Précisons ici l'algorithme qui détermine si un sous-arbre doit être masqué et remplacé par un sommet [HDM98, Sect. 5.2], dans le cas où la métrique utilisée pour l'arbre est la largeur (nombre de feuilles).

- On s'intéresse aux probabilités  $p_{n,k} = \mathcal{P}(\# \text{ feuilles} = k)$  qu'un arbre à  $n$  sommets ait  $k$  feuilles (parmi les arbres à  $n$  sommets). On calcule facilement la moyenne  $\mu_n = n/2$  (la distribution est symétrique).
- En vertu du résultat de Drmota, on peut supposer que la distribution est normale et approximer l'écart-type de la distribution par  $\sigma = \sqrt{n/8}$ . On peut aussi déterminer une valeur  $c = c_\alpha$  et construire un intervalle (symétrique) autour de la moyenne  $I_\alpha = [n/2 - c\sqrt{n/8}, n/2 + c\sqrt{n/8}]$ , tel que

$$\sum_{k \in I_\alpha} p_{n,k} = \alpha.$$

Le choix  $\alpha = 95\%$  correspondant à  $c \sim 1.96$  est un exemple classique.

- Etant donné une valeur pour  $\alpha$ , on dira qu'un arbre est « *normal* » si son nombre  $k$  de feuilles est dans l'intervalle  $I_\alpha$ , et « *anormal* » sinon. La procédure consiste donc à visiter chaque sommet de l'arbre, des feuilles vers la racine, et de vérifier à chaque fois si le sous-arbre associé à un sommet est « *anormal* ». Dans ce cas, le sous-arbre est masqué.
- On peut reprendre la procédure en considérant les sous-arbres fermés comme des feuilles et fermer de nouveau sous-arbres jusqu'à ce que plus aucun sous-arbre ne puisse être fermé.

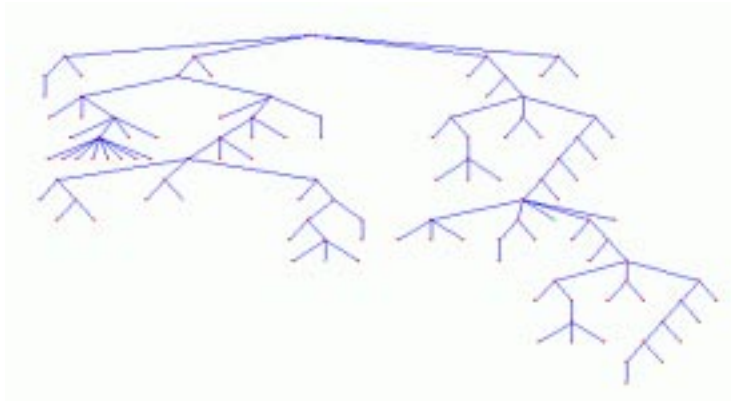


FIG. 1.14: Arbre obtenu par déploiement d'un sommet (en haut à gauche sur la figure 1.13)

Les figures 1.12 et 1.13 illustrent le passage de l'arbre original à l'arbre obtenu en fermant certains sous-arbres « *anormaux* ». La figure 1.14 montre l'arbre obtenu en dépliant l'un des sommets (en haut à gauche sur la figure 1.13) en le sous-arbre sous-jacent. Remarquez que le nouveau sous-arbre obtenu par déploiement peut lui-même contenir des sommets marqués, qui correspondent à des sous-arbres fermés lors d'appels antérieurs de la procédure.

Cette procédure peut évidemment être généralisée au cas d'une métrique et donc d'une distribution quelconque, pour laquelle l'intervalle  $I_\alpha$  puisse être calculé. La possibilité de modifier automatiquement la structure visible de l'arbre pose toutefois un problème. Le déploiement de sous-arbres fermés, et inversement le masquage de sous-arbres identifiés comme « *anormaux* », exigent que l'algorithme de dessin soit incrémental. C'est-à-dire que la représentation d'un arbre et celle d'un autre arbre obtenu du premier par masquage ou déploiement d'un sous-arbre ne doivent pas trop différer. L'utilisateur doit être en mesure de visualiser le passage de l'un à l'autre. On peut, pour palier à ce problème, animer le passage d'un arbre à l'autre [HMRD99]. Heureusement, la plupart des algorithmes d'arbres sont incrémentaux.

### 1.2.6 Vues schématiques

Le masquage des arbres *anormaux* permet, non seulement d'offrir une image plus équilibrée, mais aussi de diminuer le nombre d'éléments de la représentation d'un arbre et donc de sa complexité visuelle. Une autre approche est possible, qui

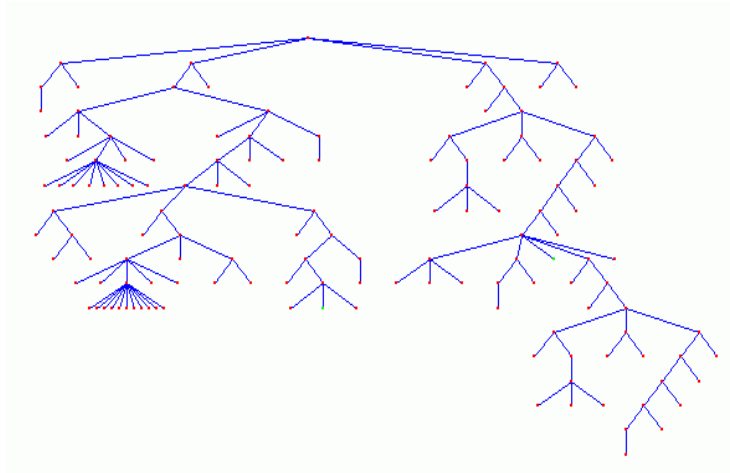


FIG. 1.15: Déploiement d'un sommet du sous-arbre déployé à l'étape précédente (cf figure 1.14)

ne repose sur aucune connaissance particulière de la distribution de la métrique utilisée. Nous l'avons précisé plus haut, les métriques que nous avons introduites sont croissantes (des feuilles vers la racine). Etant donné un arbre, on peut choisir une valeur socle qui détermine si un sommet sera ou ne sera pas affiché. La métrique étant croissante, si un sommet n'est pas affiché, c'est aussi le cas pour tout sommet appartenant à son sous-arbre associé. Le choix de la valeur socle peut être fait de diverses façons, et on peut laisser ce choix à l'utilisateur qui la fait varier interactivement à l'aide d'un curseur, par exemple.

La figure 1.17 illustre le résultat obtenu lorsqu'on applique cette méthode à l'arbre de la figure 1.16. Les arêtes sont dessinées avec couleur et épaisseur, en fonction de l'ensemble des valeurs métriques de tout l'arbre. Le résultat, pour ces sommets et arcs, est donc le même qu'à la figure 1.16. Les attributs des arêtes visibles permettent d'indiquer la complexité relative des sous-arbres masqués. C'est ce que Kimelman *et al.* nomment « *subgraph hiding* » [KLRZ94].

Une autre possibilité est de concevoir des indices spécifiques à la représentation pour indiquer la complexité des sous-arbres masqués. On peut par exemple, attacher aux racines des sous-arbres masqués une forme géométrique qui indique l'aire occupée par le sous-arbre (ou son nombre de sommets). Dans le cas où l'arbre est dessiné avec l'algorithme « radial » de Eades [Ead92], les sous-arbres sont placés dans des secteurs angulaires calculés en fonction du nombre de feuilles. On peut alors dessiner aux feuilles du sous-arbre extrait par valeur socle des portions de cercles, comme l'illustre la figure 1.18. Kimelman *et al.* [KLRZ94], parlent cette fois de « *grouping* ». Dans le cas où l'algorithme de dessin de l'arbre est l'algorithme de Reingold et Tilford, la forme des arbres est grossièrement *triangulaire* et l'algorithme en calcule la base et la hauteur. On peut alors afficher ce triangle au lieu de masquer l'arbre. C'est ce qu'illustre la figure 1.19 en page 35 plus loin. On obtient une figure qui évoque la manière classique de schématiser les arbres. Remarquez cependant que les triangles peuvent parfois se superposer. C'est la nature de l'algorithme de positionnement qui le provoque. On peut alors

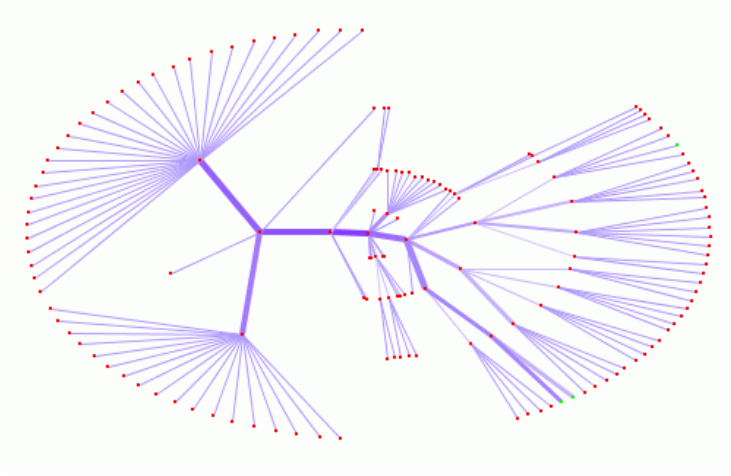


FIG. 1.16: Arbre original, enrichie avec la métrique de Strahler.

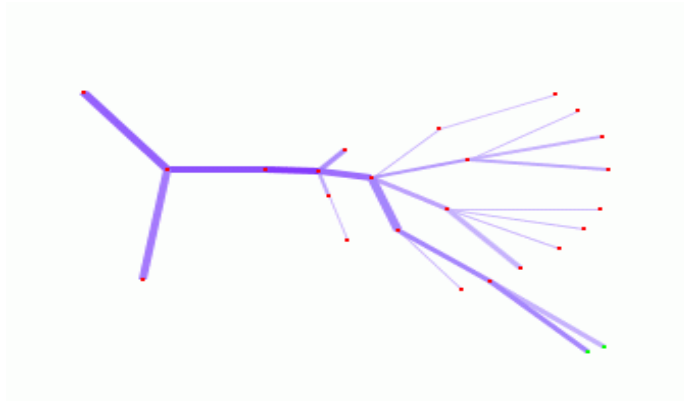


FIG. 1.17: Extraction d'un sous-arbre en utilisant une valeur socle. Seuls les sommets avec une valeur métrique assez grande sont affichés.

avoir recours à des effets de transparence pour assurer un maximum de lisibilité du schéma [HMRD99].

Kimelman *et al.* mentionnent une troisième possibilité, qu'ils appellent « *ghosting* ». Dans ce cas, tous les éléments sont affichés, mais de manière à bien faire ressortir les éléments sélectionnés et à mettre les autres en retrait. La technique décrite au paragraphe 1.2.2 consistant à dessiner les arêtes avec couleur et épaisseur tombe dans cette catégorie. Nous verrons plus loin une autre variante, appliquée au cas des graphes orientés acycliques (cf section 1.3).

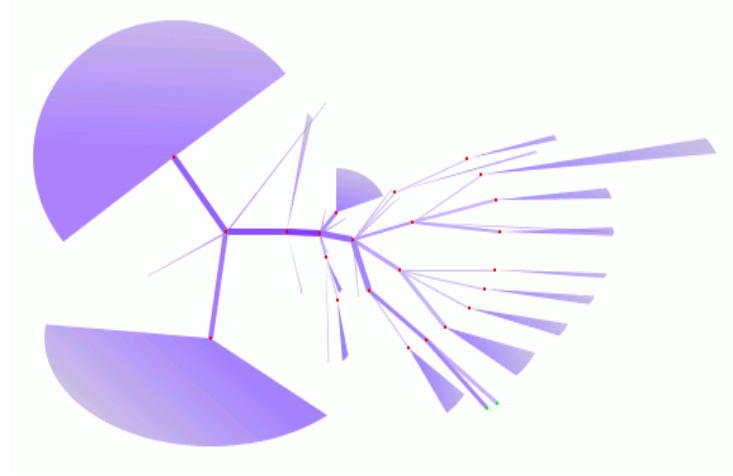


FIG. 1.18: Extraction du sous-arbre par valeur socle. Aux feuilles sont dessinées des portions de cercles indiquant la « forme » du sous-arbre masqué.

### 1.3 Extension au cas des graphes orientés acycliques

Les graphes acycliques orientés sont d'un certain point de vue une généralisation naturelle des arbres. On peut considérer les arbres, tels que nous les avons définis au paragraphe 1.2.1 en page 24, comme des graphes orientés. On considère la plupart du temps que les arêtes sont orientées de la racine vers les feuilles. La longueur de l'unique chemin menant de la racine à un sommet correspond à sa profondeur. L'absence de cycles dans les graphes orientés acycliques nous assure qu'il y existe au moins un sommet sans prédécesseur. Un sommet sans prédécesseur sera appelé un « *sommet source* ». L'existence de sommets sources nous permet de définir le « *niveau* » d'un sommet donné  $v$ . On le définit souvent comme étant la longueur du plus long chemin menant d'un sommet source jusqu'au sommet  $v$ . On dessine d'ailleurs souvent les graphes orientés acycliques en omettant les orientations des arêtes, à condition de disposer les sommets par couches selon leur niveau. Les arêtes vont alors toujours d'un niveau donné vers un niveau plus grand.

Nous allons maintenant voir comment les techniques présentées pour les arbres dans les paragraphes précédents peuvent être soit reprises telles quelles, soit généralisées et appliquées au cas des graphes orientés acycliques. De la même manière qu'il y a des sommets source, tout graphe orienté acyclique possède aussi des sommets sans successeurs, que nous appellerons des « *sommets puits* ». Ces sommets puits sont en quelque sorte analogues aux feuilles dans les arbres. Encore une fois, l'absence de cycles nous permet de généraliser les métriques (1.2), (1.3) et (1.4) en adaptant légèrement les formules qui les définissent. Il suffit *grosso modo* d'appliquer aux sommets puits ce qui était assigné aux feuilles et d'appliquer les formules aux successeurs d'un sommet plutôt qu'à ses fils. Par exemple, on peut définir le nombre de Strahler pour le sommet  $v$  d'un graphe orienté acyclique en posant :

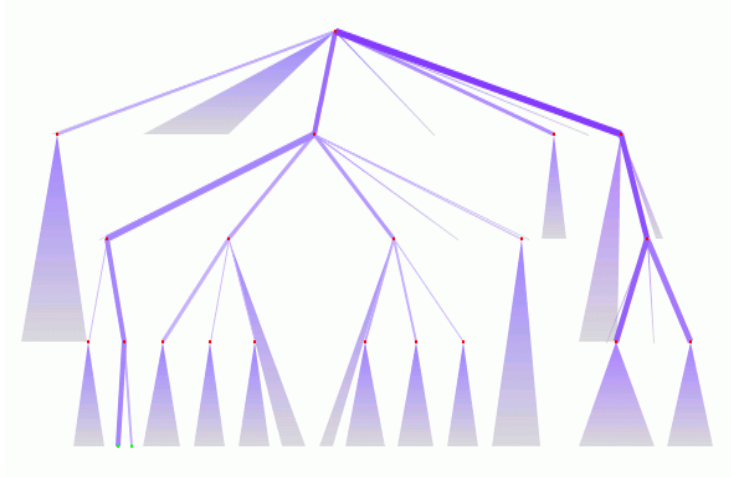


FIG. 1.19: Extraction du sous-arbre par valeur socle. Aux feuilles sont dessinées des triangles indiquant la « forme » du sous-arbre masqué.

$$s(v) = \begin{cases} 1 & \text{si } v \text{ est un sommet puit} \\ s + k - 1 & \text{si } s = s(v_1) = \dots = s(v_k) \\ \max(s(v_1), \dots, s(v_k)) + k - 2 & \text{sinon} \end{cases} \quad (1.6)$$

où  $v_1, \dots, v_k$  sont les successeurs de  $v$  dans le graphe. La figure 1.20 en page précédente illustre le calcul du nombre de Strahler des sommets d'un graphe orienté acyclique. De même on pourrait définir la largeur d'un sommet en reprenant les formules (1.4); la valeur associée à un sommet calculerait donc le nombre de sommets puits accessibles à partir de celui-ci. La profondeur (1.3) correspond ici au niveau d'un sommet (défini plus haut).

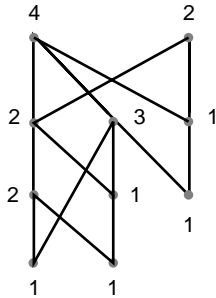


FIG. 1.20: Nombre de Strahler pour les graphes orientés acycliques.

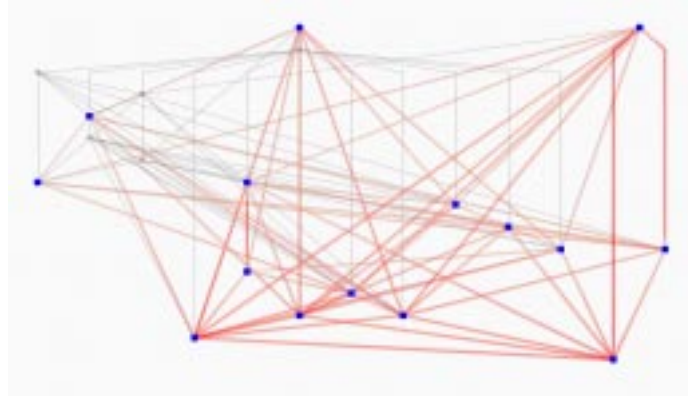


FIG. 1.21: Attributs graphiques pour les graphes orientés acycliques.

### 1.3.1 Attributs graphiques et vues schématiques

De la même manière que pour les arbres, on peut définir des attributs graphiques pour les éléments du graphe. La figure 1.21 montre le résultat obtenu lorsqu'une couleur et une épaisseur est calculée pour les arêtes, en fonction des valeurs de la métrique sur les sommets du graphe. Plus précisément, on sélectionne d'abord les sommets dont la valeur métrique excède une valeur so-cle donnée. On calcule ensuite le graphe induit par ces sommets et on affecte aux arêtes de ce graphe une couleur et une épaisseur (comme on l'a fait pour les arbres, cf section 1.2.2). Les autres sommets et les autres arêtes du graphe sont dessinés en grisés. Le résultat obtenu est une vue schématique du graphe, qui correspond à ce que Kimelman *et al.* appellent « *ghosting* » [KLRZ94]. Les figures 1.23 et 1.25 plus loin illustrent cette technique.

### 1.3.2 Métriques spécifiques aux graphes orientés acycliques

Nous présentons maintenant une métrique « naturelle » pour les graphes orientés acycliques, et qui leur est spécifique. Il nous faut introduire quelques notations afin de la définir. Etant donné un sommet  $v$  du graphe, on désignera par  $d^+(v)$  son degré sortant (son nombre de successeurs). On désignera par  $\mathcal{A}(v)$  l'ensemble de ses prédécesseurs. On définit une métrique  $K$  en posant :

$$K(v) = \begin{cases} 1 & \text{si } v \text{ est un sommet } source \\ \sum_{v' \in \mathcal{A}(v)} \frac{K(v')}{d^+(v')} & \text{sinon} \end{cases} \quad (1.7)$$

La valeur  $K(v)$  est donc calculée à partir des valeurs  $K(v')$  des prédéces-seurs de  $v$ . On peut imaginer que chaque prédécesseur du sommet  $v$  apporte une contribution et que la valeur associée à  $v$  est obtenue en sommant les contribu-tions qu'il reçoit. Un sommet divise sa valeur par son nombre de successeurs qui définit ainsi sa contribution pour chacun d'eux. La figure 1.22 illustre le calcul de cette métrique pour le même graphe qu'à la figure 1.20. On vérifie que la somme des valeurs des sommets puits est égal au nombre de sommets sources qui reçoivent chacun la valeur 1. On peut interpréter la valeur métrique d'un

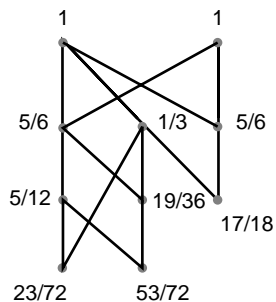


FIG. 1.22: Métrique du flot descendant pour les graphes orientés acycliques.

sommet comme le volume d'eau qui y passe, en imaginant qu'on verse une quantité égale à chaque sommet source au haut du graphe. La figure 1.23 illustre le schéma obtenu avec cette métrique (en appliquant encore une fois la méthode du *ghosting* avec une certaine valeur socle). C'est dans le sens où elle correspond à un phénomène propre au graphe orienté acyclique — l'étalement des sommets par niveaux et la possibilité pour deux sommets d'un même niveau d'avoir un successeur commun — que cette métrique leur est spécifique.

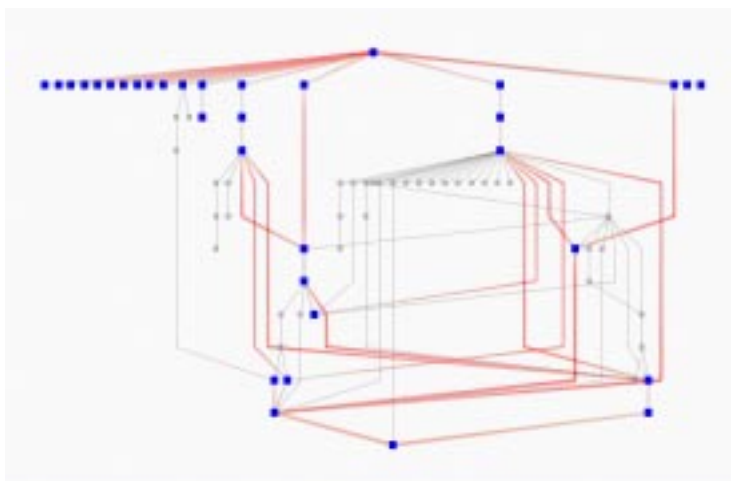


FIG. 1.23: Métrique du flot descendant pour les graphes orientés acycliques.

Remarquez aussi que la métrique (1.7) n'est pas nécessairement croissante, ni décroissante. Ainsi, le sous-graphe extrait avec la valeur socle peut très bien être non-connexe. Cela peut être une information supplémentaire à porter à l'attention de l'utilisateur.

### 1.3.3 Combinaison de métriques

Les graphes orientés acycliques présentent une symétrie naturelle; si on inverse le sens de toutes les arêtes d'un tel graphe on obtient à nouveau un graphe orienté acyclique. Cela permet, par exemple, de dualiser toute métrique définit

en fonction des successeurs (ou des prédécesseurs) d'un sommet. On peut ainsi définir la métrique (1.7) en allant plutôt vers le haut (comme si cette fois on injectait l'eau vers le haut). Dans le cas où il n'y a pas de direction privilégiée pour lire le graphe, on peut combiner une métrique et sa duale. Par exemple, on pourrait calculer la valeur moyenne de la métrique (1.7) et sa duale.

D'autres combinaisons sont possibles. La métrique (1.7) repose sur une distribution uniforme de la valeur d'un sommet vers ses successeurs. On peut pondérer ce calcul et distribuer de manière non-uniforme la valeur d'un sommet sur ses successeurs. On peut, par exemple, pondérer la somme dans la formule (1.7) en fonction du nombre de Strahler et définir une nouvelle métrique  $\bar{K}$  :

$$\bar{K}(v) = \begin{cases} 1 & \text{si } v \text{ est un sommet } source \\ \sum_{v' \in \mathcal{A}(v)} \frac{s(v)}{\sigma(v')} \bar{K}(v') & \text{sinon} \end{cases} \quad (1.8)$$

où on désigne par  $\sigma(v') = \sum_{v'' \in \mathcal{S}(v') \setminus s(v')}$  la somme des nombres de Strahler des successeurs d'un sommet  $v'$ , et par  $\mathcal{S}(v')$  l'ensemble de ses successeurs. La figure 1.25 illustre le résultat obtenu lorsqu'on calcule la vue schématique d'un graphe orienté acyclique avec cette métrique. La valeur socle utilisée est ici la même qu'à la figure 1.23. On observe que les sommets puits en haut et à gauche ont été cette fois écartés à cause de leur métrique de Strahler faible. En revanche, des sommets écartés précédemment reçoivent cette fois une contribution plus grande et viennent s'ajouter au schéma.

Les commentaires que nous avons faits plus haut concernant le choix de la métrique s'applique ici. La possibilité de combiner les métriques doit être vue comme une autre manière d'arriver à concevoir une métrique qui reflète le degré d'intérêt de l'utilisateur pour un sommet du graphe. Bien que nous n'ayons évoqué que la possibilité de combinaisons algébriques, tout autre forme de combinaisons (formule booléenne, par exemple) est aussi possible.

### 1.3.4 Bénéfices pour la navigation

L'objectif recherché est encore une fois de concevoir des indices visuels qui donnent une information quantitative sur la complexité du graphe. La technique utilisée entre dans la catégorie « Focus + contexte » puisque la vue de la totalité du graphe est maintenue tout en mettant une partie du graphe en évidence. Les

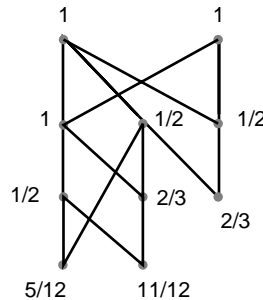


FIG. 1.24: Métrique du flot descendant pondéré par les nombres de Strahler.

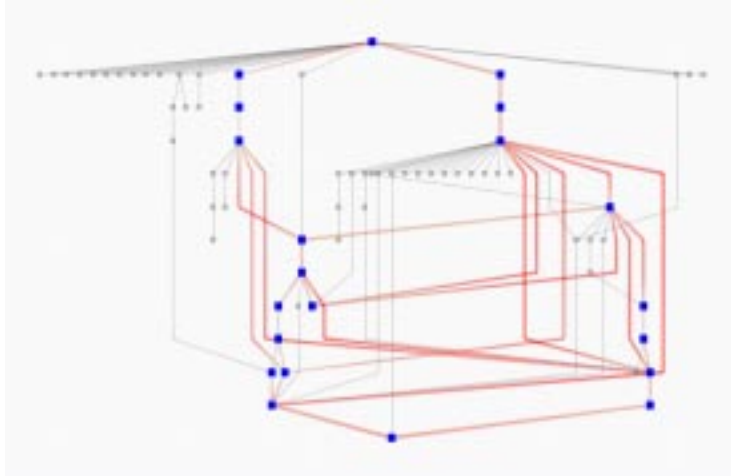


FIG. 1.25: Métrique du flot descendant pondérée par les nombres de Strahler (comparez avec la figure XX).

arcs mis en évidence donnent en quelque sorte une vue plus détaillée d'un sous-graphe. Dans le cas où la métrique correspond à une mesure pertinente par rapport au contexte, le sous-graphe extrait constitue un squelette du graphe. Il fournit à la fois une vue simplifiée du graphe et donne une indication de sa complexité en faisant ressortir les arcs et sommets les « plus importants » (par rapport à la métrique utilisée). On peut donc, lors d'une exploration du graphe, utiliser les arcs du sous-graphe extrait pour décider d'une trajectoire à suivre. L'utilité des indices visuels semble confirmé par leur application à la visualisation de certains sites web (dont on extrait un graphe orientés acycliques appropriée). Les pages les plus susceptibles d'être visitée lors d'une première visite sont mises en relief par la métrique du flot descendant (et sa variante pondérée par les nombres de Strahler). Les mêmes métriques appliquées à des graphes obtenus par déploiement de réseaux de Pétri semblent permettre d'identifier, entre autre, les états bloquants du réseaux. Ces premières réactions indiquent qu'il faut sans doute chercher à définir des métriques adaptées à ce contexte.

## 1.4 Conclusions et perspectives

Le logiciel Latour [HMRD99] qui nous a servi de plate-forme d'expérimentation sera bientôt intégré au produit offert par la société ACE b.v. à Amsterdam. Cette société avait été à l'origine du travail de visualisation des arbres. Un travail en cours explore l'utilisation des métriques dans le cadre du développement d'interfaces permettant à un utilisateur de visualiser la structure d'un ensemble de documents en situation d'apprentissage [DDG99]. La métrique permettra alors de visualiser les dépendances logiques des différentes sections du livre et les priorités de lecture en fonction des acquis du lecteur.

Malgré l'intérêt suscité par nos résultats, une conclusion s'impose. Notre commentaire sur les bénéfices pour la navigation (sections 1.2.3 et 1.3.4) est resté

relativement court. Il semble impératif d’asseoir nos conclusions sur une étude d’utilisabilité. Nous pourrions alors déterminer hors de tout doute comment l’utilisateur est aidé par les indices visuels que nous calculons. Cette remarque s’applique de manière générale à l’ensemble de la discipline [HMM99a], comme nous l’avons souligné dans l’introduction. Jusqu’à maintenant les seuls travaux concernant l’ergonomie des graphes appartiennent à la communauté du « Graph Drawing » [Pur98, PCJ95, BMK95, DC98]. Le temps et les ressources nécessaires à ce genre d’entreprise nous ont empêché jusqu’à maintenant d’envisager à court terme ce projet. Il est vraisemblable qu’une étude de ce type nous oriente vers d’autres voies à explorer.

La section 1.2.5 commentant l’utilisation de la distribution d’une métrique pour contrôler l’affichage d’un arbre se limite au cas de la métrique « largeur » (nombre de feuilles). Il faudrait certainement se pencher sur l’article de Drmota [Drm97] et étendre cette technique à d’autres métriques sur les arbres, ou sur les graphes en général. L’article de Flajolet *et al.* [FRV79] décrit le comportement asymptotique de la valeur moyenne du nombre de Strahler (dans l’ensemble des arbres binaires ayant un nombre de sommets donné) ; une description complète de la distribution semble difficile, voire impossible, à obtenir. On peut, dans un cas pareil, assouplir la procédure et laisser à l’utilisateur le choix d’un intervalle autour de la moyenne pour fixer la « normalité » d’un sous-arbre, de manière analogue à la procédure d’extraction de sous-graphe à l’aide d’une valeur socle (section 1.2.6). Cette dernière approche, laissant plus de contrôle à l’utilisateur, est à comparer avec les travaux de Furnas [Fur86], qui avait introduit une métrique permettant de contrôler l’affichage d’un arbre en faisant varier le « niveau de détail » de sa représentation.

Ces techniques, au moins du point de vue de la visualisation, s’apparentent aux algorithmes de partitionnement de graphes (« clustering », cf section 1.1.4). Lors de la manipulation d’un graphe de grande taille, on recherche souvent à en diminuer la complexité. Plusieurs algorithmes permettent de découper le graphe en sous-graphes de tailles plus petites selon certains critères choisis ou implicitement définis par l’algorithme (voir, e.g., [AK95]). On peut alors visualiser, non pas le graphe original, mais un graphe induit par le découpage du graphe original. On peut, par exemple, définir un nouveau graphe dont les sommets correspondent aux sous-graphes calculés par l’algorithme. La visualisation de ce nouveau graphe permet en quelque sorte de saisir la complexité structurelle du graphe original tout en maintenant la complexité visuelle à un niveau acceptable pour l’utilisateur. C’est ce que permet d’obtenir la technique utilisant la valeur socle pour les arbres ou les graphes orientés acycliques ; le graphe est alors partagé en plusieurs sous-graphes, dont l’un seulement est affiché. Il est clair, dans le cas des arbres, que le sous-graphe affiché rend la structure de l’arbre original. Les exemples étudiés semblent indiquer qu’on obtient un effet similaire avec les graphes orientés acycliques. Cette technique trouve tout son sens dans un environnement où l’utilisateur peut faire varier le niveau de détail du graphe, ou déployer une composante du graphe affiché. C’est ce que permet de faire le curseur faisant varier la valeur socle lors de l’affichage de l’arbre ou du graphe orienté acyclique (sections 1.2.6 et 1.3.1). A cette fin, on peut appliquer récursivement l’algorithme de partitionnement et définir une hiérarchie permettant de contrôler le niveau de détail du graphe affiché [EF96, EFL96]. Cette approche nécessite de choisir un algorithme de dessin incrémental, c’est-à-dire qui puisse adapter le dessin du graphe en y ajoutant ou en y supprimant une

composante. C'est le cas de la plupart des algorithmes d'arbres.

Nous avons mentionné, sans le décrire, notre travail d'expérimentation et de développement. Cela nous a amené à regarder de plus près certains problèmes de dessins de graphes [MH98]. La mise au point de nos idées sur les graphes orientés acycliques nous avait poussé à élaborer une méthode de génération automatique de ces graphes. Nous avons récemment regardé de près cette question et établis quelques résultats combinatoires sur les graphes orientés acycliques aléatoires [BDM99]. En particulier, nous avons mis au point un algorithme simple pour engendrer un graphe orienté acyclique aléatoire en un temps polynomial  $\mathcal{O}(n^3 \log n)$ .

L'objectif à plus long terme est d'étendre notre approche au cas des graphes généraux, orientés et non-orientés. Le premier pas dans cette direction demande qu'on identifie les métriques sur les graphes les plus utiles du point de vue de la visualisation. Le cas des graphes orientés acycliques doit aussi être revu de plus près. Les nombres de Strahler sur les arbres possèdent une interprétation très claire vu leur origine [Str52, Hor45]; les travaux de Viennot *et al.* [VEJA89] renforcent le rôle des nombres de Strahler comme paramètre descriptif pour les arbres. Bien que la formule définissant ces nombres puisse être appliquée aux sommets d'un graphe orienté acyclique, il reste que l'interprétation des nombres obtenus est moins évidente. On ne peut prétendre que ces nombres soient aussi pertinents pour ces graphes qu'ils le sont pour les arbres. En revanche, la métrique du flot descendant (1.7) semble, elle, plus naturelle. Les exemples étudiés montrent aussi que la combinaison de la métrique de Strahler et du flot descendant (1.8) donnent de bons résultats.

Le calcul des distances dans les graphes a déjà été considéré par certains auteurs [Fur86, BRS92] pour identifier certains sommets ou certaines composantes des graphes. Le problème qui *a priori* se pose avec les graphes généraux est la présence éventuelle de cycles, qui empêchent d'appliquer directement des récurrences comme celles des formules (1.2), (1.3) ou (1.4). Une approche possible est d'itérer le calcul de valeurs définies à partir des voisins d'un sommets. Le développement de cette métrique fait alors apparaître naturellement la matrice d'adjacence du graphe. Cette approche se compare aux algorithmes adaptatifs utilisés pour le routage dans les réseaux [TanXX]. Elle se compare aussi au calcul des chaînes de Markov sur les graphes qui utilisent souvent, en place de la matrice d'adjacence, une matrice stochastique. Une autre approche consiste à baser le calcul de la métrique sur un arbre recouvrant du graphe. La pertinence de ce calcul repose alors entièrement sur le choix de l'arbre recouvrant.

Deuxième partie

**Combinatoire des mots**



## 2.1 Introduction

Les deux ouvrages de M. Lothaire [Lot83], [Lotar] permettent certainement de définir la combinatoire des mots au travers des applications qu'elle trouve dans de nombreux domaines de l'informatique théorique et des mathématiques. Plusieurs problèmes, comme c'est souvent le cas en combinatoire algébrique, peuvent se ramener à des problèmes combinatoires sur des suites de lettres, ou des arbres dont les feuilles sont étiquetées par des lettres. Le problème de Burnside [MKS66, Sect. 5.12] [Lot83, Chap. 2], de Ramsey [Lot83, Chap. 3] ou la formule d'inversion de Lagrange [Lot83, Chap. 10] n'en sont que quelques exemples. Schützenberger [Sch59] avait, parmi les premiers, indiqué comment la combinatoire des mots pouvait jouer un rôle important en théorie des algèbres de Lie libres. Viennot [Vie76] a remarquablement expliqué les liens étroits et riches qu'entretiennent les factorisations du monoïde libre et les bases des algèbres de Lie libres et confirmé la place de la combinatoire des mots dans cette théorie.

Parmi toutes les factorisations, la *factorisation de Lyndon* est celle qui se prête le plus facilement aux arguments combinatoires. La raison en est certainement qu'elle est définie à partir de mots possédant certaines propriétés par rapport à l'ordre lexicographique, qui est peut-être l'ordre le plus « naturel » sur les mots. Les *mots de Lyndon* sont les mots les plus petits dans leur classe de conjugaison, par rapport à l'ordre lexicographique. Les autres factorisations sont, elles, définies à partir d'ensembles d'arbres, équipés d'ordres plus arbitraires et sont en général plus difficiles à manipuler [Reu93, Mel92]. Lothaire [Lot83, Chap. 5] introduit les mots de Lyndon comme outil combinatoire pour obtenir certains résultats centraux en théorie des algèbres de Lie libres. La preuve qu'ils permettent de donner pour le théorème de Poincaré–Birkhoff–Witt pour les algèbres de Lie libres en est une très belle illustration (voir [Lot83, Sect. 5.2] ou [MR89]). Leur étude introduit aussi des problèmes qui leur sont propres. Les algorithmes de Duval, l'un pour le calcul de la factorisation des mots en mots de Lyndon [Duv83], l'autre pour la génération des mots de Lyndon en longueur bornée [Duv88], sont à mon sens des « joyaux » de la combinatoire des mots.

Ce chapitre aborde la factorisation de Lyndon en tant qu'outil pour l'étude de mots infinis et fait un tour d'horizon de l'ensemble des résultats que nous avons obtenus sur le sujet. Le premier pas dans cette direction avait été marqué par la généralisation du théorème de factorisation de Lyndon aux mots infinis par Siromoney *et al.* [SMDS94] (cf Th. 7). Notre première contribution a été le calcul de la factorisation de certains mots infinis connus, et l'extension aux mots infinis d'un résultat sur la divisibilité des mots [Mel96a]. Nous avons aussi montré comment généraliser le théorème de factorisation des mots infinis aux factorisations de Viennot dont la factorisation de Lyndon est un cas particulier [Mel96b] (voir [Lot83, Chap. 5] pour une définition des factorisations de Viennot). Nos résultats plus récents concernent les mots sturmiens infinis et se penchent essentiellement sur le calcul explicite de leur factorisation et sur l'étude de leurs facteurs de Lyndon [Melar]. Nous avons aussi expliqué les liens entre la factorisation de Lyndon et une factorisation donnée par Wen & Wen [WW94] pour le mot de Fibonacci. Le calcul de la factorisation des mots sturmiens caractéristiques nous avait permis de définir les facteurs singuliers pour les mots sturmiens caractéristiques et généraliser certains des résultats de Wen & Wen [Mel99].

Nous avions, à partir de la factorisation de Lyndon des mots sturmiens carac-

téristiques, retrouver un résultat de Berstel et de Luca [Bd97]. Ils avaient montré que l'ensemble des facteurs de mots sturmiens qui sont des mots de Lyndon est l'ensemble des mots de Christoffel. Les facteurs de Lyndon d'un mot (fini ou infini) sont nécessairement des facteurs des mots de Lyndon apparaissant dans sa factorisation. Une description explicite des mots de Lyndon apparaissant dans la factorisation des mots sturmiens nous avaient permis de retrouver facilement le résultat de Berstel et de Luca.

Ce résultat nous avait incité à regarder de plus près l'ensemble des mots de Christoffel, en tant que mots de Lyndon. C'est d'ailleurs en utilisant les propriétés des mots de Lyndon que Borel et Laubie [BL93], et aussi Laurier [Lau95], avaient établi certains de leurs résultats sur les mots de Christoffel. Nous allons dans ce chapitre donner une caractérisation des mots de Christoffel : ce sont les mots de Lyndon sur deux lettres qui ont un unique arbre de Lyndon associé. Cette caractérisation ouvre une voie pour généraliser les mots de Christoffel à un alphabet de plus de deux lettres. Nous allons voir comment on peut obtenir, pour un alphabet fini quelconque, des résultats analogues à ceux de Borel et Laubie pour les mots de Christoffel. Cette théorie est encore incomplète, mais constitue déjà un corps de résultats intéressants qui non seulement complète nos résultats précédents, mais confirme la factorisation de Lyndon comme outil d'étude des mots infinis. On peut en effet définir l'ensemble des mots de Lyndon équilibrés infinis, qui sont les limites de suites de mots de Lyndon équilibrés (Déf. 29). L'espoir ici est de voir un lien entre ces mots infinis et d'autres qu'il reste à définir ou à trouver, tout comme il existe un lien entre mots de Christoffel infinis et mots sturmiens.

## 2.2 Factorisation de mots

La conjugaison de mots joue un rôle central en combinatoire des mots. Rappelons ici sa définition (voir aussi [Lot83, Chap. 1]). Nous désignerons par  $A$  l'*alphabet*, et par  $w = a_0 \cdots a_n$  un *mot* formé des *lettres*  $a_i \in A$  ( $i = 0, \dots, n$ ). L'ensemble de tous les mots possède une structure de *monoïde* qu'on désigne par  $A^*$  ; l'ensemble des mots *non vides* est, lui, désigné par  $A^+$ . Nous dirons que deux mots  $u, v \in A^*$  sont *conjugés* si et seulement s'il existe deux mots  $x, y \in A^*$  tels que  $u = xy$  et  $v = yx$ . On vérifie que les mots  $aabab$  et  $ababa$  sont conjugés (avec  $x = a$  et  $y = abab$ ). La relation de conjugaison des mots est une relation d'équivalence et la classe d'équivalence d'un mot est appelée une *classe de conjugaison*. Par exemple, l'ensemble des mots  $\{aabab, ababa, babaa, abaab, baaba\}$  est une classe de conjugaison. Nous dirons qu'un mot non vide est *primitif* s'il n'est pas une puissance non-triviale d'un autre mot. Le mot  $aabab$  est primitif. Remarquez que les mots d'une classe de conjugaison d'un mot primitif sont tous primitifs. Nous emprunterons dans la suite les notations habituelles en combinatoire des mots [Lot83].

L'un des résultats les plus profonds de la combinatoire des mots concernant la conjugaison et la factorisation de mots est le théorème de Schützenberger [Sch65], liant factorisation et section des classes de conjugaison de mots. L'énoncé définit ce qu'il faut entendre par factorisation ; il fait aussi appel à la notion de sous-monoïdes et d'ensemble minimal de générateurs pour lesquels on pourra consulter [Lot83, Chap. 1].

**Théorème 1** (Schützenberger [Sch65], cf [Lot83, Th. 5.4.1])

Soit  $(X_\lambda)_{\lambda \in \Lambda}$  une famille de sous-ensembles de mots non vides  $X_\lambda \subset A^+$  et totalement ordonnée. Cette famille est une factorisation du monoïde libre  $A^*$  si et seulement deux des trois conditions suivantes sont vérifiées :

(i) tout mot non vide  $w \in A^+$  peut s'écrire d'au moins une manière sous la forme

$$w = \ell_{\lambda_1} \cdots \ell_{\lambda_k}, \quad (2.1)$$

avec  $\ell_{\lambda_i} \in X_{\lambda_i}$  ( $1 \leq i \leq k$ ) et  $\ell_{\lambda_1} \geq \cdots \geq \ell_{\lambda_k}$ .

(ii) tout mot non vide  $w \in A^+$  peut s'écrire d'au plus une manière sous la forme (2.1).

(iii) Chaque classe de conjugaison  $C$  de mots non vides intersecte l'un et seulement l'un des sous-monoïdes  $X_\lambda^*$  dont l'ensemble minimal de générateurs est  $X_\lambda$  et les éléments de  $C \cap X_\lambda^*$  sont conjugués dans  $X_\lambda^*$ .

### 2.2.1 Mots de Lyndon

Les mots de Lyndon forment une section particulière des classes de conjugaison de mots primitifs. Ce sont les mots primitifs qui sont les plus petits dans leur classe de conjugaison, par rapport à l'ordre lexicographique. Le mot *aabab* est un mot de Lyndon, par exemple. Les lettres de l'alphabet sont aussi, par définition, des mots de Lyndon. Ces mots sont apparus la première fois dans un travail de Lyndon [CFL58] pour construire une base de la série centrale descendante du groupe libre. Ils donnent aussi, en vertu du théorème de Magnus [MKS66, Th. 5.12], une base de l'algèbre de Lie libre. Cette construction utilise une propriété fondamentale des mots de Lyndon qui permet de construire un arbre binaire à partir du mot lui-même. Nous étudierons plus loin les propriétés des arbres binaires obtenus des mots de Lyndon. Nous rappelons maintenant quelques résultats combinatoires fondamentaux sur les mots de Lyndon. On désignera par  $L(A)$  l'ensemble des mots de Lyndon sur l'alphabet  $A$ , ou plus simplement par  $L$  si le contexte est clair.

**Proposition 2** (Lyndon [CFL58], cf Lothaire [Lot83, Chap. 5]) Soit  $w \in A^*$ , alors  $w$  est un mot de Lyndon de longueur au moins 2 si et seulement s'il existe deux mots de Lyndon  $u$  et  $v$  tels que  $u < v$  et  $w = uv$ .

Ce résultat permet déjà de donner un algorithme simple pour engendrer les mots de Lyndon. Partant des lettres, il suffit de former, avec les mots déjà obtenus, tous les produits  $uv$  avec  $u < v$ . Ainsi, on peut donner les premiers mots pour un alphabet de deux lettres  $A = \{a, b\}$  (avec  $a < b$ ).

*a, b, ab, aab, abb*  
*aaab, aabb, abbb*  
*aaaab, aaabb, aabbb, ababb, aabab, abbbb*  
 ...

Duval [Duv88] a donné un algorithme très élégant qui engendre les mots de Lyndon de longueur au plus  $N \geq 1$  dans l'ordre lexicographique. L'algorithme,

très efficace, effectue le passage d'un mot de Lyndon à son successeur en un temps linéaire par rapport à  $N$  (voir aussi [BP94]).

**Observation 3** On tire de la proposition précédente, que si  $u, v$  sont des mots de Lyndon tels que  $u < v$  alors les mots  $u^k v$  et  $uv^k$  sont aussi des mots de Lyndon ( $k \geq 0$ ).

## 2.2.2 Théorème de factorisation

Le résultat fondamental concernant les mots de Lyndon est le théorème de factorisation :

**Théorème 4** (Lyndon [CFL58], cf Lothaire [Lot83, Chap. 5]) *Tout mot  $w \in A^*$  s'écrit de manière unique comme un produit non croissant de mots de Lyndon :*

$$w = \ell_1 \cdots \ell_k, \quad \ell_i \in L \quad (i = 1, \dots, k), \quad \ell_1 \geq \cdots \geq \ell_k, \quad \text{et } k \geq 0.$$

L'existence de la factorisation découle d'un raisonnement très simple. Tout mot peut s'écrire comme un produit de mots de Lyndon (les lettres qui le forment, par exemple). Partant d'une telle écriture,  $w = \ell_1 \cdots \ell_k$ , on peut itérativement remplacer les paires de mots  $\ell_i \ell_{i+1}$  tels que  $\ell_i < \ell_{i+1}$  par le mot  $\ell_i \ell_{i+1}$ , qui est un mot de Lyndon en vertu de la proposition 2. Après un nombre fini de réécritures on atteint nécessairement une factorisation non croissante. On peut appliquer cette procédure pour vérifier que le mot  $abaababaabaababaabab$  se factorise en  $(ab)(aabab)(aabaababaabab)$ . L'unicité de la factorisation repose sur une caractérisation des mots de Lyndon qu'il est approprié d'énoncer maintenant :

**Proposition 5** (Lyndon [CFL58], cf Lothaire [Lot83, Chap. 5]) *Un mot  $w \in A^+$  est un mot de Lyndon si et seulement s'il est strictement plus petit que tous ses suffixes propres (et non vides).*

On vérifie facilement que le mot  $aabab$  est plus petit que ses suffixes propres  $\{abab, bab, ab, b\}$ . On peut maintenant conclure la preuve du théorème 4. Supposons qu'on ait deux factorisations d'un même mot,

$$\ell_1^{(1)} \cdots \ell_k^{(1)} = \ell_1^{(2)} \cdots \ell_{k'}^{(2)}.$$

On peut supposer, sans perte de généralité, que  $\ell_1^{(1)} \neq \ell_1^{(2)}$  et que  $\ell_1^{(1)} = \ell_1^{(2)} \cdots \ell_i^{(2)} x$  où  $x$  est un préfixe de  $\ell_{i+1}^{(2)}$  ( $1 \leq i \leq k-1$ ). On a, en vertu de la proposition 5,  $\ell_1^{(1)} < x$ , si  $x$  est non vide. On a aussi,  $x \leq \ell_{i+1}^{(2)} \leq \ell_i^{(2)} \leq \cdots \leq \ell_1^{(2)} < \ell_1^{(1)}$ . On obtient donc, en combinant les inégalités, la contradiction  $\ell_1^{(1)} < \ell_1^{(1)}$ . Dans le cas où  $x$  est vide, on s'appuie plutôt sur l'inégalité  $\ell_1^{(1)} < \ell_i^{(2)}$  pour obtenir la même contradiction.

## 2.2.3 Factorisation de Lyndon de mots infinis

Nous allons maintenant voir comment le théorème 4 se généralise aux mots infinis. Le résultat repose essentiellement sur une extension appropriée de l'ordre lexicographique aux mots infinis et sur une « bonne » définition des *mots de*

*Lyndon infinis.* Introduisons d'abord certaines notations propres aux mots infinis. On désignera le plus souvent un mot infini par sa suite de lettres  $s = a_0 a_1 \dots$ , (où  $a_0, a_1, \dots \in A$ ). Le produit d'un mot fini  $w = b_0 \dots b_k$ , où  $b_0, b_1, \dots \in A$ , et d'un mot infini  $s$  résulte en le mot infini  $t = ws = b_0 \dots b_k a_0 a_1 \dots$ . Dans ce cas, on dira que le mot  $s$  est un *suffixe* du mot  $t$ , et que le mot  $w$  est un *préfixe* du mot  $t$ . Un suffixe d'un mot infini est toujours un mot infini ; un préfixe d'un mot infini est toujours un mot fini. Observez que le produit  $sw$  n'est pas défini.

On notera  $A^\infty$  l'ensemble des mots *fini ou infini*. On étend l'ordre lexicographique à tout  $A^\infty$  comme suit. Soit  $s$  et  $t$  des mots finis ou infinis. On pose  $s < t$  si et seulement si soit  $s$  est préfixe de  $t$  (ce qui n'est possible que si  $s$  est un mot fini), soit il existe un mot fini  $w$ , deux lettres  $a, b \in A$ , et deux mots (finis ou infinis selon les cas)  $s', t'$  tels que  $s = was'$ ,  $t = wbt'$  et  $a < b$ .

**Définition 6** *Un mot infini  $\ell$  est un mot de Lyndon infini si et seulement s'il admet une infinité de préfixes qui sont des mots de Lyndon finis.*

Par exemple, le mot  $ab^\infty$  est un mot de Lyndon infini. De manière générale, si  $u$  et  $v$  sont des mots de Lyndon finis tels que  $u < v$  alors  $uv^\infty$  est un mot de Lyndon infini, où  $v^\infty$  désigne le mot infini  $vv\dots$  obtenu en répétant le mot  $v$  une infinité de fois (cf observation 3).

**Théorème 7** (Siromoney *et al.* [SMDS94])

*Tout mot infini  $s$  s'écrit de manière unique sous l'une des formes :*

- (i)  $s = \ell_0 \dots \ell_k \ell$ , avec  $\ell_0 \geq \dots \geq \ell_k > \ell$ , où  $\ell_0, \dots, \ell_k$  sont des mots de Lyndon finis et  $\ell$  est un mot de Lyndon infini, ou
- (ii)  $s = \prod_{n \geq 0} \ell_n$ , avec  $\ell_0 \geq \ell_1 \geq \dots$ , où  $\ell_0, \ell_1, \dots$  sont des mots de Lyndon finis.

La preuve du théorème 7 repose sur une généralisation de la proposition 5 (cf [SMDS94, Prop. 2.2]) et suit les mêmes lignes que la preuve du cas fini.

**Exemple 8** Le calcul de la factorisation d'un mot infini donné présente un intérêt que ne présente pas nécessairement celui d'un mot fini. On peut espérer y voir émerger certaines propriétés du mot infini. A titre d'exemple, donnons la factorisation du mot de Fibonacci. Soit la suite de mots finis donnée par la récurrence

$$f_0 = b, f_1 = a \quad \text{et} \quad f_{n+1} = f_n f_{n-1} \quad (n \geq 1). \quad (2.2)$$

On a donc  $f_2 = ab$ ,  $f_3 = aba$ ,  $f_4 = abaab$ , ... Le mot de Fibonacci est le mot infini limite de cette suite

$$\begin{aligned} f &= \lim_{n \rightarrow \infty} f_n \\ &= abaababaabaababaabaabaababaabaabab\dots \end{aligned}$$

Nous avons montré [Mel96a] que le mot de Fibonacci peut s'écrire  $f = \prod_{n \geq 0} \ell_n$ , où  $\ell_0 = ab$  et  $\ell_{n+1} = \varphi(\ell_n)$  avec  $\varphi(a) = aab$  et  $\varphi(b) = ab$ . Ainsi,

$f = (ab)(aabab)(aabaababaab)\dots$ . On en déduit [Melar], en particulier, que  $\varphi(f)$  est le mot de Fibonacci sur le code  $\{aab, ab\}$ . On vérifie aussi que les facteurs  $\ell_n$  satisfont la récurrence  $\ell_{n+1} = \ell'_n \ell_n^2$ . La factorisation du mot de Fibonacci est donc du type 7.(ii).

**Exemple 9** On trouve une factorisation du type 7.(i) lorsqu'on factorise le mot de Thue-Morse dual (le mot obtenu du mot de Thue-Morse en échangeant les lettres  $a$  et  $b$ ). Ce mot est défini comme suit. On pose  $u_0 = a$ ,  $v_0 = b$  et  $u_{n+1} = u_n v_n$ ,  $v_{n+1} = v_n u_n$ . Ainsi,  $u_1 = ab$ ,  $u_2 = abba$ ,  $u_3 = abbabaab, \dots$ , et  $v_1 = ba$ ,  $v_2 = baab$ ,  $v_3 = baababba$ ,  $v_4 = baababbaabbabaab$ , etc. Le mot infini  $\mu = \lim_{n \rightarrow \infty} u_n$  est appelé le mot de Thue-Morse. Nous avons montré que la factorisation du mot « dual »  $v = \lim_{n \rightarrow \infty} v_n$ , est  $v = b\ell$  où le mot infini  $\ell$  est un mot de Lyndon [IM97].

## 2.2.4 Régularités inévitables

Mentionnons ici le lien entre factorisation de Lyndon et régularités dans les mots infinis [Mel96a]. Il est possible de déduire certaines propriétés des mots infinis lorsqu'ils possèdent une factorisation de type 7.(ii). Dans le cas où le mot est lui-même périodique, la factorisation sera ultimement périodique. En effet, il suffit alors d'écrire le mot sous la forme  $s = uw^\infty$  où  $w$  est un mot de Lyndon et  $u$  est un suffixe de  $w$ . Si  $u = \ell_1 \dots \ell_k$  est la factorisation de  $u$ , alors on a  $\ell_k > w$ , en vertu de la proposition 5, et  $s = \ell_1 \dots \ell_k w^\infty$  donne la factorisation de  $s$ .

Dans le cas où la factorisation du mot  $s$  (est du type 7.(ii) et) n'est pas ultimement périodique, alors on en tire une  $\omega$ -division. Une  $n$ -division d'un mot fini  $w$ , par rapport à l'ordre lexicographique, est une factorisation  $w = x_1 \dots x_n$  telle que  $w < x_{\sigma(1)} \dots x_{\sigma(n)}$  pour toute permutation  $\sigma$  non-triviale de  $\{1, \dots, n\}$ . On définit une  $\omega$ -division d'un mot infini en étendant cette définition. Une  $\omega$ -division d'un mot infini  $s$  est une factorisation en mots finis  $s = x_1 x_2 \dots$  telle qu'une permutation d'un nombre fini de facteurs du mot donne un mot qui soit plus grand. Reutenauer [Reu86] avait montré que la factorisation de Lyndon d'un mot  $w = (\ell_1)^{k_1} \dots (\ell_n)^{k_n}$ , où les mots de Lyndon  $\ell_1, \dots, \ell_k$  sont *distincts*, en donnent une  $n$ -division. Son argument peut-être repris pour montrer que la factorisation du mot  $s = \prod_{n \geq 0} (\ell_n)^{k_n}$ , où les mots de Lyndon finis  $\ell_n$  sont distincts, en donne une  $\omega$ -division (toute permutation donne une division d'un préfixe bien choisi de  $s$ ). Ce résultat nous avait permis de donner une nouvelle  $\omega$ -division des mots sturmiens caractéristiques [Melar] (cf Eq. 2.4), qui seront définis à la prochaine section.

## 2.2.5 Cas des mots sturmiens

Nous allons maintenant nous pencher sur le cas des mots sturmiens et en donner la factorisation de Lyndon [Melar]. Définissons d'abord la *fonction de complexité*  $p_s(n)$  d'un mot infini  $s$ . Elle donne le nombre  $p_s(n)$  de facteurs de longueur  $n$  du mot  $s$ . On peut montrer qu'un mot infini dont la fonction de complexité satisfait  $p_s(n) \leq n$  pour une valeur de  $n$  est nécessairement ultimement

périodique, c'est-à-dire de la forme  $s = uv^\infty$ , où  $u$  et  $v$  sont des mots finis.

Les mots *sturmiens* sont les mots non ultimement périodiques et qui sont de complexité minimale. Ils satisfont  $p_s(n) = n + 1$  pour tout  $n$  et sont des mots sur deux lettres puisque  $p_s(1) = 2$ . Le plus célèbre des mots sturmiens est certainement le mot de Fibonacci  $f = \lim_{n \rightarrow \infty} f_n$  (cf Ex. 8). On pourra consulter le survey de Berstel [Ber95], un article récent de de Luca [de 97], ou encore le troisième chapitre du dernier livre de M. Lothaire [Lotar]. pour en savoir plus sur les mots sturmiens. Parmi les mots sturmiens, on distingue les mots *sturmiens caractéristiques*. On peut se restreindre à leur étude lorsqu'on s'intéresse aux facteurs des mots sturmiens, puisqu'il est possible de montrer que tout mot sturmien possède un ensemble de facteurs qui coïncide avec l'ensemble des facteurs d'un mot sturmien caractéristique [Ber95, Prop. 3.2]. Les mots sturmiens caractéristiques peuvent être définis comme suit (cf [Rau84]).

**Définition 10** *Soit une suite d'entiers  $(c_n)_{n \geq 0}$  telle que  $c_0 \geq 0$  et  $c_i > 0$  pour  $i > 0$ . Le mot sturmien caractéristique associé à la suite d'entiers  $(c_n)_{n \geq 0}$  est le mot  $s = \lim_{n \rightarrow \infty} s_n$ , où*

$$s_0 = b, s_1 = a \quad \text{et} \quad s_{n+1} = s_n^{c_n-1} s_{n-1} \quad (n \geq 1). \quad (2.3)$$

Le mot de Fibonacci (2.2) est un mot sturmien caractéristique, associé à la suite  $c_n = 1$  pour tout  $n \geq 0$ . La récurrence (2.3) permet de donner pour les mots sturmiens caractéristiques le calcul explicite de leur factorisation, qui fait apparaître la suite  $(c_n)_{n \geq 0}$ . On désigne par  $wa^{-1}$  le mot obtenu de  $w$  en effaçant la dernière lettre  $a$ , si c'est possible; sinon le mot  $wa^{-1}$  est, par définition, vide. Observez que le mot  $s_{2n+1}$  se termine toujours par la lettre  $a$ .

**Théorème 11** (cf [Melar]) *La factorisation de Lyndon du mot sturmien caractéristique associé à la suite  $(c_n)_{n \geq 0}$  est :*

$$s = \prod_{n \geq 0} [(as_{2n+1}a^{-1})^{c_{2n}-1} as_{2n}s_{2n+1}a^{-1}]^{c_{2n+1}} \quad (2.4)$$

où les mots  $\ell_n = (as_{2n+1}a^{-1})^{c_{2n}-1} as_{2n}s_{2n+1}a^{-1}$  sont des mots de Lyndon tels que  $\ell_n > \ell_{n+1}$ .

Le théorème permet de retrouver les premiers facteurs de Lyndon du mot de Fibonacci mentionnés précédemment (cf Ex. 8). Dans le cas du mot de Fibonacci, les mots de Lyndon de la factorisation sont tous distincts et réduits à  $\ell_n = as_{2n}s_{2n+1}a^{-1}$ , puisqu'on a  $c_n = 1$  pour tout  $n \geq 0$ .

La preuve du théorème 11 consiste à montrer, en utilisant la récurrence (2.3), que les mots  $u_n = as_{2n+1}a^{-1}$ ,  $v_n = as_{2n}s_{2n+1}a^{-1}$  sont des mots de Lyndon tels que  $u_n = v'_n$ . On en tire que  $u_n^k v_n$  pour tout  $k \geq 0$ , en vertu de l'observation 3. On montre aussi que  $u_n^{c_{2n}-1} v_n > u_{n+1}^{c_{2n+2}-1} v_{n+1}$ . Finalement, le fait que le produit (2.4) coïncide avec le mot  $s$  s'obtient en réécrivant le produit :

$$s = s_1^{c_0} s_0 s_2^{c_1-1} s_1 s_3^{c_2-1} s_2 s_4^{c_3-1} s_3 \cdots = \ell_0^{c_1} (a s_3^{c_2-1} s_2 s_4^{c_3-1} s_3) \cdots$$

à l'aide de l'identité :

$$as_{2n+1}^{c_{2n}-1} s_{2n} s_{2n+2}^{c_{2n+1}-1} s_{2n+1} = [(as_{2n+1}a^{-1})^{c_{2n}-1} as_{2n}s_{2n+1}a^{-1}]^{c_{2n+1}} a.$$

## 2.2.6 Factorisation de Lyndon et facteurs singuliers

Jean Berstel nous avait signalé les similarités entre la factorisation décrite à l'exemple 8 et une factorisation donnée par Wen & Wen [WW94] pour le mot de Fibonacci. Dans leur article, ces auteurs considèrent la factorisation du mot de Fibonacci  $f = w_0 w_1 w_2 \dots$ , où les longueurs des mots  $w_n$  sont les nombres de Fibonacci  $F_n$  donnés par la récurrence  $F_0 = F_1 = 1$  et  $F_{n+1} = F_n + F_{n-1}$  ( $n \geq 1$ ). On a donc (cf Ex. 8)  $w_0 = a$ ,  $w_1 = b$ ,  $w_2 = aa$ ,  $w_3 = bab$ ,  $w_4 = aabaa$ , etc. Wen & Wen appellent les mots  $w_k$  les *facteurs singuliers* du mot de Fibonacci. Nous avons pu expliquer l'observation de Jean Berstel, à savoir que  $\ell_0 = ab = w_0 w_1$ ,  $\ell_1 = aabab = w_2 w_3$ , etc.

Nous avons aussi pu reformuler certains résultats de Wen & Wen en recherchant dans la factorisation de Lyndon du mot de Fibonacci les facteurs singuliers. La factorisation (2.4) des mots sturmiens caractéristiques nous avait permis de généraliser le travail de Wen & Wen et de définir pour ceux-ci des facteurs singuliers, et d'exprimer tout mot Sturmien caractéristique en termes de ses facteurs singuliers [Mel99]. Dans leur article, Wen & Wen avaient montré une propriété d'invariance de la factorisation du mot de Fibonacci en termes de facteurs singuliers  $f = w_0 w_1 w_2 \dots$ . Ils avaient montré que si, pour  $n \geq 0$  fixé, on isole les facteurs  $w_n$  dans le mot de Fibonacci (ces facteurs ne se chevauchent pas), on obtient :

$$f = \left( \prod_{j=0}^{n-1} w_j \right) w_n^{(1)} z_1 w_n^{(2)} z_2 \dots \quad (2.5)$$

où les mots  $z_n$  ( $n \geq 1$ ) sont toujours soit  $z_n = w_{n-1}$  ou  $z_n = w_{n+1}$  et que, de plus, le mot infini  $z_0 z_1 z_2 \dots$  est le mot de Fibonacci sur l'alphabet  $\{w_{n+1}, w_{n-1}\}$ . Par exemple, si on isole les occurrences du facteur singulier  $w_2 = aa$ , on trouve :

$$f = (ab) \quad aa(bab)aa(b)aa(bab)aa(bab)aa(b)aa(bab)aa(b)aa \dots$$

et on constate qu'en effet le mot  $ABAABAB \dots$  avec  $A = bab$  et  $B = b$  est bien le début du mot de Fibonacci sur  $\{bab, b\}$ .

De la même manière, nous avons montré que la factorisation de Lyndon du mot de Fibonacci admet une propriété d'invariance. Nous l'avons mentionné plus haut, le mot  $\varphi(f)$  obtenu du mot de Fibonacci en appliquant le morphisme défini par  $\varphi(a) = aab$  et  $\varphi(b) = ab$  est le mot de Fibonacci sur l'alphabet  $\{aab, ab\}$ . En réalité, on peut montrer que  $\varphi(\ell_n) = \ell_{n+1}$ ,  $\varphi^k(f) = \prod_{n \geq k} \ell_n$  et que le mot  $\varphi^k(f)$  est le mot de Fibonacci sur l'alphabet  $\{\ell'_n, \ell''_n\}$  aussi bien que sur l'alphabet  $\{\ell_n, \ell'_n\}$ . Si on exprime les mots de Lyndon  $\ell'_n$  et  $\ell''_n$  en fonction des facteurs singuliers  $w_{2n-1}$  et  $w_{2n+1}$ , cette invariance de la factorisation de Lyndon nous redonne le résultat de Wen & Wen.

Fort de ce résultat, nous avons pu proposer une généralisation des facteurs singuliers pour les mots sturmiens caractéristiques et donner, à partir d'une propriété d'invariance de la factorisation du théorème 11, un analogue de la formule (2.5) donné par Wen & Wen pour le mot de Fibonacci. Ces résultats très techniques ne seront pas détaillés.

## 2.2.7 Facteurs des mots sturmiens et mots de Christoffel

La factorisation de Lyndon (2.4) des mots sturmiens caractéristiques nous avait permis de retrouver un résultat originalement donné par Bertel et de Luca [Bd97]. Ils avaient montré que les facteurs des mots sturmiens qui sont des mots de Lyndon coïncident avec l'ensemble des mots de Christoffel primitifs positifs sur deux lettres, que nous définissons maintenant.

Soit  $p, q$  deux entiers positifs. Traçons dans le plan un chemin discret empruntant des pas est et des pas nord et suivant *au plus près* le segment de droite liant l'origine au point  $(q, p)$ . Le mot obtenu du chemin en substituant au pas est une lettre  $a$  et au pas nord une lettre  $b$  est un mot de Christoffel et tout mot de Christoffel est obtenu de cette façon. La figure 2.1 en donne un exemple.

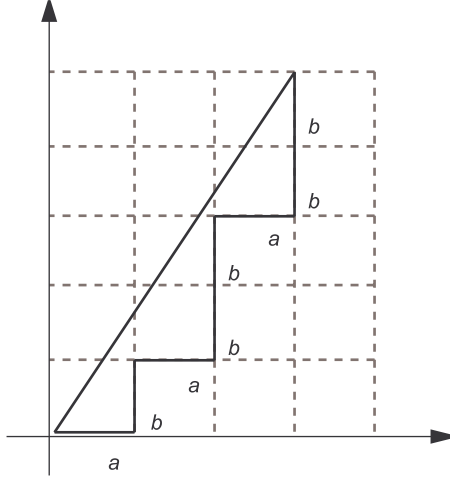


FIG. 2.1: Chemin dans le plan et mot de Christoffel primitif de pente  $5/3$

Si  $\text{pgcd}(p, q) = d$  alors le chemin intersecte les points à coordonnées entières  $(q_0, p_0), (2q_0, 2p_0), \dots, (dq_0, dp_0)$  (où  $p_0 = p/d$  et  $q_0 = q/d$ ). Si  $\text{pgcd}(p, q) = 1$  alors le chemin ne croise aucun point à coordonnées entières. Dans ce cas, on dit que le mot de Christoffel est *primitif*. Nous ne considérerons que les mots de Christoffel associés aux segments de pentes rationnelles positives, c'est-à-dire avec  $p, q \geq 0$ , appelés mots de Christoffel *positifs*. A partir de maintenant, nous ne considérerons que les *mots de Christoffel primitifs positifs*, que nous appellerons plus simplement mots de Christoffel pour alléger la discussion. Les mots de Christoffel sont donc en bijection avec les nombres rationnels positifs écrits sous forme réduite; le nombre associé au mot de Christoffel  $\ell$  est  $|\ell|_b/|\ell|_a$ .

Il existe un procédé récursif qui permet d'obtenir les mots de Christoffel. Partant des mots  $u_0 = a$  et  $v_0 = b$ , et étant donné une suite d'entiers  $(c_i)_{i=0, \dots, n}$ , on définit les mots

$$u_{i+1} = u_i v_i^{c_i} \quad \text{et} \quad v_{i+1} = u_{i+1}^{c_{i+1}} v_i. \quad (2.6)$$

Tous les mots de la suite sont des mots de Christoffel. Le dernier mot obtenu (selon la parité de  $n$ ) est le mot de Christoffel associé au nombre rationnel  $p/q$  dont la fraction rationnelle (finie) est donnée par la suite  $(c_i)_{i=0, \dots, n}$ . Par exemple, le procédé appliqué à la suite  $(1, 1, 2)$  donne la suite de mots  $u_0 = a$ ,

$v_0 = b$ ,  $u_1 = ab$ ,  $v_1 = abb$  et finalement  $u_2 = ab(abb)^2$ . On vérifie que ce mot est associé au nombre  $5/3$ , dont la fraction continue est  $[1; 1, 2]$  (cf figure 2.1). Ce passage entre nombre rationnel et mots de Christoffel utilisant les fractions continues est un élément essentiel de la théorie des mots de Christoffel [BL93, Lau95]. En particulier, l'unicité de la fraction continue pour  $p/q$  montre que la suite  $(c_i)_{i=0, \dots, n}$  définie en (2.6) amenant jusqu'au mot associé à  $p/q$  est unique.

C'est cette description des mots de Christoffel qui nous avait permis de montrer le résultat de Berstel et de Luca [Bd97]. Tout facteur d'un mot sturmien caractéristique et qui est aussi un mot de Lyndon est nécessairement un facteur d'un mot de Lyndon  $\ell_n = (as_{2n+1}a^{-1})^{c_{2n}-1}as_{2n}s_{2n+1}a^{-1}$  de la factorisation (2.4) (aucun mot de Lyndon ne peut chevaucher un produit décroissant de deux mots de Lyndon). Or, on peut montrer que les seuls facteurs de Lyndon du mot de Lyndon  $\ell_n$  sont :

$$\begin{aligned} & as_{2n+1}a^{-1} \quad \text{ou} \\ & (as_{2n+1}a^{-1})^k as_{2n}s_{2n+1}a^{-1}, \text{ avec } 0 \leq k \leq c_{2n} - 1 \quad \text{ou} \\ & as_{2n}s_{2n+1}a^{-1}. \end{aligned} \quad (2.7)$$

pour un certain  $n \geq 0$ . La suite d'exposant  $(c_n)_{n \geq 0}$  associée au mot sturmien, tronquée au rang appropriée, combinée à une récurrence définissant les mots en (2.7), permet de construire une suite comme en (2.6) pour montrer que ces mots sont des mots de Christoffel. Inversement, il est possible de donner un mot sturmien caractéristique possédant un mot de Christoffel donné comme facteur. Il suffit de considérer un mot sturmien caractéristique associé à une suite  $(c_n)_{n \geq 0}$  qui prolonge la fraction continue du nombre associé au mot de Christoffel.

La preuve complète de ce résultat nécessite qu'on étudie la factorisation standard des mots de Lyndon, que nous n'abordons qu'à la prochaine section. En regardant de près les propriétés des mots de Lyndon apparaissant dans la factorisation, on montre aussi que le produit infini (2.4) est la factorisation du mot sturmien caractéristique sur toute factorisation de Viennot (cf [Melar]).

## 2.3 Mots de de Lyndon équilibrés

Nous allons maintenant aborder l'étude des arbres de Lyndon, et étudier plus particulièrement les mots de Lyndon pour lesquels cet arbre est unique. Cette section, par rapport aux précédentes, est plus détaillée et contient des résultats nouveaux.

### 2.3.1 Factorisation standard des mots de Lyndon

La proposition 2 affirme que tout mot de Lyndon non-réduit à une lettre est un produit de deux mots de Lyndon. On a, par exemple,  $aabab = (aab)(ab)$ , et  $aababb = (aab)(abb)$ , mais aussi  $aababb = (aabab)(b) = (a)(ababb)$ . La *factorisation standard droite* d'un mot de Lyndon  $w$  est le couple  $(u, v)$  tel que  $w = uv$  et  $v$  est un mot de Lyndon de longueur maximale. On peut alors montrer que le mot  $u$  est aussi un mot de Lyndon et qu'on a  $u < uv < v$  (cf [Lot83, Chap. 5]). On peut définir, de la même manière, la *factorisation standard gauche* d'un mot de Lyndon en prenant pour  $u$  le préfixe de  $w$  de longueur maximale et

qui soit un mot de Lyndon. On désignera le plus souvent la factorisation standard (gauche ou droite) d'un mot de Lyndon en écrivant  $w = w'w''$ . L'arbre de Lyndon (gauche ou droit) associé à un mot de Lyndon est obtenu en itérant le calcul de la factorisation standard sur les facteurs de la factorisation standard (gauche ou droit). La figure 2.2 illustre l'arbre gauche et droit obtenu pour le mot  $aababb$ .

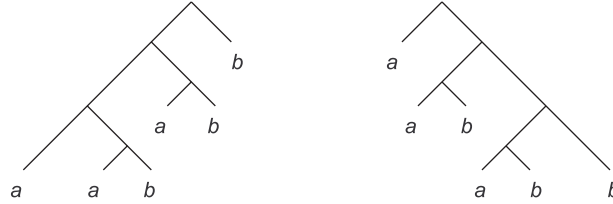


FIG. 2.2: Arbres de Lyndon gauche et droit associés au mot  $aababb$

**Proposition 12** (Lyndon [CFL58], cf [Lot83])

Soit  $u = x_1y_1$  la factorisation standard droite du mot de Lyndon  $u$ . Alors le produit  $w = uv$  est la factorisation standard droite du mot de Lyndon  $w$  si et seulement si on a  $y_1 \geq v$ .

De même, soit  $v = x_2y_2$  la factorisation standard gauche du mot de Lyndon  $v$ . Alors le produit  $w = uv$  est la factorisation standard gauche du mot de Lyndon  $w$  si et seulement si on a  $u \geq x_2$ .

Ainsi, on voit que la factorisation  $aababb = (aab)(abb)$  n'est standard ni à gauche, ni à droite puisque  $aab < ab < abb$ . On remarque aussi que le mot  $aabab$  ne possède qu'une seule factorisation  $aabab = (aab)(ab)$  qui est donc standard à gauche comme à droite (et on vérifie qu'en effet  $aab \geq a$  et  $(aab)'' = ab \leq ab$ ).

**Observation 13** Soit  $u, v$  des mots de Lyndon tels que  $u < v$  et tels que la factorisation  $uv$  soit standard à gauche (resp. à droite). Alors, pour tout  $n \geq 2$ , la factorisation  $u^n v = u(u^{n-1}v)$  est standard à gauche (resp. à droite). De même, pour tout  $n \geq 2$ , la factorisation  $uv^n = (uv^{n-1})v$  est standard à gauche (resp. à droite).

**Définition 14** Un mot de Lyndon  $w$  sur un alphabet  $A$  est équilibré si

- (i) soit  $c$  est une lettre,
- (ii) soit le mot  $w$  possède une unique factorisation  $w = w'w''$  en produit croissant de deux mots de Lyndon, et les mots  $w'$  et  $w''$  sont des mots de Lyndon équilibrés.

Par exemple, les mots  $a^n b$  et  $ab^n$  sont des mots de Lyndon équilibrés. Plus généralement, si  $u$  et  $v$  sont des mots de Lyndon équilibrés tels que la factorisation  $uv$  est standard (c'est-à-dire  $u'' \geq v$  et  $u \geq v'$ ) alors les mots  $u^n v$  et  $uv^n$  sont des mots de Lyndon équilibrés. La condition (ii) entraîne que les factorisations gauche et droite du mot  $w$  coïncident. Cela se traduit par les inégalités :

$$(w')'' \geq w'' \quad \text{et} \quad w' \geq (w'')' \quad (2.8)$$

obtenues de la proposition 12 appliquée à la factorisation  $w = w'w''$ . Nous désignerons par  $\bar{L}(A)$  l'ensemble des mots de Lyndon équilibrés sur l'alphabet  $A$ .

### 2.3.2 Le cas $A = \{a, b\}$

**Théorème 15** *L'ensemble des mots de Lyndon équilibrés sur l'alphabet  $\{a, b\}$  (avec  $a < b$ ) coïncide avec l'ensemble des mots de Christoffel primitifs positifs.*

Laurier [Lau95, p. 54] avait remarqué que les mots de Christoffel sont des mots de Lyndon équilibrés. Nous allons nous restreindre à montrer l'inclusion inverse, c'est-à-dire que tout mot de Lyndon équilibrés est un mot de Christoffel. Berstel et de Luca [Bd97] avaient donné un procédé de génération arborescent pour les mots de Christoffel primitifs positifs qui reprenaient le procédé arborescent de Stern-Brocot pour la construction des nombres rationnels positifs sous forme réduite (voir [GKP94, Sect. 4.2], par exemple). Nous allons montrer que ce même procédé permet d'engendrer les mots de Lyndon équilibrés sur  $\{a, b\}$ . Plus précisément, soit l'arbre binaire infini dont la racine est étiquetée par le couple  $(a, b)$ . On associe à chaque sommet de l'arbre un couple en suivant la règle : si un sommet de l'arbre a pour étiquette le couple  $(u, v)$  alors ses fils gauche et droit ont pour étiquettes les couples  $(u, uv)$  et  $(uv, v)$  respectivement. La figure 2.3 illustre ce procédé. Nous ferons référence à ce procédé en parlant de *l'arbre de génération des mots de Christoffel*. On retrouve alors dans l'arbre l'unique factorisation standard de tous les mots de Christoffel primitifs positifs (non réduit à une lettre). On retrouve l'arbre de Stern-Brocot si on remplace chaque couple  $(u, v)$  par le nombre rationnel  $p/q$  associé au mot de Christoffel  $uv$ . La figure 2.4 illustre l'arbre binaire infini des couples  $(q, p)$  obtenu des mots de Christoffel primitifs positifs. La démonstration du théorème 15 repose entièrement sur le lemme suivant.

**Lemme 16** *Soit  $w$  un mot de Lyndon équilibré non réduit à une lettre sur l'alphabet  $\{a, b\}$ . Alors  $w$  est soit un mot sur  $\{a, ab\}$ , soit sur  $\{ab, b\}$ . De plus,  $\bar{L}(a, ab) \cap \bar{L}(ab, b) = \{ab\}$ .*

**Observation 17** Sur l'alphabet  $A = \{a, b\}$ , les seuls mots de Lyndon équilibrés  $w$  tel que  $w' = a$  sont de la forme  $w = a^n b$  ( $n \geq 1$ ). De même, les seuls mots de Lyndon équilibrés  $w$  tels que  $w'' = b$  sont de la forme  $w = ab^n$  ( $n \geq 1$ ).

On montre cette observation facilement par récurrence. On le constate aussi facilement sur l'arbre de génération (cf figure 2.3). La seule possibilité pour que la première composante d'un couple soit la lettre  $a$ , est que le couple soit attachée à un sommet le long de la branche extrême gauche. De même, les seuls couples avec la lettre  $b$  en seconde composante sont le long de la branche extrême droite.

On montre le lemme 16 par récurrence sur la longueur des mots. On vérifie facilement l'énoncé pour la longueur 2 ; le seul mot équilibré est  $ab$ . Soit  $w$  un mot de Lyndon équilibré et  $w = w'w''$  son unique factorisation, standard à gauche et à droite. Dans le cas où  $w'$  ou  $w''$  est une lettre, on obtient le résultat en vertu

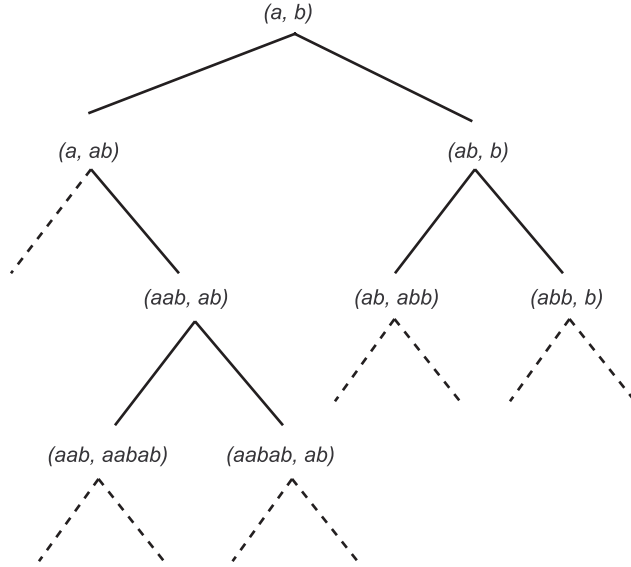


FIG. 2.3: Arbre de génération des factorisations standard des mots de Christoffel sur  $\{a, b\}$  (voir [Bd97])

de l'observation 17. On supposera donc que  $w'$  et  $w''$  sont de longueur au moins 2. Par récurrence, ce sont des mots de Lyndon équilibrés soit sur  $\{a, ab\}$  soit sur  $\{ab, b\}$ . Tout mot de  $\bar{L}(a, ab)$  débute par  $aa$  sauf le mot  $ab$ . Comme  $w' < w''$ , on ne peut avoir  $w' \in \bar{L}(ab, b)$  et  $w'' \in \bar{L}(a, ab)$ , sauf si  $w' = ab$ . De même,  $w' \in \bar{L}(a, ab)$  et  $w'' \in \bar{L}(ab, b)$  est impossible puisqu'on doit avoir  $w' \geq (w'')'$ . On conclut donc que  $w'$  et  $w''$  doivent appartenir simultanément à  $\bar{L}(a, ab)$  ou  $\bar{L}(ab, b)$ .

La condition  $\bar{L}(a, ab) \cap \bar{L}(ab, b) = \{ab\}$  établit, via une récurrence, le corollaire 1 de [BL93]. C'est aussi cette récurrence qui donne le corollaire 2 du même article affirmant que les mots de Christoffel dans le sous-arbre induit par un couple  $(u, v)$  sont des mots de Christoffel sur «l'alphabet»  $\{u, v\}$ . Laurier [Lau95] appelle le morphisme induit par  $a \mapsto u, b \mapsto v$  une *substitution standard*. Ce phénomène est familier des calculs avec les mots de Lyndon ou plus généralement avec les factorisations régulières du monoïde libre comme les factorisations de Viennot [Vie76] [Lot83, Chap.5]. Il faut aussi y voir un lien avec la propriété d'invariance de la factorisation de Lyndon des mots sturmiens caractéristiques (cf section 2.2.6).

### 2.3.3 Mots de Lyndon équilibrés à plus de deux lettres

On remarque que la définition 14 n'exige rien en ce qui concerne la cardinalité de l'alphabet. Elle permet en effet de considérer un alphabet de cardinalité quelconque, même infinie. Le théorème 15 nous permet de présenter les mots de Lyndon équilibrés comme une généralisation possible des mots de Christoffel à un alphabet de plus de deux lettres. Nous allons maintenant donner des résultats qui les confirment dans ce rôle. Nous allons ici nous concentrer sur le cas à trois

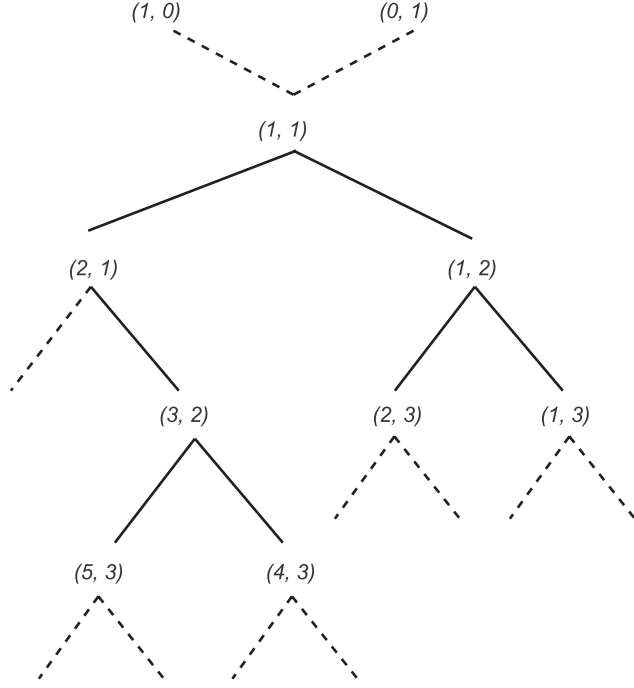


FIG. 2.4: Arbre de Stern-Brocot pour les nombres rationnels positifs (le couple  $(q, p)$  correspond au nombre  $p/q$ ).

lettres  $A = \{a, b, c\}$ , pour la simplicité de l'exposé, mais tous nos résultats peuvent être formulés pour un alphabet fini quelconque.

Notre premier résultat est l'analogie de la construction de Berstel et de Luca [Bd97] pour les mots de Christoffel primitifs positifs. Nous allons donner un procédé de génération arborescent pour les mots de Lyndon équilibrés. Considérons un arbre binaire infini dont la racine est étiquetée par le triplet  $(a, b, c)$ . On associe une étiquette à tout sommet de l'arbre comme suit. Si un sommet porte l'étiquette  $(u, v, w)$  alors ses fils gauche et droit portent les étiquettes  $(u, uw, v)$  et  $(uw, v, w)$ , respectivement. On écrira :

$$(u, v, w) \xrightarrow{L} (u, uw, v) \tag{2.9}$$

$$(u, v, w) \xrightarrow{R} (uw, v, w) \tag{2.10}$$

pour désigner ces dérivations gauche et droite dans l'arbre. Nous ferons référence à ce procédé en parlant de l'*arbre de génération des mots de Lyndon équilibrés* (cette terminologie est justifiée par le théorème 22). Nous allons maintenant expliciter certains invariants qui apparaissent dans l'arbre.

**Définition 18** *Un triplet de mots de Lyndon équilibrés  $(u, v, w)$  est standard si et seulement si :*

- (i)  *$uw$  et  $uwv$  sont des mots de Lyndon équilibrés,*

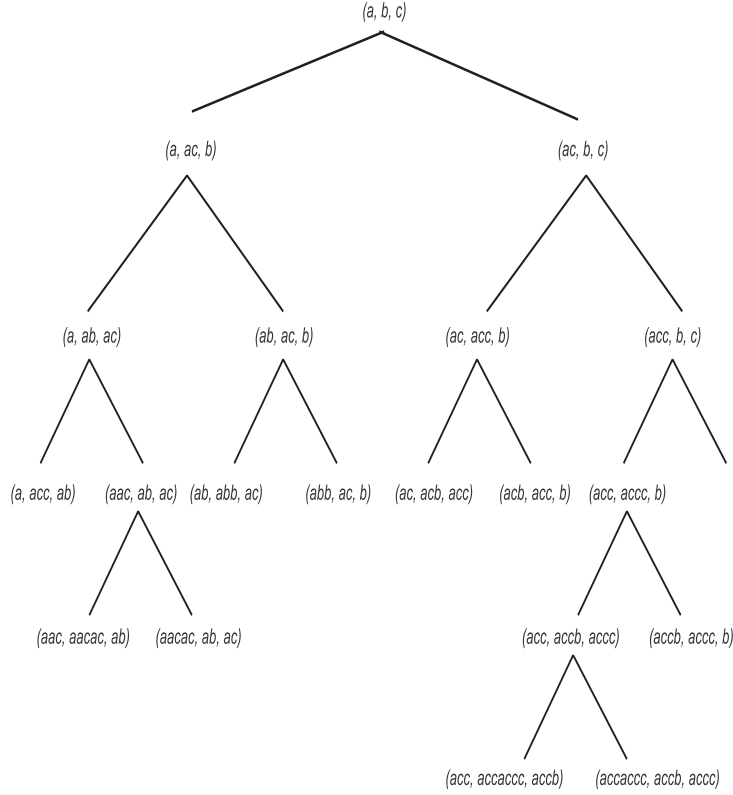


FIG. 2.5: Arbre de génération de triplets standard de mots équilibrés

- (ii) les factorisations standard (gauche et droite) de ces mots sont  $(uw)' = u$ ,  $(uw)'' = w$ , et  $(uww)' = uw$ ,  $(uww)'' = v$ .

**Proposition 19** *Les triplets de l'arbre binaire de génération sont standard. De plus, les mots équilibrés  $uw$  obtenus de triplets  $(u, v, w)$  apparaissant dans l'arbre sont des mots de  $\bar{L}(a, b, c) \setminus \bar{L}(b, c)$ .*

La preuve nécessite que l'on montre les inégalités (ii) de la définition 18 pour les mots  $uw$  et  $uww$ , ce qui découle en partie des remarques de l'observation 13. La dernière partie de l'énoncé vient du fait que le mot  $u$  commence toujours par  $a$ .

**Lemme 20** *Soit  $w$  un mot équilibré sur  $\{a, b, c\}$ . Alors  $c$ 'est un mot équilibré soit sur  $\{a, ac, b\}$ , soit sur  $\{ac, b, c\}$ . De plus, on a  $\bar{L}(a, ac, b) \cap \bar{L}(ac, b, c) = \{ac\}$ .*

Avant d'aborder la preuve du lemme, il nous faut faire une remarque.

**Observation 21** Désignons par  $X$  le code  $\{a, ac, b\}$ , (qu'on considère comme un « alphabet »). L'ordre lexicographique  $<_X$  sur le monoïde libre  $X^*$  induit par  $a < ac < b$  coïncide avec l'ordre lexicographique usuel sur  $\{a, b, c\}^*$  induit par l'inclusion  $\{a, ac, b\}^* \subset \{a, b, c\}^*$ . Il en va de même pour le monoïde libre sur  $Y =$

$\{ac, b, c\}$ . Par conséquent, les mots de Lyndon sur  $\{a, ac, b\}$  (ou  $\{ac, b, c\}$ ) sont aussi des mots de Lyndon sur  $\{a, b, c\}$  et leurs factorisations standard (gauche ou droite) dans l'un ou l'autre monoïde coïncident dans  $\{a, b, c\}^*$  (en prenant la précaution de déployer les feuilles étiquetées par  $ac$  dans  $\bar{L}(X)$  ou  $\bar{L}(Y)$  en le sous-arbre approprié). C'est là un fait connu (cf [Reu86], [Lot83, Chap. 5.4] ou [Reu93] par exemple).

*Preuve du lemme 20.*

Soit  $w$  un mot équilibré sur  $\{a, b, c\}$ . Si c'est un mot équilibré sur  $\{a, b\}$  ou sur  $\{b, c\}$  alors c'est un mot équilibré sur  $\{a, ac, b\}$  ou sur  $\{ac, b, c\}$  respectivement. Si c'est plutôt un mot équilibré sur  $\{a, c\}$ , alors c'est un mot équilibré sur  $\{a, ac\}$  ou sur  $\{ac, c\}$ , en vertu du lemme 16, donc sur  $\{a, ac, b\}$  ou sur  $\{ac, b, c\}$ . On peut donc supposer que  $|w|_a, |w|_b, |w|_c \geq 1$ .

On procède par récurrence sur la longueur des mots. On reprend les notations  $X = \{a, ac, b\}$  et  $Y = \{ac, b, c\}$  de l'observation 21. On a  $w = w'w''$ , et par récurrence les mots  $w'$  et  $w''$  sont soit dans  $\bar{L}(X)$ , soit dans  $\bar{L}(Y)$ . Supposons d'abord que  $w' \in \bar{L}(a, ac, b)$ , mais que  $w'' \in \bar{L}(ac, b, c)$ . Observons d'abord qu'on ne peut avoir  $w'' \in \bar{L}(b, c)$ , sinon on a nécessairement  $w' < (w'')'$  ce qui contredit le fait que la factorisation  $w'w''$  est standard à gauche. La « lettre »  $ac \in Y$  apparaît donc nécessairement dans  $w''$ , qui commence donc par  $ac$ . Si la lettre  $a \in X$  apparaît dans le mot  $w'$ , il commence nécessairement par  $aa$  ou par  $ab$ . Mais alors on trouve  $w' < (w'')'$  ce qui encore une fois contredit le fait que  $w'w''$  est standard à gauche. Le mot  $w'$  ne peut donc contenir la lettre  $a$ . Mais alors c'est un mot sur  $Y$  et  $w \in \bar{L}(Y)$ .

Supposons maintenant que  $w' \in \bar{L}(ac, b, c)$ , mais que  $w'' \in \bar{L}(a, ac, b)$ . On raisonne comme plus haut pour constater que le mot  $w''$  ne peut contenir la lettre  $a \in X$ . C'est alors un mot sur  $\{ac, b\}$  et donc  $w \in \bar{L}(ac, b, c)$ .

Notez que les arguments donnés plus haut échouent dans le cas où les mots sont trop courts (sur  $X$  ou  $Y$ ). La preuve peut être complétée en étudiant les cas possibles.

**Théorème 22** *Pour tout mot  $\ell$  équilibré de  $\bar{L}(a, b, c) \setminus \bar{L}(b, c)$ , et non réduit à une lettre, il existe un unique triplet standard  $(u, v, w)$  issu de  $(a, b, c)$  tels que  $\ell = uw$ .*

On reprend les notations du lemme précédent. Le mot  $\ell$  est soit dans  $\bar{L}(X)$ , soit dans  $\bar{L}(Y)$ , en vertu du lemme 20. Supposons qu'on ait  $\ell \in \bar{L}(X)$ , pour fixer les idées. Le mot  $\ell$  est, en tant que mot sur  $X$ , plus court sauf si  $|\ell|_c = 0$ , auquel cas, c'est un mot sur  $\{a, b\}$ . Mais alors, c'est un mot équilibré de  $\bar{L}(a, ab)$  ou de  $\bar{L}(ab, b)$ , qui sur  $\{a, ab\}$  ou  $\{ab, b\}$  est plus court. Comme on a  $\bar{L}(X) = \bar{L}(a, ab, ac) \cup \bar{L}(ab, ac, b)$  on trouve dans tous les cas un unique triplet  $(u, v, w)$  de mots de  $\bar{L}(X)$  issu de  $(a, ac, b)$  tel que  $\ell = uw$ . Ce triplet est aussi un triplet de mot de  $\bar{L}(a, b, c)$  et issu de  $(a, b, c)$  en vertu du lemme 20. Si  $|\ell|_c > 0$ , on obtient par récurrence un unique triplet  $(u, v, w)$  de mots équilibrés de issu de  $(a, ac, b)$  tel que  $\ell = uw$ . Ce triplet est aussi un triplet de mots équilibrés de  $\bar{L}(a, b, c)$ , accessible de  $(a, b, c)$ . Le cas  $w \in \bar{L}(Y)$  se traite de la même manière. Pour conclure, on montre l'unicité du triplet donnant la factorisation  $\ell = uw$  en remarquant que les sous-arbres issus des triplets  $(a, ac, b)$  et  $(ac, b, c)$  ne possèdent pas de triplets communs. Cela découle de la preuve du lemme 20.

### 2.3.4 Mots de Lyndon équilibrés et compositions

Nous l'avons mentionné plus haut, l'un des intérêts des mots de Christoffel est qu'ils constituent un codage des nombres rationnels positifs. Cette correspondance est très riche et rejoint la théorie des fractions continues ordinaires en théorie des nombres [BL93, Bd97]. A chaque nombre rationnel correspond un unique mot de Christoffel (cf la définition au début de la section 2.2.7) et inversement, à tout nombre rationnel  $r$  on peut associer un unique mot de Christoffel en passant par la fraction continue de  $r$  (cf Eq. (2.6)).

La généralisation que proposent les mots de Lyndon équilibrés pour les mots de Christoffel donnent une correspondance analogue que nous explicitons maintenant. Etant donné un mot de Lyndon équilibré  $\ell$  on désigne par  $|\ell|_a$ ,  $|\ell|_b$ ,  $|\ell|_c$  son nombre de lettres  $a$ ,  $b$  et  $c$ .

**Proposition 23** *L'application  $\ell \rightarrow (|\ell|_a, |\ell|_b, |\ell|_c)$  met en bijection les mots de Lyndon équilibrés de  $\bar{L}(a, b, c)$  avec les compositions en trois parts relativement premières.*

La preuve utilise une récurrence qui repose sur le lemme 20 (et sur le lemme 16 pour le cas  $|\ell|_a = 0$ ). Ce résultat confirme les mots de Lyndon équilibrés comme généralisation des mots de Christoffel primitifs positifs, dans la mesure où les compositions en trois relativement premières sont un bon analogue des nombres rationnels écrits sous forme réduite.

Le calcul du mot de Lyndon équilibré associée à une composition peut se faire en suivant la récurrence induite par le lemme 20 et en « changeant d'alphabet ». Considérez l'algorithme suivant, défini sur les compositions avec première part *non nulle* (les compositions avec une première part nulle correspondent à des mots équilibrés sur  $\{b, c\}$ ) :

$$(\alpha, \beta, \gamma) \rightarrow \begin{cases} (\alpha - \gamma, \gamma, \beta) & \text{si } \alpha > \gamma \\ (\alpha, \beta, \gamma - \alpha) & \text{si } \alpha \leq \gamma \end{cases} \quad (2.11)$$

avec la condition supplémentaire que l'algorithme s'arrête dès que l'on rencontre le triplet d'entiers  $(1, 0, 1)$ . On peut calculer le mot associé au triplet  $(\alpha, \beta, \gamma)$  en faisant suivre à chaque pas de l'algorithme un triplet standard  $(u, v, w)$  de mots équilibrés. On démarre en associant à la composition de départ le triplet standard  $(a, b, c)$ . Etant donné le triplet standard  $(u, v, w)$  associé à une composition, on associe à la composition suivante le triplet  $(u, uw, v)$  ou  $(uw, v, w)$  selon que l'on a appliqué la première ou la seconde des règles (2.11). L'algorithme s'arrête lorsqu'on rencontre la composition  $(1, 0, 1)$  accompagné du triplet de mots  $(u, v, w)$ . Le mot associé à la composition de départ est  $\ell = uw$ .

Calculons par exemple le mot associé à la composition  $(1, 2, 2)$ . On obtient la suite de compositions et de triplets standard suivants :

$$\begin{array}{ccccccc} (1, 2, 2) & \rightarrow & (1, 2, 1) & \rightarrow & (1, 2, 0) & \rightarrow & (1, 0, 2) \\ (a, b, c) & \rightarrow & (ac, b, c) & \rightarrow & (acc, b, c) & \rightarrow & (ac^2, ac^3, b) \rightarrow (1, 0, 1) \\ & & & & & & (ac^2b, ac^3, b) \end{array}$$

de sorte que le mot associé à la composition  $(1, 2, 2)$  est  $\ell = ac^2b^2$ . Remarquez que l'on peut aussi admettre un dernier pas de l'algorithme et appliquer la seconde règle pour passer de  $(1, 0, 1)$  à  $(1, 0, 0)$ . Le mot associé à la composition

est alors obtenu en prenant la première composante du triplet standard qui est alors  $(uw, v, w)$ , obtenu de  $(u, v, w)$  par la règle de réécriture (2.10). Ainsi, dans l'exemple, une étape de calcul supplémentaire utilisant la seconde règle nous amène à la paire

$$\begin{aligned} &(1, 0, 0) \\ &(ac^2b^2, ac^3, b) \end{aligned}$$

avec le mot  $\ell$  en première composante du triplet standard. La preuve du résultat suit de l'observation suivante. Supposons qu'on position  $(\alpha_0, \beta_0, \gamma_0)$  associée à un mot équilibré sur  $X = \{a, ac, b\}$ . Alors ce mot est associé, sur  $A$ , à la composition  $(\alpha, \beta, \gamma)$ , où  $\alpha = \alpha_0 + \beta_0$ ,  $\beta = \gamma_0$  et  $\gamma = \beta_0$ . On vérifie aisément que la composition  $(\alpha_0, \beta_0, \gamma_0)$  s'obtient de  $(\alpha, \beta, \gamma)$  en appliquant la règle (2.11). On peut formuler une observation similaire pour le cas  $Y = \{ac, b, c\}$ . Observez de plus que la composition  $(\alpha, \beta, \gamma)$  est en parts relativement premières si et seulement si c'est aussi le cas pour la composition  $(\alpha_0, \beta_0, \gamma_0)$ .

### 2.3.5 Arbre de Stern-Brocot pour les compositions

On peut construire un analogue de l'arbre de Stern-Brocot pour l'ensemble des composition en trois parts relativement premières, dont la première part est non-nulle. Cette condition supplémentaire vient du fait que l'arbre de génération des mots de Lyndon équilibrés sur  $\{a, b, c\}$  ne produit que les mots de  $\bar{L}(a, b, c) \setminus \bar{L}(b, c)$ . On considère un arbre binaire infini avec la composition  $(1, 0, 1)$  à la racine. La composition associée à un sommet de l'arbre est alors obtenu en sommant composante à composante les compositions du *dernier ancêtre gauche* et de l'*avant-dernier ancêtre droit* rencontré le long du chemin menant de la racine au sommet considéré. On peut vérifier cette règle de construction sur la figure 2.6. On place aussi « au-dessus » de la racine les trois compositions  $(1, 0, 0)$ ,  $(0, 1, 0)$  et  $(0, 0, 1)$ . On considèrera que la composition  $(1, 0, 0)$  est un ancêtre gauche des sommets du sous-arbre gauche, et que la composition  $(0, 1, 0)$  est un ancêtre droit des sommets du sous-arbre gauche. De même, les compositions  $(0, 1, 0)$  et  $(0, 0, 1)$  sont des ancêtres, gauche et droit respectivement, des sommets du sous-arbre droit.

Par exemple, on vérifie sur la figure 2.6 que la composition  $(3, 0, 8)$  est obtenu en sommant composante à composante les compositions  $(2, 0, 5)$  et  $(1, 0, 3)$  qui sont ses derniers ancêtre gauche et avant-dernier ancêtre droit. Les compositions à sommer pour obtenir la composition  $(1, 2, 0)$  sont  $(1, 1, 0)$ , son ancêtre gauche, et  $(0, 1, 0)$  qui joue ici le rôle d'avant-dernier ancêtre droit.

Les règles de dérivations (2.9) et (2.10) pour les triplets standard montrent que toute composition en trois parts relativement premières est associée à un unique sommet de l'arbre et que le triplet correspondant  $(u, v, w)$  de l'arbre pour les mots équilibrés donne le mot  $\ell = uw$  associée à la composition via l'application de la proposition 23.

On peut aussi donner une formulation analogue à celle de Berstel et de Luca [Bd97] en termes de chemin dans l'arbre, utilisant des matrices qui effectuent le calcul des parts des compositions. Plus précisément, considérons les matrices

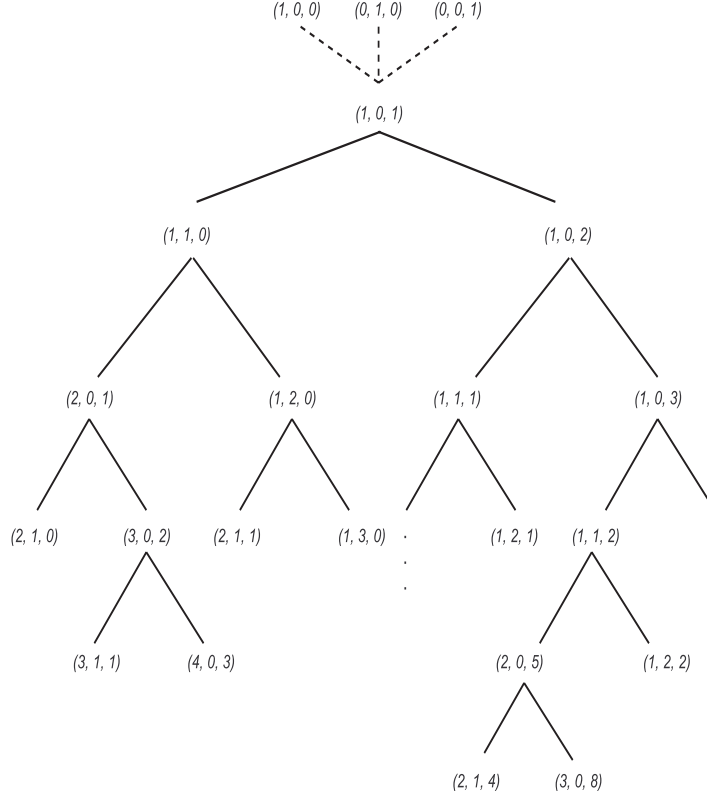


FIG. 2.6: Arbre de génération des compositions associés aux mots de Lyndon équilibrés sur  $\{a, b, c\}$

$$L = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad R = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.12)$$

Convenons d'écrire  $R^{c_0} L^{c_1} \dots L^{c_{n-1}} R^{c_n}$  pour désigner (avec les mêmes lettres) les suites qui décrivent les chemins dans l'arbre des mots de Lyndon équilibrés à partir de la racine. Une lettre  $L$  désigne un pas allant d'un sommet vers son fils gauche, et la lettre  $R$  désigne un pas allant d'un sommet vers son fils droit. Etant donné un chemin  $W$  dans l'arbre, on peut interpréter chacune des lettres du chemin comme une réécriture (2.9) ou (2.10). Ainsi, partant de la racine on obtient un triplet de l'arbre en suivant le chemin décrit par  $W$ . Le théorème peut donc être reformulé en : pour tout mot équilibré  $\ell$  il existe un unique chemin  $W$  menant au triplet standard  $W(a, b, c) = (u, v, w)$  tel que  $\ell = uw$ . On peut aussi interpréter un chemin  $W$  comme le produit des matrices associées aux lettres du chemin. Les propositions suivantes sont des conséquences immédiates du théorème 2.2 et des notations que nous venons d'introduire. Il faut les comparer à [Bd97, Prop.6.2, Th. 7.1].

**Proposition 24** *Soit un chemin  $W$  et posons  $W(a, b, c) = (u, v, w)$ . Alors la matrice obtenue de  $W$  par multiplication des matrices  $L$  et  $R$  définies en (2.12)*

est :

$$W = \begin{pmatrix} w_c & v_c & u_c \\ w_b & v_b & u_b \\ w_a & v_a & u_a \end{pmatrix}$$

Reprenons le chemin  $R^2LR$  qu'il faut suivre pour arriver au triplet  $(ac^2b, ac^3, b)$  en partant du triplet  $(a, b, c)$ . On vérifie que la matrice est :

$$R^2LR = \begin{pmatrix} 0 & 3 & 2 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

et que les colonnes, lues de haut en bas, donnent respectivement les multiplicités des lettres  $c, b$  et  $a$  dans les mots  $w = b, v = ac^3$  et  $u = ac^2b$ .

**Proposition 25** *Pour tout triplet d'entiers relativement premiers  $(\alpha, \beta, \gamma)$  tels que  $\alpha > 0$ , il existe un unique chemin  $W$  tels que*

$$\begin{pmatrix} \gamma \\ \beta \\ \alpha \end{pmatrix} = W \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

**Observation 26** Les déterminants des matrices  $R$  et  $L$  sont respectivement  $+1$  et  $-1$ . Il suit de la proposition 24 que le déterminant de la matrice associé à un triplet de mots de l'arbre de génération de  $\bar{L}(a, b, c) \setminus \bar{L}(b, c)$  est toujours  $\pm 1$ . Cette condition, ajoutée à la définition 18 des triplets standard, permet de montrer que les seuls triplets standard sont ceux apparaissant dans l'arbre de génération.

### 2.3.6 Approximation de points dans le plan

Le calcul de la fraction continue associée à un nombre est intimement lié au calcul du plus grand commun diviseur des entiers et est pour cette raison très voisin de l'algorithme d'Euclide. La première étape du calcul de la fraction continue d'un nombre rationnel  $p/q$  (avec  $p > q$ ) est  $p/q = m + r/q$  où  $m$  est le quotient de la division de  $p$  par  $q$  et  $r$  en est le reste. L'algorithme poursuit en réécrivant  $r/q = 1/(q/r)$  et en itérant le procédé sur le nombre  $q/r$ . Par exemple, le calcul de la fraction continue de  $5/3$  est :

$$\frac{5}{3} = 1 + \frac{2}{3} = 1 + \frac{1}{3/2} = 1 + \frac{1}{1 + \frac{1}{2}}$$

de sorte que la fraction continue de  $5/3$  est  $[1; 1, 2]$ .

Plusieurs généralisations des fractions continues aux dimensions supérieures ont été proposées [Bre81]. Certaines des généralisations recherchent à mettre en évidence les propriétés algébriques des nombres dont la fraction continue constitue une approximation. Il faut ici penser au résultat classique affirmant que la fraction continue (ordinaire) d'un nombre est périodique si et seulement le nombre est algébrique de degré 2.

Tout mot de Christoffel encode la fraction continue qui lui est associée. En effet, on peut tirer la fraction continue du chemin à suivre dans l'arbre de génération des mots de Christoffel. Les entiers de la fraction continue comptent alors le nombre de pas à effectuer à gauche et à droite, en succession. Par exemple, le nombre  $5/3$  se trouve dans l'arbre en partant de la racine et en suivant le chemin  $RLR$ . Les multiplicités gauche-droite sont donc 1, 1, 1 (droite, gauche, droite). On en tire la fraction continue  $[1; 1, 2]$  (il faut soustraire 1 au dernier entier de la fraction continue, voir [Bd97, Prop. 6.3]). Le calcul de la fraction continue peut aussi se faire par un algorithme soustractif, plus lent que l'algorithme d'Euclide mais équivalent. Plus précisément, on peut travailler sur le couple d'entiers  $(q, p)$  (relativement premiers) et le réécrire selon la règle (i)  $(q, p) \rightarrow (q, p - q)$  si  $q \leq p$  ou (ii)  $(q, p) \rightarrow (q - p, p)$  si  $q > p$ . Le calcul s'arrête dès que l'on atteint le couple  $(1, 0)$ . Par exemple, partant du couple  $(3, 5)$ , on obtient la suite :

$$(3, 5) \xrightarrow{\text{règle (i)}} (3, 2) \xrightarrow{\text{règle (ii)}} (1, 2) \xrightarrow{\text{règle (i)}} (1, 1) \xrightarrow{\text{règle (i)}} (1, 0).$$

On vérifie que le mot  $aabaabab$ , associé au même sommet dans l'arbre de Christoffel s'obtient en appliquant les réécritures  $(a, b) \rightarrow (ab, b) \rightarrow (ab, abb) \rightarrow (ababb, abb)$ , ou via les récurrences (2.6). La fraction continue de  $5/3$  s'obtient alors en comptant les multiplicités des règles (i) et (ii) appliquées en alternance.

Les fractions continues approximent les nombres réels, et sont les meilleures approximations possibles dans un sens bien précis (voir, par exemple, [HW38, Bre81]). On peut, de manière analogue, chercher à interpréter les mots de Lyndon équilibrés sur  $\{a, b, c\}$  comme des approximations de couples de nombres réels dans le plan. On a :

**Observation 27** Les mots de Christoffel de  $\bar{L}(a, b, c) \setminus \bar{L}(b, c)$  sont en bijection avec les points du plan à coordonnées rationnelles positives.

Ca n'est là qu'une autre version de la correspondance de la proposition 23. Le passage  $(\alpha, \beta, \gamma) \rightarrow (\frac{\beta}{\alpha}, \frac{\gamma}{\alpha})$  est bijectif. Les mots de  $\bar{L}(a, b, c) \setminus \bar{L}(b, c)$  correspondent donc à un ensemble dense de points dans le premier quadrans du plan. Cet aspect des mots de Lyndon équilibré est encore à exploiter.

Mentionnons seulement ici un résultat qui motive qu'on regarde plus en détail dans cette direction. L'algorithme qui calcule l'application inverse de la proposition 20 est invariant par multiplication des parts de la composition par un entier. En d'autres mots, si on applique l'algorithme sur un triplet d'entiers quelconques  $(\alpha, \beta, \gamma)$ , l'algorithme s'arrêtera sur le triplet  $(d, 0, 0)$  avec  $d = \text{pgcd}(\alpha, \beta, \gamma)$ . On peut de la même manière, appliquer l'algorithme sur un triplet de nombres réels quelconques  $(x, y, z)$ , qui s'arrêtera si le triplet est de la forme  $(x, y, z) = c \cdot (\alpha, \beta, \gamma)$  où  $c$  est un nombre réel quelconque et  $(\alpha, \beta, \gamma)$  est un triplet d'entiers. Sinon, l'algorithme ne s'arrêtera pas. On désignera par  $\rho : \bar{L}(a, b, c) \setminus \bar{L}(b, c) \rightarrow \mathbb{R}^2$  l'application qui associe à un mot de Lyndon équilibré  $\ell$  le couple de nombre rationnels positifs  $(\frac{|\ell|_b}{|\ell|_a}, \frac{|\ell|_c}{|\ell|_a})$  obtenus de la composition associée à  $\ell$ .

**Proposition 28** Soit  $(x, y)$  un couple de points dans le premier quadrans du plan. Soit  $(u_n)_{n \geq 0}$  la suite de mots équilibrés apparaissant en première composante des triplets de mots équilibrés obtenus par réécriture du triplet  $(a, b, c)$

en itérant l'algorithme (2.11) à partir du triplet de nombres réels  $(1, x, y)$ . Alors on a :

$$\lim_{n \rightarrow \infty} \rho(u_n) = (x, y).$$

Cette approximation peut se visualiser en une suite infinie de région du plan qui s'emboîte, un peu à la manière des intervalles standard associés aux mots de Christoffel pour les fractions continues ordinaires. Ainsi, on constate que les points obtenus de mots de Lyndon équilibrés du sous-arbre gauche issu du triplet  $(a, ac, b)$  sont dans la région  $L_{(a,b,c)} = \{(x, y) | 0 \leq x < \infty, 0 \leq y < 1\}$ . Les points obtenus des mots du sous-arbre droit sont, eux, dans la région  $R_{(a,b,c)} = \{(x, y) | 0 \leq x < \infty, 1 \leq y < \infty\}$ . Si on poursuit, on voit que les triplets  $(a, ab, ac)$  et  $(ab, ac, b)$  (issus du triplet  $(a, ac, b)$ ) induisent le découpage de la bande  $L_{(a,b,c)}$  en  $L_{(a,ac,b)} = \{(x, y) | 0 \leq x < 1, 0 \leq y < 1\}$  et  $R_{(a,ac,b)} = \{(x, y) | 1 \leq x < \infty, 0 \leq y < 1\}$ .

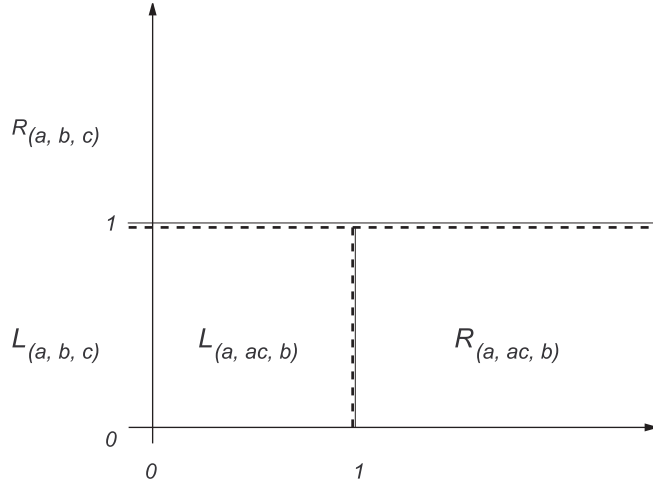


FIG. 2.7: Découpage du plan induit par l'approximation de points par les mots de Lyndon équilibrés

Par exemple, on obtient une suite d'approximations du point  $(2^{1/3}, \sqrt{2})$  en itérant le calcul de l'algorithme (2.11) à partir du triplets de nombres  $(1, 2^{1/3}, \sqrt{2})$ . La suite de pas de l'algorithme correspond au chemin débutant par :

$$LRL^2R^2LRL^5R^4LR^5L^{13}RL^{10}R^2LR^3LR^2LR^2L^9R^2L^6RL^{11}RLRL^8RL^3 \dots$$

Le même chemin, dans l'arbre de Stern-Brocot des compositions, nous amène à la composition

$$(92821476491888218, 116947732122548417, 131269390931910586)$$

qui donne l'approximation  $(1.2599210499 \dots, 1.41421356234 \dots)$  du point  $(2^{1/3}, \sqrt{2})$  exacte à neuf décimales, après 200 réécritures (2.11).

### 2.3.7 Mots de Lyndon équilibrés infinis

Nous imitons ici Borel et Laubie [BL93, Sect. III] et introduisons une famille de mots de Lyndon infinis qui jouent pour les mots de Lyndon équilibrés le même rôle que les mots de Christoffel primitifs infinis pour les mots de Christoffel.

**Définition 29** *Un mot infini est un mot de Lyndon équilibré infini s'il possède une infinité de préfixes qui sont des mots de Lyndon équilibrés.*

On peut prolonger tout mot de Lyndon équilibré fini en un mot de Lyndon équilibré infini en faisant :

$$\begin{array}{ccc} \bar{L} \setminus \{a, b, c\} & \rightarrow & \bar{L}^\infty \\ w & \mapsto & w'w^\infty \end{array}$$

Cette application est injective et permet de plonger les mots de Lyndon équilibrés finis dans l'ensemble des mots de Lyndon équilibrés infinis. Remarquons que le point du plan associé au mot  $w'w^n$  est égal à  $(\frac{|w'|_b+n|w|_b}{|w'|_a+n|w|_a}, \frac{|w'|_c+n|w|_c}{|w'|_a+n|w|_a})$ . De sorte que ce point converge vers  $(\frac{|w|_b}{|w|_a}, \frac{|w|_c}{|w|_a})$ . Il faut comparer la proposition suivante à [BL93, Prop. 4].

**Proposition 30** *Soit  $s$  un mot de Lyndon équilibré infini. La suite  $(\rho_n) = (\rho(s|_n))$  de points du plan obtenus à partir des préfixes  $s|_n$  de  $s$  (de longueur  $n$ ) est une suite convergente de points à coordonnées rationnelles et définit le point associé au mot  $s$ .*

Ce résultat nous encourage aussi à pousser l'étude des mots de Lyndon équilibrés infinis. On peut rechercher à calculer pour un mot de Lyndon infini sa *factorisation standard*, qui peut alors prendre deux formes possibles (que nous ne détaillerons pas ici). Cette factorisation standard devrait permettre, entre autre, d'obtenir une suite infini de mots de Lyndon équilibrés finis qui donne une suite d'approximations du point  $\rho(s)$ .

## 2.4 Conclusion et perspectives

Ce chapitre se concentrait sur le calcul de la factorisation de Lyndon des mots infinis. Nous avons vu comment ce calcul est lié à certaines régularités inévitables des mots infinis (section 2.2.4) et permet parfois de mettre à jour certaines propriétés des mots infinis. Le cas des mots sturmiens s'est révélé particulièrement intéressant. La factorisation de Lyndon du mot de Fibonacci nous a permis de lier ce calcul aux travaux de Wen & Wen [WW94] sur les facteurs singuliers du mot de Fibonacci. C'est cette direction de recherche qui nous avait poussé à étudier les propriétés d'invariance de la factorisation des mots sturmiens (2.4) et qui nous avait permis de définir les facteurs singuliers des mots sturmiens caractéristiques.

Il faudrait encore chercher à calculer la factorisation d'autres familles de mots infinis. Le calcul de la factorisation du mot de Thue-Morse [IM97] et de la suite de pliage [Mel96a] n'avait pas débouché sur un terrain aussi riche. Il faudrait encore chercher à voir si on peut lier la factorisation de Lyndon d'un mot infini et la propriété de récurrence des mots. Un mot récurrent est un mot qui, s'il

possède un facteur, le fait apparaître « partout » (quel que soit le rang où on se place il suffit d'aller un peu plus loin pour trouver encore une nouvelle occurrence du facteur). Cette notion s'applique évidemment aux facteurs de Lyndon du mot et on peut se demander de quelle manière elle affecte la factorisation du mot (ou sa factorisation « standard »).

La voie de recherche prometteuse et déjà abordée est formée des premiers résultats sur les mots de Lyndon équilibrés. L'espoir ici est de trouver une analogie entre les mots de Lyndon équilibrés infinis et une famille de mots infinis qu'il reste à découvrir et qui jouerait le même rôle que les mots sturmiens par rapport aux mots de Christoffel. Le récent travail de Castelli, Mignosi et Restivo [CMR99] est, de ce point de vue, plus qu'intéressant. Les auteurs construisent une famille de mots finis sur trois lettres  $\{a, b, c\}$ , qu'ils appellent  $3-PEER$  et qui satisfont certaines propriétés généralisant la condition de Fine et Wilf qui est centrale dans la théorie des mots sturmiens (cf [Md94, Bd97]). Ils montrent aussi qu'une famille de mots infinis sur  $\{a, b, c\}$  introduites par Arnoux et Rauzy [AR91] ont un ensemble de facteurs qui coïncident avec les facteurs des mots  $3-PEER$ . Leur travail utilise des réécritures de mots qui sont calqués sur le calcul du plus grand commun diviseur de trois entiers et qui y jouent un rôle clé. C'est notre sentiment que ces analogies doivent être examinées de près.

Si on itère le calcul de triplets le long de chemins infinis dans l'arbre de génération des mots de Lyndon équilibrés, on obtient une suite infinie  $(u_n, v_n, w_n)_{n \geq 0}$  de triplets standard. La suite de mots  $(u_n)_{n \geq 0}$  apparaissant en première composante converge vers un mot infini bien défini  $u = \lim_{n \rightarrow \infty} u_n$ . Les mots obtenus des mots de Lyndon équilibrés infinis en supprimant leur première lettre semblent former une famille de mots infinis qu'il faudrait étudier. Cette idée s'inspire du cas des mots de Christoffel. En effet, on obtient un mot sturmien en supprimant la première lettre d'un mot de Christoffel infini (cf [Ber95, Prop. 4.4] pour une formulation plus précise). On peut aborder ces mots infinis au travers de leur factorisation de Lyndon. On peut montrer que les facteurs de Lyndon de ces mots infinis coïncident avec l'ensemble des mots de Lyndon équilibré. Nous avons déjà abordé le calcul de la factorisation de Lyndon de ces mots. Par exemple,

**Observation 31** Considérons le chemin infini périodique  $(RL)^\infty$  et la suite de triplets standard  $(u_n, v_n, w_n)_{n \geq 0}$  obtenu de ce chemin et soit  $s$  le mot infini obtenu du mot  $u = \lim_{n \rightarrow \infty} u_n$  en supprimant sa première lettre. La factorisation de Lyndon du mot  $s$  est  $s = \prod_{n > 0} \ell_n$  où les facteurs de Lyndon sont donnés par la récurrence  $\ell_{n+1} = \ell'_n \ell_{n-1}^2$ , et les conditions initiales  $\ell_0 = c$ ,  $\ell_1 = b$  et  $\ell_2 = acc$ . De plus, on a  $\ell_{n+1} = \varphi(\ell_n)$  où  $\varphi$  est le morphisme défini par  $a \mapsto ac$ ,  $b \mapsto acc$ ,  $c \mapsto b$ .

Ce qu'on retrouve dans ce calcul est, en quelque sorte, la factorisation standard du mot de Lyndon équilibré infini associé. La récurrence pour les facteurs de Lyndon  $\ell_n$  est aussi très proche de celle des facteurs de Lyndon du mot de Fibonacci (cf page 2.2.6). Aussi, on peut montrer qu'on retrouve ainsi les mots sturmiens sur deux lettres ( $\{a, c\}$  ou  $\{a, b\}$ ) (cf la preuve du lemme 20). Par exemple, on retrouve le mot de Fibonacci sur l'alphabet  $\{a, b\}$  en itérant le long du chemin infini  $(L^2R)^\infty$  (et en supprimant la première lettre du mot de Lyndon équilibré qu'on obtient). Le mot de Fibonacci sur l'alphabet  $\{a, b\}$

s'obtient en iterant plutôt le long du chemin  $L(L^2R)^\infty$ . Cependant, la fonction de complexité de cette famille de mots infinis n'est pas donnée par l'identité  $p(n) = 2n + 1$ , comme c'est le cas pour les mots de Arnoux et Rauzy [AR91] (voir aussi Castelli *et al.* [CMR99]). Les exemples que nous avons étudiés semblent toutefois indiquer que ces mots sont de complexité linéaire.

Ce qu'il manque encore à cette théorie est une interprétation géométrique des mots de Lyndon équilibrés en termes de « chemins » dans l'espace, comme elle existe pour les mots de Christoffel. Le calcul du « déterminant » d'un mot de Lyndon équilibré (ou d'un triplet  $(u, v, w)$ , cf Prop. 24) — comme le font Borel et Laubie [BL93] pour les mots de Christoffel — est certainement la clé d'une solution et les matrices introduites à la section 2.3.5 ont sans doute un rôle à jouer ici.



# Bibliographie

## Visualisation

- [AK95] Alpert, C.J. and Kahng, A.B. Recent developments in netlist partitioning : A survey. *Integration : the VLSI Journal*, 19(1-2) :1–81, 1995.
- [BDM99] Bousquet-Mélou, M., Dutour, I., and Melançon, G. The random acyclic directed graph. Technical Report INS-XX, CWI, Amsterdam, 1999.
- [BETT94] Battista, G.d., Eades, P., Tamassia, R., and Tollis, I.G. Algorithms for drawing graphs : an annotated bibliography. *Computational Geometry : Theory and Applications*, 4(5) :235–282, 1994.
- [BETT99] Battista, G.d., Eades, P., Tamassia, R., and Tollis, I.G. *Graph Drawing : Algorithms for the Visualisation of Graphs*. Prentice Hall, 1999.
- [BKW99] Brandes U., Kenis P., and Wagner D. Centrality measures in policy networks drawings. In *International Symposium on Graph Drawing*, Lecture Notes in Computer Science. Springer Verlag, 1999.
- [BMK95] Blythe, J., McGrah, C., and Krackhardt, D. The effect of graph layout on inference from social network data. In *Symposium on Graph Drawing, GD '95*, volume 1027 of *Lectures Notes in Computer Science*, pages 40–51. Springer Verlag, 1995.
- [BRS92] Botafogo, R.A., Rivlin, E., and Schneiderman, B. Structural analysis of hypertexts : Identifying hierarchies and useful metrics. *ACM Transactions on Information Systems*, 10(2) :142–180, 1992.
- [CMS99] Card, S.K., Mackinlay, J.D., and Shneiderman, B. *Readings in Information Visualization*. Morgan Kaufmann Publishers, 1999.
- [DC98] Dengler, E.A. and Cowan, W. Human perception of laid-out graphs. In *Symposium on Graph Drawing GD'98*, volume 1547 of *Lecture Notes in Computer Science*, pages 441–443. Springer Verlag, 1998.
- [DDG99] Delest M., Domenger J.-P., and Gastel L. A graph model of knowledge acquisition, 1999.
- [DF99] Delest, M. and Fédou, J.-M. Généralisation du nombre de Stahler aux arbres généraux, 1999. (Personal communication).
- [DGK98] Duncan, C.A., Goodrich, M.T., and Kobourov, S.G. Balanced aspect ratio trees and their use for drawing very large graphs. In

- Symposium on Graph Drawing GD '98*, Lecture Notes in Computer Science, pages 111–124. Springer-Verlag, 1998.
- [DK96] Devroye, L. and Kruszewski, P. The botanical beauty of binary trees. In Brandenburg, F.J., editor, *Symposium on Graph Drawing GD '95*, volume 1027 of *Lecture Notes in Computer Science*, pages 166–177. Springer-Verlag, 1996.
- [Drm97] Drmota, M. Systems of functional equations. *Journal of Random Structures and Algorithms*, 10(1-2) :103–124, 1997.
- [Ead92] Eades, P. Drawing free trees. *Bulletin of the Institute for Combinatorics and its Applications*, 5 :10–36, 1992.
- [EF96] Eades, P. and Feng, Q.-W. Multilevel visualization of clustered graphs. In North, S., editor, *4th International Symposium on Graph Drawing*, volume 1190 of *Lecture Notes in Computer Science*, pages 101–112. Springer Verlag, 1996.
- [EFL96] Eades, P., Feng, Q.-W., and Lin, X. Straight-line drawing of hierarchical graphs and clustered graphs. In North, S., editor, *4th International Symposium on Graph Drawing*, volume 1190 of *Lecture Notes in Computer Science*, pages 113–128. Springer Verlag, 1996.
- [ES90] Eades, P. and Sugiyama, K. How to draw a directed graph. *Journal of Information Processing*, 13(4) :424–437, 1990.
- [Eve74] Everitt, B. *Cluster Analysis*. Social Science Research Council (SSRC). Heinemann Educational Books Ltd., first edition edition, 1974.
- [EW94] Eades, P. and Whitesides, S.H. Drawing graphs in two layers. *Theoretical Computer Science*, 131(2) :361–374, 1994.
- [FRV79] Flajolet, P., Raoult, J.C., and Vuillemin, J. The number of registers required for evaluating arithmetic expressions. *Theoretical Computer Science*, 9 :99–125, 1979.
- [FS98] Frécon, E. and Smith, G. Webpath - a three-dimensional web history. In *IEEE Symposium on Information Visualization*. IEEE Computer Society, 1998.
- [Fur86] Furnas, G.W. Generalized fisheye views. In *Human Factors in Computing Systems CHI '86*, pages 16–23. ACM Press, 1986.
- [GJ83] Garey, M.R. and Johnson, D.S. Crossing number is np-complete. *SIAM Journal of Algebraic and Discrete Methods*, 4(3) :312–316, 1983.
- [HDM98] Herman, I., Delest, M., and Melançon, G. Tree visualisation and navigation clues for information visualisation. *Computer Graphics Forum*, 17(2) :153–165, 1998.
- [HDWB95] Hendley, R.J., Drew, N.S., Wood, A.M., and Beale, R. Case study : Narcissus : Visualising information. In *IEEE Symposium on Information Visualization*, pages 90–96. IEEE Computer Society Press, 1995.
- [He 99] He T. Internet-based front-end to network simulator. In Gröller, E., Löffelmann, H., and Ribarsky, W., editors, *Data Visualization '99*, pages 247–252. Springer-Verlag, 1999.

- [HMM99a] Herman, I., Marshall, S., and Melançon, G. State-of-the-art report : Graph visualisation and navigation in information visualisation. In *Eurographics '99*. Eurographics Association, 1999.
- [HMM<sup>+</sup>99b] Herman, I., Marshall, S.M., Melançon, G., Duke, D., Delest, M., and Domenger, J.-P. Skeletal images as visual cues in graph visualization, 1999.
- [HMRD99] Herman, I., Melançon, G., Ruiter, B.d., and Delest, M. Latour - a tree visualization system. In Kratochvil, J., editor, *Symposium on Graph Drawing GD'99*, Lecture Notes in Computer Science. Springer-Verlag, 1999.
- [Hor45] Horton, R.E. Eroded development of streams and their drainage basins, hydrophysocal approach to quantitative morphology. *Bulletin of the Geologic Society of America*, 56 :275–370, 1945.
- [HP97] Hege, H.-C. and Polthier, K. (eds.). *Visualization and mathematics : experiments, simulations and environments*. Springer, Berlin, 1997.
- [KLRZ94] Kimelman, D., Leban, B., Roth, T., and Zernik, D. Reduction of visual complexity in dynamic graphs. In Tamassia, R. and Tollis, I.G., editors, *2nd International Symposium on Graph Drawing*, Lecture Notes in Computer Science, pages 218–225. Springer, 1994.
- [KM86] Kreweras, G. and Moszkowski, P. A new enumerative property for the narayana numbers. *Journal of Stat. Plann. Inference*, 14 :63–67, 1986.
- [LLT69] Lawler, E.L., Levitt, K.N., and Turner, J. Module clustering to minimize delay in digital networks. *IEEE Transactions on Computers*, 18 :47–57, 1969.
- [LRP95] Lamping, J., Rao, R., and Pirolli, P. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies, 1995.
- [MELS95] Misue, K., Eades, P., Lai, W., and Sugiyama, K. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6 :183–210, 1995.
- [MGB<sup>+</sup>98] Mutzel, P., Gutwengwer, C., Brockenauer, R., Fialko, S., Klau, G., Kruger, M., Ziegler, T., Naher, S., Alberts, D., Ambras, D., Koch, G., Junger, M., Bucheim, C., and Leipert, S. A library of algorithms for graph drawing. In *Symposium on Graph Drawing GD'98*, volume 1547 of *Lectures Notes in Computer Science*, pages 456–457. Springer-Verlag, 1998.
- [MH98] Melançon, G. and Herman, I. Circular drawings of rooted trees. Technical Report INS-R9817, CWI, Amsterdam, 1998.
- [MHD98] Melançon, G., Herman, I., and Delest, M. Indices visuels et métriques combinatoires pour la visualisation de données hiérarchiques. In *IHM '99*, 1998.
- [Mir96] Mirkin, B. *Mathematical Classification and Clustering*. Kluwer Academic Publishers, 1996.

- [Mun97] Munzner, T. H3 : Laying out large directed graphs in 3d hyperbolic space. In *IEEE Symposium on Information Visualization (InfoVis '97)*, pages 2–10. IEEE CS Press, 1997.
- [Mun98] Munzner, T. Drawing large graphs with h3viewer and site manager. In *Symposium on Graph Drawing GD '98*, Lecture Notes in Computer Science, pages 384–393. Springer-Verlag, 1998.
- [Nor95] North, S. Incremental layout in dynadag. In North, S., editor, *Symposium on Graph Drawing GD '95*, pages 409–418. Springer-Verlag, 1995.
- [PCJ95] Purchase, H., Cohen, R.F., and James, M. Validating graph drawing aesthetics. In *Symposium Graph Drawing GD '95*, volume 1027 of *Lectures Notes in Computer Science*, pages 435–446. Springer-Verlag, 1995.
- [Pur98] Purchase, H.C. Which aesthetic has the greatest effect on human understanding? In *Symposium on Graph Drawing GD '97*, Lecture Notes in Computer Science, pages 248–261. Springer-Verlag, 1998.
- [RT81] Reingold, E.M. and Tilford, J.S. Tidier drawing of trees. *IEEE Transactions on Software Engineering*, 7(2) :223–228, 1981.
- [SB92] Sarkar, M. and Brown, M.H. Graphical fish-eye views of graphs. In *Human Factors in Computing Systems, CHI '92 Conference Proceedings*, pages 83–91. ACM Press, 1992.
- [Sch96] Schneiderman, B. The eyes have it : A task by data type taxonomy for information visualization. In *IEEE/CS Symposium on Visual Languages (VL '96)*, pages 336–343. IEEE CS Press, 1996.
- [Str52] Strahler, A.N. Hypsometric (area-altitude) analysis of erosional topology. *Bulletin of the Geologic Society of America*, 63 :1117–1142, 1952.
- [TanXX] Tanenbaum, A. *Computer Networks*. Prentice Hall, 19XX.
- [VC85] Viennot, X. and Chaumont, M.V.d. Enumeration of rna secondary structures by complexity. In *Mathematics in biology and medicine*, volume 57 of *Lecture Notes in Biomathematics*, pages 360–365, 1985.
- [VEJA89] Viennot, X.G., Eyrolles, G., Janey, N., and Arquès, D. Combinatorial analysis of ramified patterns and computer imagery of trees. *Computer Graphics (SIGGRAPH '89)*, 23 :31–40, 1989.
- [Wil97] Wills, G.J. Niche works - interactive visualization of very large graphs. In *Symposium on Graph Drawing GD '97*, volume 1353 of *Lectures Notes in Computer Science*, pages 403–414. Springer, 1997.

## Combinatoire des mots

- [AR91] Arnoux P. and Rauzy G. Représentation géométrique de suites de complexités  $2n + 1$ . *Bulletin de la Société Mathématique de France*, 119 :199–215, 1991.

- [Bd97] Berstel J. and de Luca A. Sturmian Words, Lyndon Words and Trees. *Theoretical Computer Science*, 178(1–2) :171–203, 1997.
- [Ber95] J. Berstel. Recent results on Sturmian words. *Developpements in Language Theory 95*, pages 1–12, 1995. World Scientific.
- [BL93] Borel J. P. and Laubie F. Quelques mots sur la droite projective réelle. *Journal de Théorie des Nombres de Bordeaux*, 5 :23–51, 1993.
- [BP94] Berstel J. and Pocchiola M. Average cost of duval’s algorithm for generating lyndon words. *Theoretical Computer Science*, 1–2 :415–25, 1994.
- [Bre81] Brentjes. *Multi-dimensional Continued Fractions*. Mathematisch Centrum, Amsterdam, 1981.
- [CFL58] Chen K. T., Fox R. H., and Lyndon R. C. Free Differential Calculus, IV—The Quotient Groups of the Lower Central Series. *Annals of Mathematics*, 68 :81–95, 1958.
- [CMR99] Castelli M. G., Mignosi F., and Restivo A. Fine and Wilf’s Theorem for Three Periods and a Generalization of Sturmian Words. *Theoretical Computer Science*, 218 :83–94, 1999.
- [DDKM94] Désarménien J., Duchamp G., Krob D., and Melançon G. Quelques remarques sur les super-algèbres de Lie libres. *Compte-rendus de l’académie scientifique de Paris*, I 318(5) :419–424, 1994.
- [de 97] de Luca A. Sturmian Words : Structure, Combinatorics, and Their Arithmetics. *Theoretical Computer Science*, 1 :45–82, 1997.
- [Duv83] Duval J. P. Factorizing Words over an Ordered Alphabet. *Journal of Algorithms*, 4 :363–381, 1983.
- [Duv88] Duval J. P. Génération d’une section des classes de conjugaison et arbre de mots de Lyndon de longueur bornée. *Theoretical Computer Science*, 60 :255–283, 1988.
- [GKP94] Graham R. L., Knuth D. E., and Patashnik O. *Concrete Mathematics*. Addison-Wesley, 1994.
- [HW38] Hardy G. H. and Wright E. M. *An introduction to the theory of numbers*. Oxford at Clarendon Press, 1938.
- [IM97] Ido A. and Melançon G. Lyndon Factorization of the Thue-Morse sequence and its relatives. *Discrete Mathematics and Theoretical Computer Science*, 1 :1–10, 1997.
- [Lau95] Laurier E. *Opérations sur les mots de Christoffel*. PhD thesis, Université de Limoges, 1995.
- [Lot83] Lothaire M. *Combinatorics on Words*. Addison Wesley, 1983.
- [Lotar] Lothaire M. *Algebraic Combinatorics on Words*. Cambridge University Press, to appear. <http://www.univ-mlv/~berstel/Lothaire/>.
- [Md94] MIGNOSI F. and de LUCA A. Some combinatorial properties of sturmian words. *Theoretical Computer Science*, 136 :361–385, 1994.
- [Mel92] Melançon G. Combinatorics of Hall Trees and Hall Words. *Journal of Combinatorial Theory Series A*, 59(2) :285–308, 1992.

- [Mel93] Melançon G. Constructions des bases standard des  $K\langle A \rangle$ -modules à droite. *Theoretical Computer Science*, 117(1–2), 1993.
- [Mel96a] Melançon G. Lyndon Factorization of Infinite Words. In Puech C. and Reischuk R., editors, *STACS '96, 13th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, 1046*, pages 147–154. Springer Verlag, 1996.
- [Mel96b] Melançon G. Viennot Factorizations of Infinite Words. *Information Processing Letters*, 60 :53–57, 1996.
- [Mel99] Melançon G. Lyndon Words and Singular Factors of Sturmian Words. *Theoretical Computer Science*, 218(1) :41–59, 1999.
- [Melar] Melançon G. Lyndon Factorization of Sturmian Words. *Discrete Mathematics*, to appear. Special Issue for FPSAC'96, 8th international Conference on Formal Power Series and Algebraic Combinatorics, Stanton D. and Leroux P., eds.
- [MKS66] Magnus W., Karass A., and Solitar D. *Combinatorial Group Theory*. John Wiley, 1966.
- [MR89] Melançon G. and Reutenauer C. Lyndon words, free algebras and shuffles. *Canadian Journal of Mathematics*, 41 :577–91, 1989.
- [MR96] Melançon G. and Reutenauer C. Chains of Partitions and Free Lie Superalgebras. *Journal of Algebraic Combinatorics*, 5(4) :337–351, 1996.
- [Rau84] Rauzy G. Automata in Infinite Words. In Nivat M. and Perrin D., editors, *Mots infinis en arithmétique*, pages 164–171. Springer Verlag, 1984. Lecture Notes in Computer Science, 192.
- [Reu86] Reutenauer C. Mots de Lyndon et un théorème de Shirshov. *Annales des Sciences Mathématiques du Québec*, 10(2) :237–245, 1986.
- [Reu93] Reutenauer C. *Free Lie Algebras*. London Mathematical Society Monographs 7. Oxford University Press, 1993.
- [Sch59] Schützenberger M. P. Sur une propriétés combinatoires des algèbres de Lie libres pouvant être utilisée dans un problème de Mathématiques Appliquées. *Séminaire Dubreuil-Pisot*, 1958–59.
- [Sch65] Schützenberger M. P. On a factorization of free monoids. *Proceedings of the American Mathematical Society*, 16 :21–24, 1965.
- [SMDS94] Siromoney R., Matthew L., Dare V. R., and Subramanian K. G. Infinite Lyndon Words. *Information Processing Letters*, 50 :101–104, 1994.
- [Vie76] Viennot X. *Algèbres de Lie libres et monoïdes libres*. Number 691 in Lecture Notes in Mathematics. Springer, 1976.
- [WW94] Wen Z.-X. and Wen Z.-Y. Some Properties of the Singular Words of the Fibonacci Word. *European Journal of Combinatorics*, 15 :587–598, 1994.

# Curriculum vitae

**Nom** : Melançon

**Prénom** : Guy

**Date de naissance** : 28 avril 1961

**Etat civil** : marié, deux enfants

**Adresse** : Fregat 67, 1113 Diemen, Pays-Bas. Téléphone : +31 20 600 76 97

**Adresse professionnelle** :

CWI (Centrum voor Wiskunde en Informatica)

Kruislaan 413, P.O. Box 94079, 1090 GB Amsterdam, Pays-Bas

Téléphone : +31 20 592 41 13 - Fax : +31 20 592 41 99

Courrier électronique : Guy.Melancon@cwil.nl

**Fonctions occupées**

- 1988-1991 : Etudiant de thèse, Université du Québec à Montréal, Montréal, Canada
- 1991-1993 : Boursier post-doctoral (CRSNG, Canada), LaBRI, Université Bordeaux I / Maître de conférences associé
- 1993-1998 : Maître de conférences, Université Bordeaux I
- 1998- : Chercheur scientifique (détaché), CWI, Amsterdam

**Diplômes**

- 1988 : DEA de Mathématiques, Dép. de Math. et Informatique, Univ. du Québec à Montréal
- 1991 : Doctorat de Mathématiques, Dép. de Math. et Informatique, Univ. du Québec à Montréal (Directeur C.Reutenauer). Thèse éditée par les Publications du LaCIM (Université du Québec à Montréal), No. 8, 1991.

**Activités scientifiques diverses**

- 1995-1998 : Co-responsable du Séminaire de l'école doctorale de l'U.F.R. de Mathématiques et Informatique.
- Co-organisateur du symposium "Data Visualization '00", mai 2000, CWI, Amsterdam (Pays-Bas).
- Membre du comité de programme des journées Montoises, mars 2000, Université Marne-la-Vallée.

# Publications

## Visualisation

### Périodiques d'audience internationale

1. Graph Drawing and Navigation Techniques in Information Visualisation, à paraître dans *IEEE Transactions in Computer Graphics and Visualization* — aussi accepté comme *State of the Art Report* à Eurographics '99 (avec I. Herman et M.S. Marshall).
2. Tree Visualization and Navigational Clues for Information Visualization, *Computer Graphics Forum*, 17(2) :153–166, 1998 (avec M. Delest and I. Herman).
3. CalCo : un logiciel pour la combinatoire, Bulletin des Sciences Mathématiques du Québec, pp.30–37, mai 1994 (avec M. Delest, J.M. Fédou et N. Rouillon).

### Conférences internationales

4. Latour – a Tree Visualization System, International Symposium on Graph Drawing '99, Prague, République Tchèque, Septembre 1999 (avec M. Delest, I. Herman et B. de Ruitter).
5. Skeletal Images as Visual Cues in Graphs Visualization. IEEE/EG VisSym99 symposium, Vienna, mai 1999 (avec I. Herman, M.S. Marshall, D.J. Duke, M. Delest, J.-P. Domenger).
6. CalCo, A Visual Tool for Combinatorial mathematics, VL '93 Proceedings, IEEE/CS Symposium on Visual Languages, août 1993, Bergen University, Norvège, (avec Delest M., Fédou J.M., Rouillon N.).

### Autres catégories

7. Indices visuels et métriques combinatoires pour la visualisation de données hiérarchiques, IHM '99, Montpellier, France, novembre 1999 (avec I. Herman and M. Delest).
8. Graphes orientés acycliques aléatoires, Journées Graphes, Bordeaux, septembre 1999 (avec I. Dutour et M. Bousquet-Mélou).
9. Circular Drawings of Rooted Trees. CWI report INS-9817, décembre 1998 (avec I. Herman).
10. CalCo, Computation and Image in Combinatorics, Human Interaction for Symbolic Computation, in Texts and Monographs in Symbolic Computation, Springer-Verlag, ed. N. Kajler, 1995 (avec les membres de l'équipe *CalCo*).

## Combinatoire des mots

### Périodiques d'audience internationale

11. Lyndon Factorization of Sturmian Words, accepted (april 1997) for publication in *Discrete Mathematics*, FPSAC '96 special issue, P Leroux and D Stanton eds.
12. Lyndon Words and Singular Factors of Sturmian Words, *Theoretical Computer Science*, Words Conference special issue, (218) :41-58, 1999.
13. Lyndon factorization of the Thue-Morse word and its relatives, *Discrete Mathematics and Theoretical Computer Science*, 1(1) :43-52, 1997 (avec A Ido).
14. Factorizing infinite words using Maple, *MapleTech Special Issue : Maple in the Mathematical Sciences*, 4(1) :1-9, Birkhäuser, Boston, 1997.
15. Viennot Factorizations of Infinite Words, *Information Processing Letters*, 60 :53-57, 1996.
16. Langages de Dyck généralisés et factorisations du monoïde libre, *Annales des Sciences Mathématiques du Québec*, 21(2) :103-122, 1997 (avec H Jacquet).
17. Free Lie Superalgebras, Trees and Chains of Partitions, *Journal of Algebraic Combinatorics*, 5(4) :337-352, 1996 (avec C Reutenauer).
18. Quelques remarques sur les super-algèbres de Lie libres, *Comptes Rendus de l'Académie Scientifique de Paris*, tome 318, série I :419-424, 1994 (avec J. Désarménien, G Duchamp et D Krob).
19. Construction des bases standard des  $K\langle A \rangle$ -modules à droite, *Theoretical Computer Science*, 117 :255-272, 1993.
20. Computing Hall Exponents in the Free Group, *International Journal of Algebra and Computation*, 3(3) :275-294, 1993 (avec C Reutenauer)
21. Une présentation combinatoire de la super algèbre de Lie libre, *Comptes Rendus de l'Académie Scientifique de Paris*, tome 315, série I :1215-1220, 1992, (avec C Reutenauer)
22. Combinatorics of Hall trees and Hall words, *Journal of Combinatorial Theory*, series 'A', 59(2) :285-308, 1992.
23. Lyndon words, free algebras and shuffles", *Canadian Journal of Mathematics*, 41 :577-91, 1989.

### Conférences internationales

24. Singular Factors and Lyndon Factorization of the Fibonacci Word, 10th international Conference on Formal Power Series and Algebraic Combinatorics (FPSAC), Toronto, Canada, 14-19 june 1998.
25. Lyndon Factorization of Sturmian Words, 8th international Conference on Formal Power Series and Algebraic Combinatorics (FPSAC), Minneapolis, USA, 25-29 june 1996.

26. Lyndon Factorization of Infinite Words, 13th Symposium on Theoretical Aspects of Computer Science (STACS), Grenoble, France, 22-24 february 1996. Free Lie superalgebras and some representations of the symmetric group, Jerusalem Combinatorics, Université de Jérusalem, Israel, mai 1993.

Autres catégories

27. Conjugation, Lyndon Words and an Application to Free Lie Algebras, Conférence Words, Rouen, septembre 1999 (avec P. Andary).
28. Rubriques pour Encyclopedia of Mathematics, vol11 et ss., R.Hoksbergen ed., Kluwer Academic Publishers, 1996
29. Langages de Dyck généralisés et factorisations du monoïde libre, Colloque GASCOM, Journées d'automne à Caen, december 1996 (avec H Jacquet).
30. Bases of free Lie algebras – “are Lyndon words better ?”, 3rd SIAM Conference on Control and its Applications, Symposium on Combinatorial Methods in Control Theory, Saint-Louis, Missouri, April 1995.
31. Réécritures de mots, algorithme de calculs dans  $K\langle A \rangle$  et idéaux à droite, Journées LanFor, Aussois, April 1994.
32. A combinatorial presentation of the free Lie superalgebra, Canadian Mathematical Society 1993 Winter Meeting, Montréal, December 1993.