



HAL
open science

Modèles de compression et critères de complexité pour la description et l'inférence de structure musicale

Corentin Guichaoua

► **To cite this version:**

Corentin Guichaoua. Modèles de compression et critères de complexité pour la description et l'inférence de structure musicale. Autre [cs.OH]. Université de Rennes, 2017. Français. NNT : 2017REN1S053 . tel-01687054

HAL Id: tel-01687054

<https://theses.hal.science/tel-01687054v1>

Submitted on 18 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

École doctorale MATISSE

présentée par

Corentin Guichaoua

préparée à l'unité de recherche 6074 (IRISA)

Institut de Recherche en Informatique et Systèmes Aléatoires
(Composante universitaire)

**Modèles de compression
et critères de complexité
pour la description et
l'inférence de structure
musicale**

**Thèse soutenue à Rennes
le 19 septembre 2017**

devant le jury composé de :

François PACHET

Directeur, Spotify Creator Technology Research Lab
/ président

Moreno ANDREATTA

DR2 CNRS, IRCAM / rapporteur

Myriam DESAINTE-CATHERINE

PR CE2, Bordeaux INP / rapporteur

Matthew DAVIES

Senior Researcher, INESC TEC / examinateur

Mathieu GIRAUD

CR CNRS, CRIStAL / examinateur

Frédéric BIMBOT

DR CNRS, IRISA / directeur de thèse

Remerciements

En premier lieu, je souhaite remercier Frédéric Bimbot pour tout ce qu'il a fait pour m'encadrer tout au long de cette thèse : pour m'avoir aidé à cristalliser des intérêts encore flous en un projet de thèse, pour m'avoir accompagné et guidé dans son exploration systématique, pour m'avoir accordé sa confiance sans cesse renouvelée, pour son soutien au travers des fluctuations de motivation (et sa patience pendant), pour son extrême disponibilité lors des périodes de rush (un grand merci également à Anna pour m'avoir laissé l'accaparer ces soirs et weekends). Par *contraste*, je ne sais pas si je dois le remercier ou lui reprocher que je ne puisse plus entendre la musique comme avant.

Je remercie les membres du jury, François Pachet, Matthew Davies, Mathieu Giraud et en particulier les rapporteurs, Myriam Desainte-Catherine et Moreno Andreatta, pour leur temps, l'attention qu'ils ont porté à mes recherches, pour leurs commentaires éclairés et la richesse des points de vue avec lesquels ils ont analysé mon travail.

Cette thèse n'aurait pas pu voir le jour si je n'avais pas rencontré Frédéric, aussi je remercie Nancy Bertin et Rémi Gribonval de m'avoir orienté vers lui à l'issue de mon stage de Master, et pour leurs encouragements répétés. Je remercie également tous les autres membres et visiteurs de l'équipe METISS/PANAMA pour les échanges scientifiques et leur compagnie agréable au cours de ces 5 années, que ce soit en C301, à la cafétéria, à la cantine de Supélec ou au bar. Ce sera toujours un plaisir de vous recroiser sur Rennes ou en conférence.

Toujours à l'Irisa, je tiens à remercier Aurélie Lemaître et Bertrand Couïasnon pour m'avoir accueilli lors de mon premier stage, qui m'a donné le goût de la recherche ; François Coste, qui m'a orienté vers les travaux de Matthias Gallé qui ont été essentiels à la première partie de cette thèse ; ainsi que Kadi Bouatouch pour ses conseils bienveillants en tous genres.

London I'd like to thank Mark Plumbley for inviting me to spend a few months at the Center for Digital Music in London, and Geraint Wiggins and Marcus Pearce for welcoming me in their Computer Creativity group. Many thanks also to all the other people at Queen Mary who helped me feel at home while abroad, especially Vincent Akkermans, Stephen McGregor, Mariano Mora-McGinity, Jordan Smith, Mathieu Barthet and most of all Ioana Dalca.

Merci aussi à tous ceux que j'ai rencontré au fil des activités qui m'ont permis d'équilibrer travail et loisirs, par le hockey (à l'Irisa et à l'extérieur), les jeux de société (à l'Irisa et à l'extérieur – encore), le rock, le lindy hop ou ailleurs. Merci en particulier à Alan, Alexandre B., Alexandre R., Alexia, Alexis, Anne-Lise, Antoine, Arthur, Blandine, Benoît, Carmen, Cédric, Clélia, Constance, Edith, Élise, Élodie, Éric, Émeline, Hortense, Jean-Baptiste, Jean-Michel,

Jessica, Jérémie, Joffrey, Johanne, Julien, Justine, Marine, Martin, Marwan, Mathieu L., Mathilde, Matthieu S., Morgane, Nicolas, Noémie, Ornella, Raphaëlle, Sergei, Timothée, Wilfried, Yoann et pardon à ceux que j'aurais oubliés dans cette liste.

Enfin, je souhaite remercier ma famille, à la fois celle que je connais depuis toujours, celle que j'ai découverte récemment, et Aline, Étienne, Gaël, Hugo et William qui n'en font pas partie mais presque. Merci surtout à ma mère, qui m'a accompagné tout au long de la thèse comme elle l'a si bien fait tout au long de ma vie, et m'a permis de rendre plus accessible ce document en se soumettant à l'épreuve des premières versions.

Résumé

Une définition très générale de la structure musicale consiste à considérer tout ce qui distingue la musique d'un bruit aléatoire comme faisant partie de sa structure. Dans cette thèse, nous nous intéressons à l'aspect macroscopique de cette structure, en particulier la décomposition de passages musicaux en unités autonomes (typiquement, des sections) et à leur caractérisation en termes de groupements d'entités élémentaires conjointement compressibles.

Un postulat de ce travail est d'établir un lien entre l'inférence de structure musicale et les concepts de complexité et d'entropie issus de la théorie de l'information. Nous travaillons ainsi à partir de l'hypothèse que les segments structurels peuvent être inférés par des schémas de compression de données.

Dans une première partie, nous considérons les grammaires à dérivation unique (GDU), conçues à l'origine pour la découverte de structures répétitives dans les séquences biologiques [GALLÉ, 2011], dont nous explorons l'utilisation pour modéliser les séquences musicales. Cette approche permet de compresser les séquences en s'appuyant sur leurs statistiques d'apparition, leur organisation hiérarchique étant modélisée sous forme arborescente. Nous développons plusieurs adaptations de cette méthode pour modéliser des répétitions inexactes et nous présentons l'étude de plusieurs critères visant à régulariser les solutions obtenues.

La seconde partie de cette thèse développe et explore une approche novatrice d'inférence de structure musicale basée sur l'optimisation d'un critère de compression tensorielle. Celui-ci vise à compresser l'information musicale sur plusieurs échelles simultanément en exploitant les relations de similarité, les progressions logiques et les systèmes d'analogie présents dans les segments musicaux. La méthode proposée est introduite d'un point de vue formel, puis présentée comme un schéma de compression s'appuyant sur une extension multi-échelle du modèle Système & Contraste [BIMBOT, 2016] à des patrons tensoriels hypercubiques. Nous généralisons de surcroît l'approche à d'autres patrons tensoriels, irréguliers, afin de rendre compte de la grande variété d'organisations structurelles des segments musicaux.

Les méthodes étudiées dans cette thèse sont expérimentées sur une tâche de segmentation structurelle de données symboliques correspondant à des séquences d'accords issues de morceaux de musique pop (RWC-Pop). Les méthodes sont évaluées et comparées sur plusieurs types de séquences d'accords, et les résultats établissent l'attractivité des approches par critère de complexité pour l'analyse de structure et la recherche d'informations musicales, les meilleures variantes fournissant des performances de l'ordre de 70% de F-mesure.

En clôture de ce travail, nous en récapitulons les principales contributions et nous discutons d'extensions possibles des paradigmes étudiés, que ce soit par leur application à d'autres dimensions musicales, par l'adjonction de connaissances musicologiques, ou par leur utilisation sur des données audio.

Table des matières

1	Présentation	9
1.1	Recherche d'information musicale	9
1.1.1	Recherche d'information appliquée à la musique	10
1.1.2	Traitement automatique de la musique	11
1.2	Structure musicale	12
1.2.1	Enjeux	12
1.2.2	Multiplicité des définitions	13
1.3	Périmètre de cette thèse	17
1.3.1	Hypothèses générales	17
1.3.2	Représentation musicale considérée	17
1.3.3	Objectifs et contributions	18
2	Formalisation de l'approche présentée	21
2.1	Structure sémiotique	21
2.2	Modélisation arborescente	24
2.3	Structure musicale et théorie de l'information	26
2.3.1	Complexité de Kolmogorov	26
2.3.2	Entropie de Shannon	27
2.3.3	Critères classiques exprimés en termes d'information	28
2.3.4	Résumé de l'approche considérée	28
3	Inférence de structure musicale par plus courte grammaire	29
3.1	Modèles à base de grammaires formelles	29
3.1.1	Séquences	29
3.1.2	Grammaires formelles	30
3.1.3	Grammaires hors-contexte	31
3.1.4	Grammaires à Dérivation Unique (GDU)	32
3.2	Minimisation de la longueur de la grammaire	33
3.2.1	Remplacement itératif des répétitions (algorithme IRR)	34
3.2.2	Optimisation des règles (algorithme MGP)	35
3.2.3	Élagage de règle (algorithme CLEANUP)	38
3.2.4	Combinaison des algorithmes	39
3.3	Mise en pratique	40
3.3.1	Ensembles de données	40
3.3.2	Sélection de la règle pour IRR	42
3.3.3	Combinaison des algorithmes	45
3.3.4	Application à la découverte de structure dans du texte	48
3.3.5	Estimation de la structure linéaire	52
3.3.6	Comparaison de structure hiérarchique	56

4	Grammaires à éditions et critères de complexité	61
4.1	Extension à des grammaires à édition	61
4.1.1	Variations au sein d'une classe de segments	61
4.1.2	Formalisation des grammaires à éditions à dérivation unique	63
4.1.3	Adaptation des algorithmes	64
4.2	Critères de complexité	69
4.2.1	Définition de critères avancés	70
4.2.2	Adaptation des algorithmes	70
4.3	Expériences	71
4.3.1	Impact de l'ajout des éditions	72
4.3.2	Utilisation des critères avancés	77
5	Principes de compression tensorielle	81
5.1	Modèle Système & Contraste	82
5.2	Patron tensoriel	83
5.3	Graphe de redondance	84
5.4	Types et déformations de patrons tensoriels	84
5.5	Critère de complexité	86
5.5.1	Schéma général	88
5.5.2	Encodage par réplication	90
5.5.3	Adjonction d'une contrainte d'antériorité	92
5.6	Inférence de la structure interne d'un segment	93
5.7	Inférence de la structure sémiotique d'un morceau	95
6	Compression tensorielle par systèmes d'analogie dans les segments musicaux	99
6.1	Analogies sur des patrons hypercubiques	99
6.1.1	Propriétés et relations	99
6.1.2	Cas du système carré	100
6.1.3	Patron cubique	103
6.1.4	Cas tesseractique	110
6.1.5	Cas général du système hypercubique	119
6.2	Comparaison de paires d'accords	119
6.3	Complexité d'un graphe de redondance	121
6.3.1	Gestion des déformations	122
6.3.2	Algorithme	122
6.4	Évaluation des méthodes à encodage tensoriel	124
6.4.1	Influence du modèle de relations	124
6.4.2	Sensibilité aux hyper-paramètres	125
6.4.3	Seuil de tolérance et base d'annotation	127
6.4.4	Exemples de segmentation	128
7	Conclusions et perspectives	135

Chapitre 1

Présentation

Les avancées technologiques des dernières décennies sont la source de multiples bouleversements dans le paysage de la musique. Les progrès de l'industrie sur le matériel et les méthodes d'enregistrement et de post-traitement ont permis une démocratisation de la production musicale, de telle façon qu'aujourd'hui, le nombre d'artistes ayant les moyens de publier leur musique est plus élevé que jamais.

Parallèlement, le développement d'Internet et des formats de stockage numériques a grandement facilité la diffusion de la musique. En particulier, l'apparition des plateformes de streaming (telles que Spotify ou Deezer) qui proposent un accès gratuit à leur contenu a permis de diminuer considérablement la barrière financière à la découverte de nouveaux morceaux.

Combinées les unes aux autres, ces évolutions ont mené à une augmentation spectaculaire du nombre de titres accessibles. Au lieu de catalogues de l'ordre de quelques milliers de titres chez un disquaire local, le public a maintenant accès à plusieurs dizaines de millions de titres sur une multitude de plateformes globales comme iTunes, Amazon et d'autres à venir.

Face à ces quantités de données dont l'écoute dépasserait largement ce dont il est possible de faire l'expérience en une vie humaine, la difficulté principale n'est plus la disponibilité des données, mais la pertinence des données accédées.

1.1 Recherche d'information musicale

La recherche d'information et le traitement automatisé de la musique traitent des problèmes qui peuvent apparaître comme deux branches distinctes répondant à des problèmes différents. La première considère les données musicales de façon abstraite, se basant principalement sur des métadonnées, tandis que l'autre reste proche du signal musical, le caractérisant précisément ou le transformant. Toutefois, avec l'apparition de tâches plus complexes, un besoin commun se dégage pour des représentations d'un niveau intermédiaire, capturant une conceptualisation, même partielle, de la musique.

Parmi les concepts se prêtant à de telles représentations intermédiaires, la structure musicale joue un rôle important, et c'est autour de celle-ci que gravitent les problématiques de cette thèse.

1.1.1 Recherche d'information appliquée à la musique

L'abondance de documents n'est pas un problème spécifique à la musique. D'autres domaines, en particulier la gestion de bibliothèques et, plus récemment, la recherche de pages web, ont constitué une force motrice derrière le développement de méthodes permettant de trouver des documents selon divers types de requêtes, menant ainsi à l'apparition du domaine de la recherche d'information. L'application de méthodes issues de la recherche d'information au domaine de la musique a permis d'apporter des solutions à de nombreux cas d'utilisation.

L'approche la plus répandue repose sur une indexation selon des données définies au moment de la publication du morceau, telles que le titre du morceau, son compositeur, ses interprètes et son année de parution. Ces méthodes sont très efficaces dès lors que l'information sur les morceaux recherchés est connue précisément, mais il s'agit également de leur principale limitation : elles ne permettent de trouver que des titres dont certaines caractéristiques sont connues précisément, ce qui exclut un grand nombre de cas d'utilisations.

Une autre limitation de ce type d'approche est que leur utilisation est limitée aux données déjà renseignées, qui peuvent varier selon les sources. Cette limitation est particulièrement présente dans le cas des métadonnées déclaratives, telles que le titre ou l'auteur d'un morceau ou les lieux et dates d'enregistrement, qui à l'inverse des données descriptives ne peuvent pas être déduites de l'écoute du morceau¹.

À l'échelle de millions de morceaux, l'annotation d'une nouvelle catégorie de métadonnées est une tâche colossale qui requiert de nombreux annotateurs disposant des connaissances contextuelles ou capables de décrire les morceaux. Plusieurs initiatives de collecte collaborative de données, telles que MusicBrainz ou Last.fm, ont vu le jour afin de relever le défi. En une quinzaine d'années, ces bases de données ont accumulé une grande quantité d'informations, recensant plus de 16 millions d'enregistrements dans le cas de MusicBrainz [MUSICBRAINZ, 2017]. Malgré leur ampleur, ces initiatives peinent toutefois à suivre la masse de données publiées, en particulier pour les artistes moins populaires.

Outre les métadonnées de nature déclarative ou descriptive, de plus en plus de plateformes collectent des données relatives au contexte d'écoute des morceaux. Ces métadonnées dites sociales permettent d'établir des profils d'écoute pour les utilisateurs, les morceaux ou les artistes, qui peuvent ensuite servir de base pour des recommandations ou des comparaisons. Ces métadonnées présentent l'avantage d'être collectées de façon automatique, mais par leur nature tendent à être concentrées sur les morceaux les plus populaires, laissant les morceaux inconnus dans l'ombre.

Comme en atteste le succès des plateformes de distribution en ligne, qui sont basées sur ce type d'approches, la recherche d'information a permis de grandes avancées dans l'accessibilité et la découvrabilité de la musique. Actuellement, une des principales difficultés techniques est l'existence et la disponibilité des bases de données permettant l'indexation de la multitude des morceaux de musique publiés. Le perfectionnement des méthodes d'extraction automatique de métadonnées descriptives est en passe d'apporter une solution sur ce plan.

1. S'il est parfois possible de dater approximativement un morceau, d'y reconnaître le style d'un artiste ou d'en inférer le titre, ces informations sont d'une fiabilité faible.

Parallèlement, une autre limitation importante est que l'échelle de description utilisée dans ces approches est généralement le morceau entier. Si d'un point de vue applicatif, il s'agit de l'échelle appropriée dans une majorité de cas, du point de vue descriptif elle est souvent insuffisante car elle ne permet pas de rendre compte des changements au cours du déroulement du morceau. Là encore, les interactions avec les méthodes de traitement automatique de la musique apportent d'éventuelles solutions, à travers une utilisation, directe ou indirecte, du signal musical.

De nombreux travaux récents se focalisent sur des approches axées sur le signal musical pour des problèmes de recherche d'information. La définition et la formalisation de modèles de niveau intermédiaire sont alors très utiles afin de capturer l'essentiel de l'information contenue dans le signal tout en bénéficiant d'un niveau d'abstraction élevé.

1.1.2 Traitement automatique de la musique

Aujourd'hui, on cherche de plus en plus à appliquer des traitements automatiques à la musique. Qu'il s'agisse de producteurs cherchant à automatiser certains aspects de la production musicale, d'un simple individu voulant connaître le titre d'une chanson en l'absence d'une personne suffisamment renseignée ou encore d'un musicien désirant bénéficier d'un accompagnement pour répéter seul, les technologies relatives à la musique sont de plus en plus présentes.

Ces traitements automatiques de la musique, ainsi que les méthodes de recherche d'information spécifiques à la musique, à travers des rapprochements entre ces deux branches, sont regroupés dans le domaine de la *Music Information Retrieval*² (MIR).

Modélisation sur données symboliques

Une distinction importante qui peut être opérée parmi les approches à base de données musicales concerne le type des données traitées. On trouve d'une part des données audio, basées directement sur l'enregistrement de la musique, et d'autre part des données symboliques, qui sont issues d'un premier processus d'abstraction. La nature précise de cette abstraction peut prendre différentes formes, suivant de façon générale les conventions occidentales d'annotation de la musique et se focalisant sur une facette précise de la musique, comme le rythme ou l'harmonie.

Si la plupart des applications finales sont principalement basées sur des données d'entrée sous forme audio afin de pouvoir proposer une expérience complète, le traitement de données symboliques demeure une part critique du domaine. En effet, l'utilisation de données symboliques permet de séparer les tâches applicatives en de multiples sous-tâches, et ainsi d'améliorer individuellement chacune de ces tâches.

Ce découpage rend possible l'élaboration de méthodes d'un haut niveau d'abstraction, qui réutilisent des données issues d'autres algorithmes, sans avoir besoin d'attendre que ces algorithmes atteignent des niveaux de performances élevés. Le niveau d'abstraction fourni par les données symboliques permet d'isoler les difficultés spécifiques aux problèmes de haut niveau et de les surmonter individuellement.

2. Le terme se traduit littéralement par recherche d'information musicale; cependant, son sens a rapidement dérivé avec l'inclusion dans le domaine de tâches et de méthodes déconnectées de la recherche d'information et il n'existe pas d'équivalent francophone faisant l'objet d'un consensus.

À terme, l'objectif reste généralement une intégration au sein d'une chaîne de traitement complète. Dans les cas les plus avancés, des approches traitant les différents problèmes en interaction (plutôt que successivement) peut amener les solutions les plus performantes, comme cela a été le cas dans le domaine de la reconnaissance automatique de la parole avec les modèles de langue.

1.2 Structure musicale

La musique en général est un type de signal très riche, présentant de nombreuses facettes. Un des éléments principaux qui nous permet de distinguer la musique du bruit est le caractère fortement structuré de la musique. Cette structure se retrouve à toutes les échelles du signal musical, depuis les relations entre les fréquences sonores au sein d'une note de piano jusqu'à l'organisation en mouvements d'une symphonie. Lorsque l'on parle de structure musicale, il est généralement question de la structure macroscopique de la musique, qui peut être vue comme l'organisation des relations entre les divers éléments musicaux, tels que les notes, les phrases, ou les sections.

1.2.1 Enjeux

La connaissance de la structure d'un morceau, qui peut prendre de nombreuses formes, ouvre la voie à de nombreuses possibilités pour des applications qui s'appuient sur cette information. Les paragraphes suivants décrivent une poignée d'exemples de telles applications, mais de nombreux autres types d'applications existent [LIUTKUS et al., 2012; McVICAR et ELLIS, 2014; GROSCHE et al., 2012].

Création de vignettes musicales L'utilisation de courts extraits musicaux permet à un utilisateur d'avoir un aperçu rapide d'un morceau, sans pour autant l'écouter intégralement [LOGAN et CHU, 2000]. De telles vignettes musicales peuvent être utiles dans de nombreux contextes, présentant un équivalent musical au feuilletage d'un livre ou d'une revue. Ces extraits, afin d'être représentatifs, doivent souvent être définis manuellement. Des approches automatiques s'appuyant sur la structure musicale peuvent produire des résultats allant dans le même sens, utilisant par exemple les répétitions afin d'identifier les passages les plus marquants d'un morceau [AUCOUTURIER et SANDLER, 2002].

Guidage de remixes Une pratique de plus en plus courante, notamment dans des milieux amateurs, est la création de versions dérivées d'un ou plusieurs morceaux. L'utilisation d'interfaces de visualisation et d'édition faisant ressortir clairement la structure des morceaux originaux peut ainsi simplifier la tâche [DAVIES et al., 2013], notamment pour la création de remixes³, de mash-ups⁴ ou de medleys⁵. Il est également possible d'utiliser cette information afin de réaliser la transformation de façon entièrement automatique, permettant ainsi la création de versions dérivées potentiellement infinies, à l'exemple de l'Infinite Jukebox [LAMERE, 2012].

Réparation de passages manquants En traitement du signal audio, une tâche récurrente est la réparation d'enregistrements endommagés, où certaines parties du signal sont manquantes ou

3. Un remix est une version souvent radicalement différente d'un morceau, réalisée à partir de l'enregistrement original via des techniques d'édition audio.

4. Un mash-up est un morceau dérivatif réalisé par la fusion par superposition de pistes provenant de plusieurs autres morceaux.

5. Un medley, ou pot-pourri, est une juxtaposition de sections musicales issues de différents morceaux, généralement avec l'objectif d'effectuer des enchaînements fluides.

corrompues [ADLER et al., 2012]. Si les méthodes classiques fonctionnent bien lorsque les données détériorées sont peu groupées, la réparation de blocs macroscopiques (de l'ordre d'une demi-seconde ou plus) nécessite des méthodes spécifiques. En s'appuyant sur la structure musicale, il est possible de rechercher un passage homologue du morceau qui peut alors être copié et inséré pour masquer le manque de données fiables [MARTIN, 2012].

Génération automatique de musique Afin de pouvoir générer des morceaux imitant de façon crédible une composition ou une improvisation humaine, il est essentiel que les systèmes de génération automatique de musique (par exemple Continuator [PACHET, 2003] ou Antescofo [CONT, 2008]) intègrent des modèles permettant à la musique générée d'avoir une structure proche de celle générée par des humains.

À travers ces quelques exemples d'applications pratiques de la structure musicale se dégage un rôle de compréhension de l'articulation des morceaux de musique. Pour les vignettes, il s'agit de comprendre quelles sections jouent un rôle principal; pour les remixes, il s'agit de comprendre quels sont les points de jonction permettant de passer d'une section à une autre sans raccord flagrant; pour les réparations, il s'agit de comprendre comment sélectionner et altérer un passage de remplacement pour le greffer à la place des données manquantes. Cette forme de compréhension de la musique peut également permettre de compléter l'audio pour l'estimation des données intermédiaires comme la position temporelle des temps [DANNENBERG, 2005] ou la transcription en accords [MAUCH, NOLAND et DIXON, 2009]. On a alors une synergie entre plusieurs types de données, où les estimations des données intermédiaires informent l'estimation de la structure et réciproquement, qui se prête à une estimation jointe [PAUWELS et PEETERS, 2013], à l'instar de ce qui a été réalisé avec les modèles de langue en reconnaissance de la parole.

Dans cette thèse, nous nous focalisons sur des définitions et des considérations algorithmiques, mais l'examen des forces et faiblesses de ces approches peuvent présenter un intérêt pour l'analyse musicologique [DERUTY, BIMBOT et VAN WYMEERSCH, 2013] ou la psychologie de la musique.

1.2.2 Multiplicité des définitions

Si l'existence et l'intérêt de la structure musicale sont communément admis, il est en revanche plus difficile de trouver des approches explicitant de façon univoque ce concept [SMITH, 2014].

En musicologie, les analyses structurelles des morceaux sont en général réalisées avec un grain de détail très fin, en faisant appel à différents paradigmes de description structurelle. Ces différents paradigmes mènent à des descriptions qui sont complémentaires, mais souvent difficiles à comparer. Dans cette thèse, nous ne traiterons pas directement des approches de description de la structure issues de la musicologie.

Le projet SALAMI s'est intéressé à la production d'une grande base de données d'annotations structurelles. En raison de la souplesse des instructions d'annotations⁶, ces annotations permettent d'avoir un aperçu de la conception intuitive des annotateurs de la structure musicale. Smith et collègues [SMITH et al., 2011] rapportent un taux d'accord entre les annotations variant entre 70% et 90% selon la méthodologie de mesure pour de échelles d'annotation comparables, ce qui illustre à nouveau la variété des définitions de la structure musicale, cette fois au niveau

6. Les annotateurs sont tenus de respecter un format pré-déterminé, mais ils sont libres de choisir leurs propres critères pour déterminer le résultat de l'annotation.

des représentations intuitives. Ce constat s’inscrit dans le contexte de travaux récents [BRUDERER, MCKINNEY et KOHLRAUSCH, 2006 ; WIERING et al., 2009 ; PEETERS et DERUTY, 2009], et notamment des travaux effectués dans le cadre du projet QUAERO [BIMBOT, LE BLOUCH et al., 2010 ; BIMBOT et al., 2011 ; BIMBOT et al., 2012], qui visent à définir en des termes reproductibles les éléments qui président à la perception de la structure musicale.

En MIR, les travaux sur la structure musicale tendent à se focaliser sur les aspects applicatifs. Cela peut permettre de se dispenser de l’explicitation de la définition de la structure musicale en se reposant sur une perception intuitive, mais ne s’affranchit pas pour autant de la multiplicité de ces définitions. La définition implicitement considérée de la structure musicale est alors fortement conditionnée par le type d’application et la méthode utilisée pour parvenir au résultat final.

On peut distinguer les modes de description formelle de la structure musicale selon deux axes principaux : le formalisme utilisé pour la représentation de la structure, et les critères qui guident le processus d’inférence de la structure musicale, qu’elle soit effectuée par un humain ou une machine.

À ces distinctions principales vient s’ajouter la considération des strates musicales prises en compte. Ainsi, deux approches, dont l’une prend seulement en compte l’harmonie d’un morceau et l’autre considère exclusivement l’instrumentation et le rythme, aboutiront probablement à deux représentations différentes de la structure musicale. Les implications précises du choix des strates musicales relèvent de la musicologie et ne seront pas traitées en détail dans cette thèse. L’impact des choix spécifiques relatifs aux strates musicales effectués dans les travaux présentés sera toutefois discuté en section 1.3.2.

1.2.2.1 Types de représentation

Le processus principal régissant la plupart des approches d’analyse structurelle musicale est le regroupement d’éléments de bas niveau en entités de plus haut niveau permettant une description plus globale du morceau. On peut distinguer plusieurs familles selon le type de regroupements qui sont considérés dans l’approche.

Structure linéaire/séquentielle La représentation la plus simple de la structure musicale, et la plus utilisée en MIR, est celle d’une suite de segments. Dans ce type d’approche, les regroupements s’effectuent en général à une échelle unique et aboutissent à une succession de segments sans recouvrement ni interruption de grande ampleur. Cette segmentation est souvent accompagnée d’étiquettes pour les segments, visant à indiquer une ressemblance entre certains segments ou un rôle particulier dans l’organisation du morceau. À travers sa simplicité et son essor dans la communauté MIR, ce type de représentation bénéficie de l’existence d’un grand nombre d’outils et de méthodes permettant la création de bases de données, la comparaison de différentes annotations et l’évaluation de méthodes d’inférence. En particulier, la simplicité des représentations séquentielles rend possible et dans une certaine mesure pertinente la comparaison entre des approches basées sur des critères différents.

Toutefois, cette simplicité restreint dramatiquement la capacité à représenter l’ensemble de la structure musicale. Il est ainsi impossible dans ce formalisme d’indiquer qu’un segment se compose de plusieurs sous-segments, ou à l’inverse que plusieurs segments forment un développement à long terme. Si les similitudes entre segments peuvent être partiellement représentées, l’annotation de relations plus élaborées n’est pas possible.

Structure hiérarchique/arborescente À un niveau de description plus détaillé, on trouve des approches qui considèrent la structure musicale comme étant de nature hiérarchique. Selon ce paradigme, les éléments de base peuvent être regroupés en entités de plus en plus haut niveau. On aboutit ainsi à des structures dans lesquelles des éléments sont imbriqués les uns dans les autres, qui se prêtent bien à une modélisation sous forme arborescente. On trouve dans cette catégorie la théorie de [LERDAHL et JACKENDOFF, 1985], et l'analyse Schenkerienne [SCHENKER, 1935] peut également y être incluse [MARSDEN, 2010].

À l'heure actuelle, l'utilisation de cette catégorie de représentation est encore limitée en MIR par le manque de méthodes universelles de comparaison et la faible disponibilité de bases d'annotations pouvant servir de points de comparaison. Cependant, le compromis apporté entre la maniabilité des modèles pour des méthodes automatisées et le niveau de détail des représentations en explique l'attention grandissante qui leur est attribuée [BARBAR, DESAINTE-CATHERINE et MINIUSSI, 1993; BEURIVÉ et DESAINTE-CATHERINE, 2001; MARTIN, 2012; DE HAAS, 2012; MCFEE et ELLIS, 2014].

Structure relationnelle / associationnelle À l'opposé des représentations séquentielles, on trouve les représentations relationnelles. Pour ce type de structure, les éléments ne sont pas strictement regroupés en entités de niveau supérieur, et la description se focalise sur les relations qui les lient. Ces relations peuvent être de multiples types : contiguïté, similarité, inclusion, "sémantique", etc. On peut placer dans cette catégorie le modèle d'Implication-Réalisation décrit par Narmour [NARMOUR, 1990; NARMOUR, 1992].

Afin de maintenir un niveau de représentation qui puisse rester compréhensible, l'intégralité des relations ne peut pas être considéré, et les graphes obtenus ne sont alors comparables qu'entre des méthodes d'annotations très similaires. Ce type de modélisation permet un degré de description très fin, mais sa complexité limite sa popularité en MIR.

1.2.2.2 Critères d'organisation

La distinction selon les critères de détermination de la structure musicale est moins évidente, car ces critères sont souvent implicites, dictés par la méthodologie de l'approche utilisée. Dans leur état de l'art des méthodes de segmentation audio, Paulus et collègues [PAULUS, MÜLLER et KLAPURI, 2010] caractérisent les approches selon 3 critères : la répétition, l'homogénéité et la nouveauté. Ces critères, qui s'appliquent également sur des données symboliques, ont en commun d'être des critères de type *intra-opus*, c'est à dire qu'ils ne font appel qu'à des éléments faisant partie du morceau étudié. On retrouve également des approches utilisant des critères *inter-opus*, basés sur des connaissances expertes issues de la musicologie et donc bâties à partir de l'examen de nombreux autres morceaux. À ces approches, nous ajouterons l'approche par complexité, que nous développerons dans cette thèse et nous précisons comment elle peut être formulée pour exprimer les autres critères.

Répétition La répétition est un élément clé dans la musique, particulièrement dans la musique populaire. L'essentiel des méthodes utilisant un critère de répétition fait l'hypothèse que les éléments musicaux qui sont systématiquement répétés ensemble sont constitutifs d'un élément structurel correspondant à l'ensemble de la portion répétée.

En pratique, cette hypothèse tend à être vérifiée lors de comparaisons avec des formes intuitives de la structure musicale, mais certains écarts systématiques peuvent être observés. Par exemple,

lors de longues sections répétées, le critère de répétition ne permet pas, en l'absence d'information supplémentaire, de distinguer des sous-sections. Il en résulte alors un problème d'ambiguïté de "phase", où il devient impossible, sur la seule base du critère de répétition, de déterminer précisément les instants de début et de fin des segments structurels à une échelle donnée.

Homogénéité Les approches basées sur l'homogénéité cherchent à regrouper ensemble les éléments contigus dont les propriétés sont similaires. Les notions de similarité utilisées dans ce type d'approche s'appuient généralement sur des propriétés telles que le timbre ou la dynamique pour lesquelles il est possible d'adapter des méthodes de clustering en intégrant une contrainte de continuité temporelle.

Une divergence fréquente des méthodes à base d'homogénéité avec les structures intuitées est l'agglutination de plusieurs sections similaires contigües. À l'inverse, il est fréquent que la fin d'une section contraste avec le reste de la section, et une sensibilité trop élevée peut mener à les considérer comme des éléments structurels distincts.

Nouveauté Les critères de nouveauté présentent beaucoup de similitudes avec les critères d'homogénéité. Là où les critères d'homogénéité visent à obtenir des éléments qui ont des propriétés stables au cours de leur durée, les approches basées sur la nouveauté cherchent à identifier les instants où se produisent des changements rapides dans les propriétés étudiées pour marquer les séparations entre les éléments.

Si les méthodes d'inférence appliquant les deux principes sont assez différentes d'un point de vue pratique, la dualité entre homogénéité et nouveauté fait que les définitions implicites de la structure musicale qui en résultent sont très similaires.

Connaissances expertes Un autre type d'approche consiste à rechercher un certain nombre de motifs prédéterminés par des études musicologiques, censés correspondre à des séquences d'événements typiques de segments structurels. La structure ainsi obtenue doit respecter des règles musicologiques, telles que la présence d'une cadence en fin de section, et la définition exacte de la structure dépend des règles considérées, qui peuvent varier en pertinence selon les genres musicaux. À la différence des autres approches qui sont fréquemment combinées entre elles, les méthodes basées sur des connaissances expertes intègrent rarement des critères d'autres types.

Complexité Le type d'approche que nous étudions dans cette thèse est basé sur l'obtention d'une structure permettant de décrire simplement le morceau étudié, en accord avec la définition proposée par Bent [BENT et DRABKIN, 1987]. La principale difficulté de ce type d'approche réside dans le passage de la définition intuitive de la complexité d'une structure à une définition formelle et calculable.

La théorie de l'information permet d'apporter un certain nombre de concepts pouvant être utilisés afin d'aboutir à la définition de critères optimisables. Comme nous le détaillons dans le prochain chapitre après avoir introduit quelques bases de théorie de l'information, cette approche peut être utilisée pour exprimer des formulations modifiées des critères de répétition, d'homogénéité et de nouveauté, et sous certaines hypothèses intégrer un certain nombre de règles expertes. Pour une étude sous un angle musicologique de ce type de définition de la structure, on peut se référer à [LÉVY, 2004].

1.3 Périmètre de cette thèse

Au regard d'un besoin général de méthodes traitant la musique directement par son contenu plutôt que par des descripteurs externes, la structure musicale, joue un rôle critique dans la compréhension et l'explication des morceaux de musique. Sa description apparaît comme une étape essentielle dans le développement de nouvelles générations de méthodes de traitement automatisé reposant sur la capacité à appréhender le message musical à des échelles intermédiaires et globales.

Afin d'avancer vers ce but, la définition de modèles permettant simultanément une description détaillée et un niveau d'abstraction élevé ouvre la voie à un rapprochement entre les méthodes automatiques et les descriptions structurelles des musicologues, dont le niveau de précision dépasse parfois ce qui peut être traité efficacement par des machines à l'heure actuelle.

Avec l'apparition de ces modèles avancés mais destinés à une exploitation automatique, les travaux de formalisation des concepts définissant la structure musicale prennent une importance toujours plus grande. Toutefois, l'état de l'art fait ressortir qu'il n'existe pas une unique définition de la structure musicale, mais une multitude, permettant la description de la musique sous de nombreux angles différents.

Parmi ces approches de description, une définition qui semble particulièrement adaptée à l'utilisation dans des algorithmes est la définition par le biais d'un critère de complexité. Selon cette définition, la structure d'un morceau est le modèle qui permet de décrire celui-ci de façon la plus économique, au sein d'une famille de modèles prédéterminés.

1.3.1 Hypothèses générales

Dans cette thèse, nous nous intéressons à la définition de modèles non séquentiels et d'algorithmes d'inférence pour ce type de structure. La première approche que nous étudions, dans les chapitres 3 et 4, est basée sur un modèle arborescent représenté sous la forme d'une grammaire formelle, et utilise des méthodes issues de l'algorithmique du texte afin de trouver une représentation compacte. La deuxième approche (chapitres 5 et 6) repose sur l'organisation des éléments musicaux en graphes de redondance, dont la forme est basée sur des patrons tensoriels hypercubiques.

1.3.2 Représentation musicale considérée

Dans l'optique de focaliser nos efforts sur la modélisation et l'inférence du niveau intermédiaire de la musique, nous avons opté pour utiliser des données de type symbolique.

De plus, afin de pouvoir traiter ces données comme des séquences régulières, nous avons choisi de traiter des données synchrones aux temps musicaux. La notion de temps musical est parfois ambiguë, et il est fréquent que le tempo soit estimé au double ou à la moitié de sa valeur réelle. Un des objectifs de conception des approches présentées ici est la robustesse à de telles variations.

Pour cette étude, on se concentre principalement sur des morceaux de musique populaire. Toutefois, les concepts et les méthodes présentées restent potentiellement applicables au reste de la musique conventionnelle occidentale.

Même si les deux approches que nous présentons ici sont conçues pour rester génériques, et intégrer directement le minimum de connaissances musicologiques explicites, le choix de la dimension musicale considérée conserve une grande influence sur le résultat final. Pour nos expériences, nous avons choisi d'utiliser des séquences d'accords. Ce choix découle de plusieurs propriétés de l'harmonie, en particulier dans la musique populaire :

- L'harmonie se prête bien à une représentation symbolique. Dans une majorité de contextes, les accords sont représentés par leurs noms et non par la liste des notes qui les forment. En MIR, cette abstraction est fréquemment poussée encore plus loin, réduisant le vocabulaire des accords aux deux formes les plus fréquentes (accords majeurs et mineurs) croisées avec chacune des 12 classes de notes fondamentales, pour un total de 24 accords (25 symboles avec l'inclusion d'un symbole pour les passages sans harmonie).
- La transcription automatique d'accords est une tâche très étudiée en MIR. Par conséquent, il existe de nombreuses bases d'annotations manuelles, dont certaines décrivant des ensembles de données pour lesquels des annotations structurales existent également. De plus, la grande disponibilité d'algorithmes de transcription d'accords laisse envisager une intégration verticale des algorithmes d'inférence de la structure permettant de traiter des données audio.
- En musique populaire, l'harmonie présente un bon compromis entre une forte variabilité entre des segments de types différents et une faible variabilité des segments de même classe. En ce sens, ce choix reflète celui effectué dans [MÜLLER et KURTH, 2007] de sélectionner une représentation d'entrée robuste face aux variations attendues.

Cependant, le choix d'une seule strate musicale pose de fortes limitations sur la détection de la structure. Par exemple, le timbre et la mélodie, qui sont des indices très forts de la structure en musique populaire, sont complètement ignorés. L'extension des méthodes présentées à une multiplicité de strates musicales est l'une des principales pistes d'extension envisageables, mais non couvertes par la présente thèse.

Le choix de travailler à partir de séquences d'accords issues d'une annotation manuelle (considérée exacte) rend difficile la comparaison directe avec des méthodes travaillant à partir du signal audio. D'une part, la justesse des annotations permet d'atteindre de meilleurs résultats que des algorithmes devant procéder à leur propre estimation, mais d'autre part le fait de défausser une grande partie de l'information plafonne les performances.

1.3.3 Objectifs et contributions

D'un point de vue applicatif, la tâche d'inférence d'une segmentation structurale linéaire reste de loin la plus étudiée. Basés sur des approches qui délinéarisent le contenu musical, les modèles développés dans cette thèse permettent de mieux tenir compte de la richesse multi-échelle du contenu musical lors de la modélisation de la structure, tout en veillant à préserver la tractabilité calculatoire du processus d'inférence⁷.

Dans cette thèse, nous abordons la tâche de segmentation structurale de la musique selon 4 axes principaux :

7. Les différents algorithmes d'inférence s'exécutent en des durées allant de l'ordre de la seconde à environ 50% de la durée du morceau étudié. Si la partie haute de cette fourchette ne permet pas une application facile à de grandes bases de données, peu de pistes d'optimisation du temps de calcul ont été développées, et il est probable que des améliorations sensibles puissent être réalisées.

- Nous proposons de formaliser les méthodes d'inférence de la structure musicale par des principes sous-jacents de compression d'information qui peuvent être assimilés à des méthodes d'encodage de données sans perte.
- Dans ce cadre, nous étudions une première famille de méthodes d'inférence de structure, à base de Grammaires à Dérivation Unique, initialement utilisées pour la découverte de structure dans l'ADN [GALLÉ, 2011]. Nous proposons une extension de ce modèle de grammaire afin de mieux prendre en compte les répétitions inexactes et les variations présentes dans les séquences musicales.
- Nous définissons ensuite une méthode d'estimation de la complexité d'une séquence musicale à l'échelle du segment structurel reposant sur un principe de compression tensorielle. Fondée sur une extension du modèle Système & Contraste [BIMBOT et al., 2012], la modélisation tensorielle permet de délinéariser la séquence des éléments musicaux en les répartissant dans des patrons à base hypercubique selon des systèmes de similarité et/ou d'analogie multi-échelle.
- Enfin, une quatrième contribution de cette thèse est la constitution d'une base de données de séquences d'accords, métriquement synchrones, sur les 100 chansons RWC-Pop. Ces données permettent la validation expérimentale et la comparaison des méthodes étudiées dans cette thèse sur un ensemble de test substantiel.

Chapitre 2

Formalisation de l'approche présentée

2.1 Structure sémiotique

Un des buts de cette thèse est de définir des approches et des algorithmes permettant de décrire la structure "sémiotique" d'un morceau de musique, au sens défini dans [BIMBOT et al., 2012], c'est-à-dire sous forme d'une suite de symboles telle que :

$$ABCDEFBCDEGDEDEH$$

où

- chaque étiquette correspond à des segments contigus de taille comparable,
- les segments portant une même étiquette appartiennent à une même classe "d'équivalence",
- le nombre de segments et de classes de segments reste limité.



FIGURE 2.1 – Exemple de structure sémiotique (réduite) pour le morceau RWC-Pop 001. Les largeurs des blocs sont proportionnelles aux durées des segments correspondants.

Comme la plupart des descriptions de la structure utilisées en MIR, l'approche par structure sémiotique est basée sur un découpage en segments contigus (une *segmentation*), qui est complémentaire au processus de groupement décrit dans le chapitre précédent. Ces segments doivent avoir une certaine autonomie et peuvent être écoutés séparément sans apparaître comme incomplets. Cela implique que les segments possèdent une structure interne suffisante pour assurer leur cohérence autonome.

Les segments sont juxtaposés en général, mais occasionnellement ils peuvent se recouvrir partiellement. La représentation de tels recouvrements est limitée au cas où celui-ci est significatif. Dans le cas plus fréquent des anacrouses¹ et autres chevauchements mineurs, le chevauchement est ignoré dans le formalisme².

1. Une anacrouse se produit lorsqu'une des voix musicales commence une section de façon anticipée par rapport aux autres.

2. Idéalement, des méthodes plus avancées permettraient de séparer explicitement l'anacrouse du segment précédent pour la considérer comme faisant partie du segment qui la suit. En pratique, l'anacrouse doit être considérée comme faisant partie du segment pour les considérations d'autonomie et de cohérence interne, mais le début du segment est annoté sans l'anacrouse.

Les segments sont regroupés dans des classes d'équivalence partageant une même étiquette reflétant leur similarité. Cela permet d'identifier facilement les répétitions qui sont un des fondements de la musique conventionnelle. La définition de ces classes d'équivalence est soumise à un compromis entre le nombre de classes et la fidélité de la représentation. Dans un morceau qui présente beaucoup de variations, avoir un seuil élevé pour les classes d'équivalence pourra aboutir à une segmentation de type

ABCDEFGHIJKLM

où chaque segment est le seul représentant de sa classe ; une telle annotation ne donne que peu d'information sur le morceau et sa structure sémiotique, et dans le cas où il est possible de rapprocher certaines classes d'après des points communs, on lui préférera une annotation avec des classes plus larges, telle que :

ABCDEABCGEIJKAB

A l'inverse, un morceau très homogène avec peu de variations risquerait d'être annoté

AAAAAAAAAAAAAAAA

ce qui, encore une fois, ne procure que peu d'information sur la structure du morceau. On préférera alors trouver des caractéristiques permettant de distinguer des classes plus détaillées, aboutissant à une annotation du type :

ABABABCDABABEF

Afin de pouvoir comparer les différents segments entre eux et de pouvoir décrire leurs relations, il est utile qu'ils soient de taille similaire. Comme établi dans en section 1.2.2.1, la structure musicale est présente à de nombreuses échelles, de telle sorte qu'il est généralement possible d'identifier une échelle de description principale. Cela n'exclut pas que certains segments puissent dévier de la taille visée, mais ces segments restent occasionnels et du même ordre de grandeur.

Le choix d'une échelle de description appropriée est alors crucial : si la granularité est trop fine, la séquence d'étiquettes sera redondante et comportera un nombre élevé de segments sans qu'ils apportent une information utile :

AA'BB'CC'DD'EE'FF'BB'CC'DD'EE'GG'DD'EE'DD'EE'HH'.

À l'inverse, si la granularité est trop grossière, il faudra diminuer la régularité des segments :

A BC DE F BC DE G DE DE H

ou avoir des classes moins pertinentes :

AB CD EF BC DE GD ED EH.

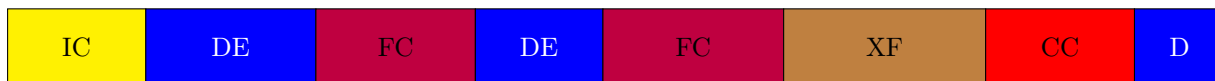
Ce type de compromis est illustré en figure 2.2 sur un exemple réel.



(a) Exemple de sur-segmentation



(b) Exemple de sous-segmentation avec tailles irrégulières



(c) Exemple de sous-segmentation avec classes irrégulières



(d) Exemple de segmentation équilibrée

FIGURE 2.2 – Différentes possibilités de représentation pour le morceau RWC Pop 001

Ces critères peuvent être vus sous l'angle du concept de simplicité. Le but, lors de l'annotation de ce type de structure, est d'obtenir un bon compromis entre la simplicité de la représentation et sa fidélité. La simplicité, ou de façon complémentaire, la complexité de cette représentation se sépare en deux composantes principales :

La complexité de la séquence d'étiquettes Intuitivement, une séquence d'étiquettes tend à être plus complexe si elle comporte beaucoup de symboles. À l'inverse, une séquence d'étiquettes, même longue, peut être simple si elle n'est composée que de quelques symboles différents. D'autres régularités comme la répétition cyclique peuvent également contribuer à rendre une séquence d'étiquettes simple.

La complexité des classes représentées par les étiquettes Différents facteurs contribuent à la complexité d'une classe d'équivalence. Un premier facteur est le nombre de segments qui composent la classe, et le niveau de variabilité entre ces segments : une classe dont tous les membres sont exactement identiques est plus simple qu'une classe dont les membres présentent de fortes dissimilarités. La régularité des membres, en particulier l'archétype de la classe, sont un autre facteur. On peut ici distinguer deux types de régularité : 1. Une régularité intrinsèque du segment, sa *complexité* : un segment plus homogène, dont les propriétés varient peu, peut en général être considéré comme étant plus simple, de même qu'un segment dans lequel on peut identifier une succession logique d'éléments. 2. Une régularité extrinsèque du segment, sa *typicalité* : la réalisation de motifs musicaux récurrents, tels qu'une grille de blues ou une cadence parfaite, peut également contribuer à rendre un segment "simple", au sens où ses propriétés sont prédictibles, particulièrement si ce motif est fréquemment employé dans le genre du morceau.

L'inférence de la structure sémiotique de morceaux, qu'il s'agisse d'un processus manuel ou automatique, doit résulter d'un compromis entre les différents facteurs de complexité évoqués ci-dessus, pour aboutir à une chaîne simple d'éléments simples.

De façon semi-formelle, on peut écrire que l'on cherche à trouver la séquence d'étiquettes $S = s_1 s_2 \dots s_n$ qui minimise une formule du type :

$$C_s(S) + \sum_{s_i \text{ différents}} C_c(s_i), \quad (2.1)$$

où $C_s(S)$ représente la complexité de la séquence d'étiquettes de S et $C_c(s_i)$ la complexité de la classe de l'étiquette s_i .

Cette formulation est intentionnellement très générique, mais les différentes approches présentées dans les chapitres suivants de cette thèse peuvent être vues comme diverses façons de définir en pratique ces termes de complexité et de mettre en œuvre des algorithmes correspondants en permettant la minimisation.

On notera d'ailleurs que la structure sémiotique est une conception de la structure musicale principalement basée sur l'organisation logique à moyenne et grande échelle du morceau. Par conséquent, elle n'a pas pour but de modéliser des segments remplissant une fonction musicale particulière ou de décrire précisément les interactions entre les éléments musicaux de bas niveau.

2.2 Modélisation arborescente

Une alternative au problème de sélection d'une échelle appropriée est d'opter pour un modèle plus élaboré, permettant de tenir compte de l'existence des multiples échelles de structure musicale. Une telle modélisation permet de décrire l'organisation interne des sections décrites dans une structure sémiotique, ainsi que la façon dont ces sections s'articulent en structures de plus haut niveau.

De manière générale, les frontières des éléments de grande taille s'alignent avec celles d'éléments de plus petite taille qui les composent. Ainsi, la structure musicale multi-échelle se prête bien à une représentation hiérarchique reliant les éléments de petite échelle aux éléments de grande échelle qu'ils composent. Dans cette optique, nous utilisons des arbres pour représenter la structure musicale multi-échelle, comme en figure 2.3.

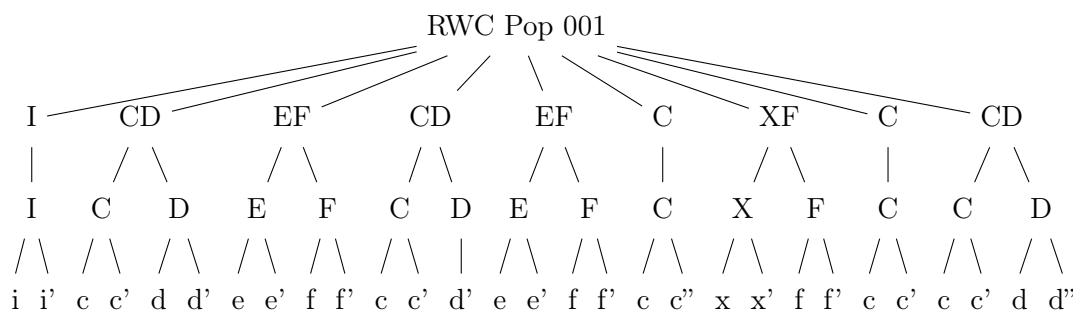


FIGURE 2.3 – Arbre de structure musicale simplifié pour le morceau RWC Pop 001. Voir figure 2.4 pour une version plus détaillée.

Les principes de la représentation sémiotique s'appliquent également à ce type de représentation. Le principe d'homogénéité de taille des segments est celui qui est le plus modifié par

l'extension du système à une description multi-échelle. Au lieu de viser une taille cible pour l'ensemble des segments, de multiples tailles sont requises pour accommoder les différentes échelles. L'objectif devient alors de faire en sorte que les éléments d'un même niveau soient de tailles comparables mais, le principe d'homogénéité s'applique toujours, échelle par échelle.

Ainsi, toutes choses égales par ailleurs, pour une section de taille 16, on préférera un sous-découpage régulier 4/4/4/4 à un sous-découpage 5/3/6/2. De plus, il est souhaitable que les différentes échelles soient clairement différenciées par leur taille. Un découpage 9/7 sera ainsi préférable à un découpage 14/2 (mais un découpage 8/8 sera encore plus favorisé).

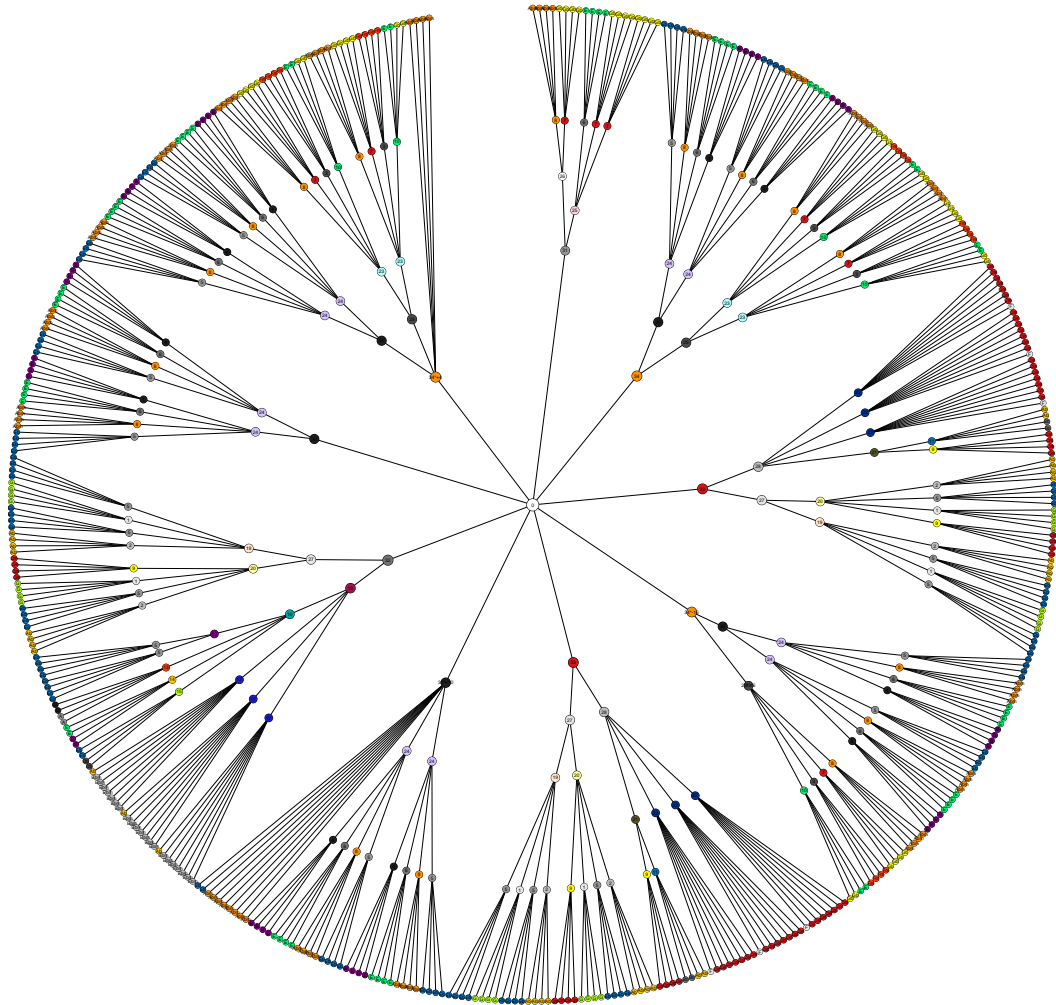


FIGURE 2.4 – Arbre de structure musicale pour le morceau RWC-Pop 001 (*Eien no replica*). L'axe temporel est enroulé sur un cercle dans le sens horaire pour permettre une représentation compacte et sans intersections. La distance de chaque nœud avec la circonférence du cercle est proportionnelle au logarithme de la longueur du segment qu'il représente.

Les classes d'équivalence restent inchangées. Le principe est toujours de traduire autant que possible les redondances du morceau dans la modélisation de sa structure, tout en évitant de regrouper des segments trop différents au sein d'une même classe.

La minimisation du nombre de segments et du nombre de classes de segments perd de son importance car elle jouait un rôle critique lors de la sélection de l'échelle cible. Même si cette étape disparaît, cette minimisation reste un principe sous-jacent important de la modélisation.

2.3 Structure musicale et théorie de l'information

Les critères évoqués dans les sections précédentes peuvent être regroupés sous le concept de complexité. Chacun des concepts vise à atteindre une représentation qui soit la plus simple possible, en sacrifiant de façon contrôlée la précision de la représentation. En ce sens, le problème de la modélisation de la structure musicale peut être vu comme un problème de compression. Il est alors intéressant de poser ce problème sous un angle de théorie de l'information.

L'utilisation de la compression de façon indirecte, de manière à mettre en évidence des propriétés du signal compressé plutôt que pour réduire sa taille, a déjà été utilisée avec succès, par exemple avec le concept de distance de compression [LI, CHEN et al., 2004; CILIBRASI et VITÁNYI, 2005].

Dans cette section, nous décrirons brièvement les concepts sur lesquels reposent les deux branches principales de la théorie de l'information et comment ils peuvent s'appliquer à la modélisation de la structure musicale. Pour un traitement plus complet de la complexité de Kolmogorov, de l'entropie de Shannon et des parallèles qui peuvent être établis entre ces concepts, le lecteur est invité à consulter [LI et VITÁNYI, 1991] et [GRÜNWALD et VITÁNYI, 2003].

2.3.1 Complexité de Kolmogorov

La première approche repose sur le concept de complexité de Kolmogorov [KOLMOGOROV, 1965; KOLMOGOROV, 1968]. Intuitivement, la complexité de Kolmogorov permet d'exprimer le fait que la séquence

$$S_1 = \text{adibadibehmpgehmppgadibadib}$$

est de complexité moindre que la séquence

$$S_2 = \text{adibehmphaabgpampbpmppghmibp}$$

bien qu'elles comportent toutes les deux 26 caractères dont 9 différents. En effet, S_1 peut être décrite comme "3 alternances de 2 fois *adib* et 2 fois *ehmpg*", alors que S_2 peut difficilement être décrite plus simplement qu'en la spécifiant intégralement.

Formellement, pour une machine de Turing M donnée, la complexité de Kolmogorov³ $K_M(x)$ d'un objet x est la longueur du plus petit programme qui produit x lorsqu'il est exécuté sur M .

Une première propriété bien connue de la complexité de Kolmogorov est le théorème d'invariance, qui établit que les complexités de Kolmogorov selon différentes machines de Turing sont équivalentes, à une constante additive près, qui ne dépend que des machines utilisées.

3. La complexité de Kolmogorov est aussi connue entre autres sous les noms de complexité de Chaitin [CHAITIN, 1966] ou de Solomonoff [SOLOMONOFF, 1964] en raison de leur formulation indépendante du concept, ou de complexité algorithmique. Il est important de la distinguer de la complexité calculatoire étudiée en théorie de la complexité, qui décrit les ressources requises pour l'exécution d'un algorithme.

Une deuxième propriété est la non-calculabilité de la complexité de Kolmogorov. Cependant, pour des applications pratiques, il est souvent suffisant de chercher à s'en approcher. Une façon de faire consiste à restreindre la classe de programmes considérés : s'il existe une description courte de la séquence dans la classe restreinte, cette description établit une borne supérieure de la complexité de Kolmogorov.

On peut y trouver là un parallèle avec la complexité intuitive : une séquence issue d'un générateur pseudo-aléatoire, même très longue, peut être décrite simplement à partir de l'algorithme de génération et de la graine utilisée pour amorcer la génération, toutefois un humain ne pourra probablement pas procéder à cette inférence et percevra la séquence comme complexe.

Un autre concept important est la notion de *complexité conditionnelle*, qui permet de mesurer la capacité d'une séquence à assister la description d'une autre. Formellement, $K(X|Y)$ est la longueur du plus court programme prenant comme entrée Y qui génère X .

2.3.2 Entropie de Shannon

A l'inverse de l'approche basée sur la complexité de Kolmogorov qui considère les objets décrits en isolation, l'autre approche de la théorie de l'information considère les objets comme issus d'une source aléatoire connue. Le concept clé de cette branche, l'entropie de Shannon [SHANNON, 1948], peut être décrit intuitivement comme étant une mesure de l'incertitude relative à la production de cette source. Ainsi, une source déterministe sera parfaitement prédictible et aura une entropie nulle, et un dé pipé aura une entropie plus faible qu'un dé équilibré.

Formellement, l'entropie de Shannon est définie comme suit :

Définition 1

Soit X une variable aléatoire prenant des valeurs sur l'ensemble dénombrable⁴ \mathcal{X} avec la distribution P . L'entropie de Shannon (en bits) de la variable X est

$$H(X) = - \sum_{x \in \mathcal{X}} P(X = x) \log_2 P(X = x)$$

La quantité d'information contenue dans un objet peut alors être interprétée comme étant la longueur du mot de code nécessaire pour distinguer cet objet des autres productions possibles de la source.

En pratique, la distribution réelle n'est souvent pas connue, particulièrement dans le cas de données issues de phénomènes réels. Il est alors nécessaire d'approcher la distribution réelle avec des modèles, pré-établis ou estimés de façon dynamique en fonction des observations, ce qui résulte en moyenne en une surestimation de l'entropie.

Ce besoin d'une approximation peut être mis en parallèle avec la non-calculabilité de la complexité de Kolmogorov. Dans un cas comme dans l'autre, les concepts ne peuvent être utilisés en pratique qu'à travers des bornes supérieures.

4. La définition peut également être adaptée au cas continu.

2.3.3 Critères classiques exprimés en termes d'information

Ces concepts peuvent être appliqués à l'inférence de la structure musicale [MEREDITH, 2012 ; BIMBOT, 2016], par l'application du principe de longueur de description minimale [RISSANEN, 1978 ; GRÜNWARD, 2007] (MDL en anglais). Ce principe, issu du paradigme du rasoir d'Occam, préconise la sélection du modèle qui minimise la complexité de la paire constituée par le modèle et la description des données selon ce modèle⁵. Le très similaire principe de simplicité a également été suggéré comme étant un des principes fondamentaux de la cognition [CHATER, 1999].

Comme nous l'avons mentionné au chapitre précédent, les critères classiques de segmentation peuvent être exprimés en terme d'information et de complexité :

- Le critère de répétition correspond à l'existence de sous-séquences distantes qui présentent chacune une complexité conditionnelle faible en fonction de l'autre.
- L'homogénéité des segments se traduit par une complexité plus faible des sous-séquences qui correspondent à un segment. À l'inverse, une forte complexité conditionnelle de l'élément suivant correspond à une rupture de l'homogénéité et permet de retrouver le critère de nouveauté. Sur cet axe, on peut trouver la dynamique de l'information, étudiée par [PEARCE, 2005] et [ABDALLAH et PLUMBLEY, 2009 ; ABDALLAH, EKEUS et al., 2012], qui permet d'analyser la répartition de l'information dans le morceau.
- L'utilisation de règles basées sur des connaissances musicales peut quant à elle s'interpréter comme des connaissances sur la source qui génère les segments [PEARCE et WIGGINS, 2006 ; WIGGINS, MÜLLENSIEFEN et PEARCE, 2010] qui montrent alors une entropie de Shannon plus faible que les sous-séquences qui ne sont pas alignées avec les segments.

2.3.4 Résumé de l'approche considérée

Dans cette thèse, nous nous intéressons à l'inférence par compression de la structure musicale. Plus particulièrement, nous considérons la structure musicale dans sa définition sémiotique, mais également sur son aspect multi-échelle.

Les méthodes d'inférence que nous présentons s'appuient sur des éléments de théorie de l'information, en particulier sur les concepts de complexité de Kolmogorov et de complexité conditionnelle. Les modèles introduits permettent également l'intégration de connaissances musicales, sur des bases d'entropie de Shannon, même si ces aspects ne sont que peu explorés dans ces travaux.

5. On peut toutefois relever des objections à son utilisation, voir [DOMINGOS, 1999].

Chapitre 3

Inférence de structure musicale par plus courte grammaire

Comme annoncé précédemment, une des approches étudiées dans cette thèse consiste à modéliser la structure d'un morceau de musique par le biais de grammaires formelles. Le formalisme des grammaires permet d'obtenir de très nombreuses représentations alternatives de la séquence correspondant au morceau étudié. Parmi ces représentations alternatives, certaines sont considérablement plus économiques en termes d'information que la représentation naïve sous forme de séquence. D'après l'hypothèse principale de cette thèse, les régularités qui permettent ces représentations économiques sont des éléments constitutifs de la structure du morceau.

Ce chapitre est donc consacré à la formalisation du problème de la plus courte grammaire et à son exploration pour la modélisation de séquence issues de morceaux de musique. La section 3.1 établit les diverses définitions et notations nécessaires. La section 3.2 reprend les travaux existants, notamment ceux de la thèse de Gallé [GALLÉ, 2011]. La section 3.3 procède à l'application des méthodes existantes au domaine de la musique et établit un constat des spécificités rencontrées. Ce modèle sera ensuite repris au chapitre 4 afin d'en présenter une version étendue.

Il est important d'effectuer une distinction entre les grammaires utilisées ici comme outil de représentation d'un morceau et les grammaires musicales qui font l'objet d'études comme [ROADS, 1979; BARONI, MAGUIRE et DRABKIN, 1983], qui visent à définir des grammaires universelles, permettant de décrire une large variété de morceaux selon un même modèle.

3.1 Modèles à base de grammaires formelles

Les grammaires formelles sont un outil classique dans le traitement des séquences. Nous repreneons ici certaines définitions classiques de Chomsky [CHOMSKY, 1956; CHOMSKY, 1957] selon un formalisme usuel, avant de présenter des définitions plus spécifiques à partir de la section 3.1.4.

3.1.1 Séquences

Une *séquence* (ou *chaîne*) est la concaténation d'un nombre fini de symboles issus d'un *alphabet* Σ . On note Σ^* l'ensemble des séquences utilisant les symboles de Σ , et Σ^+ ce même ensemble privé de la *chaîne vide* ϵ .

Dans cette thèse, les séquences seront généralement désignées soit par des lettres minuscules grecques, soit par la concaténation des symboles ou des sous-séquences qui les composent. Les séquences sont indexées à partir de 0 et leurs symboles individuels sont désignés par des indices, de sorte que $\alpha = \alpha_0\alpha_1 \dots \alpha_{|\alpha|-1}$, où $|\alpha|$ désigne la *longueur* (c'est-à-dire le nombre de symboles) de la séquence α . Les sous-séquences sont désignées par une paire d'indices $\alpha_{i,j}$; on parle d'un *préfixe* si $i = 0$ et d'un *suffixe* si $j = |\alpha| - 1$. La paire (i, j) désigne une *occurrence* de $\alpha_{i,j}$ dans α entre ces deux positions.

3.1.2 Grammaires formelles

Une grammaire formelle est un système de réécriture à base de règles permettant de générer un ensemble de chaînes de symboles à partir d'un seul symbole. Formellement, une grammaire est un quadruplet $\langle \mathcal{N}, \Sigma, \mathcal{R}, S \rangle$:

- \mathcal{N} et Σ sont des ensembles finis de symboles, généralement disjoints, respectivement appelés *symboles non-terminaux* et *symboles terminaux*. Par la suite, nous noterons leur union $\mathcal{V} = \mathcal{N} \cup \Sigma$; sur les exemples abstraits, les symboles seront désignés par des lettres romaines, majuscules pour les non-terminaux et minuscules pour les terminaux.
- \mathcal{R} est un ensemble fini de *règles de production*, définies sur $\mathcal{V}^+ \times \mathcal{V}^*$ et notées sous la forme $\alpha \rightarrow \beta$ où α et β sont appelés membres respectivement de gauche et de droite;
- S est un non-terminal particulier appelé *axiome* ou *symbole de départ*, désigné S ou 0 par convention.

$$S \rightarrow Aab \tag{1}$$

$$Aa \rightarrow bSa \tag{2}$$

$$A \rightarrow bb \tag{3}$$

FIGURE 3.1 – Une grammaire composée de 3 règles ($\mathcal{N} = \{S, A\}, \Sigma = \{a, b\}$)

L'application d'une règle de production se fait en remplaçant dans une chaîne une occurrence du membre de gauche d'une règle par le membre de droite de la même règle. L'application successive d'une ou plusieurs règles constitue le processus de *dérivation*. Ce processus est illustré en figure 3.2. On appelle *langage engendré* par la grammaire l'ensemble des chaînes de Σ^* pouvant être atteintes par dérivation depuis l'axiome de la grammaire (on parle alors d'un *mot* du langage). La figure 3.3 montre un exemple de grammaire et le langage qu'elle permet de générer.

S		S	
Aab	(1)	Aab	(1)
$bSab$	(2)	$bbab$	(3)
$bAabab$	(1)		
$bbbabab$	(3)		

FIGURE 3.2 – Exemples de dérivations sur la grammaire de la figure 3.1. Les nombres en fin de ligne indiquent la règle utilisée pour transformer la ligne précédente.

$$\begin{aligned}
S &\rightarrow abcS \\
S &\rightarrow \epsilon \\
ab &\rightarrow ba \\
ba &\rightarrow ab \\
ac &\rightarrow ca \\
ca &\rightarrow ac \\
bc &\rightarrow cb \\
cb &\rightarrow bc
\end{aligned}$$

FIGURE 3.3 – Une grammaire ($\mathcal{N} = \{S\}, \Sigma = \{a, b, c\}$) dont le langage engendré est l'ensemble des chaînes de $\{a, b, c\}^*$ constituées d'un nombre égal de a, b et c .

Afin de considérer une grammaire formelle comme un programme au sens de la complexité de Kolmogorov, il est nécessaire d'inclure une séquence complémentaire, qui indique dans quel ordre et à quelles positions appliquer les règles. En termes de critère MDL, la grammaire constitue le modèle et la séquence complémentaire la description selon le modèle.

Les grammaires formelles sont un outil très puissant, qui permet de générer tous les langages reconnaissables par des machines de Turing. Toutefois, cette puissance les rend difficiles à utiliser, et il est commun de se restreindre à un sous-ensemble plus simple de grammaires, tel que les grammaires hors-contexte [HIGUERA, 2009].

3.1.3 Grammaires hors-contexte

Une grammaire hors-contexte est une grammaire formelle telle que son ensemble de règles \mathcal{R} appartient à $\mathcal{N} \times \mathcal{V}^*$, c'est-à-dire que toutes ses règles de production ont pour membre de gauche un unique non-terminal. La principale conséquence de cette restriction est que la dérivation d'une sous-chaîne se fait de façon indépendante du reste de la chaîne (d'où le nom de hors-contexte).

Si l'on considère uniquement les grammaires réduites (c'est-à-dire que tous les symboles apparaissent dans la dérivation d'au moins un mot du langage engendré), il est possible de dénoter de façon interchangeable la grammaire et son ensemble de règles de production en admettant pour convention que la première règle énumérée a l'axiome pour membre de gauche.

En effet, les membres non énumérés peuvent alors être déduits de la liste des règles : l'ensemble de non-terminaux est l'ensemble des symboles apparaissant dans les membres de gauche, l'ensemble des terminaux est composé du reste des symboles, et l'axiome est déduit de la convention. Nous dénoterons donc les grammaires hors-contexte de façon simplifiée, par la seule liste de leurs règles de production, comme en figure 3.4.

L'indépendance de la dérivation d'une sous-chaîne du reste de la chaîne permet également une représentation du processus de dérivation sous la forme d'un arbre, où les feuilles représentent les symboles terminaux générés et les nœuds représentent les non-terminaux utilisés dans la dérivation. Un exemple d'arbre de dérivation est présenté en figure 3.5. Ces arbres peuvent être interprétés comme un découpage hiérarchique de la séquence découlant du choix de la grammaire.

$S \rightarrow aSb$ $S \rightarrow \epsilon$ <p>(a) Grammaire générant le langage $\{a^n b^n, n \geq 0\}$</p>	$S \rightarrow G_n V_t G_n$ $G_n \rightarrow D_m N_m$ $G_n \rightarrow D_f N_f$ $D_m \rightarrow \text{le}$ $D_m \rightarrow \text{un}$ $D_f \rightarrow \text{la}$ $D_f \rightarrow \text{une}$ $N_m \rightarrow \text{canard}$ $N_m \rightarrow \text{chat}$ $N_f \rightarrow \text{souris}$ $N_f \rightarrow \text{lionne}$ $V_t \rightarrow \text{suit}$ $V_t \rightarrow \text{mange}$
	(b) Grammaire générant quelques phrases simples

FIGURE 3.4 – Exemples de grammaires hors-contexte

Pour passer d'une grammaire hors-contexte à un programme au sens de Kolmogorov, générant un mot spécifique du langage engendré par la grammaire, il suffit cette fois d'une séquence complémentaire déterminant quelle règle utiliser pour chaque non-terminal rencontré. Il est en effet possible de définir une convention sur l'ordre de dérivation (par exemple, toujours dériver le non-terminal apparaissant en premier dans la chaîne) sans affecter le résultat final de la dérivation.

Afin de rendre possible l'inférence d'une grammaire à partir d'un unique mot faisant partie de son langage, nous nous concentrons sur une classe de grammaires hors-contexte encore plus restreinte, les grammaires à dérivation unique.

3.1.4 Grammaires à Dérivation Unique (GDU)

Une grammaire à dérivation unique (ou *Straight-Line Grammar (SLG)* selon la terminologie de Gallé¹) est une grammaire hors-contexte telle que chaque non-terminal peut être dérivé en une seule chaîne de terminaux (appelée *chaîne constitutive*). Cela s'exprime par deux contraintes :

1. chaque non-terminal apparaît dans exactement un membre de gauche ;
2. aucun non-terminal ne peut apparaître dans une de ses propres dérivations.

La première contrainte assure l'unicité des chaînes constitutives. La deuxième assure son existence en interdisant toute récursion, qui serait nécessairement infinie à cause de la première contrainte.

La conséquence directe de cette définition est que chaque GDU génère un seul mot. Chaque GDU peut alors être considérée comme une représentation exacte de la séquence qu'elle génère : il

1. Bien que l'utilisation de cette classe de grammaire soit antérieure aux travaux de Gallé [NEVILL-MANNING, 1996 ; SAKAMOTO, 2005 ; YANG et HE, 2003], à notre connaissance aucune autre approche ne nomme cette classe de grammaires.

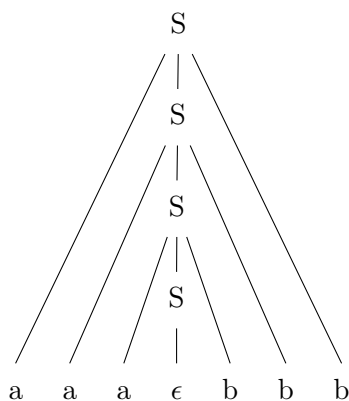


FIGURE 3.5 – Arbre de dérivation pour le mot $aaabbb$ selon la grammaire de la figure 3.4a

n'est plus nécessaire de disposer d'une séquence complémentaire spécifiant quelle règle appliquer pour chaque non-terminal car aucun choix n'est possible (à l'exception de l'ordre de dérivation, qui n'a pas d'influence). Dans le critère MDL, le terme correspondant à la description de la séquence selon le modèle disparaît, laissant uniquement la longueur de description du modèle, c'est-à-dire la GDU.

De plus, comme la dérivation est unique (à l'ordre près), il existe un unique arbre de dérivation de la chaîne constitutive, et la GDU peut aussi être considérée comme une représentation de la structure hiérarchique impliquée par l'arbre. Il est également possible de passer de l'arbre de dérivation, même si les nœuds ne sont pas étiquetés, à une GDU unique avec les hypothèses suivantes :

- La GDU ne comporte pas de règles redondantes, c'est-à-dire toutes les règles ont un membre droit différent².
- Deux GDU équivalentes à un renommage des non-terminaux près sont fonctionnellement identiques.

Il est alors possible d'utiliser de façon interchangeable une GDU, son arbre de dérivation et la structure hiérarchique correspondante.

3.2 Minimisation de la longueur de la grammaire

Selon la définition de la complexité de Kolmogorov, une chaîne est aussi complexe que le programme le plus simple la générant. L'obtention d'une GDU simple générant une chaîne donnée, si elle ne permet pas d'en évaluer directement la complexité de Kolmogorov, permet d'en obtenir une borne supérieure [CERRA et DATCU, 2010]. De plus, l'examen de cette GDU permet de révéler une structure de la chaîne qui permet une génération simple. Ainsi, il est possible d'obtenir une estimation de la structure d'une chaîne à travers la recherche de la GDU la plus simple la générant.

La longueur d'une GDU G est définie comme étant la somme de la longueur des membres droits de ses règles :

$$|G| = \sum_{N \rightarrow \alpha \in \mathcal{R}} |\alpha|$$

2. La simplification d'une GDU comportant des règles redondantes peut s'effectuer trivialement en fusionnant les non-terminaux correspondants au paires de règles redondantes.

Dans un premier temps, nous considèrerons cette mesure comme étant représentative de la complexité d'une grammaire. Cette formulation a fait l'objet d'études théoriques, notamment [LEHMAN et SHELAT, 2002; CHARIKAR, LEHMAN et LIU, 2002; CHARIKAR, LEHMAN, LIU et al., 2005]. Dans le prochain chapitre, nous examinerons des formulations plus avancées de la complexité d'une grammaire.

Dans le cadre applicatif présenté dans ce chapitre et le suivant, cette mesure de complexité sert de critère à minimiser pour trois algorithmes. Ces algorithmes, appliqués successivement, transforment incrémentalement une GDU triviale (composée d'une seule règle) en une nouvelle GDU générant la même séquence, mais ayant une complexité inférieure ou égale. De façon simplifiée, les transformations effectuées par ces algorithmes sont :

- l'introduction de nouvelles règles (algorithme IRR), permettant de regrouper des sous-séquences identiques ;
- la ré-écriture des règles existantes (algorithme MGP), qui ré-écrit les règles en un minimum de symboles, sans changer leurs séquences constitutives ;
- la suppression de règles existantes (algorithme CLEANUP), éliminant les règles trop rarement utilisées dans la grammaire.

3.2.1 Remplacement itératif des répétitions (algorithme IRR)

La plupart des algorithmes de compression à base de GDU (par exemple SEQUITUR [NEVILL-MANNING et WITTEN, 1997] ou REPAIR [LARSSON et MOFFAT, 2000]) ont un fonctionnement similaire : une première grammaire triviale est construite à partir de la séquence analysée, puis de nouvelles règles sont ajoutées de façon itérative en remplaçant des répétitions dans la grammaire jusqu'à épuisement des possibilités (ou un point arbitraire de l'exécution).

Les algorithmes de compression à base de macros (par exemple [STORER et SZYMANSKI, 1982; ALLISON, EDGOOSE et DIX, 1998; APOSTOLICO et LONARDI, 2000]), même s'ils ne sont pas initialement formulés à l'aide de grammaires, peuvent également y être rattachés car les règles d'une GDU sont assimilables à des macros récursives. Cette classe d'algorithmes inclut notamment les algorithmes LZ77 [ZIV et LEMPEL, 1977] et LZ88 [ZIV et LEMPEL, 1978], qui sont la fondation de nombreux outils actuels de compression sans perte, dont l'usage est largement répandu.

Dans sa thèse, Gallé propose une généralisation regroupant ces algorithmes sous une même formulation, présentée dans l'algorithme 1 sous forme de pseudocode.

Algorithme 1 Remplacement itératif des répétitions (IRR)

Entrée: une séquence α , une fonction de score f
Initialiser G avec une règle $N_0 \rightarrow \alpha$
tant que $\max f(\beta, G) > 0$ **faire**
 Trouver $\hat{\beta}$ la séquence qui maximise $f(\beta, G)$
 Remplacer chaque occurrence de $\hat{\beta}$ dans G par N_i
 Ajouter une nouvelle règle $N_i \rightarrow \hat{\beta}$ à G
fin tant que
Sortie: G , une GDU compacte générant α

Cet algorithme est constitué d'une série principale d'opérations, répétées jusqu'à convergence :

- l'algorithme identifie une séquence répétée à factoriser, selon une stratégie caractérisée par une fonction de score f ;
- toutes les occurrences de la séquence à factoriser sont remplacées par un nouveau non-terminal ;
- une nouvelle règle est ajoutée, permettant d'obtenir la séquence remplacée par dérivation du nouveau non-terminal.

Les différentes stratégies de remplacement correspondant aux divers algorithmes peuvent alors être implémentées à travers différentes versions la fonction de score. Par exemple, si f associe à chaque paire de symbole son nombre de répétitions dans G (et 0 aux séquences de plus de 2 symboles), on retrouve un comportement analogue à celui de l'algorithme REPAIR [LARSSON et MOFFAT, 2000].

Un cas particulier apparaît lorsque plusieurs occurrences d'une sous-séquence se recouvrent, comme c'est le cas par exemple pour la sous-séquence *aba* dans la séquence *abababababa*. Il est impossible dans ce type de situation de remplacer toutes les occurrences. Les occurrences doivent alors être sélectionnées afin d'obtenir des occurrences distinctes. Une méthode simple, appliquée dans notre implémentation, consiste à conserver la première occurrence non traitée et à éliminer les autres occurrences la recouvrant, en continuant ainsi jusqu'à épuisement des occurrences. On peut démontrer que cette méthode conduit à conserver un nombre maximum d'occurrences, mais la solution n'est pas unique³.

Cet algorithme converge, à condition que la fonction de score soit bien conçue, c'est-à-dire de telle sorte qu'un remplacement qui augmente la taille de la grammaire soit associé à un score négatif.

Toutefois, le résultat ne correspond pas nécessairement à un minimum global, en raison de la nature gloutonne de l'algorithme, car certaines substitutions effectuées en début d'exécution s'avèrent sous-optimales.

Accélération de la recherche de répétitions La recherche naïve de répétitions dans une séquence est un processus coûteux en temps de calcul. Une solution utilisée en algorithmique du texte est la construction d'une table triée des suffixes [ABOUELHODA, KURTZ et OHLEBUSCH, 2004]. Un exemple de cette structure est présenté en figure 3.6. Cette structure de données regroupe toutes les répétitions de chaque sous-séquence sur des positions contiguës. Ce regroupement permet ensuite trouver à faible coût toutes les occurrences de toutes les sous-séquences. Gallé propose dans [GALLÉ, PETERLONGO et COSTE, 2008] une optimisation supplémentaire basée sur une mise à jour incrémentale de la table des suffixes, ce qui évite de recalculer la table entière après chaque itération. Après expérimentation, cette optimisation ne s'avère pas efficace dans notre cas, car les séquences étudiées sont beaucoup plus courtes, et les répétitions remplacées en couvrent une proportion plus élevée.

3.2.2 Optimisation des règles (algorithme MGP)

Afin de réduire davantage la longueur de la grammaire finale, Gallé et collègues [CARRASCOSA et al., 2010 ; CARRASCOSA et al., 2011b ; CARRASCOSA et al., 2011a] présentent un algorithme

3. On peut par exemple commencer par la dernière occurrence.

Index	Suffixe	Index	Suffixe
0	<i>abaabcdabcaab</i>	10	<i>aab</i>
1	<i>baabcdabcaab</i>	2	<i>aabcdabcaab</i>
2	<i>aabcdabcaab</i>	11	<i>ab</i>
3	<i>abcdabcaab</i>	0	<i>abaabcdabcaab</i>
4	<i>bcdabcaab</i>	7	<i>abcaab</i>
5	<i>cdabcaab</i>	3	<i>abcdabcaab</i>
6	<i>dabcaab</i>	12	<i>b</i>
7	<i>abcaab</i>	1	<i>baabcdabcaab</i>
8	<i>bcaab</i>	8	<i>bcaab</i>
9	<i>caab</i>	4	<i>bcdabcaab</i>
10	<i>aab</i>	9	<i>caab</i>
11	<i>ab</i>	5	<i>cdabcaab</i>
12	<i>b</i>	6	<i>dabcaab</i>

(a) Table des suffixes

(b) Table des suffixes triée selon l'ordre lexicographique

FIGURE 3.6 – Construction d'une table triée des suffixes pour la séquence *abaabcdabcaab*

consistant à fixer les symboles non-terminaux et leurs chaînes constitutives pour trouver une réécriture optimale des règles de production.

Pour chacune des règles $N_i \rightarrow \alpha_i$ de la GDU, on construit un graphe représentant toutes les séquences de symboles (terminaux et non-terminaux) dont la dérivation aboutit à la séquence constitutive de N_i . On sélectionne ensuite parmi ces séquences celle qui minimise la longueur de la grammaire. La figure 3.7 montre un exemple d'un de ces graphes et de sa construction, et le cas général est décrit en pseudocode dans l'algorithme 2.

Ces graphes peuvent être interprétés de la façon suivante :

- chaque arc étiqueté d'un terminal correspond à un symbole de la séquence constitutive du non-terminal associé au graphe ;
- chaque nœud correspond à une position entre deux symboles de la séquence constitutive. Le premier et le dernier nœud correspondent respectivement aux positions avant le premier symbole et après le dernier symbole ;
- chacun des arcs étiquetés d'un non-terminal correspond à une façon de modéliser la sous-séquence entre les positions correspondant à son nœud source et à son nœud cible ;
- chaque chemin du nœud initial au nœud final correspond à une reformulation possible de la règle.

Une fois ces graphes construits⁴, on recherche pour chacun le chemin le plus court du premier au dernier nœud. La recherche du plus court chemin dans un graphe est un problème classique, pour lequel on peut utiliser un algorithme classique, comme l'algorithme 3. Dans un dernier temps, l'algorithme remplace chaque partie droite des règles par la séquence des étiquettes des arcs du plus court chemin sur le graphe correspondant.

L'algorithme 4 récapitule le déroulement de l'algorithme MGP.

4. En pratique, seul le graphe Γ_0 correspondant à l'axiome doit être construit car les autres sont des sous-graphes de celui-ci.

$$\begin{aligned}
S &\rightarrow AadbcabB \\
A &\rightarrow abc \\
B &\rightarrow cad
\end{aligned}$$

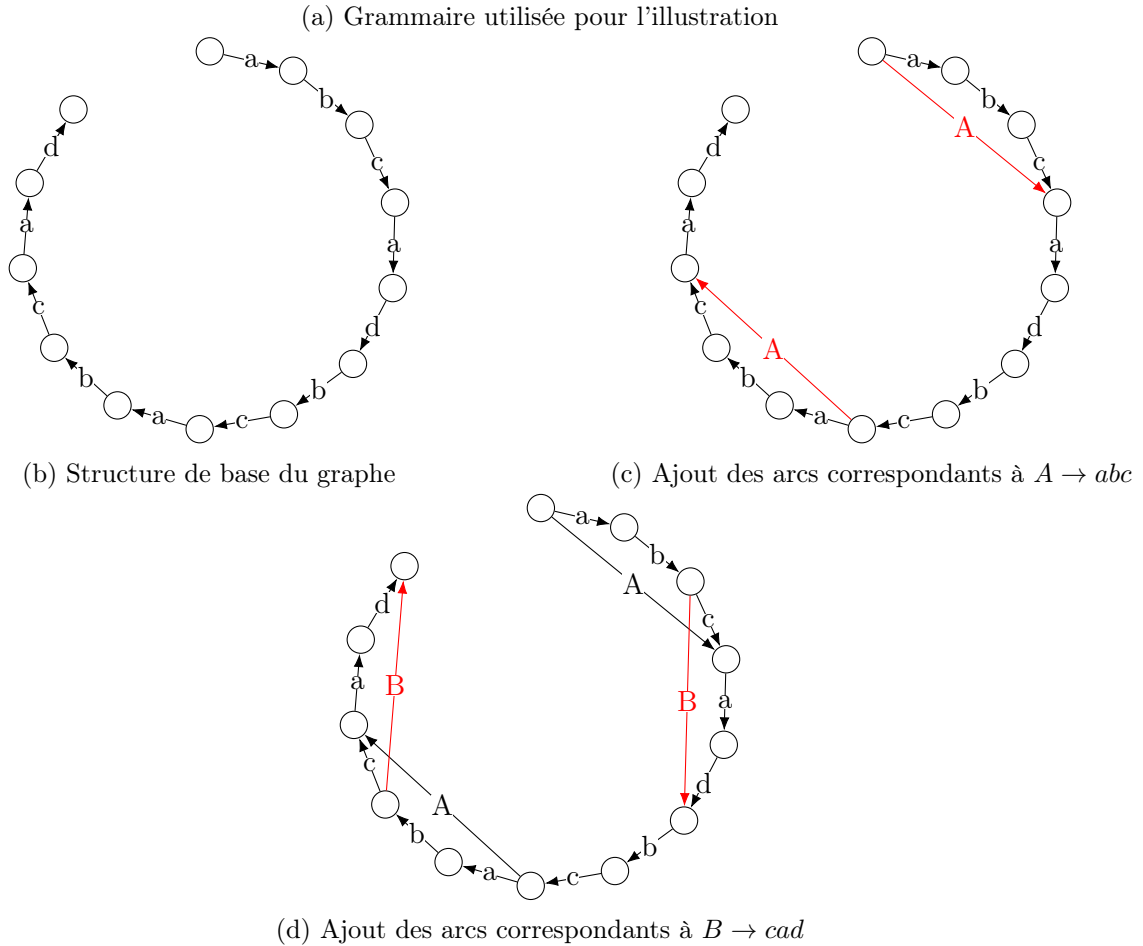


FIGURE 3.7 – Illustration de la construction de Γ_0 pour une grammaire simple

Algorithme 2 Construction de Γ_0 , graphe de ré-écriture de l'axiome.

Entrée: une GDU G générant la séquence α

Créer Γ_0 avec un noeud n_0

pour i de 1 à $|\alpha|$ **faire**

 Ajouter à Γ_0 un noeud n_i

 Ajouter à Γ_0 un arc de n_{i-1} à n_i , étiqueté α_i

fin pour

pour tout $N_i \rightarrow \beta \in G$ **faire**

 Trouver γ_i la séquence constitutive de N_i

pour tout occurrence de γ_i entre α_j et α_k **faire**

 Ajouter à Γ_0 un arc de n_{j-1} à n_k , étiqueté N_i

fin pour

fin pour

Sortie: Γ_0 , le graphe d'optimisation de G

Algorithme 3 Breadth-First Search (BFS) pour le plus court chemin dans un graphe

Entrée: un graphe dirigé G , un noeud de départ n_0 et un noeud cible n_Ω

$V = \{n_0\}$

▷ Recherche de n_Ω dans le graphe

initialiser une file (first in - first out) F avec n_0

tant que $n_\Omega \notin V$ **faire**

 prendre n en tête de F

pour tout n' successeur de n **faire**

si $n' \notin V$ **then**

$V := V \cup \{n'\}$

$\text{pred}[n'] := n$

 Ajouter n' en queue de F

fin si

fin pour

fin tant que

$n := n_\Omega$

▷ Reconstruction du chemin emprunté

$C := [n_\Omega]$

répéter

$n := \text{pred}[n]$

 Préfixer C de n

jusqu'à $n = n_0$

Sortie: C , chemin le plus court de n_0 à n_ω dans G

Algorithme 4 MGP

Entrée: une GDU G

pour tout $N_i \rightarrow \alpha^i \in G$ **faire**

 Construire Γ_i le graphe de ré-écriture pour la séquence constitutive de N_i

 Trouver $\hat{\alpha}^i$ la séquence associée au plus court chemin traversant Γ_i

 Remplacer dans G la règle $N \rightarrow \alpha^i$ par $N \rightarrow \hat{\alpha}^i$

fin pour

Sortie: G , une GDU au moins aussi compacte et générant la même séquence qu'initialement.

3.2.3 Élagage de règle (algorithme CLEANUP)

Un des effets de bords de l'algorithme précédent est que certaines règles auparavant utiles peuvent se retrouver insuffisamment utilisées pour compenser la longueur de leur définition.

Selon la stratégie de remplacement utilisée dans IRR, l'ajout de règles peut également rendre coûteuses certaines règles antérieures.

L'impact sur la longueur de la grammaire de la suppression (ou développement) d'une règle dépend de sa longueur l et de son nombre d'utilisations n :

- la suppression de la règle permet de réduire la taille de la grammaire de l symboles ;
- le remplacement dans le reste de la grammaire du non-terminal associé à la règle par la séquence correspondante augmente la taille de la grammaire de $l - 1$ symboles pour chaque occurrence.

Pour la suppression d'une règle de longueur l utilisée n fois, on a ainsi

$$\Delta|G| = (l - 1)(n - 1) - 1 \tag{3.1}$$

Si et seulement si $l < 2$ ou $n < 2$, la règle est coûteuse et sa suppression permettrait de réduire la taille de la grammaire⁵.

Une étape supplémentaire (désignée CLEANUP) peut donc être ajoutée consistant à supprimer les règles coûteuses.

3.2.4 Combinaison des algorithmes

Ces trois algorithmes (IRR, MGP et CLEANUP) peuvent être combinés de différentes façons. IRR est requis dans tous les cas, car MGP et CLEANUP nécessitent que des règles soient créées au préalable. MGP et CLEANUP peuvent être exécutés après chaque itération de IRR, ou être exécutés seulement une fois que IRR a convergé.

Gallé retient 4 choix de combinaison :

1. IRR seul,
2. IRR avec MGP à chaque itération (IRR+MGP),
3. IRR avec MGP et CLEANUP à chaque itération (IRR+MGP+C), et
4. une alternance de IRR jusqu'à convergence et de MGP et CLEANUP (cf algorithme 5 – IRRMGP*).

Algorithme 5 IRRMGP*

Entrée: une séquence α , une fonction de score f

Initialiser G avec une règle $N_0 \rightarrow \alpha$

répéter

tant que $\max f(\beta, G) > 0$ **faire** ▷ IRR

 Trouver $\hat{\beta}$ la séquence qui maximise $f(\beta, G)$

 Remplacer les occurrences de $\hat{\beta}$ dans G par N_i

 Ajouter une nouvelle règle $N_i \rightarrow \hat{\beta}$ à G

fin tant que

pour tout $N_k \in \mathcal{N}$ **faire** ▷ MGP

 Construire Γ_k

 Trouver $\hat{\gamma}^k$ la séquence associée au plus court chemin de n_1^k à $n_{|\gamma_k|+1}^k$

 Remplacer dans G la règle $N \rightarrow \gamma^k$ par $N \rightarrow \hat{\gamma}^k$

fin pour

pour tout $N_k \in \mathcal{N}$ **faire** ▷ CLEANUP

si N_k a moins de deux occurrences dans G **then**

 Retirer $N_k \rightarrow \gamma^k$ de G

 Remplacer toute occurrence de N_k dans G par γ^k

fin si

fin pour

jusqu'à G n'a pas changé depuis l'itération précédente

Sortie: G , une GDU compacte générant α

5. Avec la définition de longueur employée par Gallé, une règle de longueur 2 utilisée 2 fois est également coûteuse car la suppression de la règle réduit la taille de la grammaire de $l + 1$.

Gallé rapporte des expériences sur un corpus de séquences d'ADN et un corpus de textes, qui montrent un avantage pour les méthodes utilisant MGP, avec un avantage plus marqué en faveur d'IRRMGP* pour l'ADN.

À ce stade, une généralisation de ces résultats à des séquences issues de morceaux de musique serait toutefois hasardeuse. De plus, cette comparaison est limitée à la taille des grammaires inférées ; il est tout à fait envisageable qu'une grammaire de taille supérieure permette de déduire une meilleure estimation de la structure d'un morceau qu'une autre grammaire, plus petite. Des expériences sur d'autres contenus s'imposent avant de pouvoir tirer des conclusions à ce sujet.

3.3 Mise en pratique

Afin de juger de la pertinence de l'application de l'approche par inférence de grammaire à des séquences musicales, nous avons mené à bien plusieurs types d'expériences :

- Les premières expériences visent à déterminer si les tendances observées par Gallé sur la compression de séquences longues (de dizaines de milliers à plusieurs millions de symboles), d'origine biologique ou textuelle, se transposent aux séquences relativement courtes (quelques centaines de symboles) rencontrées en musique.
- Dans un deuxième temps, nous procédons à une évaluation qualitative de la pertinence des structures découvertes par l'algorithme IRRMGP* dans du texte.
- La troisième série d'expériences évalue quantitativement les structures inférées sur de la musique, en s'appuyant sur une projection de la structure hiérarchique vers une structure linéaire.
- Enfin, nous recensons les types d'erreurs récurrentes présentes dans les structures inférées par l'algorithme, afin de pouvoir les compenser à l'aide des modifications apportées dans le chapitre suivant.

L'application d'algorithmes de compression par GDU (ou par macros) à la musique a également été étudiée dans des travaux précédents [PACHET, 1999] et concomitants à cette thèse. Ainsi, Nevill-Manning [NEVILL-MANNING, 1996] a (brièvement) appliqué l'algorithme SEQUITUR pour la détection de motifs répétés dans des chorales de Bach. Dans [LOUBOUTIN et MEREDITH, 2016], les algorithmes LZ77, LZ78 et Burrows-Wheeler [BURROWS et WHEELER, 1994] sont utilisés pour la classification de mélodies folks et la découverte de sujets et contre-sujets dans les fugues de Bach.

3.3.1 Ensembles de données

Pour les expériences sur des séquences musicales présentées dans cette thèse, nous avons utilisé des ensembles d'annotations portant sur les morceaux de la base RWC-Pop [GOTO et al., 2002].

Cette base est constituée de 100 morceaux originaux, dans le style des musiques pop américaines et japonaises entre 1980 et 2000. Elle est fréquemment utilisée pour l'évaluation d'algorithmes de segmentation musicale, notamment dans le cadre de la campagne annuelle d'évaluation MIREX [EHMANN et al., 2011].

3.3.1.1 Séquences d'accords

Pour les séquences utilisées en entrée de nos algorithmes, nous avons choisi de nous focaliser sur des séquences d'accords, échantillonnées à chaque temps musical. Bien que les algorithmes présentés soit indifférents à la nature des séquences analysées, les séquences d'accords se prêtent bien à une représentation symbolique et offrent un bon compromis entre disponibilité, abstraction, stabilité et variabilité.

En effet, il est possible de représenter l'ensemble des accords sous forme symbolique avec un vocabulaire limité. Dans notre cas, nous utilisons une projection sur l'ensemble des triades majeures et mineures, avec un symbole pour chaque fondamentale et un symbole supplémentaire pour les passages sans harmonie.

Les accords permettent en outre de distinguer dans la plupart des cas les segments n'étant pas apparentés, sans pour autant être trop variables au sein d'une même classe de segments. Une exception notable est le cas des transpositions, qui nécessiteraient une gestion particulière par les algorithmes.

On peut distinguer plusieurs types de représentations des séquences d'accords :

- Les progressions d'accords, qui décrivent uniquement l'ordre dans lequel les accords s'enchaînent. Dans ce type de représentation, la dimension temporelle n'est considérée que pour donner une notion de chronologie, sans chercher à quantifier la durée de chaque accord. Ainsi, un accord est annoté de la même façon dans la séquence, qu'il soit tenu pendant un demi-temps ou plusieurs mesures.
- Les segmentations en accords sont le format privilégié pour les méthodes de transcription automatique. Elles prennent la forme de progressions d'accords pour lesquelles on a indiqué pour chaque accord son temps de début et de fin par rapport à un enregistrement précis. Avec ce type de représentation, on peut distinguer un accord tenu longtemps d'un accord tenu brièvement, mais l'assignation d'une durée musicale à la durée temporelle indiquée est ambiguë.
- Les grilles d'accords sont une représentation schématique de l'harmonie, pour laquelle on sélectionne une échelle discrète de description (souvent la mesure, parfois le temps), et on assigne à chaque unité temporelle l'accord principal qui est tenu sur cette unité. Cette représentation combine la modélisation de la durée des accords avec une discrétisation pertinente musicalement, mais elle peut s'avérer insuffisamment précise si l'échelle de description sélectionnée est trop grande, ou redondante si l'échelle est petite.

Pour notre application, la forme synchrone des grilles d'accords en font un candidat privilégié. Nous retenons l'échelle d'un accord par temps : même si la représentation obtenue est redondante, les algorithmes que nous utilisons ne sont affectés par cette redondance qu'au niveau des temps de calcul. Le choix d'une échelle encore plus courte permettrait une description encore plus précise, mais les cas où cette précision est utile ne sont pas suffisamment fréquents pour justifier l'augmentation des temps de calcul qui l'accompagneraient.

La transcription automatique d'accords est un problème très étudié en MIR. Une des conséquences de la popularité de cette tâche est que les annotations d'accords sont un des types d'annotations les plus répandus. Ainsi, [CHO, 2011] met à disposition des annotations d'accords pour RWC-Pop. Toutefois, ces annotations prennent la forme d'une segmentation en accords.

Une conversion vers une représentation de type grille est nécessaire pour les utiliser avec les algorithmes décrits dans cette thèse.

Afin d’obtenir des séquences à l’échelle d’un accord par temps, les annotations d’intervalles temporels ont été croisées avec les annotations de temps fournies avec RWC-Pop [GOTO, 2006]. Dans une première passe, les intervalles pour chaque accord ont été arrondis au temps le plus proche. Pour chaque temps, une valeur d’accord a ensuite été sélectionnée, en ignorant les éventuels intervalles vides (obtenus si les instants de début et de fin d’un accord sont assignés au même temps).

Les séquences ainsi obtenues constituent une première base, que nous désignons AUTO. On peut cependant noter que le processus de conversion ne produit pas toujours la même séquence pour des passages comportant des changements d’accords à contre-temps, en raison des arrondis. Une passe de correction manuelle a été effectuée pour homogénéiser les inexactitudes introduites par la quantification. Nous désignons cette base corrigée CORRECTED.

Une révision supplémentaire de l’annotation a été effectuée par Frédéric Bimbot. Cette révision se focalise principalement sur la cohérence des annotations entre les différentes occurrences d’une même classe de segments. Ainsi, lorsqu’un accord ambigu est présent dans plusieurs segments d’une même classe, l’ambiguïté est systématiquement résolue de la même façon pour chacune des occurrences. Cette nouvelle annotation apporte également quelques corrections ponctuelles sur la segmentation des morceaux. Nous dénotons cette base MANUAL.

Un exemple d’annotation est présenté en figure 3.8.

```

Abm Abm Abm Abm F# F# F# F# E E E E F# F# F# F# F# F# F# F# Eb Eb Eb Eb Abm Abm
Abm Abm E E E E B B B B Eb Eb Eb Eb Abm Abm Abm Abm E E E E F# F# Eb Eb Abm Abm
Abm Abm F# F# F# F# C# C# C# C# E E F# F# Abm Abm Abm Abm F# F# F# F# C# C# C#
C# E E F# F# Cm Cm Cm Cm Cm Cm Cm F Cm Cm Cm Cm Cm Cm Cm Bb Cm Cm Cm Cm Cm Cm Cm
F Ab Ab Bb Bb Cm Cm Cm Cm Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm Cm Cm Cm Ab Ab Ab
Ab Eb Eb Eb Eb G G G G Eb Eb Eb Eb Abm Abm Abm Abm E E E E B B B B Eb Eb Eb Eb
Abm Abm Abm Abm E E E E F# F# Eb Eb Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E
F# F# Cm Cm Cm Cm Cm Cm Cm F Cm Cm Cm Cm Cm Cm Cm Bb Cm Cm Cm Cm Cm Cm Cm F Ab
Ab Bb Bb Cm Cm Cm Cm Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm Cm Cm Cm Ab Ab Ab Ab Eb
Eb Eb Eb G G G G Eb Eb Eb Eb Abm Abm Abm Abm E E E E B B B B Eb Eb Eb Eb Abm Abm
Abm Abm E E E E F# F# Eb Eb Abm Abm Abm Abm Abm Abm E Eb Ebm Ebm Ebm Ebm Ebm Ebm
Ebm Ab Ebm Ebm Ebm Ebm Ebm Ebm Ebm C# Ebm Ebm Ebm Ebm Ebm Ebm Ebm Ab C# C# F# F#
B B E E A A D D Eb Eb Eb Eb Eb Eb Eb Eb Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm Cm Cm
Cm Ab Ab Ab Ab Eb Eb Eb Eb G G G G Eb Eb Eb Eb Abm Abm Abm Abm E E E E B B B B
Eb Eb Eb Eb Abm Abm Abm Abm E E E E F# F# Eb Eb Abm Abm Abm Abm E E E E B B B B
Eb Eb Eb Eb Abm Abm Abm Abm E E E E F# F# Eb Eb Abm Abm Abm Abm F# F# F# F# C#
C# C# C# E E F# F# Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E F# F# F# F# Abm N
N N

```

FIGURE 3.8 – Grille d’accords correspondant au morceau RWC-Pop 001 pour la base MANUAL.

3.3.2 Sélection de la règle pour IRR

Un premier degré de liberté dans la méthode décrite dans ce chapitre se situe dans le choix de la règle à ajouter à chaque itération de l’algorithme IRR.

On considère 3 stratégies simples de sélection :

ML MAXIMUM-LENGTH, qui sélectionne la répétition la plus longue, quel que soit son nombre de répétitions (au moins 2 occurrences).

MF MAXIMUM-FREQUENCY, qui sélectionne la paire de symboles la plus fréquente.

MR MAXIMUM-REDUCTION, qui sélectionne à chaque étape la séquence qui réduirait le plus la taille de la grammaire si une règle correspondante était ajoutée.

Si $\text{occ}(\alpha)$ est le nombre d'occurrences dans la grammaire courante de α et $|\alpha|$ sa longueur, on a alors

$$\begin{aligned}\hat{\alpha}_{ML} &= \arg \max_{\text{occ}(\alpha) > 1} |\alpha| \\ \hat{\alpha}_{MF} &= \arg \max_{|\alpha|=2} \text{occ}(\alpha) \\ \hat{\alpha}_{MR} &= \arg \max_{\alpha} (\text{occ}(\alpha) - 1) (|\alpha| - 1)\end{aligned}$$

où $\hat{\alpha}_{ML}, \hat{\alpha}_{MF}, \hat{\alpha}_{MR}$ sont les séquences sélectionnées respectivement par MAXIMUM-LENGTH, MAXIMUM-FREQUENCY et MAXIMUM-REDUCTION.

On peut noter que plusieurs séquences peuvent obtenir le même score, auquel cas les séquences sont départagées de façon arbitraire⁶.

Le critère MF tend à faire apparaître des règles qui sont rendues coûteuses par l'introduction de règles ultérieures. Afin d'éliminer cette tendance qui ne se produit pas avec les critères ML et MR, on examine également l'algorithme MFC, qui consiste à insérer une étape de CLEANUP après chaque itération de MF.

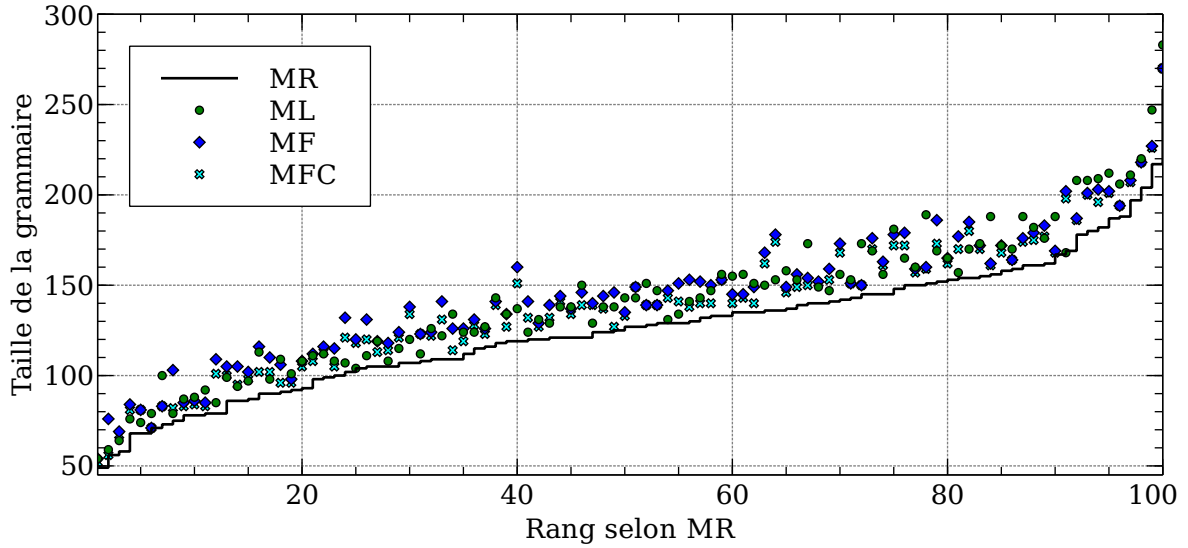
La figure 3.9 et la table 3.1 donnent les longueurs des grammaires finales pour chaque version d'IRR. Comme la figure le montre, la variance élevée est principalement due aux différences entre les morceaux. Afin d'isoler la part de la variance liée au choix de l'algorithme, nous avons choisi de considérer les différences de longueur par rapport à MR plutôt que les longueurs absolues.

Données testées \ Version	MR	ML	MF	MFC
AUTO	126.5 (± 36.3)	+12.6% ($\pm 6.0\%$)	+14.6% ($\pm 8.0\%$)	+10.2% ($\pm 5.9\%$)
MANUAL	95.8 (± 25.0)	+14.5% ($\pm 7.0\%$)	+26% ($\pm 9.0\%$)	+13.3% ($\pm 8.4\%$)

TABLE 3.1 – Tailles moyennes (et écarts-types) des grammaires en fin d'exécution. Les valeurs pour la version MR considèrent la taille absolue, les autres sont données relativement à MR pour isoler l'effet dû à l'algorithme.

6. Dans le cas de notre implémentation, chaque classe de symbole se voit assigner un identifiant numérique, négatif pour les terminaux, positif pour les non-terminaux, de valeur absolue croissante en fonction de leur ordre d'apparition. Cet identifiant est utilisé pour définir un ordre lexicographique sur les séquences, et la première selon cet ordre est utilisée en cas d'égalité.

(a) Tailles pour le corpus AUTO.



(b) Tailles pour le corpus MANUAL.

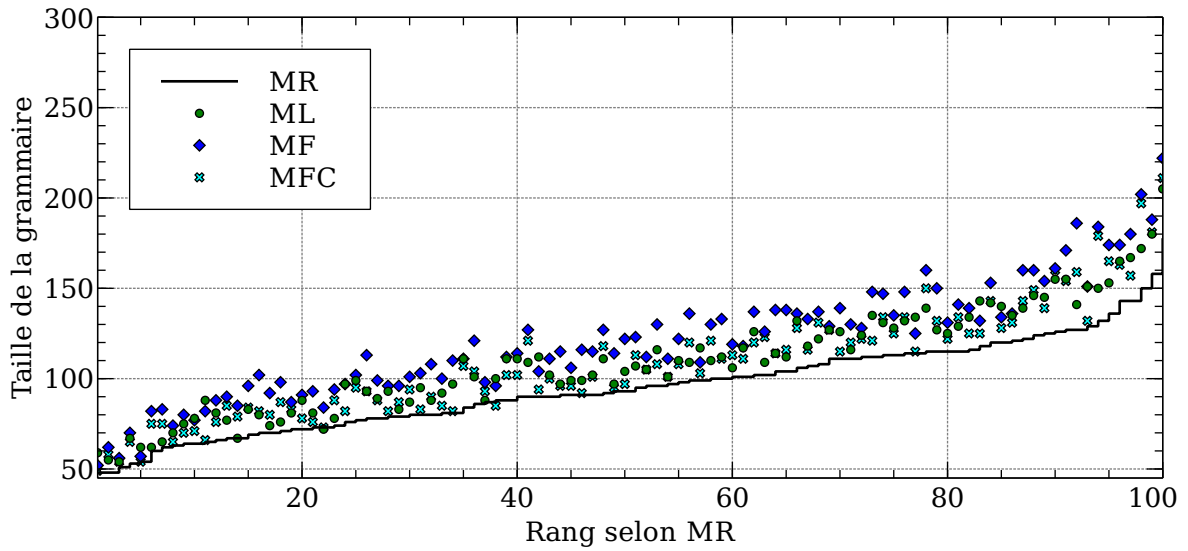


FIGURE 3.9 – Tailles des GDU obtenues selon la fonction de sélection de règle pour les 100 morceaux de RWC-Pop. Les morceaux ont été réordonnés par ordre croissant de taille selon la méthode MR afin d’améliorer la lisibilité.

Un premier constat, qui confirme les résultats de Gallé, est que la version MR produit les grammaires les plus courtes : seule la version MFC produit ponctuellement des grammaires plus courtes. Ces résultats ponctuels laissent entrevoir la possibilité d'autres versions plus efficaces. On pourrait ainsi envisager des stratégies de remplacement moins gloutonnes, qui pénaliseraient l'ajout de règles qui rentrent en conflit avec d'autres règles potentielles.

On peut toutefois noter que la version ML reste relativement proche (+12.6%, du même ordre que les versions MF et MFC) de la version MR, contrairement aux expériences de Gallé qui la plaçaient à +25% sur des séquences de texte et +55% sur des séquences ADN [GALLÉ, 2011, p. 36]. Une explication possible de cette différence est la longueur bien plus faible des séquences, qui donne moins d'importance à des règles courtes mais fréquentes.

3.3.3 Combinaison des algorithmes

Comme mentionné précédemment, chacun des algorithmes IRR, MGP et CLEANUP correspond à un type d'opération incrémentale sur la grammaire, respectivement l'ajout, la modification et la suppression de règles.

La modification et la suppression de règle supposent qu'il existe des règles à modifier ou supprimer, ce qui, pour une initialisation par la grammaire triviale ne comportant que son axiome, impose la présence de l'algorithme IRR en tête de la chaîne d'exécution, mais il s'agit de la seule contrainte stricte sur l'enchaînement de ces algorithmes.

L'exécution de chacun des algorithmes résulte en une réduction de la taille de la grammaire⁷, ce qui garantit que tous les enchaînements finissent par converger. Toutefois, les minima locaux atteints par divers enchaînements peuvent différer.

La complexité calculatoire de ces algorithmes varie fortement. Typiquement, sur des séquences musicales et avec les optimisations décrites en section 3.2.1, une exécution de MGP est plus longue qu'une itération d'IRR d'un ordre de grandeur, qui à son tour est plus longue qu'une exécution de CLEANUP de plusieurs ordres de grandeur.

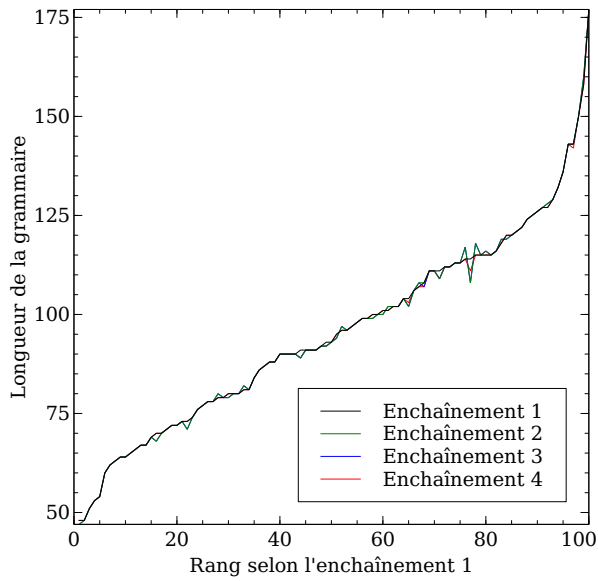
Comme nous l'avons mentionné dans la section 3.2.4, Gallé a examiné sur des séquences massives 4 boucles d'enchaînement des algorithmes : (i) IRR seul, (ii) IRR puis MGP à chaque itération (IRR+MGP), (iii) IRR puis MGP et CLEANUP à chaque itération (IRR+MGP+C), et (iv) une alternance de IRR jusqu'à convergence et de MGP et CLEANUP (IRRMGP*). Ces enchaînements sont détaillés sous forme d'algorigrammes dans la figure 3.11.

La figure 3.10 et la table 3.2 présentent les résultats obtenus pour les différentes méthodes sur les corpus AUTO et MANUAL.

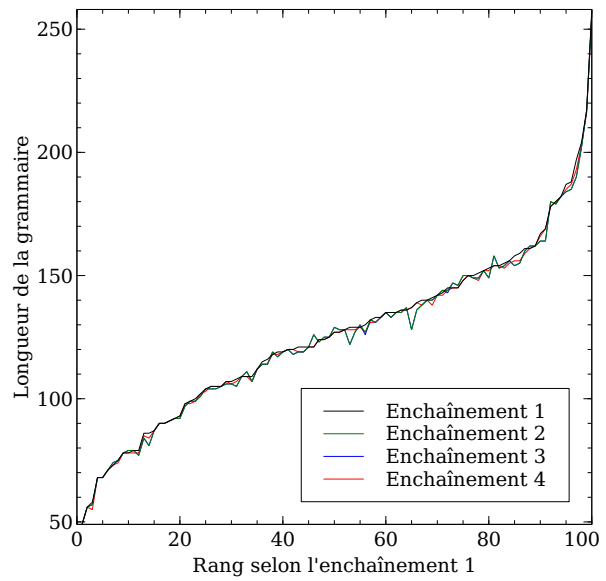
On remarque que l'ajout des algorithmes MGP et Cleanup dans l'optimisation de la longueur de la grammaire a un effet très faible, en particulier sur la base MANUAL. Par construction, IRRMGP* ne produit jamais une grammaire plus longue que IRR⁸, mais ne raccourcit pas la grammaire dans plus de la moitié des cas. L'enchaînement 3 ne diffère de l'enchaînement 2 que

7. Cette réduction n'est pas stricte, cependant IRR et CLEANUP ne modifient pas la grammaire si la réduction est nulle, et l'application répétée de MGP ne change pas son résultat.

8. IRRMGP* commence par une exécution complète d'IRR et ne peut que réduire la taille de la grammaire par la suite.



(a) Longueurs des grammaires sur MANUAL



(b) Longueurs des grammaires sur AUTO

FIGURE 3.10 – Taille des GDU obtenues selon la méthode pour les 100 morceaux de RWC-Pop. Les morceaux ont été réordonnés par ordre croissant de taille selon IRR seul afin d'améliorer la lisibilité.

Enchaînement	1	2	3	4
Données testées				
AUTO	126.5 (± 36.0)	-0.6% ($\pm 1.5\%$)	-0.6% ($\pm 1.5\%$)	-0.5% ($\pm 0.8\%$)
MANUAL	95.8 (± 25.0)	-0.1% ($\pm 0.9\%$)	-0.1% ($\pm 0.9\%$)	-0.1% ($\pm 0.4\%$)

TABLE 3.2 – Comparaison des tailles des grammaires inférées différents enchaînements des algorithmes. Les tailles de grammaires sont données relativement à la méthode 1 (IRR seul) sauf la méthode 1 qui est donnée en longueur absolue (intervalles de 1σ).

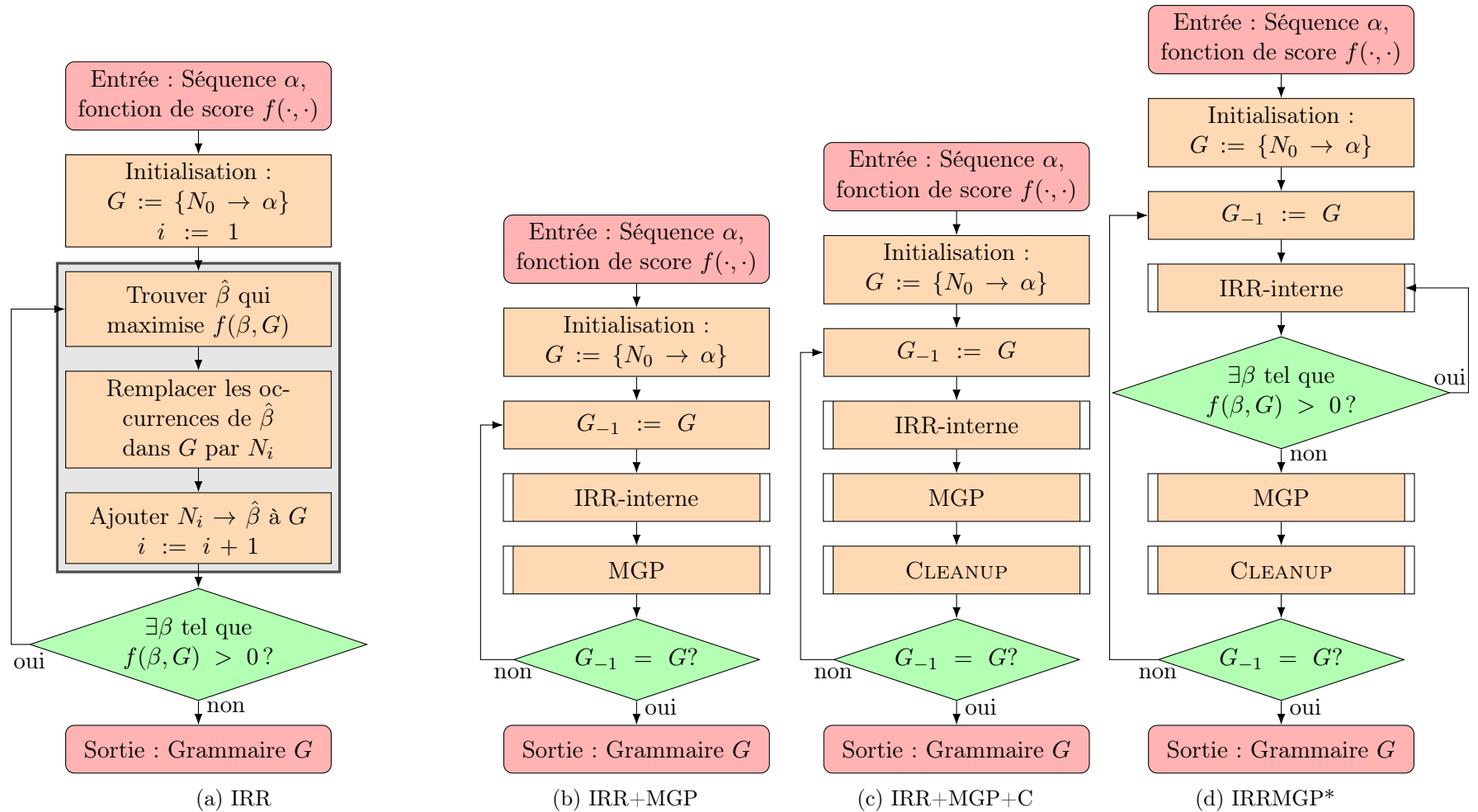


FIGURE 3.11 – Algorithmes correspondant aux enchaînements d'opérations considérés

sur quelques séquences, et ne raccourcit alors la longueur de la grammaire que d'un symbole. Ces résultats contrastent avec ceux de Gallé [GALLÉ, 2011, pp. 78-82], qui trouve une réduction de la taille des grammaires de l'ordre de 4 à 10% sur les séquences ADN.

Cette différence d'efficacité de l'algorithme MGP pour réduire la taille de la grammaire peut s'expliquer par plusieurs facteurs :

- en raison de la faible longueur des séquences d'accords, le nombre de règles introduites reste faible, ce qui limite fortement le nombre de décompositions possibles : on retrouve également une tendance dans les résultats de Gallé vers une réduction plus importante sur les séquences plus longues ;
- le nombre de symboles terminaux est plus élevé que pour l'ADN : les répétitions non-pertinentes sont moins fréquentes, réduisant encore le nombre de décompositions possibles ;
- les séquences d'accords sont très denses en répétitions, ce qui donne un poids très important à l'algorithme IRR : une ré-écriture utilisant seulement les règles déjà sélectionnées ne permet généralement pas de sortir des optima locaux atteints par IRR.

L'influence infime de l'algorithme CLEANUP est attribuable en grande partie au faible nombre de changements dans la grammaire lors de l'étape MGP. De plus, l'optimisation effectuée par MGP ne prend pas en compte une éventuelle suppression des règles non-utilisées.

3.3.4 Application à la découverte de structure dans du texte

Avant de procéder à l'évaluation de ces algorithmes pour l'inférence de structure sur des séquences musicales, nous procédons à une étude préliminaire sur du texte.

L'objectif de cette expérience est de jauger la capacité de ces algorithmes à inférer des éléments structurels dans un texte en français. Les GDU ne permettent aucune généralisation, ce qui restreint fortement le cadre de la structure que l'on peut attendre de voir émerger d'une analyse sur du texte. Ainsi, malgré leur nom de grammaires, les GDU ne permettent pas de modéliser les processus grammaticaux qui ont lieu dans le langage naturel, car les notions abstraites de classes grammaticales ne font pas partie de ce qui peut être modélisé par les GDU. *A fortiori*, la structure sémantique, qui comporte entre autres le regroupement des phrases en paragraphes et en chapitres, dépasse largement le cadre de cette découverte de structure.

Les éléments structurels qui nous intéressent ici sont les structures plus rigides, qui sont réutilisées à de multiples reprises au fil du corpus. Parmi ces structures, on peut notamment distinguer les morphèmes, les mots et les locutions courantes.

Pour cette expérience, nous nous basons sur le texte brut de *Notre-Dame de Paris*, par Victor Hugo. Cette œuvre se situe dans le domaine public, et le texte intégral peut être obtenu en ligne, par exemple sur wikisource⁹. Le texte a été pré-traité afin de retirer toute trace de la structure existante :

- les titres de chapitres ont été retirés,
- les chapitres et paragraphes ont été concaténés,
- la ponctuation et les espaces ont été supprimées,

9. https://fr.wikisource.org/wiki/Notre-Dame_de_Paris

- la casse des lettres a été homogénéisée de sorte à ce que tous les caractères soient en minuscules ;
- les accents et autres diacritiques ont toutefois été préservés.

Le résultat de ce pré-traitement est une chaîne ininterrompue de 799447 caractères alphabétiques.

Nous avons ensuite appliqué sur cette chaîne l'algorithme IRRMGP*, décrit dans la section précédente, qui a produit une GDU correspondante totalisant 24471 règles. La reproduction intégrale de cette GDU dans ce document est inenvisageable, mais la figure 3.12 en présente deux extraits : le premier correspond aux 50 premières¹⁰ règles de la grammaire, le deuxième montre les 10001^e à 10050^e règles¹¹.

Comme attendu, les règles obtenues correspondent à des séquences relativement courtes : la longueur médiane de la chaîne constitutive d'une règle est de 7 caractères, et la règle la plus longue correspond à une chaîne de 75 caractères (une citation dans le texte) :

`unespritpassadevantmafaceetjentendisunpetitsouffleetclepoildemachairsehériss`

Il s'agit cependant d'un exemple extrême car seules 21 règles (0.1%) dépassent 30 caractères, comme le montre la figure 3.13. La faible fréquence de ces règles longues reflète le fait que des phrases entières sont rarement répétées à l'identique. L'absence de structures de haut niveau se constate également dans la longueur de l'axiome, qui compte 121070 symboles.

On peut remarquer que les premières règles correspondent à des paires, voire des triplets de lettres qui sont fréquemment co-occurentes en français, tandis que les règles du reste de la grammaire correspondent souvent à des mots ou des groupements de mots. Bien que peu représentés dans les extraits, on trouve aussi des groupements qui correspondent à des radicaux ou à des terminaisons.

Une tendance ressort en particulier de l'examen de cette GDU : les mots ou groupes de mots sont occasionnellement amputés ou augmentés d'un faible nombre de caractères, comme `[s]ubitement` (règle 10531) ou `ogiv[e]` (règle 10524). L'origine de ces troncations peut souvent être retracées aux premières règles.

On peut pour s'en convaincre étudier de façon simplifiée l'exemple de `[s]ubitement`. Le mot *subitement* apparaît dans une multitude de contextes dans le texte brut, dont entre autres

`...changeasubitementlecours...`
`...comptesubitementdeces...`
`...souvritsubitementetsereferma...`
`...larefermasubitementcommeépouvantée...`

10. L'ordre indiqué pour les règles fait ici référence à l'ordre d'introduction des règles, car les règles ne sont pas formellement ordonnées dans la grammaire.

11. On peut noter que les numéros attribués aux règles ne correspondent pas exactement à la plage indiquée, et que certains (10509 par exemple) semblent manquer. Ce décalage correspond à des règles qui ont été introduites, mais ont été ultérieurement retirées par l'algorithme CLEANUP.

1 -> e s	e s		
2 -> e n	e n	10493-> 670 3940	d u s i è c l e
3 -> a i	a i	10494-> 20778 e	h a u t a i n e
4 -> o u	o u	10495-> 12366 e	d u c i e
5 -> l e	l e	10496-> 678 1384	d u c a p i t
6 -> r e	r e	10497-> 10495 194	d u c i e l e t
7 -> d e	d e	10498-> 681 13	a j o u t e r
8 -> q u	q u	10499-> 5122 15781	a j o u t a t i l
9 -> l a	l a	10500-> 682 5	i m p l e
10-> o n	o n	10501-> 682 93	i m p o s
11-> a n	a n	10502-> 7320 e	i m p o r t e
12-> e t	e t	10503-> 690 32 22330	f a i s c e q u e t u v o u d r a s
13-> e r	e r	10504-> 9661 527	u n e p l a c e
14-> 3 t	a i t	10505-> 694 2687	a v e c u n g e s t e
15-> i l	i l	10506-> 694 3131	a v e c u n a c c e n t
16-> 2 t	e n t	10507-> 2199 t	r e g a r d e t
17-> u n	u n	10508-> 3017 250	e t d e m e u r
18-> i s	i s	10510-> 2551 e	v i v a n t e
19-> i t	i t	10511-> 706 79	a u p o n t
20-> a r	a r	10512-> 706 334	a u p i è r e
21-> i n	i n	10513-> 10512 s	a u p i è r e s
22-> a u	a u	10514-> 706 1980	a u p i l o r i
23-> c h	c h	10515-> 706 3151	a u p o i n g
24-> e l	e l	10516-> 710 79	q u i l s o n t
25-> u r	u r	10518-> 20849 146 39	r e n d e z l a m o i
26-> 5 s	l e s	10519-> 3371 141	r e n d u e
27-> t e	t e	10520-> 20851 775 3	v o u s e x p l i q u e r a i
28-> a s	a s	10522-> 715 7341 s 5916	d e s p o i g n é e s d e c h e v e u x
29-> o r	o r	10523-> 3735 s	d e l a p o r t e s
30-> 4 r	o u r	10524-> 727 155	o g i v
31-> o m	o m	10525-> 5867 406	r o u e s
32-> c e	c e	10526-> 20881 19	r o u g i t
33-> 7 s	d e s	10527-> 729 11	r é p a n
34-> a v	a v	10528-> 729 6097	r é p o n s e s
35-> d 173	d a n s	10529-> 2741 1104	d a n s l e c o e u r
36-> u s	u s	10530-> 735 34 915 2835	d e l a s a v a t e r i e
37-> 9862 19	é t a i t	10531-> 14016 58	u b i t e m e n t
38-> 11 t	a n t	10532-> 739 89	i c i p h o e b u s
39-> o i	o i	10533-> 5639 é	q u i t t é
40-> p r	p r	10534-> 14018 3284 308	q u i t i e n t l e m i l i e u e n t r e
41-> 8 e	q u e	10535-> 741 325	c e q u i f a i s a i t
42-> s e	s e	10536-> 742 m	f o u r m
43-> i e	i e	10537-> 7340 é	f o u r r é
44-> m e	m e	10538-> 747 598	a r t è r e s
45-> t r	t r	10539-> 753 388	e n c a d
46-> 4 s	o u s	10540-> 754 255	q u e l e j o u r
47-> 3 s	a i s	10541-> 15975 15833 14262	n i c o l a s f l a m e l
48-> 1 t	e s t	10542-> 6747 2110	g a u c h e e t
49-> 3549 r	p a r	10543-> 761 526	m i l l e a u t r e s
50-> 17 e	u n e	10544-> 762 89	r é p o n d i t p h o e b u s
		10546-> 765 38	e n t e n d a n t

(a) 50 premières règles de la gram-
maire

(b) 50 règles du milieu de la grammaire

FIGURE 3.12 – Extraits de la grammaire issue de l'exécution de IRRMGP* sur le texte pré-traité de *Notre-Dame de Paris*. Les règles sont accompagnées des séquences constitutives correspondantes.

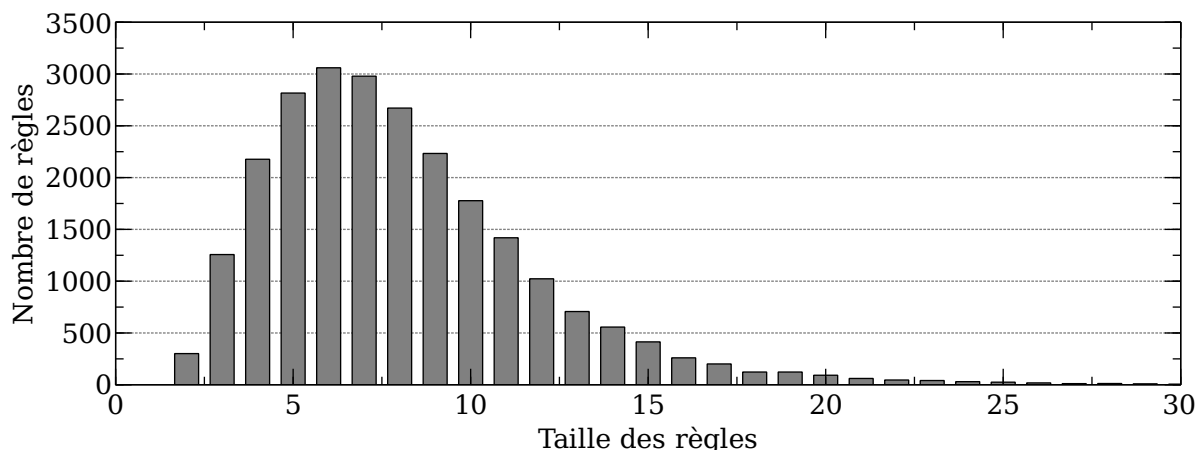


FIGURE 3.13 – Répartition de la taille des chaînes constitutives des règles de la grammaire inférée. Les 21 règles ayant une chaîne constitutive de plus de 30 caractères ne sont pas représentées.

Une fois le remplacement des règles 1 (*es*) et 28 (*as*) effectués, ces contextes deviennent

```

...change28ubitementle cours...
...compt1ubitementdec1...
...souvritsubitementetsereferma...
...lareferm28ubitementcommeépouvantée...

```

Ces contextes peuvent donner lieu à l’ajout de règles correspondant à *28ubitement* ou *ubitement*, mais *subitement*, apparaissant moins fréquemment, n’est pas retenu.

Si le mot apparaît dans un nombre suffisant de contextes différents, une règle peut être ultérieurement ajoutée. Ainsi, dans la grammaire finale, on trouve des règles correspondant à *asubitement*, *subitement* et *ubitement*. Ces trois règles, dont deux sont superflues, sont toutes conservées car elles sont localement stables par rapport aux algorithmes MGP et CLEANUP, pris individuellement : MGP utilise les règles superflues qui permettent occasionnellement un chemin marginalement plus court, et CLEANUP conserve ces règles car elles sont suffisamment utilisées.

Malgré quelques erreurs de ce type, les résultats illustrent que l’algorithme IRRMGP* présente une certaine efficacité pour l’inférence de structure dans des données discrètes. Il faut toutefois noter que l’application à un texte long est un cas favorable à ce type d’approche, à plusieurs égards :

- La quantité de données : avec plus de 170000 mots, *Notre Dame de Paris* permet d’observer de nombreuses répétitions des unités structurales recherchées, même si le vocabulaire est large ; on peut opposer cela à la longueur typique d’une chanson, qui ne permet d’observer les phrases ou sections musicales qu’un nombre très limité de fois (même si ces répétitions sont plus denses).
- La qualité des répétitions : même si les noms, adjectifs et verbes sont sujets à des inflexions, l’orthographe d’un même mot ou d’une même locution reste pratiquement constante sur l’ensemble du texte. Les variations autour d’une même racine peuvent être considérées comme des unités différentes, et sont dans la plupart des cas récurrentes. À l’inverse, dans la musique, deux sections homologues ne sont souvent pas identiques, mais seulement similaires. De plus, les variations observées sont souvent uniques.

- La variété des contextes : les unités structurelles en matière de texte sont relativement indépendantes, de telle sorte que chaque mot et chaque locution est utilisée dans des contextes variés, ce qui permet de les identifier et de les distinguer au sein des éléments de niveau supérieur. Les unités structurelles pour la musique, en plus d’avoir un nombre limité d’occurrences, sont souvent encapsulées dans les mêmes sections ou développements.

3.3.5 Estimation de la structure linéaire

La tâche la plus utilisée pour évaluer la performance d’algorithme de découverte de structure musicale est la segmentation linéaire du morceau. Pour cette tâche, on cherche à produire des estimations de la structure musicale qui divisent le morceau considéré en segments sans recouvrement.

La campagne d’évaluation MIREX conduit ainsi annuellement une comparaison d’algorithmes sur cette tâche. Les résultats de ces évaluations fournissent un point de référence utile à titre indicatif, mais il n’est pas possible de les comparer directement aux résultats présentés ici, pour plusieurs raisons :

- les estimations sont effectuées à partir des enregistrements audio des chansons et non d’annotations symboliques des accords ; cela signifie d’une part que beaucoup plus d’information est présente¹², mais d’autre part son extraction est souvent imparfaite, ce qui introduit des perturbations qui ne sont pas aussi présentes dans les annotations manuelles.
- les segmentations sont exprimées sur une échelle de temps continue et non discrète ; les métriques d’évaluation décrites en section 3.3.5.2 sont calibrées pour être similaires aux métriques en temps absolu, mais peuvent différer selon le tempo de la musique.

Annotations structurelles linéaires Les annotations utilisées sont celles mises à disposition par [BIMBOT, SARGENT et al., 2014] pour RWC-Pop. Ces annotations ont été produites selon une méthodologie qui reflète la structure sémiotique, selon une définition proche de celle présentée au chapitre 2.

Comme les annotations d’accords mentionnées précédemment, ces données sont présentées sous le format d’intervalles de temps absolus. Les frontières des segments ont donc été adaptées de la même façon que les séquences d’accords : selon la même conversion automatique pour la base AUTO, selon une correction manuelle pour la base CORRECTED et elles ont été éventuellement retouchées lors de l’annotation pour la base MANUAL.

Le format retenu indique pour chaque segment son temps musical de début (voir figure 3.14 pour un exemple). Le début du premier segment étant systématiquement le premier symbole de la séquence, il est ignoré lors des comparaisons pour éviter une surévaluation systématique des performances.

1 25 53 85 117 149 177 193 225 257 293 337 369 397 425 459

FIGURE 3.14 – Annotation des frontières pour le morceau RWC-Pop001 (version MANUAL)

12. Le timbre, la dynamique et la mélodie notamment contiennent des indices très pertinents pour l’inférence de structure musicale.

3.3.5.1 Conversion

La représentation de la structure musicale résultant de la méthode IRR-MGP ne correspond pas directement avec un format de structure linéaire. Une conversion est nécessaire avant de pouvoir comparer les résultats aux annotations de la base.

La méthode que nous avons retenue consiste à considérer comme frontières les limites de tous les nœuds de l'arbre dont la séquence constitutive dépasse une certaine taille. La figure 3.15 montre un exemple minimal de ce processus.

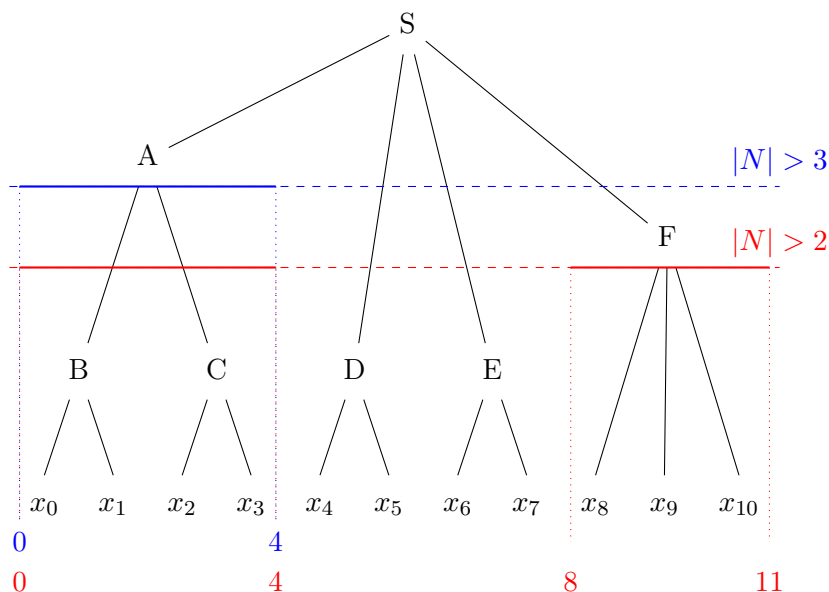


FIGURE 3.15 – Exemple du processus de conversion d'arbre structurel en segmentation linéaire. En rouge, la conversion avec un seuil de 2 symboles retient les non-terminaux A et F, pour des segments explicites de x_0 à x_3 et de x_8 à x_{10} et un segment implicite de x_4 à x_7 . En bleu celle obtenue avec un seuil de 3 symboles ne retient que le non-terminal A, ce qui correspond à un segment explicite de x_0 à x_3 et un segment implicite de x_4 à la fin de la séquence. En pratique, on ne fait pas de différence entre les segments explicites et implicites.

Le choix du seuil n'étant pas l'objet principal de l'évaluation, il a été effectué morceau par morceau par un oracle maximisant la F-mesure (définie dans la section suivante). Son introduction entraîne toutefois un biais à la hausse des F-mesures observées. La figure 3.16 reproduit le profil de la F-mesure pour quelques morceaux en fonction du seuil choisi.

3.3.5.2 Métriques

Pour les comparaisons de segmentation sur des signaux audio, les métriques principales utilisées sont les mesures de précision et de rappel sur les frontières et leur moyenne harmonique, la F-mesure. Les frontières entre segments étant définies sur des valeurs continues, ces notions sont définies de façon légèrement différente de leurs définitions conventionnelles en recherche d'information, de façon à inclure un seuil de tolérance.

La valeur centrale de cette définition est le nombre de frontières de segments correctement détectées. Une frontière entre deux segments de l'annotation de référence est définie comme

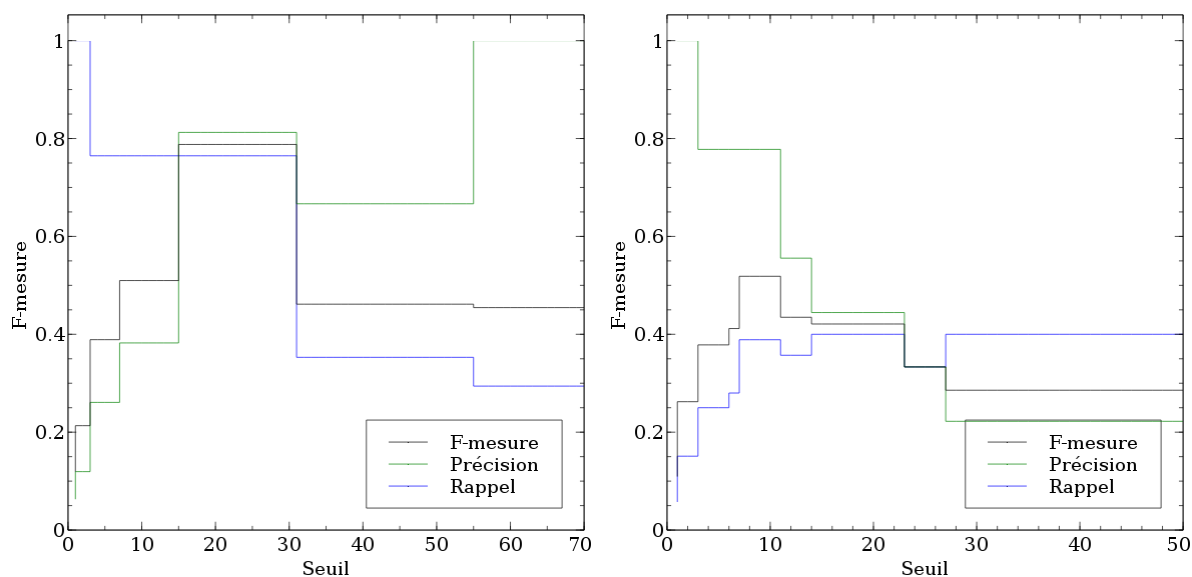


FIGURE 3.16 – Évolution de la F-mesure en fonction du seuil de segmentation pour les morceaux RWC Pop 026 et 065 (annotation MANUAL).

correctement détectée avec une fenêtre de tolérance de t secondes s’il existe une frontière dans la segmentation évaluée à moins de $t/2$ secondes.

Les notions de rappel, de précision et de F-mesure sont alors définies de façon classique :

$$R = \frac{\text{nombre de frontières correctement détectées}}{\text{nombre de frontières de la segmentation de référence}}$$

$$P = \frac{\text{nombre de frontières correctement détectées}}{\text{nombre de frontières de la segmentation évaluée}}$$

$$F = 2 \frac{R \times P}{R + P}$$

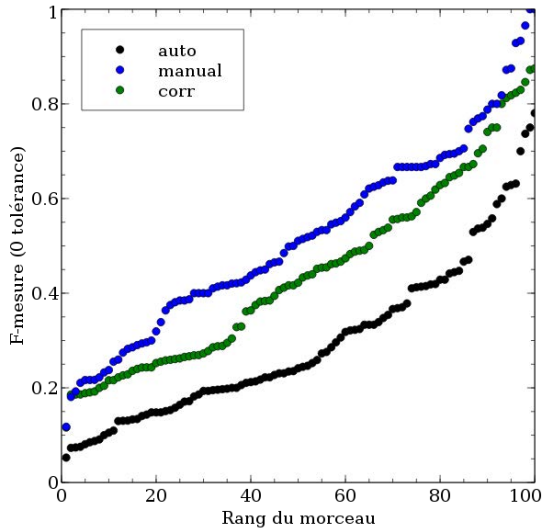
Ces mesures sont typiquement déclinées sous deux formes, avec des fenêtres de tolérance de 0.5 et 3 secondes.

Dans l’optique d’obtenir des résultats d’ordre comparable avec ceux présentés dans les travaux existants, nous avons opté pour ce type de définition bien que les estimations de frontières produites par nos algorithmes soient discrètes.

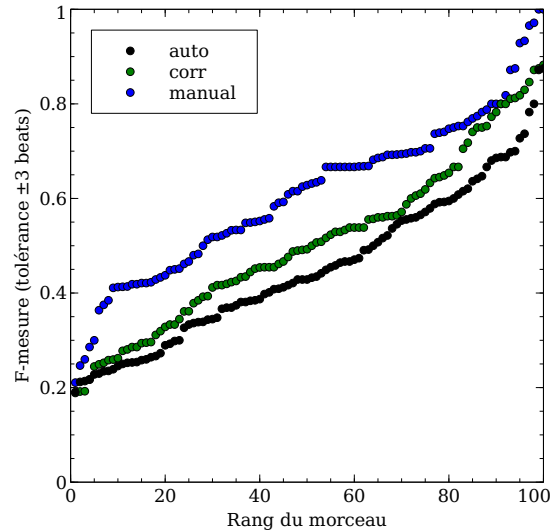
D’après les méta-données fournies avec RWC-Pop, les morceaux de la base ont un tempo moyen de 111 bpm. Afin d’obtenir des seuils de tolérance du même ordre que les 0.5 s et 3 s conventionnellement utilisés, on utilise ainsi une tolérance nulle (les 0.5 s de tolérance étant alors intégrées au processus de quantification des temps) et une tolérance de ± 3 temps.

3.3.5.3 Résultats

La figure 3.17 montre la distribution des F-mesures obtenues sur chacun des 3 ensembles d’annotations.



(a) F-mesure sans tolérance



(b) F-mesure avec tolérance

FIGURE 3.17 – F-mesure pour l’algorithme IRRMGP sur chacun des 100 morceaux de RWC-Pop selon l’annotation utilisée. Pour plus de lisibilité, les morceaux ont été ré-ordonnés par F-mesure croissante selon chaque annotation.

Ensemble d’annotations	AUTO	CORRECTED	MANUAL
F-mesure ± 3 beats	0.45	0.51	0.61
F-mesure sans tolérance	0.30	0.44	0.52

TABLE 3.3 – F-mesures moyennes avec l’algorithme IRRMGP.

Comme la longueur des grammaires en fin d’exécution, la F-mesure a une grande étendue sur l’ensemble du corpus. Certains morceaux sont très mal segmentés ($F=0.2$), tandis que quelques uns correspondent parfaitement avec l’annotation.

La F-mesure montre une forte progression avec les révisions successives des annotations, ce qui traduit une sensibilité élevée de la méthode au bruit ou aux variations sur les données d’entrée. En effet, une occurrence inexacte d’une répétition en raison d’une erreur dans l’annotation invalide l’ensemble de l’occurrence ; les répétitions longues (souvent pertinentes) ayant un nombre réduit d’occurrences sont alors plus difficilement sélectionnées par la phase IRR. On peut toutefois relever quelques morceaux pour lesquels la F-mesure diminue sur les annotations corrigées. Cela peut se produire si les corrections introduisent des répétitions ne correspondant pas à des segments musicaux et qui sont sélectionnées à la place de répétitions correspondant à des segments.

L’ajout d’une fenêtre de tolérance a un impact notable, quoique moindre sur les performances, en particulier sur les annotations corrigées manuellement, qui présentent une différence de moins de 10 points de pourcentage. Ces différences sont à mettre en contraste avec celles des campagnes MIREX, pour lesquelles la plupart des algorithmes obtiennent des écarts de 20 à 30 points de pourcentage entre la tolérance faible (0.5 s) et la tolérance moyenne (3 s).

3.3.6 Comparaison de structure hiérarchique

Idéalement, la correspondance entre les arbres inférés et la structure musicale hiérarchique serait évaluée en comparant les résultats de l’algorithme à des annotations manuelles arborescentes. Toutefois, deux éléments sont manquants pour une telle comparaison : des annotations complètes de la structure hiérarchique sémiotique, et une méthode de comparaison entre arbres adaptée à la représentation de la musique.

3.3.6.1 Disponibilité d’annotations

La production d’annotations de structure musicale représente un effort important, et il n’existe à notre connaissance pas d’ensemble d’annotations représentant la structure musicale de façon hiérarchique. On peut relever l’existence des annotations produites dans le cadre du projet SALAMI [SMITH et al., 2011], qui comportent deux échelles d’annotations, correspondant approximativement aux sections et aux phrases musicales. Toutefois, ces deux échelles ne suffisent pas pour pouvoir considérer ces annotations comme une représentation arborescente de la structure musicale avec laquelle effectuer des comparaisons.

Comme mentionné dans le chapitre précédent, ce problème de disponibilité d’annotations est amplifié par la multiplicité des définitions de la structure musicale. Cette ambiguïté s’accroît lorsque l’on s’éloigne des échelles couramment ciblées par les définitions. En effet, si les différentes définitions tendent à se rejoindre à l’échelle de la section musicale, les façons dont peuvent être regroupées ces sections en développements longs sont encore plus floues. Un problème similaire se pose en descendant à des échelles de descriptions plus détaillées.

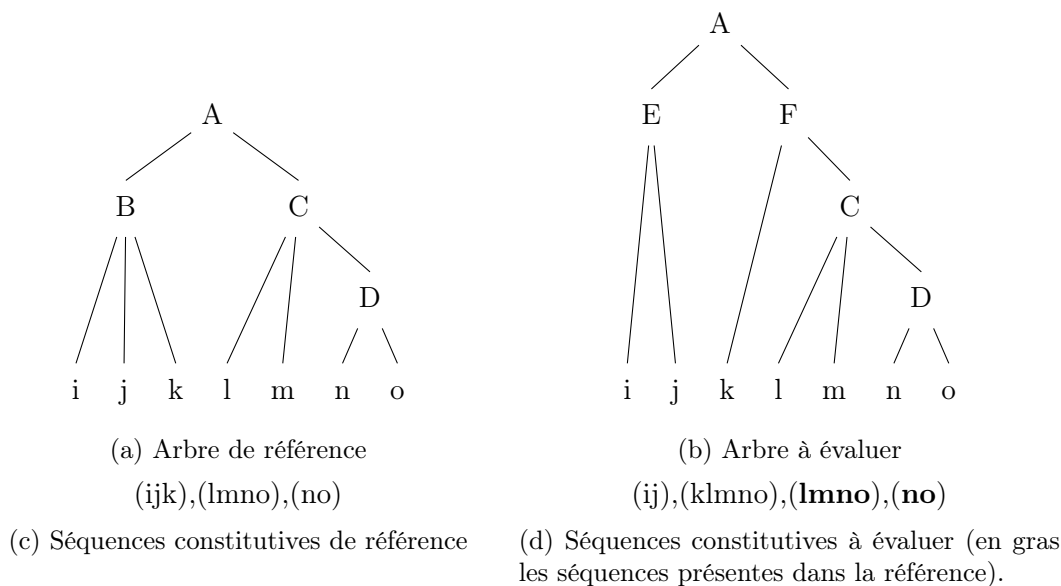
Afin de disposer de points de référence pour des comparaisons manuelles, nous avons produit des annotations complètes de structure hiérarchique pour un faible nombre de morceaux de la base RWC-Pop. Ces annotations ne peuvent toutefois être utilisées qu’à titre indicatif, par leur faible nombre d’une part, mais surtout par leur caractère *ad hoc*. Un important travail de formalisation et de définition de la structure musicale hiérarchique reste nécessaire afin de disposer de méthodologies non ambiguës pour la production d’annotations fiables.

3.3.6.2 Similarité de structure musicale hiérarchique

Le deuxième obstacle pour une évaluation directe de la structure musicale hiérarchique est l’existence de métriques adaptées. Comme le note Martin [MARTIN, 2012, p. 130], il n’existe à l’heure actuelle pas de méthodologie reconnue pour l’évaluation de représentations hiérarchiques de la structure musicale. On peut cependant trouver des méthodes de comparaison d’arbres, utilisées pour l’évaluation des analyses grammaticales en traitement des langues naturelles [BILLE, 2005 ; GALLÉ, 2010].

L’une des plus utilisées est la méthode PARSEVAL [KLAVANS et al., 1991]. Cette méthode consiste à comparer les nœuds des arbres à travers les chaînes constitutives correspondantes (voir exemple figure 3.18). Ces méthodes sont toutefois très limitées dans leur transposition à la structure hiérarchique musicale.

Ainsi, une des critiques soulevées contre la méthode PARSEVAL est son inadéquation lorsque les méthodologies d’annotation varient, même légèrement [REHBEIN et GENABITH, 2007], ce qui inclut le niveau de variation lié à l’ambiguïté de la structure musicale discutée dans la section précédente.



$$P = \frac{2}{4} = 0.5$$

$$R = \frac{2}{3} \approx 0.66$$

$$F = 2 \frac{PR}{P + R} \approx 0.57$$

FIGURE 3.18 – Exemple d'évaluation selon la méthode PARSEVAL.

La définition d'une erreur peut en outre varier fortement d'un domaine à l'autre : sur un regroupement d'éléments, un décalage d'un mot n'est pas équivalent à un décalage d'un temps ; le premier peut généralement être considéré strictement comme une erreur, alors que le second peut être considéré comme presque juste. On peut toutefois noter l'apparition récente de méthodes de comparaisons pour la structure musicale hiérarchique [MCFEE, NIETO et BELLO, 2015], mais elles n'ont pas été explorées dans le cadre de cette thèse.

En raison de ces difficultés, nous avons choisi pour l'évaluation quantitative de la structure inférée de nous limiter à la tâche de segmentation linéaire, pour laquelle des données et des méthodologies de référence existent déjà.

3.3.6.3 Évaluation qualitative

Comme nous venons de le voir, il est difficile d'évaluer formellement l'exactitude des structures hiérarchiques inférées par les algorithmes. Il reste cependant possible de procéder à un examen manuel des données, et d'identifier quels sont les problèmes principaux rencontrés.

La figure 3.19 juxtapose les arbres correspondants à une annotation manuelle et à l'inférence de la structure par IRRMGP* pour un même morceau. Comme sur le reste du corpus examiné, on peut observer visuellement un niveau de similarité relativement élevé. Dans une optique de diagnostic et afin de localiser les fortes densités d'erreurs, nous allons nous focaliser sur les points de divergence entre les structures annotées et les structures inférées.

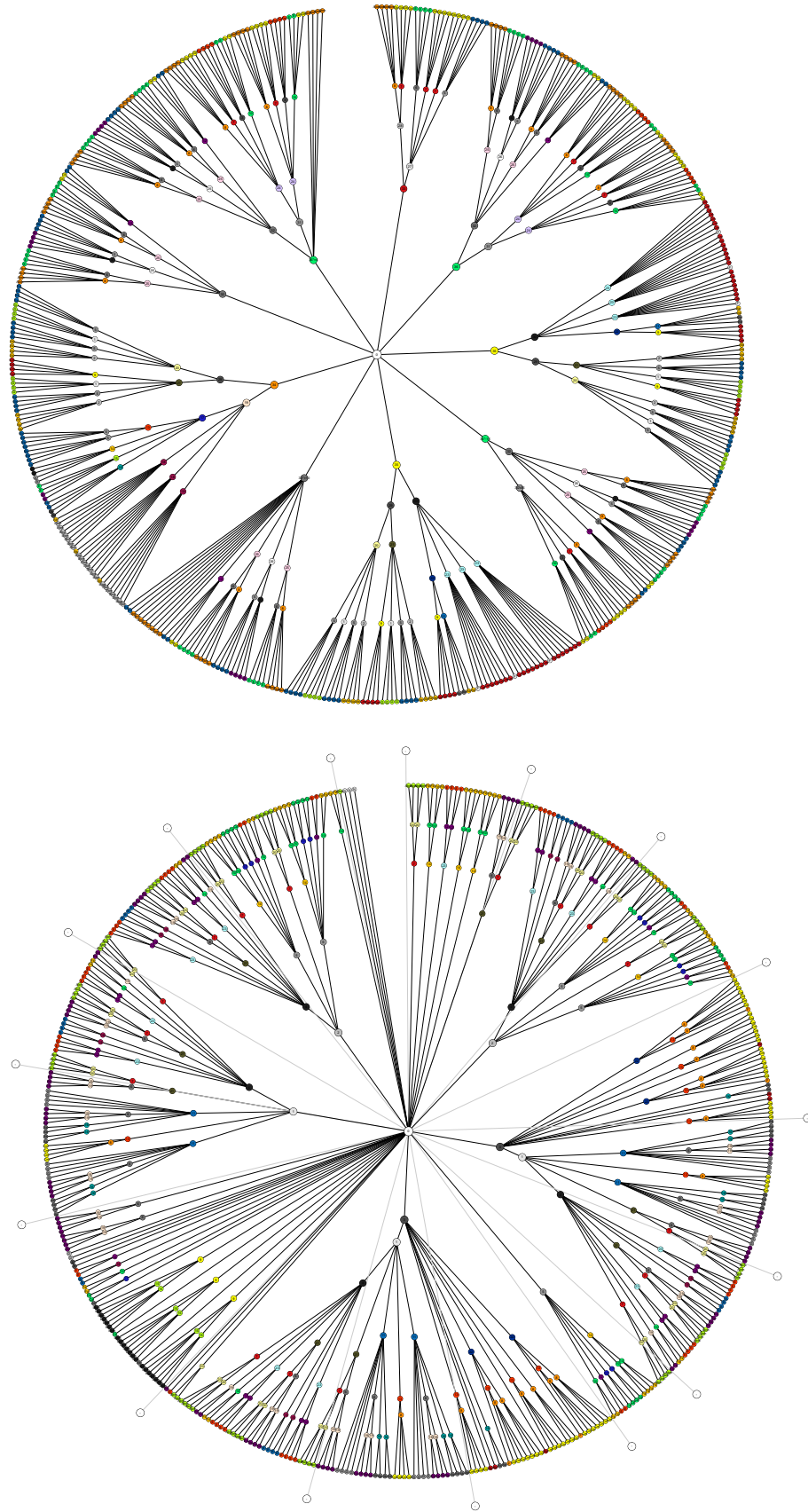


FIGURE 3.19 – Arbres correspondants à l’annotation manuelle (haut) et à l’inférence par IRRMGP* (bas) pour le morceau RWC-Pop 001

D'après l'étude manuelle des résultats sur les vingt premiers morceaux de la base RWC-Pop, plusieurs types d'erreurs récurrentes sont identifiables :

- l'absence totale de regroupement sur une partie du morceau,
- le décalage temporel d'une partie des regroupements,
- la fragmentation d'un segment,
- un manque de sous-divisions dans un regroupement.

Ces différents types d'erreurs peuvent être mis en relation avec les situations dans lesquelles elles ont lieu. En analysant ces situations, on peut mieux cerner où se situent les faiblesses de l'approche par grammaire à dérivation unique.

Segments non répétés Si la plupart des segments sont répétés plusieurs fois au cours d'un morceau, ce n'est pas le cas de tous. Une exception fréquente est le cas du solo qui possède souvent une progression harmonique qui lui est propre. La séquence d'accords, n'étant pas répétée, ne peut alors pas être factorisée par l'introduction d'une règle correspondante.

Phase d'une répétition cyclique Comme pour l'expérience précédente sur du texte, on observe occasionnellement que les regroupements effectués par l'algorithme sont en décalage par rapport aux segments attendus. Ces décalages sont fréquemment causés par la répétition bout-à-bout d'un même segment, accompagné d'un autre segment commençant ou se terminant de la même façon, ce qui crée une ambiguïté sur l'instant de début et de fin de la répétition. Par exemple, sur la séquence

... abcdefghabcdefghabcdefghabcdijkl ...

il y a une ambiguïté si la séquence répétée est **abcdefgh**, **efghabcd** ou une autre permutation circulaire.

Phase sur des séquences partageant un préfixe ou un suffixe Une variante du problème précédent peut se produire lorsque de nombreux segments commencent ou se terminent de la même façon. Cette situation se produit fréquemment (dans la majorité des 20 premiers morceaux de RWC-Pop, plus de la moitié des sections commencent par le même accord), mais ne conduit pas toujours à une erreur. Sur un exemple du type

... dabadacabadcdababadaccd ...

il peut être difficile de séparer les hypothèses

$$S \rightarrow \dots DBDCBdcDBBDccd \dots$$

$$B \rightarrow ba$$

$$C \rightarrow ca$$

$$D \rightarrow da$$

et

$$S \rightarrow \dots dBDCBDcdBBDCcd \dots$$

$$B \rightarrow ab$$

$$C \rightarrow ac$$

$$D \rightarrow ad$$

Un nombre limité d’occurrences de ces sections dans des contextes différents peut suffire à résoudre cette ambiguïté, mais une erreur précoce de ce type dans l’algorithme IRR peut causer des erreurs supplémentaires en cascade.

Sous-segments Il arrive fréquemment que la structure interne d’une section soit sous-segmentée. Ce manque de subdivisions peut s’expliquer par le faible nombre de contextes différents d’occurrence des subdivisions. Si l’on considère sur la séquence `aabbaaccdefgaabbaacc` un exemple simplifié :

$$\begin{aligned} S &\rightarrow AdefgA \\ A &\rightarrow BbbBcc \\ B &\rightarrow aa \end{aligned}$$

Une fois la séquence réduite, les séquences `bb` et `cc` ne sont plus répétées, et leur regroupement sous les règles correspondantes n’est plus souhaitable selon le critère de la longueur de la grammaire. Ces regroupements permettraient cependant de rendre la définition de la règle associée à `A` plus régulière, en accord avec les principes énoncés au chapitre 1.2.2.1.

Variations Les répétitions des sections musicales ne sont souvent pas effectuées à l’identique. Dans une forte proportion des cas, cette différence est masquée par la représentation sous la forme d’une séquence d’accords. Un nombre non-négligeable de variations se traduisent toutefois par une séquence d’accords qui diffère partiellement de la séquence observée ailleurs dans le morceau. Lorsque cela se produit, la séquence correspondante ne peut pas être factorisée dans une règle.

Si la différence est ponctuelle, les parties non couvertes par la différence constituent toujours des séquences répétées, ce qui résulte en une fragmentation du segment. On peut noter que ce type de variation reflète à la fois des variations réelles dues à un changement dans la musique et des variations dues à la présence d’erreurs dans la séquence d’entrée. Si la différence est plus générale, comme dans le cas relativement fréquent d’une transposition, la séquence observée est entièrement nouvelle et ne peut pas être factorisée si elle n’est pas répétée sous cette forme altérée.

On peut déduire des différentes situations dans lesquelles se produisent des erreurs d’inférence que les limites de la méthode décrite dans ce chapitre sont de trois ordres principaux :

- le modèle basé sur les GDU est très rigide et ne permet pas toujours de rendre compte efficacement de la structure des morceaux ;
- le choix de la longueur de la grammaire comme critère à optimiser est simpliste et ne reflète que partiellement l’objectif annoncé d’une structure permettant une description simple du morceau ;
- le caractère glouton de l’algorithme IRR entraîne occasionnellement l’inférence dans de minima locaux peu satisfaisants dont il est très difficile de sortir.

Il en résulte donc l’intérêt de concevoir des extensions de cette méthode visant à dépasser ces limites. Ces extensions étant destinées à apporter d’une part un pouvoir de modélisation plus fort des GDU et d’autre part les outils nécessaires à la définition d’un critère plus adapté à l’inférence de la structure musicale hiérarchique. C’est l’objet du prochain chapitre.

Chapitre 4

Grammaires à éditions et critères de complexité

Comme les études de cas du chapitre précédent le montrent, une application directe des algorithmes IRR et MGP à des grammaires à dérivation unique ne permet pas d'obtenir des résultats pleinement satisfaisants pour l'analyse de structure musicale. Les problèmes rencontrés sont de plusieurs ordres :

- la rigidité des répétitions ne permet pas de prendre en compte les variations ou le bruit d'annotation ;
- les répétitions sont souvent décalées à cause de répétitions à une échelle supérieure ;
- les règles correspondant à différentes échelles d'une même structure se concurrencent entre elles ;
- les segments non répétés ne peuvent être identifiés qu'à travers leurs voisins.

Dans ce chapitre, nous proposons des solutions à certains de ces problèmes. Dans un premier temps, nous étendons la classe des grammaires à dérivation unique afin d'autoriser plus de flexibilité sur les règles, tout en maintenant leur compacité. Dans un second temps, nous nous penchons sur la validité de l'utilisation de la longueur d'une grammaire comme mesure de sa complexité et nous considérons d'autres mesures plus élaborées.

4.1 Extension à des grammaires à édition

4.1.1 Variations au sein d'une classe de segments

En musique occidentale conventionnelle, il est fréquent que la répétition d'une section ne soit pas effectuée à l'identique. Un cas très commun est la modification du dernier quart d'une section. Dans cette situation, la partie commune peut être modélisée par un non-terminal, mais la partie qui diffère ne peut pas être intégrée dans l'élément structurel, même si elle fait bien partie de la section.

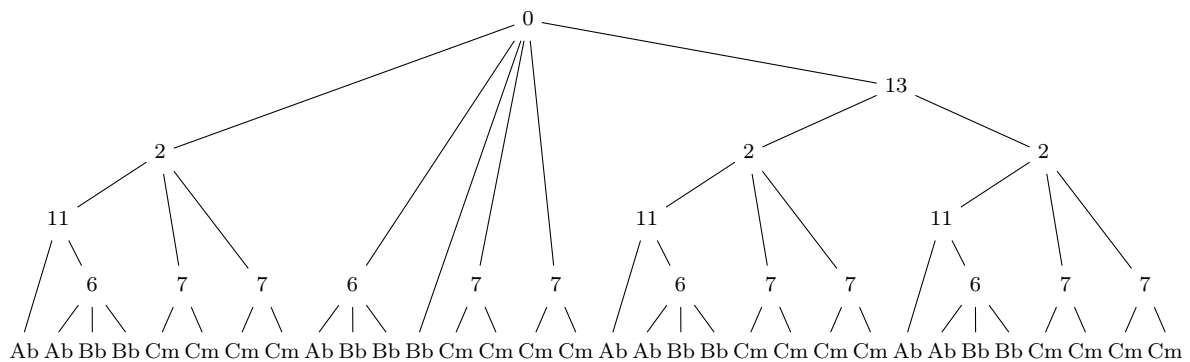
Un autre cas fréquent est une variation mineure, comme l'anticipation d'un changement d'accord au milieu d'une section. Il est possible de modéliser ces cas avec un non-terminal pour le préfixe commun et un non-terminal pour le suffixe commun, mais cette solution est coûteuse en information et ne correspond pas à la réalité de la structure. Ce type de situation peut également se produire lorsque la séquence d'entrée comporte des erreurs d'annotation, comme cela est le cas si elle est issue d'une transcription automatique. Cette situation est illustrée en figure 4.1.

Ab Ab Bb Bb Cm Cm Cm Cm Ab Bb Bb Bb Cm Cm Cm Cm Ab Ab Bb Bb Cm Cm Cm Cm Ab Ab Bb
 Bb Cm Cm Cm Cm

(a) Extrait de la séquence d'accords de la base AUTO pour le morceau RWC-Pop-015. Le changement d'accord au 10^e temps (de Ab vers Bb) a été quantifié 1 temps trop tôt.

0 → [...] 2 6 Bb 7 7 13
 2 → 11 7 7
 6 → Ab Bb Bb
 7 → Cm Cm
 11 → Ab 6
 13 → 2 2

(b) Extrait de la grammaire inférée sur le morceau RWC-Pop 015 (version AUTO). L'altération empêche le regroupement de la première moitié de l'extrait sous le non-terminal 13. On remarque également que l'altération entraîne la présence du non-terminal 6 qui n'est pas un regroupement pertinent musicalement.



(c) Arbre de dérivation correspondant

FIGURE 4.1 – Exemple d'échec de modélisation des variations (extrait de RWC-Pop 015, version AUTO).

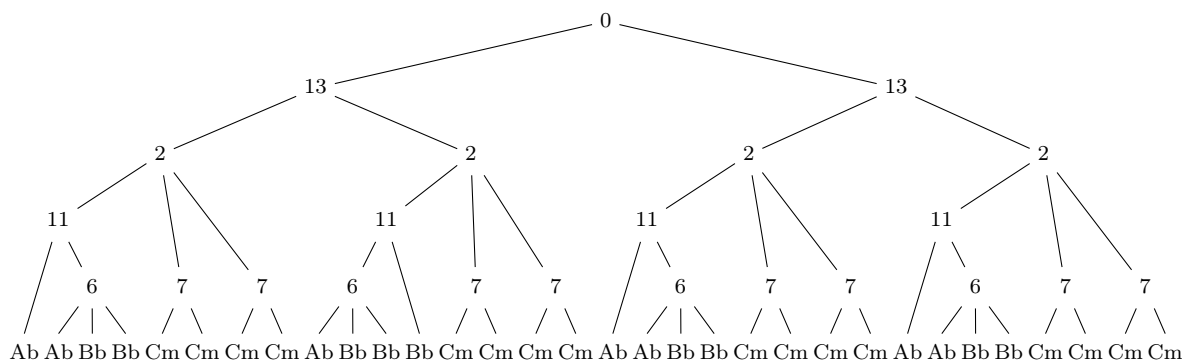
4.1.2 Formalisation des grammaires à éditions à dérivation unique

L'idée générale motivant l'introduction des grammaires à éditions est de regrouper sous un même non-terminal non seulement les répétitions identiques, mais aussi les répétitions similaires. Une approche utilisant le formalisme des grammaires hors-contexte serait d'introduire des règles supplémentaires dans la grammaire, correspondant aux variantes observées d'une même classe de segments.

Afin de conserver l'unicité de la dérivation, il est alors nécessaire d'accompagner l'utilisation des non-terminaux comportant plusieurs règles d'une chaîne complémentaire, indiquant quelle règle doit être utilisée lors de la dérivation. Alternativement, cette chaîne complémentaire peut être omise afin d'obtenir un niveau d'abstraction plus élevé, comme illustré en figure 4.2.

$0 \rightarrow [\dots] 13 \ 13$
 $2 \rightarrow 11 \ 7 \ 7$
 $6 \rightarrow Ab \ Bb \ Bb$
 $7 \rightarrow Cm \ Cm$
 $11 \rightarrow Ab \ 6$
 $11 \rightarrow 6 \ Bb$
 $13 \rightarrow 2 \ 2$

(a) Grammaire hors contexte générant l'extrait de la figure 4.1a



(b) Arbre de dérivation pour la séquence de l'extrait.

FIGURE 4.2 – Modélisation de variations à l'aide d'une grammaire hors contexte sur l'exemple de la figure 4.1. Le non-terminal 11 permet ici de rattacher le troisième groupe de 4 accords de l'extrait aux autres groupes de 4 accords similaires. On peut toutefois noter que d'autres séquences peuvent être dérivées.

Toutefois, cette approche ne permet pas de tirer pleinement parti des redondances entre les différentes variantes d'une même classe.

Une autre approche consiste à considérer une classe de segments comme un ensemble de versions déformées d'un prototype. On peut ainsi représenter la classe à l'aide d'une règle associant le symbole non-terminal au prototype de la classe, et accepter un certain nombre de déformations autour de ce prototype.

Type	Notation
Insertion de x terminaux	A^{+x}
Suppression de x terminaux	A^{-x}
Substitution de x terminaux par y terminaux	$A^{x/y}$

TABLE 4.1 – Notations pour les éditions

On définit ainsi l'ensemble des Grammaires à Éditions à Dérivation Unique (GEDU), que l'on peut subdiviser en sous-classes, selon le type d'éditions que l'on considère. Dans le cadre de cette thèse, nous nous limitons aux éditions unitaires de la distance de Levenshtein [LEVENSHTEIN, 1966], à savoir l'insertion, la suppression et la substitution. Une différence notable est que nous considérons ces opérations à l'échelle de sous-séquences contigües et non des symboles individuels. Ainsi, la suppression de 4 symboles contigus est considérée comme une seule opération de suppression et non 4 opérations de suppression distinctes. La longueur des sous-séquences impliquées est tout de même considérée lors de la définition du coût de l'opération. On obtient ainsi une classe de grammaires très similaire aux grammaires paramétriques utilisées dans [GIRAUD et STAWORKO, 2015].

L'utilisation de types d'éditions plus riches et musicalement informées est toutefois une des pistes d'extension les plus prometteuses de cette approche. On peut notamment envisager des éditions de type consolidation et fragmentation [MONGEAU et SANKOFF, 1990], ou la gestion de transpositions [ALLALI et al., 2007].

À nouveau, une information complémentaire est nécessaire pour maintenir l'unicité de la dérivation. Cette information peut être omise si la réversibilité n'est pas nécessaire pour la modélisation. Pour cette approche toutefois, nous conservons les informations définissant les éditions effectuées. Afin de ne pas surcharger les notations dans ce document, seuls les types d'édition seront indiqués en exposant, bien que le modèle sous-jacent conserve l'intégralité des informations sur les éditions. Les notations correspondantes sont indiquées en table 4.1.

La figure 4.3 donne un exemple partiel d'une GEDU et de sa dérivation sur l'exemple de la figure 4.1.

L'extension proposée du formalisme pour les GEDU n'altère pas le fonctionnement de base des GDU ; les algorithmes s'appliquant aux GDU restent donc compatibles, mais ils ne tirent pas parti de cette extension.

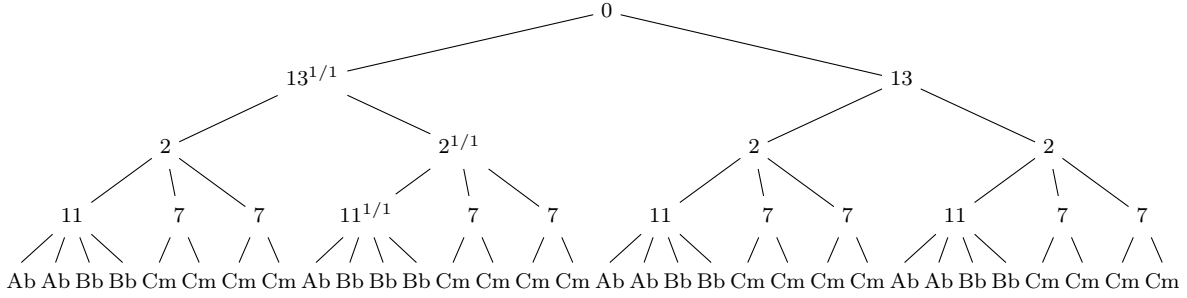
4.1.3 Adaptation des algorithmes

Afin d'utiliser les degrés de liberté introduits avec les éditions, il est nécessaire d'adapter les algorithmes d'inférence de sorte que les règles inférées puissent inclure des éditions.

Parmi les algorithmes décrits dans le chapitre précédent, deux se prêtent à des adaptations pour inclure des éditions par leur fonction : la phase de recherche d'IRR et la phase de construction du graphe de MGP.

$0 \rightarrow [\dots] 13^{1/1} 13$
 $2 \rightarrow 11 \ 7 \ 7$
 $7 \rightarrow \text{Cm} \ \text{Cm}$
 $11 \rightarrow \text{Ab} \ \text{Ab} \ \text{Bb} \ \text{Bb}$
 $13 \rightarrow 2 \ 2$

(a) GEDU correspondant au cas de la figure 4.1. La règle du non-terminal 6 a été supprimée car le non-terminal 6 n'était présent que dans la règle du non-terminal 11.



(b) Arbre de dérivation correspondant

FIGURE 4.3 – Une Grammaire à Éditions à Dérivation Unique et la dérivation correspondante

4.1.3.1 Adaptation de MGP

L'adaptation de la construction du graphe de parsing est assez directe. En plus de rechercher les occurrences exactes de la séquence considérée, les arcs correspondants aux occurrences approximatives de la séquence sont ajoutés au graphe, avec un coût défini par une fonction de score modifiée pour tenir compte de la sévérité de la déformation (voir section 4.2.1 pour la fonction de score modifiée).

Appliquée simplement, cette adaptation peut toutefois résulter en une grammaire dont les règles se référencent mutuellement, comme celle de la figure 4.4. Afin d'éviter ce type de situation, une solution consiste à filtrer quels sont les non-terminaux pouvant être utilisés pour la réécriture d'une règle. Ainsi, si un non-terminal A est présent dans la dérivation d'un autre non-terminal B , alors B et ses variantes ne peuvent pas être utilisés pour la ré-écriture de A .

En raison de cette modification, l'ordre de ré-écriture des règles peut changer le résultat de la ré-écriture si deux règles indépendantes gagnent à être ré-écrites en utilisant l'autre. En pratique, cette différence ne semble pas être significative. Pour notre implémentation, nous ré-écrivons les règles dans leur ordre d'introduction.

Pour notre implémentation de l'algorithme adapté, nous limitons la recherche d'occurrences approximatives aux variantes comportant une seule édition, affectant moins de la moitié de la règle modifiée. Dans certains cas, il est possible que la phase IRR factorise une séquence avec plusieurs non-terminaux modifiés, ce qui résulte en une variante comportant plusieurs éditions. Afin d'assurer que la phase de ré-écriture produit une grammaire qui est au pire équivalente à la grammaire précédente selon la fonction de score, les arcs correspondant aux variantes utilisées sont ajoutés au graphe, même s'ils ne font pas partie de l'espace de recherche des variantes.

$S \rightarrow AAB$	$S \rightarrow AAB$
$A \rightarrow Bef$	$A \rightarrow Bef$
$B \rightarrow abcd$	$B \rightarrow A^{-2}$
(a) Grammaire avant ré-écriture naïve	(b) Grammaire après ré-écriture naïve

FIGURE 4.4 – Ré-écriture incorrecte produisant une grammaire non-dérivable

L’algorithme 6 rassemble ces modifications et décrit la version adaptée de la construction du graphe pour MGP.

Algorithme 6 Construction du graphe pour MGP avec éditions. Les modifications par rapport à l’algorithme 2 sont indiquées en bleu italique.

Entrée: une GDU G , comportant la règle $A \rightarrow \beta$, Δ *l’ensemble des éditions admissibles*

Trouver α la séquence constitutive de A

Créer Γ_A avec un noeud n_0

pour i de 1 à $|\alpha|$ **faire**

Ajouter à Γ_A un noeud n_i

Ajouter à Γ_A un arc de n_{i-1} à n_i , étiqueté α_i

fin pour

Trouver \mathcal{N}' l’ensemble des non-terminaux de G *n’utilisant pas A dans leur dérivation*

pour tout $N_i \in \mathcal{N}$ **faire**

Trouver γ_i la séquence constitutive de N_i

pour tout occurrence de γ_i entre α_j et α_k **faire**

Ajouter à Γ_0 un arc de n_{j-1} à n_k , étiqueté N_i

fin pour

pour tout occurrence de γ_i entre α_j et α_k à l’édition $\delta \in \Delta$ *près faire*

Ajouter à Γ_0 un arc de n_{j-1} à n_k , étiqueté N_i^δ

fin pour

fin pour

Ajouter à Γ_A les arcs correspondant à β

Sortie: Γ_A , le graphe d’optimisation de la règle de G associée à A

Ces modifications ont un effet modéré sur le temps d’exécution. Le faible nombre de séquences considérées compense la recherche peu optimisée des occurrences approximatives, et le nombre accru d’arcs n’augmente que peu le temps d’exécution de la phase de recherche du chemin optimal. Un exemple est présenté en figure 4.5.

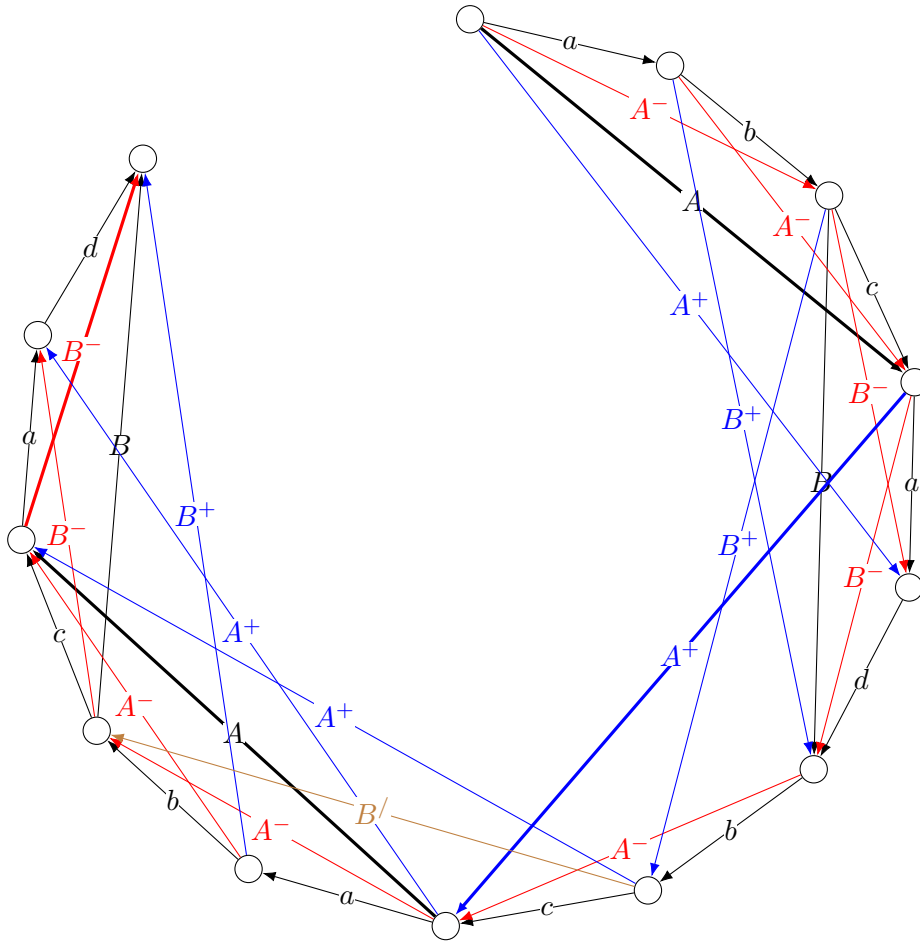
Notre implémentation effectue une recherche naïve des occurrences approchées, ce qui nous contraint à limiter la recherche à une édition par occurrence afin de conserver un temps d’exécution raisonnable. La recherche d’occurrences approximatives est cependant un problème très étudié, en particulier dans les séquences musicales [KILIAN et HOOS, 2004; CAMBOUROPOULOS et al., 2002; CLIFFORD et ILIOPOULOS, 2004]. Si les approches spécifiques à la musique requièrent fréquemment un format particulier, il est également possible d’appliquer des algorithmes indépendants de la représentation tels que BLAST [ALTSCHUL et al., 1990] pour trouver

(a) Grammaire utilisée pour l'illustration

$$S \rightarrow AadbcabB$$

$$A \rightarrow abc$$

$$B \rightarrow cad$$



(b) Graphe de MGP avec des éditions. En rouge les suppressions, en bleu les insertions, en beige les substitutions. Un chemin court est indiqué en gras.

FIGURE 4.5 – Construction du graphe de ré-écriture avec des éditions (limitées à une édition d'un seul symbole). La construction du graphe sans éditions (en noir) est présentée en figure 3.7.

des occurrences sur des séquences musicales [MARTIN et al., 2012]. L'intégration de ces algorithmes permettrait d'étendre le champ des variantes acceptées.

4.1.3.2 Adaptation de IRR

Une adaptation directe à IRR pose plus de difficultés.

Premièrement, un nombre de séquences beaucoup plus important est examiné lors de cette étape, rendant la recherche de séquences approchées beaucoup plus coûteuse algorithmiquement. Ce problème est amplifié par les méthodes d'optimisation de la recherche de répétition exacte, qui ne se transpose que partiellement aux éditions considérées : la table de suffixes ne permet une recherche rapide que pour certains types d'édition et l'élagage de l'espace des solutions n'est pas transposable.

Deuxièmement, l'espace de recherche des nouvelles règles est élargi : une sous-séquence n'a plus besoin d'être répétée (ni même présente) pour permettre une réduction de la taille de la grammaire en introduisant la règle correspondante.

Enfin, l'addition des éditions entraîne la présence de nombreux chevauchements entre des occurrences dont le remplacement a un bénéfice variable, ce qui demande un algorithme supplémentaire afin de sélectionner un ensemble optimal d'occurrences compatibles pour évaluer l'introduction de la règle correspondante.

L'ensemble de ces difficultés, bien qu'aucune ne semble insurmontable, nous a conduit à rediriger nos efforts sur les autres axes d'amélioration, ainsi que sur l'approche alternative présentée au chapitre suivant.

Un autre problème se pose également pour la gestion des éditions existantes pour la recherche de répétitions. On peut soit considérer toutes les versions d'une édition comme étant équivalentes, ce qui permet aux éditions de généraliser, soit considérer chaque édition comme un symbole différent, ce qui permet de spécialiser une variante récurrente d'un non-terminal.

Dans le premier cas, que nous retenons pour notre implémentation, il est nécessaire de propager l'édition aux échelles supérieures lors d'un remplacement, afin que la dérivation de la grammaire reste constante. Si plusieurs non-terminaux de la répétition sont des variantes, on peut ainsi obtenir un non-terminal comportant plusieurs éditions. On peut alors avoir une règle qui n'a aucune occurrence "pure" dans la séquence initiale, ce qui ne permet pas d'obtenir le graphe de reparsing simplement par filtrage du graphe initial.

Les variantes obtenues lors de l'ajout de règles ne sont pas nécessairement incluses dans l'espace de recherche de la ré-écriture, principalement lorsque plusieurs éditions sont présentes sur un même non-terminal. Comme mentionné dans la section précédente, il faut alors ajouter une étape supplémentaire visant à insérer les arcs correspondant aux variantes utilisées dans la version actuelle de la grammaire, afin d'assurer que la phase MGP ne fait pas croître la taille de la grammaire.

4.2 Critères de complexité

Comme nous l'évoquons en section 3.2, la longueur d'une grammaire est une mesure assez limitée de sa complexité. Elle ne permet par exemple pas de prendre en compte des régularités de l'arbre au-delà des répétitions, telles qu'un facteur de branchement constant ou l'équilibre des branches. La figure 4.6 permet d'illustrer ces faiblesses.

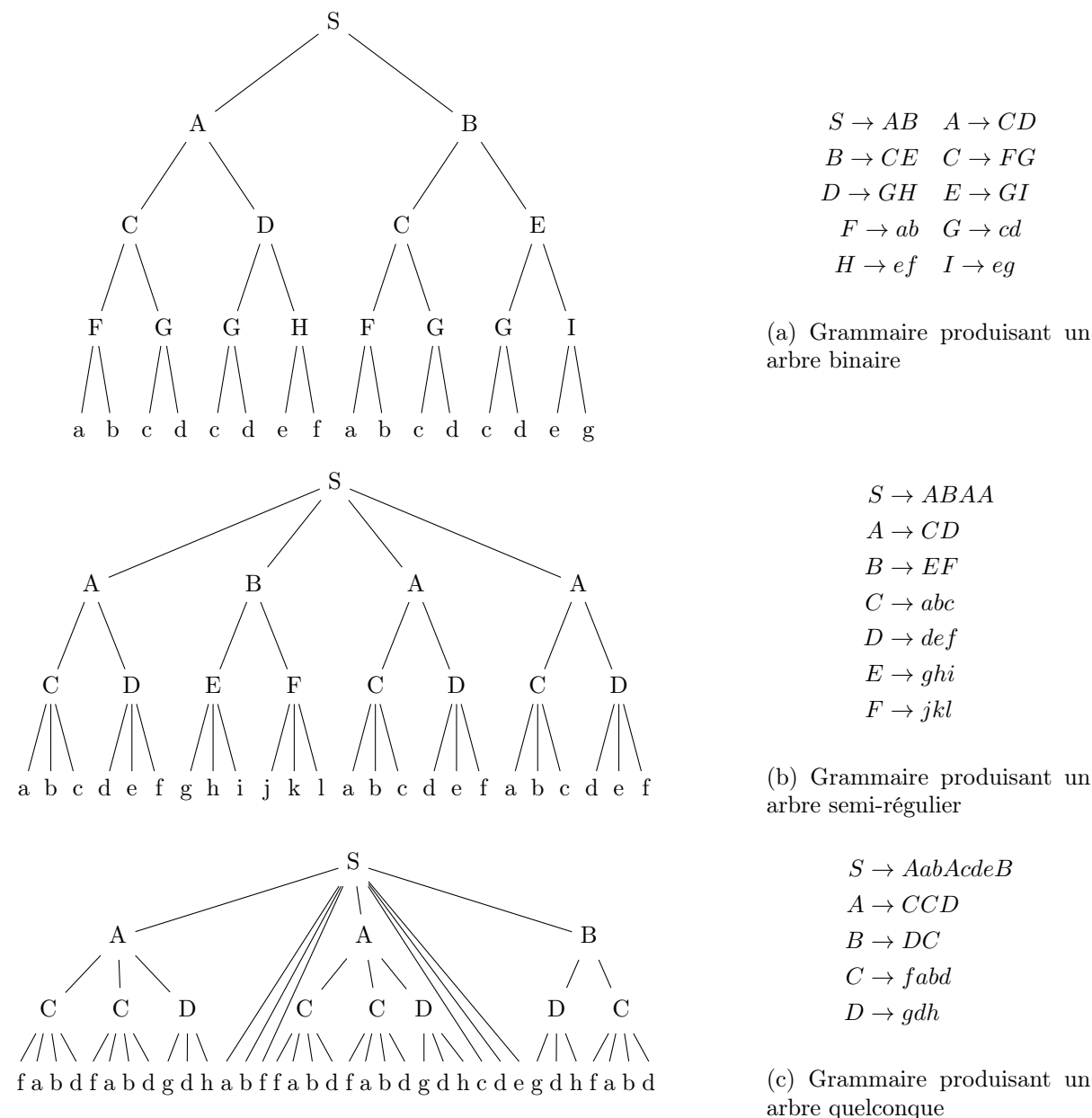


FIGURE 4.6 – 3 GDU de longueur 20, produisant des arbres de complexité différente

Cette faiblesse est renforcée par la nature des données musicales. En effet, les séquences étudiées sont relativement courtes (de l'ordre de quelques centaines de symboles), et les répétitions pertinentes à la structure ne sont observées qu'un nombre réduit de fois, ce qui peut les rendre difficile à distinguer des co-occurrences fortuites sur la seule base de la réduction de la taille de la grammaire.

4.2.1 Définition de critères avancés

On pose :

$$\mathcal{C}(G) = \sum_{A \rightarrow \alpha \in G} \mathcal{C}(A \rightarrow \alpha) \quad (4.1)$$

c'est-à-dire que le coût d'une GEDU est composée du coût de ses règles.

On choisit de distinguer deux composantes du coût d'une règle, que l'on appelle coût de description et coût de conformité.

$$\mathcal{C}(A \rightarrow \alpha) = \mathcal{C}_d(A \rightarrow \alpha) + \mathcal{C}_c(A \rightarrow \alpha)$$

Coût de définition Ce coût représente les coûts relatifs aux symboles qui composent le membre droit de la règle, α . On peut le décomposer à nouveau, symbole par symbole :

$$\mathcal{C}_d(A \rightarrow \alpha) = \sum_{\alpha_i \in \alpha} \mathcal{C}(\alpha_i).$$

Coût de conformité Ce coût permet de modéliser la concordance de la règle par rapport à un a priori sur sa structure ou son contenu.

Ces coûts peuvent être instanciés de multiples façons, dont nous allons souligner quelques versions.

Longueur de la grammaire Une première façon d'instancier ces coûts est de prendre $\mathcal{C}(a) = 1$ pour tous les symboles et de fixer le coût de conformité à 0. On retrouve alors comme critère la longueur de la grammaire.

Longueur par codage entropique Une tendance que l'on peut observer avec un critère basé sur la longueur de la grammaire est la prolifération de nombreuses règles très courtes remplaçant toutes les co-occurrences répétées. Si ces règles courtes n'ont que peu d'incidence sur la longueur de la grammaire, elles étendent l'ensemble des symboles, ce qui affecte à la hausse la complexité de la grammaire. On peut tenter de compenser ce problème en simulant un codage arithmétique de la grammaire, assignant à chaque symbole un coût proportionnel au logarithme de sa fréquence d'apparition.

Pénalisation des éditions Une version d'un symbole modifiée par une édition peut être considérée comme étant plus complexe que sa version prototype, car il s'agit d'une déviation par rapport aux attentes, qui nécessite d'être qualifiée pour être reproduite. La quantité d'information supplémentaire dépend du type d'édition : pour une suppression, il suffit de spécifier la plage de symboles à supprimer, mais pour une insertion ou une substitution, les symboles insérés doivent être décrits en plus de la position d'insertion. Cette information supplémentaire peut être traduite dans le coût de définition de chaque symbole édité.

Distribution a priori sur des méta-propriétés Le coût de conformité permet de traduire des connaissances a priori sur le type de morceau analysé. Ainsi, on peut favoriser des sous-segments de longueur plus régulière ou pénaliser plus ou moins fortement des variations selon leur position dans le morceau ou dans le segment, à travers un ajustement de ce terme.

4.2.2 Adaptation des algorithmes

Les algorithmes présentés dans le chapitre 3 étant prévus pour minimiser la longueur de la grammaire, un certain nombre d'adaptations est nécessaire afin de pouvoir traiter des critères avancés.

Coûts variables par symbole La modification du critère de complexité rend obsolète un certain nombre d’optimisations basée sur l’élagage anticipé des remplacements clairement suboptimaux. Selon la version du critère de complexité, il est encore possible de procéder à un élagage, mais la formulation plus élaborée du critère rend l’élagage moins efficace. Parallèlement, l’algorithme BFS (algorithme 3) ne permet plus de rechercher le chemin le plus court dans le graphe de MGP. Cependant, on dispose par construction d’un ordonnancement topologiquement trié du graphe¹, ce qui permet d’utiliser un autre algorithme efficace de recherche du plus court chemin, détaillé dans l’algorithme 7.

Algorithme 7 Plus Court Chemin sur un graphe topologiquement trié

Entrée: Γ un graphe dirigé, $N = \{n_0, \dots, n_\Omega\}$ un ordonnancement topologiquement trié des nœuds, *longueur* une fonction qui donne la distance entre deux nœuds reliés par un arc
cout $[n_0] := 0$ ▷ Parcours du graphe
pour tout $n_i \in N \setminus \{n_0\}$ **faire**
 cout $[n_i] := \inf$
 pour tout n_j prédécesseur de n_i **faire**
 si *cout* $[n_j] + \text{longueur}(n_j, n_i) < \text{cout}[n_i]$ **then**
 pred $[n_i] := n_j$
 cout $[n_i] := \text{cout}[n_j] + \text{longueur}(n_j, n_i)$
 fin si
 fin pour
fin pour
 $n := n_\Omega$ ▷ Reconstruction du chemin emprunté
 $C := [n_\Omega]$
répéter
 $n := \text{pred}[n]$
 Préfixer C de n
jusqu’à $n = n_0$
Sortie: C , chemin le plus court de n_0 à n_ω dans Γ

Coûts de conformité non-séparables Le caractère non-séparable du coût de conformité pose un problème supplémentaire à l’application d’un algorithme de plus court chemin au graphe de MGP. Toutefois, si le terme de conformité reste faible par rapport au terme de définition, il est possible de contourner le problème en calculant avec le coût de définition non seulement le plus court chemin mais aussi un nombre limité d’autres candidats, puis en reclassant les candidats en incluant le coût de conformité. Cette modification requiert une adaptation de l’algorithme 7 afin de conserver plus de solutions, ce qui est possible avec une augmentation modérée du temps de calcul (voir algorithme 8).

4.3 Expériences

Pour les expériences de ce chapitre, nous utilisons de nouveau les ensembles décrits dans le chapitre précédent, section 3.3.1. Les expériences de segmentation suivent également la même méthodologie, décrite en section 3.3.5.

1. Un ordonnancement des nœuds d’un graphe dirigé acyclique est dit topologiquement trié si pour tous les arcs l’origine apparaît avant la destination.

Algorithme 8 k plus courts chemins sur un graphe topologiquement trié

Entrée: Γ un graphe dirigé, $N = \{n_0, \dots, n_\Omega\}$ un ordonnancement topologiquement trié des nœuds, **longueur** une fonction qui donne la distance entre deux nœuds reliés par un arc, k le nombre de chemins à trouver

chemins $[n_0] := [([n_0], 0)]$ \triangleright Pour chaque nœud, on conserve au plus k chemins et leurs longueurs

pour tout $n_i \in N \setminus \{n_0\}$ **faire**

Initialiser un tas H du type de **chemins** $[n]$ trié selon la longueur du premier chemin

pour tout n_j prédécesseur de n_i **faire**

temporaire $:=$ **chemins** $[n_j]$

Ajouter n_i aux chemins de **temporaire** et **longueur** (n_j, n_i) à leurs longueurs

Insérer **temporaire** sur le tas T

fin pour

répéter

\triangleright Fusion partielle de plusieurs listes triées

Prendre **temporaire** sur le dessus du tas H

Retirer le premier chemin de **temporaire** et l'ajouter à **chemins** $[n_i]$

si **temporaire** n'est pas vide **then**

Ré-insérer **temporaire** dans H

fin si

jusqu'à T est vide ou **chemins** $[n_i]$ contient k chemins

fin pour

Sortie: **chemins** $[n_\Omega]$, les k chemins les plus courts de n_0 à n_Ω dans Γ

4.3.1 Impact de l'ajout des éditions

Dans un premier temps, on évalue l'influence de l'introduction de chaque type d'édition. Cette première phase permet également de régler individuellement les fonctions de coût pour les différents types d'éditions.

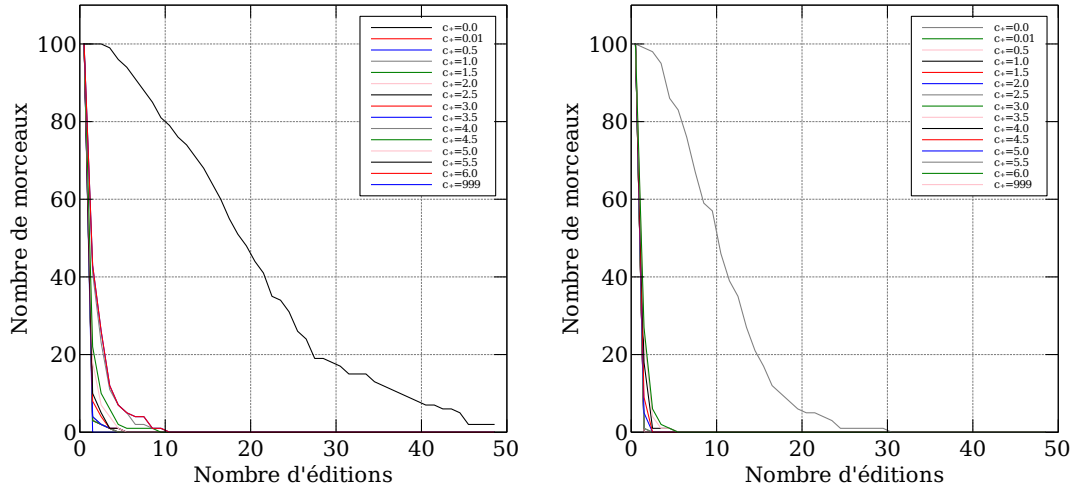
Pour chacun des types d'édition, on examine le comportement selon les deux enchaînements IRR+MGP+C et IRRMGP*. On se limite ici aux éditions ajoutant et supprimant des symboles terminaux. Des éditions permettant d'ajouter ou de supprimer des symboles non-terminaux permettraient une modélisation plus riche, mais nécessitent un nombre plus élevé d'adaptations.

4.3.1.1 Insertions

Complexité d'édition Afin de spécifier une insertion, il est nécessaire de décrire les symboles qui sont insérés. On définit donc le coût d'une insertion comme la longueur de description de la séquence insérée. À cette longueur de description, on ajoute un terme fixe représentant l'irrégularité introduite par l'insertion. Ce terme pourrait être paramétré selon la position de l'édition, afin de, par exemple, favoriser des insertions en fin de segment. Toutefois nous n'avons pas exploré cette option pour limiter le nombre de paramètres. Le coût de l'insertion de la séquence β est donc défini comme

$$\mathcal{C}(+\beta) = c_+ + \mathcal{C}(\beta) \quad (4.2)$$

où c_+ est un paramètre du modèle de coût et $\mathcal{C}(\beta)$ est le coût de description de la séquence β , ici sa longueur.



(a) MGP après chaque itération

(b) MGP après convergence

FIGURE 4.7 – Nombre d’édits par morceau (décumulatif) pour le corpus AUTO selon l’enchaînement d’IRR et MGP utilisé

La figure 4.7 montre le nombre d’insertions présentes dans les grammaires finales selon la version de l’algorithme utilisée. On peut constater que le nombre d’édits décroît très rapidement avec l’augmentation de la pénalisation.

À l’exception du cas où $c_+ = 0$, l’utilisation des insertions est très ponctuelle : même pour le cas où elles sont les plus utilisées, seulement un tiers de représentations font appel à une insertion, et l’utilisation de plusieurs insertions est très rare. Dans le cas où $c_+ = 0$, l’insertion en fin de segment est équivalente en terme de coût à la juxtaposition des deux chaînes : $\mathcal{C}(A^+\beta) = \mathcal{C}(A\beta)$. Cette égalité conduit à une utilisation systématique des insertions, dont une majorité ne correspond pas à des variantes pertinentes. Inversement, lorsque $c_+ > 0$, une insertion en fin de segment n’est jamais moins coûteuse, bien que ce type de variante soit un des plus fréquents. Ces cas de figures montrent qu’un critère de complexité plus nuancé est requis afin d’aboutir à des représentations en accord avec la structure intuitive.

TABLE 4.2 – Écarts de F-mesure (± 3 temps) avec et sans insertions (positif = meilleur avec insertions)

c_+	∞	0	1	2
Auto, IRR+MGP+C	45.4%	+2.1%	-0.4%	-0.4%
Auto, IRRMGP*	45.4%	+0.6%	+0.1%	-0.1%
Manuel, IRR+MGP+C	61.1%	-2.1%	-0.1%	-0.2%
Manuel, IRRMGP*	60.9%	-0.3%	+0.1%	+0.1%

La table 4.2 détaille l’influence des insertions sur la segmentation obtenue. Comme la majorité des grammaires inférées ne comporte pas d’insertions, seul un faible nombre de segmentations est affecté.

Celles qui le sont peuvent conduire à de fortes différences de F-mesure, mais l’effet moyen est à la fois très faible et variable selon la base de données et la version de l’algorithme utilisée,

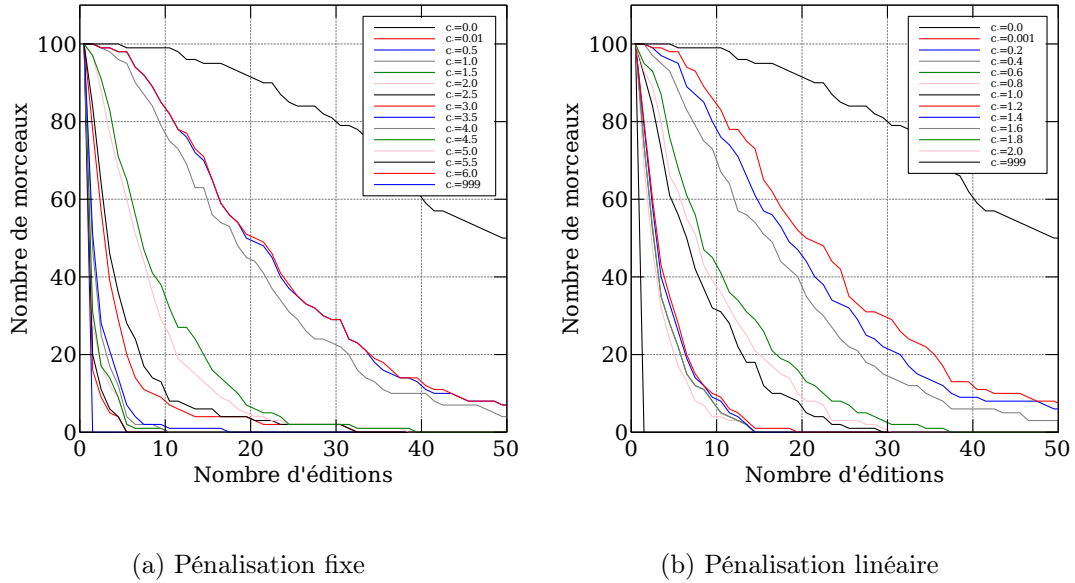


FIGURE 4.8 – Nombre de suppressions par morceau (décumulatif) pour le corpus AUTO en fonction du type de pénalisation.

et il n'est pas possible de conclure à une amélioration ou une détérioration de la qualité de la segmentation.

4.3.1.2 Suppressions

Complexité d'édition Pour les suppressions, l'information requise pour transformer la séquence se limite à la position et à la taille de la suppression. On peut toutefois souhaiter pénaliser plus fortement les suppressions de grande taille. Dans cette optique, on teste deux fonctions pour la complexité de l'édition : la première associant un coût fixe à toutes les suppressions et la seconde variant linéairement avec la taille de la suppression. Pour la suppression de la chaîne β on a donc :

$$\mathcal{C}_1(-\beta) = c_- \quad (4.3)$$

$$\mathcal{C}_2(-\beta) = c'_- \times |\beta| \quad (4.4)$$

où c_- et c'_- sont les paramètres du modèle de coût contrôlant la pénalisation des suppressions.

Les figures 4.8 montrent la répartition du nombre de suppression par morceau dans les grammaires finales. Contrairement aux insertions, les variantes à base de suppressions sont beaucoup sélectionnées par l'algorithme, y compris pour des coûts non nuls. Cette différence peut s'expliquer entre autres par la complexité relativement faible de l'opération de suppression.

Les tendances relatives au corpus et à la version de l'algorithme utilisés sont les mêmes que pour les insertions : les éditions sont plus fréquentes avec la version automatique des séquences d'accords et lorsque la ré-écriture des règles est effectuée après chaque nouvelle règle.

La figure 4.3 montre l'évolution du score de segmentation avec l'ajout des suppressions. Là encore, les effets sont trop faibles et irréguliers pour dégager une tendance.

TABLE 4.3 – Écarts de F-mesure avec et sans suppressions (positif = meilleur avec insertions)

c_-	∞	0	0.2	0.4	1	2
Linéaire, Auto, IRR+MGP+C	45.4%	+0.8%	+0.5%	+0.9%		
Linéaire, Auto, IRRMGP*	45.4%	+0.8%	+1.0%	+0.8%		
Linéaire, Manuel, IRR+MGP+C	61.1%	-3.0%	-2.1%	-1.8%		
Linéaire, Manuel, IRRMGP*	60.9%	+0.8%	+0.5%	+0.1%		
Fixe, Auto, IRR+MGP+C	45.4%	+0.8%			+0.5%	+0.1%
Fixe, Auto, IRRMGP*	45.4%	+0.8%			+0.8%	+0.2%
Fixe, Manuel, IRR+MGP+C	61.1%	-3.0%			-2.8%	-2.0%
Fixe, Manuel, IRRMGP*	60.9%	+0.8%			+0.3%	+0.1%

4.3.1.3 Substitutions

Complexité d'édition Comme une suppression, une substitution nécessite de spécifier la plage remplacée ; comme une insertion, il est nécessaire de décrire la séquence insérée. Pour évaluer la complexité d'une substitution, on utilise ainsi des fonctions de complexité qui combinent celle des insertions et celles des suppressions. Formellement, pour la substitution de la chaîne β par la chaîne γ on a :

$$\mathcal{C}_1(\beta/\gamma) = c_j + \mathcal{C}(\gamma) \quad (4.5)$$

$$\mathcal{C}_2(\beta/\gamma) = c'_j \times |\beta| + \mathcal{C}(\gamma) \quad (4.6)$$

où c_j et c'_j sont les paramètres des critères de complexité respectifs et $\mathcal{C}(\gamma) = |\gamma|$ dans notre cas. Une formulation plus élaborée pourrait prendre en compte l'adéquation entre la séquence qui est remplacée et celle qui la remplace, mais afin d'éviter de devoir introduire un modèle supplémentaire estimant cette adéquation, cette piste n'a pas été explorée.

Dans notre implémentation, les substitutions sont les éditions qui affectent le plus l'exécution de l'algorithme, atteignant dans le pire cas un temps d'exécution d'une vingtaine de minutes avec une utilisation de la mémoire vive d'environ 3 gigaoctets, là où les autres versions ne dépassent pas 20 secondes pour 60 mégoctets de mémoire vive. Cette augmentation est principalement due à une implémentation relativement naïve de la recherche d'occurrences approximatives pour le temps d'exécution, et à l'existence de nombreuses occurrences avec une substitution triviale remplaçant une sous-séquence par une copie identique (qui est alors trivialement plus coûteuse que l'occurrence exacte).

Un premier constat qualitatif pour les substitutions est que l'algorithme sélectionne fréquemment des substitutions pour remplir le rôle de suppressions², par exemple en remplaçant la sous-séquence $abcde$ par le symbole a , pour un effet équivalent à la suppression de la sous-séquence bcd . Ces substitutions détournées ne posent pas de problème pour une utilisation simultanée des substitutions et des suppressions, car sauf dans le cas d'un paramétrage très différent, elles sont plus coûteuses que les suppressions équivalentes et ne sont pas retenues. Toutefois, elles perturbent l'analyse des substitutions en isolation en confondant les deux types d'édition.

La figure 4.9 et la table 4.4 présentent les résultats obtenus avec les substitutions. Les constats qui peuvent en être tirés sont très similaires au cas des suppressions, en partie en raison des

2. Ce type de comportement est également possible pour des insertions, mais comme établi précédemment les insertions sont beaucoup moins sélectionnées que les suppressions.

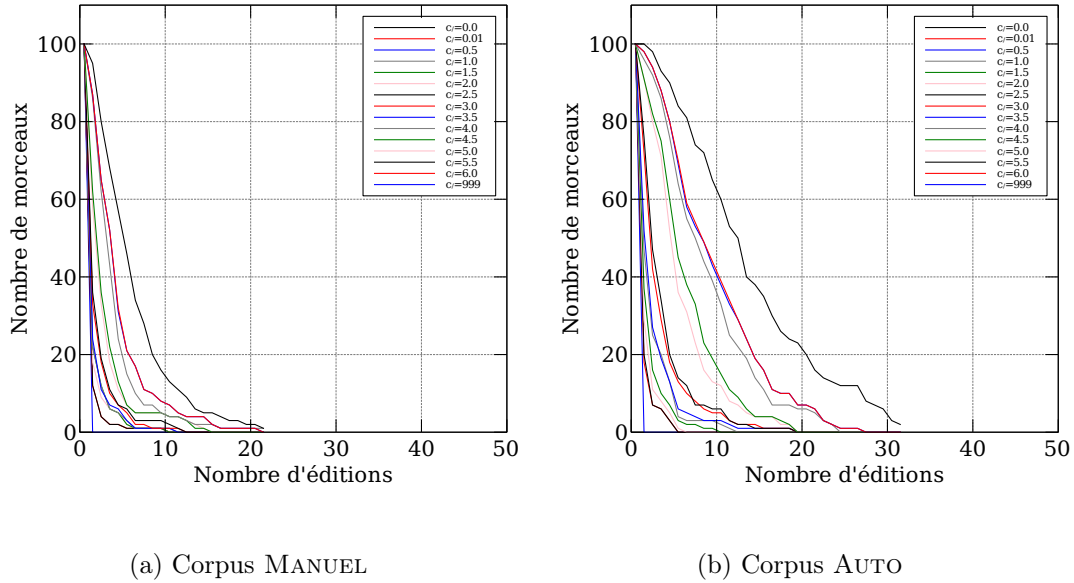


FIGURE 4.9 – Nombre de substitutions par morceau (décumulatif) pour la ré-écriture systématique avec coût fixe selon la base utilisée.

TABLE 4.4 – Répartitions des écarts de F-mesure avec et sans substitutions (positif = meilleur avec insertions)

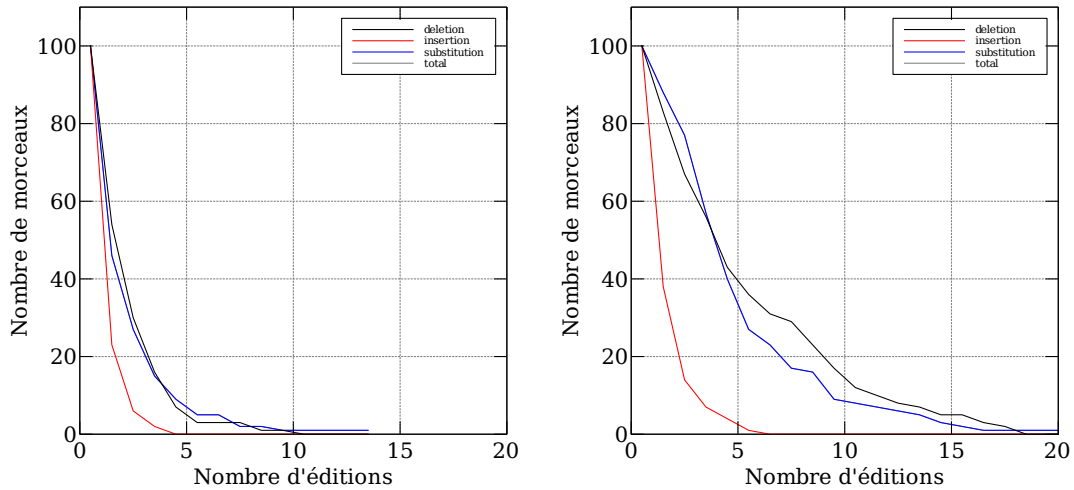
c_+	∞	0	0.2	0.4	1	2
Linéaire, Auto, IRR+MGP+C	45.4%	+2.2%	+3.1%	+2.9%		
Linéaire, Auto, IRRMGP*	45.4%	+2.5%	+1.6%	+1.6%		
Linéaire, Manuel, IRR+MGP+C	61.1%	-3.0%	-2.1%	-1.8%		
Linéaire, Manuel, IRRMGP*	60.9%	+0.2%	-0.1%	$\pm 0.0\%$		
Fixe, Auto, IRR+MGP+C	45.4%	+2.2%			+2.8%	+1.3%
Fixe, Auto, IRRMGP*	45.4%	+2.5%			+1.7%	+0.6%
Fixe, Manuel, IRR+MGP+C	61.1%	-3.0%			-2.2%	-2.4%
Fixe, Manuel, IRRMGP*	60.9%	+0.2%			$\pm 0\%$	$\pm 0\%$

substitutions détournées : les substitutions sont assez fréquentes, surtout pour le corpus AUTO et avec la variante effectuant plus de ré-écritures, mais les différences de score sur la segmentation sont trop variables pour conclure.

4.3.1.4 Combinaison des éditions

Complexité d'édition Les expériences précédentes n'ayant pas permis de déterminer des valeurs privilégiées pour les paramètres du critère de complexité, nous avons choisi pour l'expérience combinée de régler les coûts fixes comme équivalents à celui d'un symbole normal. Ce paramétrage permet d'éviter le problème des substitutions détournées comme suppressions (ainsi que le détournement, moins courant, des substitutions en insertions).

La figure 4.10 montre le nombre d'édits obtenus dans les grammaires finales pour chaque type. Les tendances observées individuellement pour les éditions sont également observables ici : les insertions sont assez rares, les suppressions sont fréquentes, et les substitutions ont une fréquence intermédiaire.



(a) Corpus MANUEL

(b) Corpus AUTO

FIGURE 4.10 – Nombre d'édits par morceau (décumulatif) pour IRR+MGP+C avec coût fixe

TABLE 4.5 – F-mesure en utilisant des insertions, des suppressions et des substitutions

Auto, IRR+MGP+C	48.3%
Auto, IRRMGP*	46.4%
Manuel, IRR+MGP+C	60.3%
Manuel, IRRMGP*	60.8%

Comme le montre la table 4.5, les écarts de F-mesure sont encore une fois très faibles une fois agrégés.

D'après un examen manuel des résultats, les variantes dont la différence avec la séquence de référence est située en milieu de séquence sont en général bien identifiées, cependant ce type de variante représente une minorité des cas. Pour les modifications qui se situent en début ou en fin de séquence, le critère de complexité n'a pas une granularité suffisante pour distinguer si l'utilisation d'une édition est pertinente. Ainsi, selon le choix des paramètres, l'algorithme ne sélectionne jamais d'insertions en forme de suffixe ou les sélectionne autant que possible. De façon similaire, la résolution de l'ambiguïté entre une séquence tronquée et une séquence avec une terminaison différente dépend uniquement des paramètres du critère de complexité et non du contexte d'occurrence comme on pourrait le souhaiter.

4.3.2 Utilisation des critères avancés

Comme décrit dans la section 4.2.1, le coût d'une règle selon des critères avancés peut être séparé en deux composantes : le coût de description, qui prend une forme locale, et dépend principalement du nombre et du type de symboles utilisés, et le coût de conformité, qui s'exprime à l'échelle de la règle et permet d'exprimer la correspondance entre la règle et des a priori sur les caractéristiques de la règle.

4.3.2.1 Coût de description

Pour nos expériences sur le coût de description, nous avons utilisé une formulation du critère de complexité qui simule un codage entropique de la grammaire finale. Dans un tel codage, chaque symbole se voit attribuer un code de longueur négativement proportionnelle au logarithme de sa fréquence d'apparition. Ainsi, les symboles les plus fréquents se voient attribuer un code court tandis que les symboles peu fréquents sont représentés par un code long, ce qui permet une représentation plus courte qu'un encodage naïf avec des codes de longueur égale.

Formellement, le critère de complexité s'exprime comme

$$\mathcal{C}(G) = \sum_{A \rightarrow \alpha \in G} \mathcal{C}(\alpha) \quad (4.7)$$

$$\mathcal{C}(\alpha) = \sum_{\alpha_i \in \alpha} -\log O_G(\alpha_i) \quad (4.8)$$

où $O_G(\alpha_i)$ est le nombre d'occurrence du symbole α_i dans la grammaire G .

La principale différence des grammaires inférées selon le critère simulant un codage entropique est un nombre plus faible de règles. L'introduction de règles est en effet plus coûteuse selon ce critère : la réduction du nombre de symboles utilisés s'accompagne par une augmentation du coût unitaire moyen des symboles³.

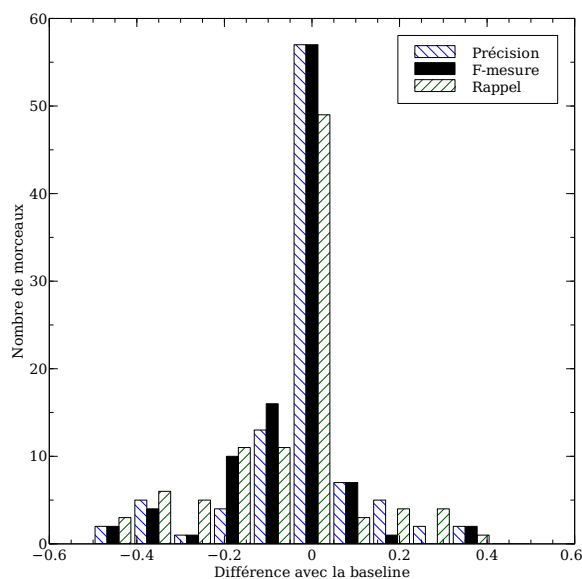


FIGURE 4.11 – Écarts des mesures de segmentation avec le critère simulant un codage entropique

Comme le montre la figure 4.11, cette différence se traduit par une baisse de la F-mesure pour la segmentation, principalement entraînée par une baisse du rappel.

3. Sauf dans des cas particuliers ne correspondant pas aux séquences observées en pratique

Une partie de cette baisse de performance peut s'expliquer par une optimisation moins efficace du critère de complexité. En effet, dans 70% des cas, la grammaire obtenue par optimisation gloutonne de la longueur entropique a une longueur entropique plus élevée que celle obtenue par optimisation de la longueur classique (voir table 4.6). Les sélections de règle en début d'exécution reposent sur des fréquences d'occurrence des symboles qui sont souvent très différentes des fréquences finales, ce qui tend à entraîner l'algorithme vers des minima locaux moins faibles.

TABLE 4.6 – Complexité moyenne des grammaires finales selon chaque critère

Critère optimisé \ Critère mesuré	Longueur (symboles)	Longueur entropique (bits)
Longueur	95.7	451.6
Longueur entropique	117.3	458.6

4.3.2.2 Coût de conformité

Parmi les facteurs qui contribuent à la complexité d'une grammaire, on s'intéresse particulièrement aux points sur lesquels les grammaires inférées par l'algorithme IRRMGP dévient des structures intuitives.

Une de ces tendances est le non-regroupement des sous-séquences non répétées. Cette tendance concerne principalement les sections de type solo ou pont, mais aussi les sous-divisions de sections répétées qui ne se trouvent que dans un type de contexte et ne sont alors relevées une fois ce contexte factorisé. Afin de favoriser le regroupement de ces séquences, on introduit dans le coût de conformité une pénalité pour les règles comportant un grand nombre de symboles, de la forme

$$\mathcal{C}_c(A \rightarrow \alpha) = \begin{cases} \lambda \cdot (|\alpha| - \theta) & \text{si } |\alpha| > \theta \\ 0 & \text{sinon} \end{cases} \quad (4.9)$$

où θ et λ permettent respectivement de contrôler le seuil à partir duquel la pénalité s'applique et sa sévérité.

Une autre tendance est la présence de règles asymétriques, c'est à dire composées de symboles dont la taille des séquences constitutives varie beaucoup. On trouve ainsi des règles où des non-terminaux représentant plusieurs dizaines de temps sont juxtaposés à des terminaux représentant un unique temps. Une formulation possible d'un coût de conformité adressant ce problème est d'utiliser l'écart-type des tailles des séquences constitutives :

$$\mathcal{C}_c(A \rightarrow \alpha) = \lambda \cdot \sigma(\{|\alpha_i|\}) \quad (4.10)$$

L'intégration de ces critères dans l'algorithme n'a toutefois pas été concluante : selon la façon de les intégrer, ces critères entrent en jeu trop tard pour influencer la grammaire qui est quasiment stabilisée, ou trop tôt, les critères s'appliquant alors à un stade de l'inférence où leur formulation n'est pas pertinente.

Ainsi, lorsque ces critères sont implémentés dans la phase MGP de l'algorithme selon la version IRRMGP*, leur influence est très limitée, car la phase MGP ne permet plus que des ajustements mineurs sur la grammaire, à moins d'accorder un poids extrême au coût de conformité. À l'inverse, avec la version IRR+MGP+C ou une implémentation dans la phase IRR de l'algorithme,

ces pénalités s'appliquent dès la grammaire triviale, de façon souvent contre-productive par rapport à l'objectif de l'introduction du critère. Par exemple, sur la règle $N \rightarrow aaaabbbbbbaaaa$, le critère d'homogénéité des tailles pénalisera fortement l'introduction d'une règle $A \rightarrow aaaa$, alors que l'introduction ultérieure d'une seconde règle $B \rightarrow bbbb$ permettrait d'obtenir la règle homogène $N \rightarrow ABBA$.

Ces difficultés montrent à nouveau les limites de l'algorithme glouton au cœur de la méthode IRRMGP. Si cette méthode permet d'obtenir des segmentations modérément correctes lorsque appliquée avec un critère simple, l'utilisation de critères plus avancés rend l'espace d'optimisation plus irrégulier et entrave la recherche d'un optimum.

Conclusion du chapitre

À l'issue du chapitre précédent, nous avons identifié certaines faiblesses de l'application des algorithmes IRR et MGP pour l'inférence de structure musicales. En particulier, la rigidité du modèle de Grammaires à Dérivation Unique ne permet pas de prendre en compte les variations entre les différentes occurrences de segments homologues, et la mesure de la complexité d'une GDU par sa longueur est réductrice, ce qui résulte souvent en des règles courtes mais complexes.

Dans ce chapitre, nous avons défini les Grammaires à Édition à Dérivation Unique, un nouveau modèle utilisant le formalisme des GDU, incluant de plus la possibilité de modifier ponctuellement l'application d'une règle par l'application de transformations d'un type prédéfini. Après avoir adapté l'algorithme MGP afin de tirer parti des fonctionnalités de ce nouveau modèle, nous avons étudié son application avec des transformations standards (insertion, suppression et substitution) pour l'inférence de structure musicale. Nos résultats montrent que si les suppressions et substitutions sont abondantes dans les grammaires inférées, ces utilisations des transformations ne sont pas particulièrement pertinentes et ne permettent pas d'améliorer de façon fiable la segmentation car elles sont introduites trop tard dans le processus.

L'autre axe développé dans ce chapitre est la formulation d'un critère de complexité pour les GDU (et les GÉDU) plus précis que la longueur de la grammaire. Nous avons ainsi proposé une distinction de ce critère de complexité en plusieurs composantes distinctes, permettant une large gamme de critères, ainsi que des adaptations pour les algorithmes IRR et MGP pour tenter d'optimiser ces critères composites. Toutefois, le résultat principal de nos expériences sur ces critères est que la stratégie gloutonne employée dans l'algorithme IRR ne permet pas d'optimiser efficacement des critères plus détaillés que la longueur de la grammaire.

Au vu des résultats sur les deux axes traités dans ce chapitre, une voie d'amélioration possible pour les méthodes d'inférence de structure par recherche de la plus simple grammaire serait de remplacer l'algorithme IRR par un autre algorithme pour l'introduction de règles, plus adapté à des critères composites et conçu pour la recherche d'occurrences transformées.

Pour la suite de cette thèse, nous avons opté pour l'étude d'une seconde famille d'approches, complémentaire de l'approche par plus simple grammaire, basées plus explicitement sur la modélisation des éléments structurels comme des éléments autonomes et construits selon des suites logiques.

Chapitre 5

Principes de compression tensorielle

Dans les chapitres précédents, nous avons développé une méthode d'estimation de la structure musicale s'appuyant principalement sur la compression des répétitions dans un morceau de musique. Les limites de cette méthode ont permis de mettre en évidence l'importance d'un critère de "conformité" des éléments structurels. Les tentatives visant à intégrer de tels critères dans la méthode des chapitres précédents ont révélé que l'aspect glouton de son algorithme principal IRR se prêtait peu à l'optimisation de ce type de critères.

Dans ce chapitre, nous décrivons une nouvelle méthode d'estimation de la structure musicale s'appuyant sur la cohérence interne des segments. Plus précisément, on cherche à obtenir une suite de segments indépendants, de telle sorte que chacun des segments peut être décrit économiquement en exploitant ses régularités internes.

Le paradigme proposé ici consiste à modéliser la structure "mésoscopique"¹ des segments musicaux sous l'hypothèse que la description macroscopique optimale du morceau résulte de l'économie de chacun de ses segments structurels. On fait donc l'hypothèse qu'un segment structurel se caractérise par le groupage de constituants élémentaires qui se décrivent conjointement de façon économique.

Pour réaliser cette description économe, nous nous appuyons sur une extension multi-échelle du modèle Système & Contraste (S&C) [BIMBOT et al., 2012], dont nous résumons la forme originale en section 5.1. La section 5.2 introduit le concept de patron tensoriel, qui permet de représenter des prototypes de l'organisation interne des éléments d'un segment. Nous y identifions en particulier une famille régulière d'organisations sur laquelle nous focalisons notre étude : les patrons hypercubiques. En section 5.5, nous établissons un cadre pour la description économe d'une séquence en s'appuyant sur un patron tensoriel, pour lequel nous proposons successivement la formulation de deux critères provisoires, avant d'en proposer une version plus complète qui fait l'objet du prochain chapitre.

Ce cadre est ensuite utilisé pour définir simplement une méthode de description économe de segments individuels en section 5.6 et une méthode d'estimation de segmentation en section 5.7.

1. L'échelle mésoscopique se définit comme une échelle intermédiaire entre l'échelle microscopique (qui caractérise les atomes, les molécules, les constituants élémentaires...) et l'échelle macroscopique (qui caractérise les corps dans leur ensemble).

5.1 Modèle Système & Contraste

Le modèle *Système & Contraste* est un modèle qui permet de décrire l'organisation interne d'un groupe de 4 éléments selon les relations qui existent entre les propriétés de ces éléments.

Dans sa forme la plus simple, le modèle Système & Contraste se compose de 4 éléments consécutifs x_{00}, x_{01}, x_{10} et x_{11} , qui peuvent être arrangés en carré comme cela est illustré sur la figure 5.1². Ce carré permet de mettre en évidence une relation d'analogie entre les différents éléments du système :

- x_{01} est à x_{00} ce que x_{11} est à x_{10}
- et/ou
- x_{10} est à x_{00} ce que x_{11} est à x_{01} .



FIGURE 5.1 – Forme de base du modèle S&C et illustrations de systèmes non-contrastifs pour des lieux, des formes et des nombres. On peut noter que la forme abstraite est elle-même un système non-contrastif portant sur les indices de x .

Alternativement, cet ensemble d'éléments peut être décrit par l'élément x_{00} , que l'on appelle le *primer*, et les relations (ou transformations) f et g telles que $x_{01} = f(x_{00})$, $x_{10} = g(x_{00})$ et $x_{11} = (f \circ g)(x_{00})$. Ces transformations peuvent porter sur toutes, sur une partie ou sur aucune (fonction identité) des propriétés de x_{00} , comme le montrent les illustrations de la figure 5.1. Cette description par un triplet (x_{00}, f, g) est appelée *système porteur*.

La forme simple où $x_{11} = (f \circ g)(x_{00})$ est appelée système *non-contrastif*, car le dernier élément ne dévie pas de la suite logique établie par le reste du système. En pratique, on observe souvent une déviation partielle ou totale du dernier élément par rapport à l'élément impliqué $x_{11} = \gamma((f \circ g)(x_{00}))$. On note alors \widehat{x}_{11} l'élément virtuel qui aurait complété le système de façon non contrastive. Cette déviation γ , qui s'oppose à la logique établie par le système porteur constitue le *contraste* du modèle Système & Contraste.

Un certain nombre de variantes autour de ce système carré ont été définies pour permettre de modéliser des segments basés sur un nombre différent d'éléments. On y trouve notamment des variantes permettant de modéliser des systèmes de 3, 5 ou 6 éléments, en déformant le carré support du système. Ces variantes sont détaillées dans [BIMBOT et al., 2016] et permettent de rendre compte de structures musicales variées.

Un des objectifs du modèle présenté dans la suite de cette thèse est d'apporter une généralisation du modèle S&C pour une représentation multi-échelle, s'appuyant sur un plus grand nombre de relations, dont le besoin est illustré dans [DERUTY, BIMBOT et VAN WYMEERSCH, 2013]. Il se place ainsi comme un modèle apparenté à celui présenté dans [LOUBOUTIN et BIMBOT, 2016 ; LOUBOUTIN et BIMBOT, 2017a ; LOUBOUTIN et BIMBOT, 2017b], élaboré de façon concomitante

2. Pour des exemples musicaux, le lecteur est invité à consulter [BIMBOT et al., 2016].

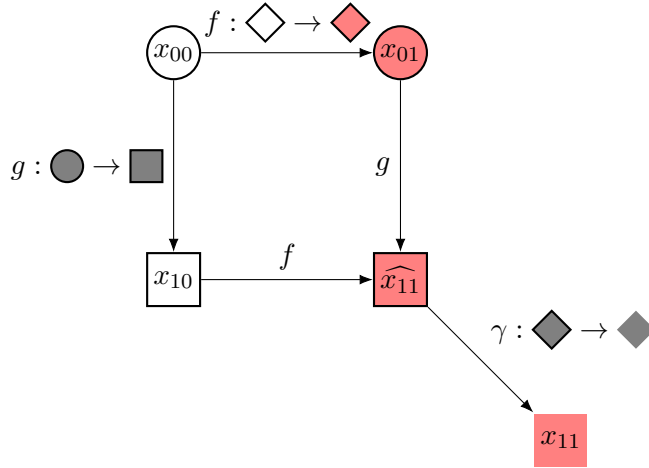


FIGURE 5.2 – Visualisation des relations entre les éléments du modèle Système & Contraste sur un système d'éléments géométriques. Les relations portent sur le remplissage, la forme et le contour des éléments. L'élément \widehat{x}_{11} est un élément virtuel qui ne fait pas partie de la séquence observée.

dans l'équipe de recherche PANAMA³. Toutefois, dans le présent travail, le paradigme proposé permet de gérer des segments de taille variable et l'application visée est centrée sur la tâche de segmentation structurelle.

5.2 Patron tensoriel

La première partie de l'extension du modèle Système & Contraste consiste à s'affranchir des limitations de la forme carrée utilisée pour arranger les éléments de la séquence étudiée. D'une part, l'organisation carrée limite intrinsèquement la modélisation à des séquences de 4 éléments (avec d'éventuels infixes ou suffixes). D'autre part, les relations entre les éléments sont limitées à 2, le nombre de dimensions du plan dans lequel s'inscrit le carré.

On peut voir l'organisation en carré utilisée dans le modèle S&C comme une représentation dans une matrice 2×2 d'une séquence de 4 éléments. Pour étendre le nombre de relations et le nombre d'éléments, une possibilité consiste à d'augmenter le nombre de dimensions de la représentation, et d'organiser les éléments non plus selon une matrice, mais selon un tenseur sur 3 dimensions ou plus. On assigne alors les éléments de la séquence à certaines positions du tenseur, ce qui détermine un *patron tensoriel* qui permettra de conditionner l'analyse :

Définition 2 (Patron tensoriel)

Un patron tensoriel de capacité n et de dimension $d < n$ est constitué d'une fonction bijective dite d'indexation associant une séquence de n éléments à autant de positions d'un tenseur sur d dimensions.

Il faut préciser qu'un patron tensoriel ne remplit pas nécessairement complètement le tenseur sous-jacent. Ainsi, on peut associer un élément à la position $(1, 2, 1)$ sans qu'un élément soit associé à la position $(1, 2, 0)$.

3. On peut ainsi établir un parallèle entre certaines notions, comme les polytopes supports et les graphes de relations latentes, avec les notions qui sont introduites dans la suite de ce chapitre, respectivement les patrons tensoriels et les graphes de redondance.

Définition 3 (Admissibilité d'un patron)

On dit qu'un patron tensoriel est admissible pour une séquence donnée si la longueur de la séquence est égale à la capacité du patron tensoriel, c'est-à-dire que l'on peut indexer exactement la séquence selon les positions du patron tensoriel.

5.3 Graphe de redondance

De la même façon que sur la matrice carrée du modèle S&C on fait l'hypothèse que les éléments voisins verticalement ou horizontalement forment un système d'analogie (x_{01} est à x_{00} ce que x_{11} est à x_{10}), on peut émettre l'hypothèse que les paires d'éléments voisins portées par des arêtes parallèles dans le patron tensoriel sont analogues (par exemple, que x_{201} est à x_{200} ce que x_{122} est à x_{121}). On regroupe les relations formées par ces paires d'éléments dans des classes d'équivalence appelées *classes d'homologies* (on dit également que les paires d'éléments sont homologues). L'ensemble de ces relations permet de définir un graphe, que l'on appelle *graphe de redondance*.

Définition 4 (Graphe de redondance)

Le graphe de redondance est l'ensemble des relations entre éléments voisins selon une seule dimension dans un patron tensoriel (celles-ci étant entendues comme étant orientées dans l'ordre chronologique).

Définition 5 (Classe d'homologie)

Une classe d'homologie est l'ensemble des relations portées par des arêtes parallèles dans un patron tensoriel.

5.4 Types et déformations de patrons tensoriels

La définition que nous avons donnée pour les patrons tensoriels est extrêmement générale et permet de considérer un grand nombre d'organisations différentes des éléments de la séquence. Dans cette thèse, nous considérons non seulement les patrons tensoriels réguliers de dimension d (c'est-à-dire les patrons hypercubiques) mais aussi nous prenons en compte la possibilité que ces patrons réguliers subissent un certain nombre de déformations par insertion et/ou omission de certains de ses éléments à des positions prédéterminées.

Définition 6 (Patron hypercubique)

Le patron hypercubique de dimension d , ou patron d -cubique, est le patron tensoriel correspondant à un hypercube de dimension d . Chaque sommet du d -cube correspond à une position du patron d -cubique.

Pour la fonction d'indexation d'une séquence sur un patron hypercubique, on remarque que les indices des positions dans le tenseur sont binaires. En associant arbitrairement⁴ chacune des coordonnées de l'index à un bit, on obtient pour chaque position un nombre représenté sur d bits. La fonction d'indexation associe à chaque position l'élément dont le rang dans la séquence est égal à ce nombre. Ainsi, sur un patron hypercubique de dimension 4, l'élément x_3 est associé à la position $(0, 0, 1, 1)$ et l'élément x_{13} est associé à la position $(1, 1, 0, 1)$.

La figure 5.3 montre deux exemples de patrons hypercubiques, l'un basé sur un cube (dimension 3) et l'autre sur un tesseract (dimension 4).

4. L'hypercube étant symétrique, ce choix n'a pas d'influence.

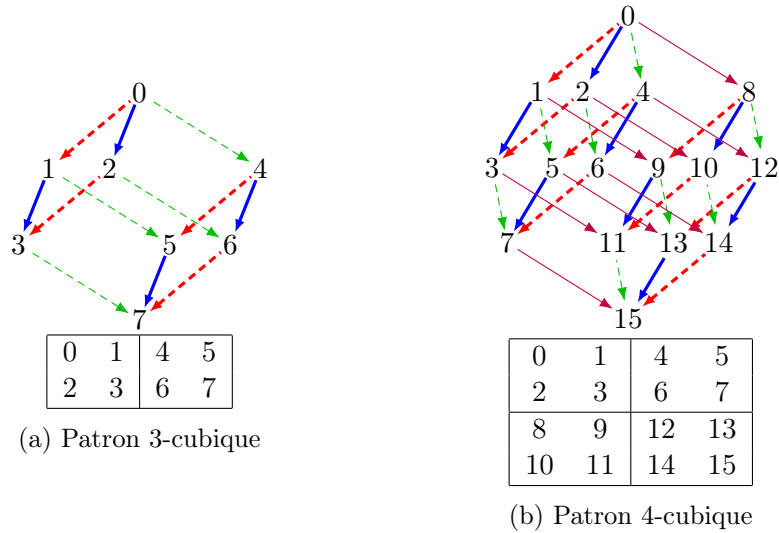


FIGURE 5.3 – Patrons hypercubiques, sous forme géométrique et de table. Les classes d’homologie sont mises en évidence par les couleurs et le motif de trait. Sur les représentations en table, les positions alignées verticalement ou horizontalement d’une même case sont connectées, de même que les éléments aux mêmes positions de cases voisines. Chaque type d’alignement correspond à une classe d’homologie.

Bien évidemment, le patron n -cubique est admissible seulement pour les séquences de 2^n éléments. Afin de pouvoir modéliser des séquences d’autres longueurs tout en conservant une partie de la régularité de l’hypercube, on introduit deux types de déformation du patron hypercubique, l’une permettant de modéliser des séquences plus longues, l’autre des séquences plus courtes.

Définition 7 (Omission sur un patron hypercubique)

Une m -omission sur le patron n -cubique ($0 \leq m < n$) consiste à réduire le patron n -cubique en supprimant les positions correspondant à un hypercube de dimension m . La fonction d’indexation de la séquence est adaptée pour ignorer les positions ainsi supprimées.

Définition 8 (Insertion sur un patron hypercubique)

Une m -insertion sur le patron n -cubique ($0 \leq m < n$) consiste à étendre le patron n -cubique en dupliquant les positions correspondant à un hypercube de dimension m .

L’indexation de la séquence est adaptée pour incorporer les positions ainsi créées, en veillant à préserver la cohérence des classes d’homologies.

La figure 5.4 montre des exemples de patrons avec une omission, une insertion ou une combinaison des deux types d’éditations (on parle alors de patrons *mixtes*, et l’omission prend priorité sur l’insertion pour les positions communes). On se restreint ici aux omissions (respectivement insertions) pour lesquelles les positions supprimées (respectivement insérées) incluent la dernière position du patron (située à l’index $[1, 1 \dots, 1]$). La première position supprimée (respectivement insérée) suffit alors à identifier l’omission (respectivement l’insertion).

On peut remarquer que les cas où $m = n - 1$ se simplifient et peuvent donc être ignorés :

- les patrons n -cubiques avec une $(n - 1)$ -omission sont équivalents au patron $(n - 1)$ -cubique,
- les patrons n -cubiques avec une $(n - 1)$ -insertion sont équivalents à des patrons $(n + 1)$ -cubiques avec une $(n - 1)$ -omission.

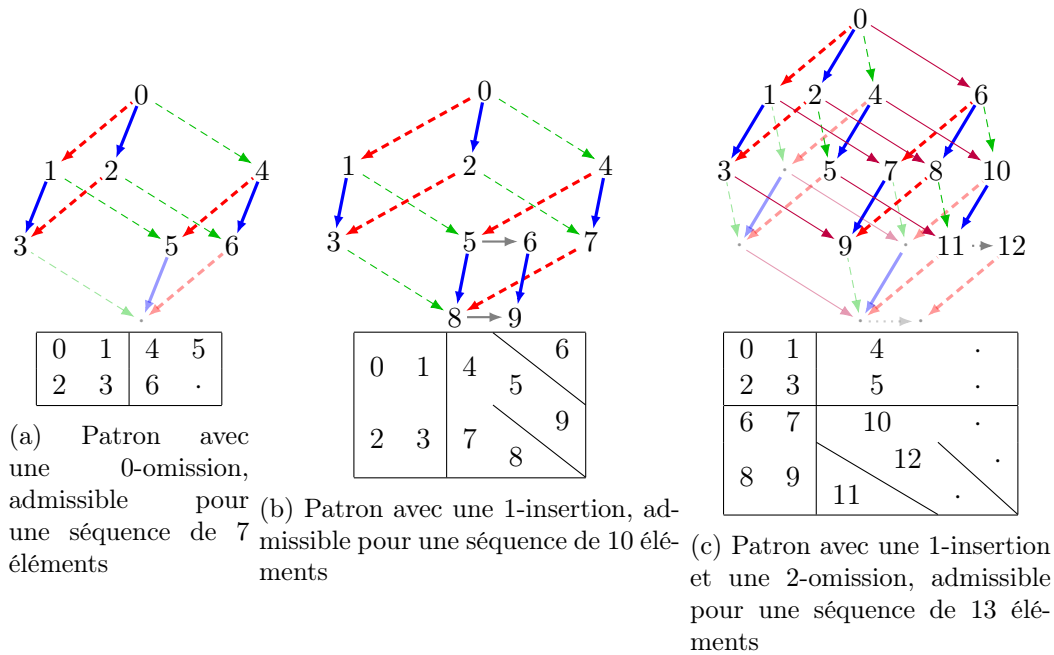


FIGURE 5.4 – Exemples de patrons avec omission, avec insertion et avec une combinaison des deux.

Si l'on considère ensemble les patrons n -cubiques, les patrons avec une insertion, une omission et les patrons avec une omission et une insertion, il est possible de couvrir un nombre bien plus élevé de longueurs et d'organisations internes de séquences qu'avec les seuls patrons hypercubiques. La table 5.1 détaille le nombre de patrons admissibles pour des longueurs de segment de 6 à 64. On peut noter que certaines tailles demeurent non-admissibles pour ces types de patrons, telles que 39 ou 44. Toutefois on observe empiriquement que ces tailles de segments sont relativement rares⁵.

5.5 Critère de complexité

Une fois un patron tensoriel déterminé, on peut assigner à chaque position du patron un élément de la séquence étudiée par le biais de la fonction d'indexation. Il en découle un graphe de redondance, qui met en correspondance les éléments voisins sur le patron tensoriel.

Si le patron est "bien adapté" à la séquence, ces correspondances sont prédisposées à permettre une description simple de la séquence. La figure 5.5 montre deux projections d'une même séquence, dont une seule permet une description simple. En ce sens, le patron tensoriel peut être apparenté à une hypothèse sur la structure interne du segment considéré.

Dans cette section, nous développons cette idée et nous étudions la définition d'une mesure formelle de complexité de la description d'une séquence en s'appuyant sur les concepts de patron tensoriel et de graphe de redondance. Cette mesure permettra ensuite de servir de critère pour choisir entre les différents patrons tensoriels admissibles, et de définir un critère à optimiser pour grouper les éléments musicaux et obtenir ainsi une segmentation d'un morceau de musique.

5. Sur le corpus MANUAL, 21 segments (1.6%) ont une longueur pour laquelle aucun des patrons décrits ici n'est admissible, dont 16 pour la seule longueur 44.

Taille de segment	Patrons n-cubiques	Patrons avec extension	Patrons avec omission	Patrons mixtes	Total
6	0	0	3	6	9
7	0	0	1	7	8
8	1	0	0	3	4
9	0	1	0	0	1
10	0	3	0	0	3
11	0	0	0	0	0
12	0	0	6	24	30
13	0	0	0	12	12
14	0	0	4	32	36
15	0	0	1	19	20
16	1	0	0	16	17
17	0	1	0	12	13
18	0	4	0	6	10
19	0	0	0	0	0
20	0	6	0	0	6
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	10	80	90
25	0	0	0	20	20
26	0	0	0	60	60
27	0	0	0	10	10
28	0	0	10	100	110
29	0	0	0	30	30
30	0	0	5	100	105
31	0	0	1	51	52
32	1	0	0	55	56
33	0	1	0	30	31
34	0	5	0	70	75
35	0	0	0	10	10
36	0	10	0	30	40
37	0	0	0	20	20
38	0	0	0	10	10
39	0	0	0	0	0
40	0	10	0	0	10
41	0	0	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	0	0	0	0	0
45	0	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	15	240	255
49	0	0	0	30	30
50	0	0	0	120	120
51	0	0	0	15	15
52	0	0	0	180	180
53	0	0	0	0	0
54	0	0	0	60	60
55	0	0	0	0	0
56	0	0	20	280	300
57	0	0	0	60	60
58	0	0	0	180	180
59	0	0	0	60	60
60	0	0	15	330	345
61	0	0	0	60	60
62	0	0	6	312	318
63	0	0	1	141	142
64	1	0	0	156	157

TABLE 5.1 – Nombre de patrons admissibles pour des segments de longueur 6 à 64.

ABAABA

(a) La séquence considérée (sous sa forme séquentielle)

A	B	B	A	A	B	A	B
A	A	·	·	A	·	A	·

(b) Indexation résultant en un modèle irrégulier (c) Indexation résultant en un modèle régulier

FIGURE 5.5 – Exemples d’indexation d’une séquence sur différents patrons tensoriels

5.5.1 Schéma général

Dans la lignée des chapitres précédents, on utilise la compressibilité d’une séquence comme mesure de sa complexité⁶. On se situe dans le cadre d’un codage de type progressif, c’est-à-dire que chaque élément de la séquence est encodé individuellement, en se basant exclusivement sur les éléments qui le précèdent.

Si on note $(x_0, x_1, \dots, x_{|X|-1})$ les éléments de la séquence X et $\mathcal{H}(x_i) = \{x_h : 0 \leq h < i\}$ l’ensemble des éléments précédant chronologiquement x_i (c’est-à-dire son *historique*), on peut écrire la complexité de la séquence X selon le patron tensoriel P comme la somme de l’information nécessaire pour encoder chaque élément :

$$\mathcal{C}(X|P) = \sum_{x_i \in X} \mathcal{C}(x_i|P, \mathcal{H}(x_i)), \quad (5.1)$$

où $\mathcal{C}(x_i|P, \mathcal{H}(x_i))$ désigne l’information nécessaire à encoder l’élément x_i en s’appuyant sur le patron tensoriel P et sur l’historique de x_i , c’est-à-dire $\mathcal{H}(x_i)$.

On peut utiliser le patron tensoriel afin de restreindre cet historique aux éléments occupant une position homologue à l’élément considéré, ainsi qu’aux éléments menant à la définition de ces éléments homologues.

On définit à cette fin la relation d’adjacence tensorielle \prec_1 telle que $x_i \prec_1 x_j$ si $i < j$ et si l’indexation des éléments x_i et x_j sur le patron tensoriel diffère d’exactement une unité sur une dimension et une seule. On note \prec la fermeture transitive de \prec_1 , c’est-à-dire :

$$x_i \prec x_k \text{ si } \begin{cases} x_i \prec_1 x_k \\ \text{ou} \\ \exists x_j : x_i \prec_1 x_j \wedge x_j \prec x_k \end{cases} \quad (5.2)$$

Ces relations permettent de définir pour chacun des éléments de la séquence les ensembles :

$$\sigma(x_i) = \{x_j : x_i \prec_1 x_j\} \quad (5.3)$$

$$\sigma^*(x_i) = \{x_j : x_i \prec x_j\} \quad (5.4)$$

$$\pi(x_j) = \{x_i : x_i \prec_1 x_j\} \quad (5.5)$$

$$\pi^*(x_j) = \{x_i : x_i \prec x_j\} \quad (5.6)$$

— $\sigma^*(x_i)$ est l’ensemble des *successeurs* de x_i

6. On ne s’intéresse toutefois au code représentant la séquence d’entrée que pour sa longueur ; un modèle abstrait de l’encodage, qui permet d’effectuer des simplifications, est donc suffisant.

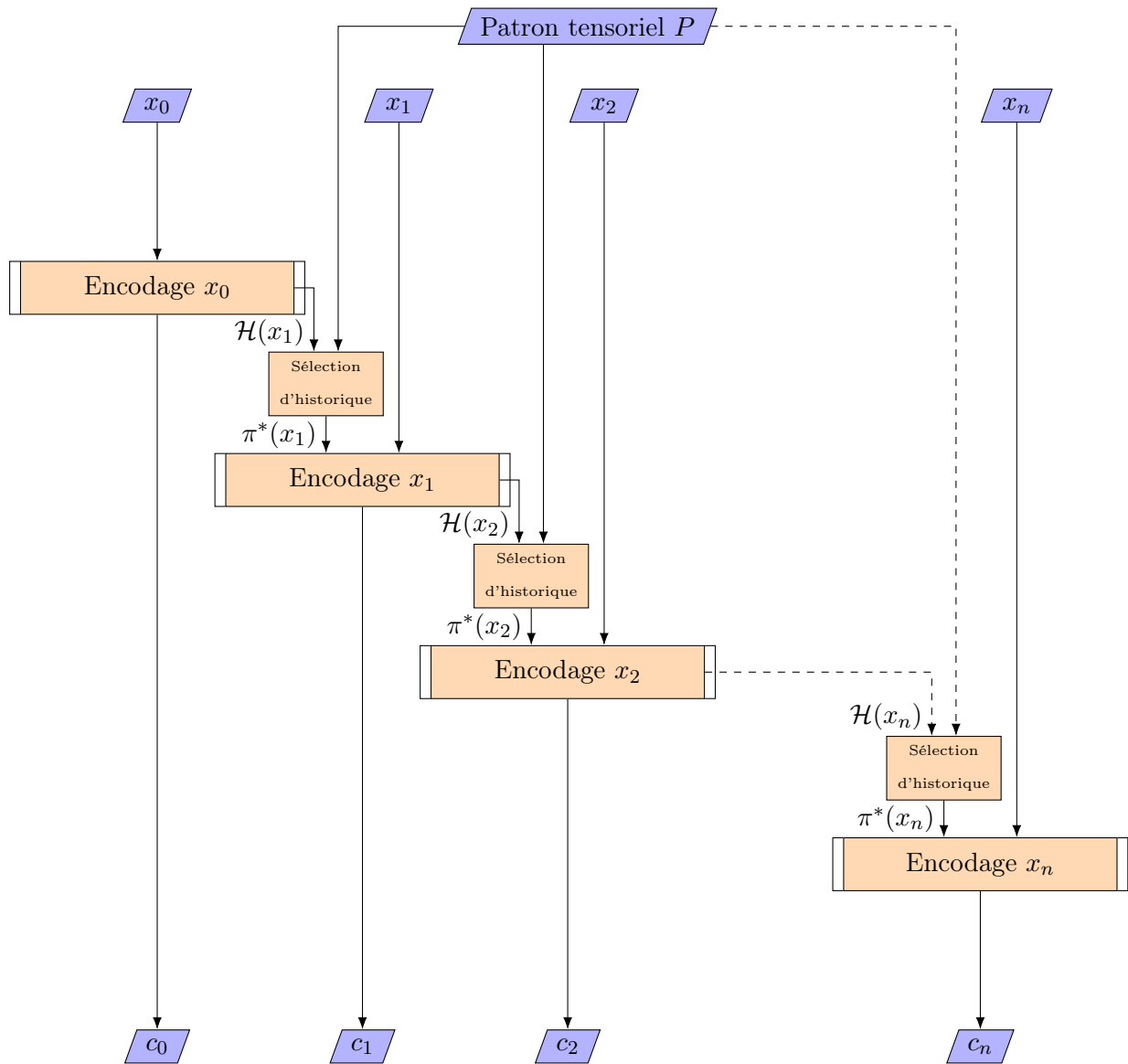


FIGURE 5.6 – Représentation schématique de l’encodage à historique tensoriel. Ce schéma illustre le fait que le patron tensoriel sert à sélectionner un sous-ensemble des éléments de l’historique pour l’établissement de la fonction de compression et le calcul du critère de complexité.

- $\sigma(x_i)$ est l'ensemble des successeurs *directs* de x_i
- $\pi^*(x_i)$ est l'ensemble des *prédécesseurs* de x_i
- $\pi(x_i)$ est l'ensemble des prédécesseurs *directs* de x_i .

Par extension, on notera $\sigma^*(i)$, $\sigma(i)$, $\pi^*(i)$ et $\pi(i)$ les positions correspondantes dans le patron tensoriel.

Ces concepts permettent de formuler de façon synthétique les hypothèses effectuées sur les modèles à base de graphe de redondance pour la suite de ce travail. En l'espèce, seuls les prédécesseurs d'un nœud contribuent à son contexte. On a alors :

$$\mathcal{C}(x_i|P, \mathcal{H}(x_i)) = \mathcal{C}(x_i|\pi^*(x_i)) \quad (5.7)$$

$$\mathcal{C}(X|P) = \sum_{x_i \in X} \mathcal{C}(x_i|\pi^*(x_i)) \quad (5.8)$$

Ainsi, le patron tensoriel sert à sélectionner dans l'historique de x_i le sous-ensemble des éléments qui sont pris en compte pour son encodage, tel que cela est illustré dans la figure 5.6.

La définition de $\mathcal{C}(x_i|\pi^*(x_i))$ est le problème clé de l'établissement d'un critère de complexité basé sur l'utilisation d'un patron tensoriel. Dans les deux sous-sections suivantes, nous proposons une formulation naïve de cette valeur, puis une variante corrigeant une partie des défauts de la formulation initiale. Toutefois, c'est dans le chapitre 6 que nous développons une expression plus complète de $\mathcal{C}(x_i|\pi^*(x_i))$.

5.5.2 Encodage par réplication

Une première formulation simplifiée du critère de complexité proposé dans la section précédente consiste à faire deux hypothèses simplificatrices :

- d'une part, n'utiliser pour chaque élément que le premier ordre du contexte, c'est à dire les prédécesseurs directs de cet élément :
- d'autre part, de ne considérer que l'identité ou la dissemblance entre les éléments comparés pour le calcul du critère de complexité.

Ainsi, conformément à la première de ces deux hypothèses :

$$\mathcal{C}(x_i|\pi^*(x_i)) = \mathcal{C}(x_i|\pi(x_i)) \quad (5.9)$$

ce à quoi vient s'ajouter une nouvelle simplification pour encoder x_i à partir de $\pi(x_i)$, qui consiste à sélectionner un de ses prédécesseurs directs x_h , puis à encoder x_i en s'appuyant sur x_h .

Le coût d'encodage de x_i peut alors se décomposer en deux termes :

- la quantité d'information $\mathcal{Q}(h|i)$ nécessaire pour spécifier la position h d'un des prédécesseurs directs de x_i ;
- la quantité d'information $\mathcal{C}(x_i|x_h)$ nécessaire pour décrire x_i à partir de x_h .

Le choix de x_h étant libre, on prend celui qui permet de minimiser la somme de ces deux quantités.

$$\mathcal{C}(x_i|\pi(x_i)) = \min_{h \in \pi(i)} (\mathcal{C}(x_i|x_h) + \mathcal{Q}(h|i)) \quad (5.10)$$

Si l'on considère une modélisation basique des propriétés des éléments, où l'encodage par la réplication de l'élément de référence est trivial et tout encodage d'une dissemblance est équivalent à un encodage sans référence $\mathcal{C}(x_i)$, on a :

$$\mathcal{C}(x_i|x_h) \simeq \begin{cases} 0 & \text{si } x_h = x_i \\ \mathcal{C}(x_i) = c & \text{sinon} \end{cases} \quad (5.11)$$

où c est une constante dépendant du nombre de valeurs que peut prendre chaque élément. On dit qu'un élément est *concordant* (selon P) si on peut trouver un élément x_h qui permet une description triviale, et qu'il est *discordant* (selon P) sinon.

Ultimement, une simplification⁷ est appliquée sur la formule 5.10, conduisant à l'expression suivante :

$$\mathcal{C}(x_i|\pi(x_i)) \simeq \begin{cases} 0 & \text{si } \exists x_h \in \pi(x_i) : x_h = x_i \\ c & \text{sinon} \end{cases} \quad (5.12)$$

En fixant $c = 1$, la formule de l'équation 5.12 revient à calculer le critère de complexité comme étant le nombre d'éléments discordants dans la séquence considérée, c'est-à-dire (dans le cas présent) le nombre d'éléments dont tous les prédécesseurs directs sont dissemblables.

La figure 5.7 montre un exemple d'application de cette fonction de complexité à plusieurs séquences réelles. Sur ces exemples, les séquences synchronisées sur des segments structurels tendent à obtenir des scores de complexité plus faibles que les autres.

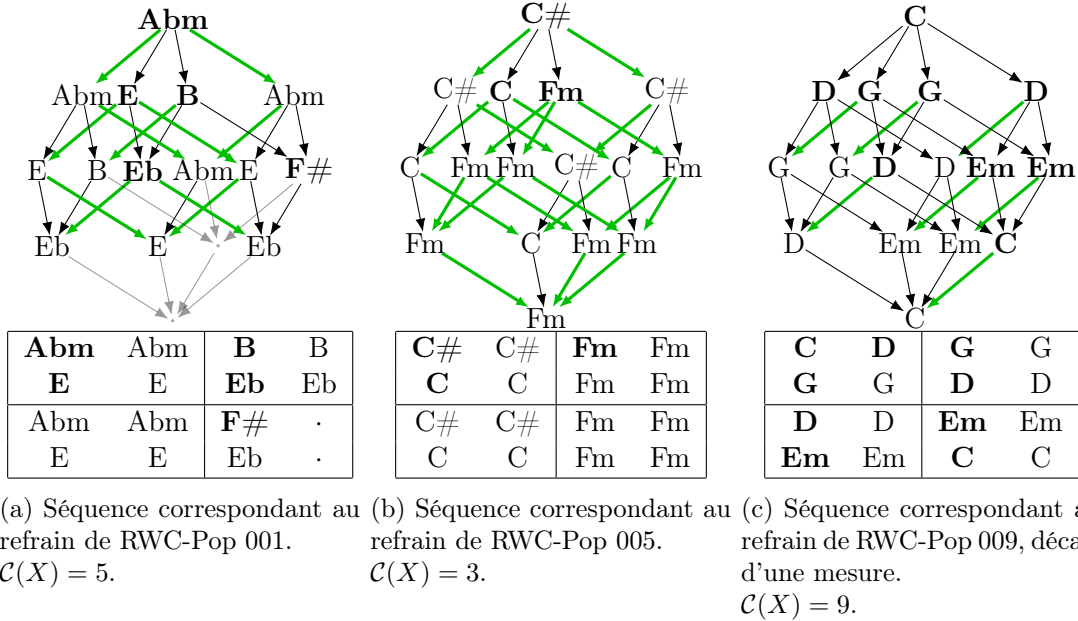


FIGURE 5.7 – Exemples de systèmes de 16 accords évalués à l'aide de la fonction d'encodage par réplication. Les relations permettant un encodage par réplication sont mises en évidence (en vert), ainsi que les éléments discordants (en gras).

7. Cette simplification s'appuie sur l'hypothèse que les variations de $\mathcal{Q}(h|i)$ sont négligeables devant celles de $\mathcal{C}(x_i|x_h)$. La contribution des $\mathcal{Q}(h|i)$ est alors approximativement constante sur l'ensemble de la séquence et n'influe pratiquement pas sur l'optimisation de la segmentation, menant à leur omission dans l'expression du critère.

5.5.3 Adjonction d'une contrainte d'antériorité

Si cette première définition d'un critère de complexité par réplication permet d'associer des scores faibles à des séquences simples, elle produit également des scores relativement faibles pour des séquences avec une organisation peu intuitive, parfois même plus faibles que pour d'autres séquences présentant une organisation beaucoup plus régulière. La figure 5.8 détaille un exemple.

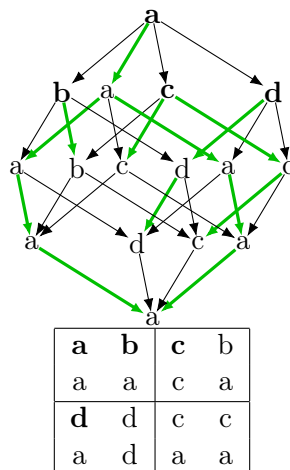


FIGURE 5.8 – Exemple artificiel d'un système très irrégulier donnant lieu à un score faible ($C = 4$) par le critère de réplication de la section 5.5.2.

Si l'on compare la représentation de cet exemple aux séquences régulières de la figure 5.7, on constate que les classes d'homologies utilisées pour les éléments concordants sont beaucoup plus désordonnées. En particulier, certaines relations sont utilisées pour déterminer un élément de référence alors que toutes les autres paires d'éléments de la même classe d'homologie sont dissemblables ; dans un tel cas, l'identité des éléments voisins n'est pas un indice de régularité et apparaît plutôt comme anecdotique.

Pour remédier à cette difficulté, on peut introduire une contrainte, dite d'*antériorité*, qui consiste à invalider certains prédécesseurs. En effet, si le prédécesseur x_h d'un élément x_i permet un encodage par réplication mais qu'une paire homologue observée antérieurement dans le graphe de redondance présente une dissemblance, l'identité entre x_i et x_h constitue une forme de contradiction.

Au-delà de la similarité des éléments au sein d'un segment musical, on peut faire l'hypothèse qu'un critère de compression doit prendre en compte les similarités des *relations* au sein d'une même classe d'homologie, c'est-à-dire l'ensemble des paires d'éléments voisins dans le tenseur portées par des arêtes parallèles.

Pour formuler cette propriété, on introduit une contrainte, qui consiste à considérer comme prédécesseurs directs *valides* uniquement les prédécesseurs directs x_h de l'élément à encoder x_i pour lesquels toutes les paires homologues à la paire (x_h, x_i) rencontrées dans $\pi^*(x_i)$ – l'historique tensoriel de x_i – sont des paires d'éléments identiques.

En d'autres termes, dès qu'une relation d'identité entre deux éléments est infirmée dans le graphe de redondance, toutes les relations entre paires d'éléments homologues au sein de ses

successeurs sont définitivement invalidées et ne peuvent plus être utilisées pour la compression par réplication.

Formellement, on a :

$$\mathcal{C}(x_i|\pi^*(x_i)) = \min_{h \in V_i} (\mathcal{C}(x_i|x_h) + \mathcal{Q}(h|i)) \quad (5.13)$$

C'est à dire l'équation 5.10, où l'historique des prédécesseurs directs $\pi(x_i)$ a été remplacé par le sous-ensemble de prédécesseurs valides V_i (qui dépendent désormais de la séquence X) selon la formule :

$$V_i = \{h \in \pi(i) : \forall a, b \in \pi^*(i), (x_a, x_b) \equiv (x_h, x_i) \implies x_a = x_b\} \quad (5.14)$$

c'est-à-dire les prédécesseurs directs x_h pour lesquels toutes les relations (x_a, x_b) dans l'historique $\pi^*(i)$ qui sont homologues à la relation (x_h, x_i) vérifient $x_a = x_b$.

On peut ensuite appliquer les mêmes simplifications que dans la section précédente. Mais du fait de la similitude de l'équation 5.13 avec l'équation 5.10, le critère obtenu s'énonce exactement de la même façon que pour l'encodage par réplication sans contrainte, en ajustant simplement le concept d'élément concordant, par restriction des éléments de référence éligibles en vertu de la contrainte d'antériorité.

La figure 5.9 illustre l'application de ce nouveau critère à l'exemple de la figure 5.8. On note que la contrainte d'antériorité fait croître le coût de compression sensiblement, ce qui reflète l'absence de concordance des relations dans les classes d'homologie.

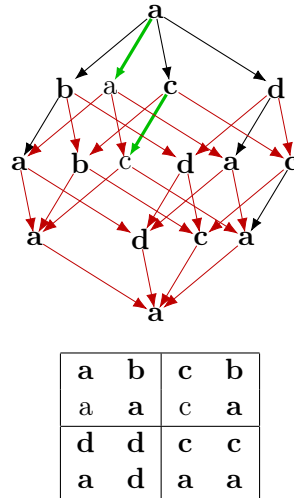


FIGURE 5.9 – Système de la figure 5.8, évaluée avec la fonction à contrainte d'antériorité de la section 5.5.3 ($\mathcal{C} = 14$). Les arêtes en trait rouge et fin indiquent les prédécesseurs invalidés.

5.6 Inférence de la structure interne d'un segment

Les définitions de mesures de complexité d'une séquence selon un patron établies dans la section précédente permettent de définir une méthode de sélection du patron tensoriel représentant au mieux la structure interne d'une séquence. On considère que le meilleur patron pour une

séquence donnée est celui qui permet de minimiser la somme de la complexité du patron et de la complexité de la séquence selon le patron :

$$\hat{P}(X) = \arg \min_{P \text{ admissible pour } X} \mathcal{C}(P) + \mathcal{C}(X|P) \quad (5.15)$$

On reconnaît là un schéma de type MDL, qui consiste à sélectionner le patron tensoriel selon un critère d'économie de la description des données observées que permet ce patron. Le score $\mathcal{C}(X)$, correspondant à \hat{P} fournit une mesure de la complexité de la séquence.

Comme mentionné en section 5.4, des insertions ou des omissions sont possibles dans le patron tensoriel d -cubique, de sorte à rendre admissibles des chaînes différentes de 2^d . Toutefois, seul un nombre limité de configurations sont possibles vu les restrictions apportées à ces déformations dans le cadre du présent travail. En effet, on contraint les insertions et les omissions à être elles-mêmes supportées par un hypercube de dimension $m < d - 1$, incluant la dernière position du patron principal. Ainsi, la première position insérée (ou supprimée) suffit à caractériser la déformation, plusieurs déformations pouvant ensuite être combinées.

On pose donc que les patrons d'un même type ont une complexité comparable, et que la complexité des déformations s'additionne à la complexité du patron hypercubique. Pour les types de patrons considérés, une formulation possible de $\mathcal{C}(P)$ est alors

$$\mathcal{C}(P) = \begin{cases} p_0 & \text{si } P \text{ est hypercubique} \\ p_0 + p_+ & \text{si } P \text{ est avec extension} \\ p_0 + p_- & \text{si } P \text{ est avec omission} \\ p_0 + p_+ + p_- & \text{si } P \text{ est mixte} \end{cases}$$

où p_0, p_+ et p_- sont des constantes.

Dans le cadre de comparaisons entre différents patrons, on peut ignorer la constante p_0 qui est présente dans toutes les formulations. p_+ et p_- sont laissées comme paramètres de la méthode, et déterminées de façon empirique, comme détaillé au chapitre 6.4.

La table 5.1 récapitule le nombre de patrons admissibles pour des longueurs de séquences de l'ordre de celles rencontrées en pratique. Cette table montre que les patrons admissibles sont suffisamment nombreux pour admettre une grande diversité de tailles de séquences, mais qu'ils restent en quantité "raisonnable", ce qui permet aisément leur examen exhaustif de façon automatique ; d'autant que l'admissibilité d'un patron pour une séquence ne dépendant que de la taille de la séquence, il est même possible de pré-calculer pour chaque taille quels sont les patrons à examiner.

Les algorithmes 9 et 10 détaillent une sélection du patron optimal au moyen du critère d'encodage par réplique, respectivement sans et avec contrainte d'antériorité. La version avec contrainte d'antériorité est légèrement plus élaborée, avec notamment le maintien des ensembles V_i qui indiquent pour chaque position les prédécesseurs pour lesquels la contrainte d'antériorité est vérifiée. Ce maintien nécessite que les éléments soient traités dans un ordre topologiquement trié selon \prec , comme par exemple l'ordre chronologique.

Algorithme 9 Optimisation du patron tensoriel selon la critère d'encodage par réplication

Entrée: Une séquence X

pour tout P admissible pour la séquence X **faire**

meilleurScore := ∞

score := 0

pour tout x_i élément de X **faire**

si $\forall x_h \in \pi(x_i), x_h \neq x_i$ **then**

score := **score** + 1

sinon

score := **score** + 0

fin si

fin pour

si P a une omission **then**

score := **score** + c_-

fin si

si P a une extension **then**

score := **score** + c_+

fin si

si **score** < **meilleurScore** **then**

meilleurScore := **score**

\hat{P} := P

fin si

fin pour

Sortie: \hat{P} , le patron optimal pour X et **meilleurScore**, une mesure de la régularité de X

5.7 Inférence de la structure sémiotique d'un morceau

Passant maintenant de la description mésoscopique des segments structurels à la structure macroscopique d'un morceau dans son ensemble, on peut s'appuyer sur la définition de la complexité d'une structure sémiotique proposée dans le chapitre 2 par la formule qui peut être approximée par :

$$\hat{S} = \arg \min_S \left(\mathcal{C}_s(S) + \sum_{i=1}^n \mathcal{C}(s_i) \right), \quad (5.16)$$

où $S = (s_1, s_2, \dots, s_n)$ est une hypothèse de segmentation macroscopique, $\mathcal{C}_s(S)$ représente la complexité de la séquence d'étiquettes de S et $\mathcal{C}(s_i)$ la complexité intrinsèque de la séquence s_i .

Une façon basique de procéder est d'abandonner le terme $\mathcal{C}_s(S)$ et de considérer simplement la complexité d'une segmentation comme égale à la somme des complexités des segments. Cette approche néglige les redondances à long terme, mais permet de simplifier considérablement le processus d'optimisation.

De ce fait, les scores de complexité étant déterminés indépendamment les uns des autres, il est possible de résoudre ce problème en utilisant un algorithme standard de recherche du plus court chemin afin de trouver une séquence de segments minimisant la complexité totale, comme par exemple l'algorithme 7, mentionné au chapitre précédent.

Pour cette recherche de plus court chemin, on applique toutefois quelques a priori sur les hypothèses de segmentation :

Algorithme 10 Optimisation du patron tensoriel selon la critère d'encodage par réplication avec contrainte d'antériorité

Entrée: Une séquence X

pour tout P admissible pour la séquence X **faire**

meilleurScore := ∞

score := 0

pour tout x_i élément de X **faire**

$V_i = \pi^*(x_i)$

fin pour

score := 0

pour tout x_i élément de X , dans un ordre trié topologiquement selon \prec **faire**

tmp := 1

pour tout $x_h \in V_i$ **faire**

si $x_h = x_i$ **then**

tmp := 0

sinon

pour tout $x_k \in \sigma(x_i)$ **faire**

$V_k := \{x_j \in V_k : \overrightarrow{x_j x_k} \equiv \overrightarrow{x_h x_i}\}$

fin pour

fin si

fin pour

score := **score** + **tmp**

fin pour

si P a une omission **then**

score := **score** + c_-

fin si

si P a une extension **then**

score := **score** + c_+

fin si

si **score** < **meilleurScore** **then**

meilleurScore := **score**

$\hat{P} := P$

fin si

fin pour

Sortie: \hat{P} , le patron optimal pour X et **meilleurScore**, une mesure de la régularité de X

- une taille cible de segment est privilégiée (en l’occurrence 32 éléments dans nos expériences), ce qui n’interdit pas une certaine variation autour de cette taille cible, mais entraîne potentiellement une pénalité de déviation,
- les segments trop courts (moins de 8 éléments) sont ignorés, pour éviter une sur-segmentation, sauf en début et fin de morceau (les intro et outro des morceaux de musique pop étant fréquemment irréguliers et/ou s’écartant de la taille cible) ;
- les segments trop longs (plus de 64 éléments) sont ignorés afin d’éviter la prise en considération coûteuse d’hypothèse de segments éloignées la taille ciblée ;
- pour certaines tailles de séquences (39, 44, etc), la mesure de complexité n’est pas définie car aucun des patrons que nous étudions n’est admissible (voir tableau 5.1) ; rien ne s’oppose toutefois à l’ajout de nouvelles formes de patrons qui permettrait leur admissibilité.

Récapitulation et prélude au prochain chapitre

L’ensemble des principes présentés dans ce chapitre seront mis en œuvre dans la section 6.4 en donnant lieu à une évaluation expérimentale, qui permet d’en établir le périmètre de validité pratique sur des données musicales réelles.

Pour l’heure, nous avons défini la notion de patron tensoriel, qui étend l’organisation en carré du modèle Système & Contraste à un nombre arbitraire de relations et d’éléments. Nous avons identifié en particulier un sous-ensemble de ces patrons tensoriels, les patrons hypercubiques et leurs déformations, sur lesquels nous concentrons notre étude.

Nous avons défini une méthode d’estimation de la structure musicale d’un morceau basée sur la segmentation de ce morceau en unités dont la cohérence des éléments constitutifs repose sur un critère de compressibilité conjointe, c’est-à-dire pouvant être décrits de façon économe, à l’aide de peu d’information.

Afin d’estimer cette complexité, nous avons élaboré un protocole général d’encodage de la séquence s’appuyant sur la sélection d’un patron tensoriel, chaque élément étant encodé successivement, selon le contexte défini par sa position dans ce patron et selon un graphe de redondance représentant les relations de similarité entre les éléments à différentes échelles.

La façon dont chaque élément est encodé en fonction du contexte défini est le point crucial de cette méthode. Nous en avons décrit deux formulations simplifiées, qui permettent d’exprimer partiellement les concepts d’analogie régissant les similarités et les relations entre les éléments musicaux d’un même segment. Dans le chapitre suivant, nous détaillons un modèle plus complet permettant de généraliser le modèle Système & Contraste au cas tensoriel.

Nous mettons en évidence les correspondances entre les positions géométriques des éléments du patron tensoriel et les fonctions que les éléments musicaux occupent dans la construction des systèmes logiques au sein des segments structurels. Nous formalisons une méthode permettant d’anticiper la complétion de suites logiques complexes régies par de multiples implications résultant de systèmes d’analogie à différentes échelles.

Nous terminons par un ensemble de résultats expérimentaux qui permettent de positionner les approches tensorielles les unes par rapport aux autres, ainsi que de les comparer avec les approches par grammaires étudiées dans les chapitres 3 et 4.

Chapitre 6

Compression tensorielle par systèmes d’analogie dans les segments musicaux

Pour présenter les principes de la compression tensorielle, nous nous sommes focalisés sur des graphes de redondance où les relations se limitent à une distinction binaire entre éléments identiques et éléments dissemblables.

Un des fondements du modèle Système & Contraste est l’existence de relations sur certaines paires d’éléments qui peuvent être transposées par analogie à d’autres éléments en position homologue conduisant à l’implication projective d’un élément ultérieur, dit élément virtuel. En pratique, l’élément virtuel peut être effectivement réalisé (ce qui résulte en un système non-contrastif) ou bien donner lieu à un déni d’implication qui est alors appelé contraste (et le système est dit contrastif). On retrouve là le principe d’Implication-Réalisation introduit par Narmour [NARMOUR, 1992 ; NARMOUR, 2000] sur les séquences musicales, tel que généralisé par [BIMBOT, 2016] sur les systèmes d’analogie.

Dans le cas d’une modélisation binaire des relations entre éléments, seule la relation identité peut être transposée à une autre paire d’éléments, ce qui limite fortement le champ d’application du modèle Système & Contraste. Dans ce chapitre, nous décrivons l’extension des graphes de redondance à des relations non triviales entre éléments, que l’on peut utiliser pour construire un élément projeté selon une suite logique.

6.1 Analogies sur des patrons hypercubiques

Dans cette section, nous reformulons le modèle Système & Contraste selon le formalisme des graphes de redondance. Nous étudions successivement l’application de ses principes à des patrons hypercubiques de dimension croissante, chacun apportant de nouvelles configurations. Ces extensions aboutissent à une formulation générale permettant l’utilisation d’une forme étendue du modèle Système & Contraste dans des patrons hypercubiques de dimension quelconque.

6.1.1 Propriétés et relations

L’établissement d’un système d’analogie suppose la définition implicite d’un ensemble de propriétés et de relations entre les propriétés des éléments qui composent le système d’analogie.

On définit un modèle de représentation des propriétés comme étant composé de deux ensembles \mathcal{P} et \mathcal{R} :

- \mathcal{P} définit les valeurs que peuvent prendre les éléments du système, comme la note fondamentale ou la modalité pour les accords ;
- \mathcal{R} est l'ensemble des relations retenues comme potentiellement pertinentes entre deux valeurs de \mathcal{P} , comme les écarts entre notes fondamentales ou les changements de modalité pour les accords. Cet ensemble inclut nécessairement la relation identité, que l'on note $\vec{0}$.

On s'intéresse plus précisément dans ce travail au cas où \mathcal{R} permet de décrire de façon complète et non-ambiguë les relations entre les paires de valeurs de \mathcal{P} .

Hypothèse 1 (Non-ambiguïté)

Chaque paire d'éléments de \mathcal{P} vérifie une unique relation de \mathcal{R}

$$\forall (A, B) \in \mathcal{P}^2, \quad \exists! \vec{r} \in \mathcal{R}, \quad A \vec{r} B.$$

On note \vec{AB} la relation vérifiée pour la paire (A, B) .

Pour les illustrations de cette section, nous employons le modèle de représentation des accords et de leurs relations décrit en section 6.2, qui est basé sur le nombre de degrés de séparation entre les accords sur un cercle des tierces.

Nous allons maintenant décliner les procédés de définition des systèmes d'analogies dans des hypercubes de dimension croissante.

6.1.2 Cas du système carré

Le système carré correspond au cas familier du modèle Système & Contraste dans sa forme de base. On rappelle le rôle des éléments du système sur la figure 6.1 : le premier élément, appelé *primer*, permet d'ancrer le système et de définir son point d'origine ; le second et le troisième établissent les relations sur lesquels se construit le système d'analogie, appelées *relations primordiales* ; le quatrième élément (*final*) occupe la seule et unique position contrastive dans cette configuration à 4 éléments.

Formellement, sur le système X on note $\vec{r}_1 = \overline{x_0x_1}$ et $\vec{r}_2 = \overline{x_0x_2}$ pour les relations *primordiales* et plus généralement $\vec{r}_k = \overline{x_0x_k}$ pour la relation d'un élément quelconque avec le primer, que cette relation soit primordiale ou non.

Lorsque x_0 , x_1 et x_2 sont présentés tous les trois, il est possible de construire un système logique d'analogie afin d'anticiper la valeur de l'élément final x_3 .

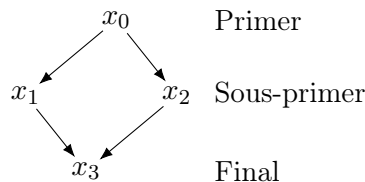


FIGURE 6.1 – Les types d'éléments sur un patron carré.

En toute généralité, cette analogie peut s'établir soit en anticipant que la relation entre x_1 et x_3 sera équivalente à la relation primordiale entre x_0 et x_2 , ce que l'on notera $\overrightarrow{x_0x_2} = \overrightarrow{x_1x_3}^1$, soit en anticipant que la relation entre x_2 et x_3 sera équivalente à la relation primordiale entre x_0 et x_1 , ce que l'on notera $\overrightarrow{x_0x_1} = \overrightarrow{x_2x_3}^2$. On utilise donc un exposant pour distinguer les projections résultant des deux schémas d'analogie possibles, mais comme on le verra un peu plus loin, ces deux projections sont équivalentes quand les relations commutent.

En complément de l'hypothèse 1, on pose les hypothèses suivantes :

Hypothèse 2 (Composition sur \mathcal{R})

L'espace des relations est tel que l'on peut définir sur \mathcal{R} une loi de composition interne \oplus , dont $\vec{0}$ est l'élément neutre, qui respecte la relation de Chasles : $\overrightarrow{AB} \oplus \overrightarrow{BC} = \overrightarrow{AC}$.

NB : la relation identité $\vec{0}$ étant l'élément neutre de cette loi de composition, on parlera de relation *nulle* pour la relation identité et de relation *non-nulle* pour les autres relations.

Hypothèse 3 (Commutativité de \oplus)

La loi de composition \oplus est commutative :

$$\forall(\vec{r}_1, \vec{r}_2) \in \mathcal{R}^2, \quad \vec{r}_1 \oplus \vec{r}_2 = \vec{r}_2 \oplus \vec{r}_1$$

Hypothèse 4 (Existence et unicité de l'analogie)

On peut déterminer de façon unique l'élément qui entretient une relation de \mathcal{R} donnée avec un élément de \mathcal{P} donné.

$$\forall(A, \vec{r}) \in \mathcal{P} \times \mathcal{R}, \quad \exists! B \in \mathcal{P}, \quad \overrightarrow{AB} = \vec{r}$$

On note $B = A + \vec{r}$.

Cet ensemble d'hypothèses rappelle la définition d'un espace affine, ce qui motive l'utilisation de notations faisant écho aux vecteurs géométriques.

Ce formalisme et ces hypothèses permettent d'exprimer d'une façon bien codifiée l'ensemble des principes qui régissent l'établissement des systèmes d'analogie.

À partir de l'hypothèse 2, on peut exprimer la projection obtenue par analogie avec $\overrightarrow{x_0x_2}$:

$$\overrightarrow{x_0x_3}^2 = \overrightarrow{x_0x_2} \oplus \overrightarrow{x_2x_3}^2 \quad (6.1)$$

$$\overrightarrow{x_0x_3}^1 = \overrightarrow{x_0x_2} \oplus \overrightarrow{x_0x_1} \quad (6.2)$$

$$\overrightarrow{x_0x_3}^2 = \vec{r}_2 \oplus \vec{r}_1 \quad (6.3)$$

Similairement, l'analogie duale s'exprime par $\overrightarrow{x_0x_3}^1 = \vec{r}_1 \oplus \vec{r}_2$.

En vertu de l'hypothèse 4, il existe un unique élément impliqué qui vérifie les relations d'analogie, ce qui permet d'écrire les équations précédentes plus simplement :

$$\widehat{x_3}^2 = x_2 + \vec{r}_1 \quad (6.4)$$

$$\widehat{x_3}^1 = x_1 + \vec{r}_2 \quad (6.5)$$

Et plus généralement, on admettra que :

$$\widehat{x_p}^k = x_{p-k} + \vec{r}_k \quad (6.6)$$

Si l'hypothèse de commutativité de \oplus est vérifiée, les deux systèmes d'analogie sont équivalents $\widehat{x}_3 = \widehat{x}_3^1 = \widehat{x}_3^2$.

La comparaison entre l'élément impliqué \widehat{x}_3 et l'élément observé x_3 détermine le caractère contrastif du système. On note $\vec{\gamma}_3 = \overrightarrow{\widehat{x}_3 x_3}$ le *contraste* défini par la relation entre ces deux éléments. Si $\vec{\gamma}_3 = \vec{0}$, on rappelle que l'élément x_3 , et par extension le système, sont dits non-contrastifs. Si $\vec{\gamma}_3 \neq \vec{0}$, ils sont dits contrastifs.

La représentation *in extenso* du système, consiste à énumérer ses éléments $\{x_0, x_1, x_2, x_3\}$. Une représentation alternative du système (appelée *représentation systémique*) consiste à décrire le système par le quadruplet $\{x_0, \vec{r}_1, \vec{r}_2, \vec{\gamma}_3\}$, constitué du primum, des deux relations primordiales et du contraste.

Dans un système carré, tous les éléments x_i et toutes les relations $\overrightarrow{x_i x_j}$ peuvent être exprimés comme une combinaison des éléments de sa représentation systémique. Notamment on a :

$$x_3 = x_0 + \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{\gamma}_3 \quad (6.7)$$

<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">G</td></tr> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">D</td></tr> </table>	G	G	G	D	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">Em</td></tr> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">D</td></tr> </table>	G	Em	G	D	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">Em</td></tr> <tr><td style="padding: 2px 5px;">D</td><td style="padding: 2px 5px;">Bm</td></tr> </table>	G	Em	D	Bm	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">Bbm</td></tr> <tr><td style="padding: 2px 5px;">Eb</td><td style="padding: 2px 5px;">A</td></tr> </table>	G	Bbm	Eb	A
G	G																		
G	D																		
G	Em																		
G	D																		
G	Em																		
D	Bm																		
G	Bbm																		
Eb	A																		
<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">$\vec{0}$</td></tr> <tr><td style="padding: 2px 5px;">$\vec{0}$</td><td style="padding: 2px 5px;">+2</td></tr> </table>	G	$\vec{0}$	$\vec{0}$	+2	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">-1</td></tr> <tr><td style="padding: 2px 5px;">$\vec{0}$</td><td style="padding: 2px 5px;">+3</td></tr> </table>	G	-1	$\vec{0}$	+3	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">-1</td></tr> <tr><td style="padding: 2px 5px;">+2</td><td style="padding: 2px 5px;">$\vec{0}$</td></tr> </table>	G	-1	+2	$\vec{0}$	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="padding: 2px 5px;">G</td><td style="padding: 2px 5px;">+11</td></tr> <tr><td style="padding: 2px 5px;">-8</td><td style="padding: 2px 5px;">+1</td></tr> </table>	G	+11	-8	+1
G	$\vec{0}$																		
$\vec{0}$	+2																		
G	-1																		
$\vec{0}$	+3																		
G	-1																		
+2	$\vec{0}$																		
G	+11																		
-8	+1																		

FIGURE 6.2 – Exemples de systèmes carrés, en représentation *in extenso* (moitié haute) et systémique (moitié basse) en référence à l'ensemble de relations du cercle des tierces (voir section 6.2). Les accords en gras indiquent les positions pour lesquelles les éléments sont discordants, au sens où les relations permettant de les définir sont différentes de l'identité.

La figure 6.2 montre des exemples sur des séquences d'accords. On peut remarquer que plus la séquence est régulière au sens du modèle Système & Contraste, plus un nombre élevé de relations sont égales à l'identité. Ainsi, pour les représentations systémiques, une séquence plus simple se traduit par un nombre plus faible de relations non-nulles dans la représentation, ce qui est une composante importante et facilement mesurable de sa complexité. Par extension de la notion de discordance définie dans le chapitre 5, on dira que de tels éléments sont discordants vis-à-vis du système.

Dans le cas d'objets simples comme des accords, la description d'un élément discordant est du même ordre de complexité que la description d'un nouvel élément. La représentation systémique $(x_0, \vec{r}_1, \vec{r}_2, \vec{\gamma}_3)$ n'est alors pas significativement plus compacte que la représentation *in extenso* (x_0, x_1, x_2, x_3) si toutes les relations élémentaires sont non-nulles.

L'exemple de la figure 6.3 montre un cas impliquant des objets plus complexes comportant plusieurs propriétés. Les relations \vec{r}_1 , \vec{r}_2 et $\vec{\gamma}_3$ ne portent chacune que sur la modification d'une seule de ces propriétés, respectivement le remplissage, la forme et la bordure. Si on considère que ces relations sont moins complexes que les objets complets, la représentation systémique est moins complexe que la représentation *in extenso*, bien qu'aucune des relations élémentaires ne soit nulle.

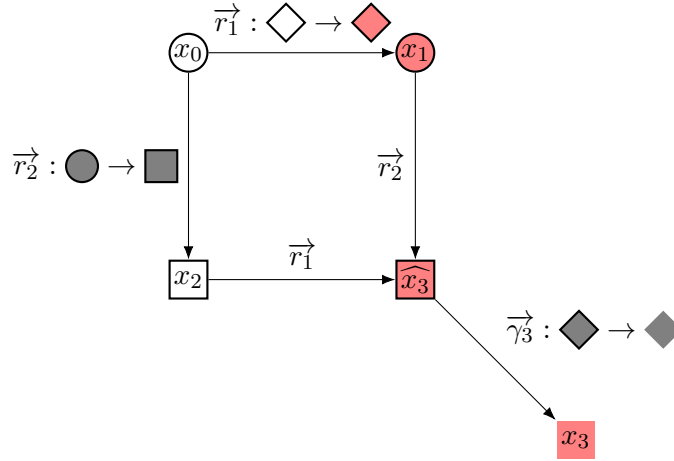


FIGURE 6.3 – Exemple d’un système carré contrastif d’éléments géométriques et les relations primordiales correspondantes. La relation \vec{r}_1 porte sur le remplissage des éléments géométriques, tandis que la relation \vec{r}_2 porte sur leurs formes. Le contraste $\vec{\gamma}_3$ porte sur la bordure de l’élément géométrique.

Ce cas illustre l’intérêt de disposer d’une mesure de complexité précise pour les relations entre les éléments comme exploré dans [LOUBOUTIN et BIMBOT, 2016]. Dans cette thèse, on se limite toutefois à l’étude d’une complexité mesurée par le nombre de relations non-nulles.

6.1.3 Patron cubique

Nous étudions maintenant l’extension du modèle Système & Contraste à un patron tensoriel défini non plus par un carré mais par un cube.

On constate que l’on retrouve dans le système cubique un certain nombre de carrés, correspondant aux faces du cube, formant des sous-systèmes élémentaires de dimension inférieure.

6.1.3.1 Éléments constitutifs

Définition 9 (Sous-système)

Un sous-système est un système inscrit dans un système de dimension supérieure (dit système principal) dont tous les éléments et toutes les relations existent dans le système principal.

On peut noter un sous-système soit de façon extensive, par exemple (x_0, x_1, x_4, x_5) , soit, afin d’alléger la notation, sous la forme $[x_i \diamond x_j]$, où x_i est le primum du sous-système et x_j son élément final (par exemple $[x_0 \diamond x_5]$).

Comme la figure 6.4 le montre, x_0 reste le primum du système cubique, et (x_0, x_1, x_2, x_3) en constitue le premier sous-système carré¹. On retrouve ainsi :

$$\vec{r}_1 = \overrightarrow{x_0 x_1} \tag{6.8}$$

$$\vec{r}_2 = \overrightarrow{x_0 x_2} \tag{6.9}$$

1. Toutefois, x_3 n’est pas le dernier élément du système cubique considéré dans son ensemble, et est donc qualifié d’élément *intermédiaire* au lieu de final.

$$\overrightarrow{x_0x_3} = \overrightarrow{r_1} \oplus \overrightarrow{r_2} \quad (6.10)$$

$$\overrightarrow{\gamma_3} = \overrightarrow{x_3x_3} \quad (6.11)$$

Le cinquième élément, x_4 , a pour seul prédécesseur x_0 , comme les éléments x_1 et x_2 . Il définit ainsi une nouvelle relation primordiale $\overrightarrow{r_4}$ qui, combinée avec $\overrightarrow{r_1}$ ou $\overrightarrow{r_2}$, définit deux nouveaux sous-systèmes carrés, où (respectivement) x_5 et x_6 occupent les positions contrastives. Formellement, on a donc :

$$\overrightarrow{r_4} = \overrightarrow{x_0x_4} \quad (6.12)$$

$$\overrightarrow{x_0x_5} = \overrightarrow{r_1} \oplus \overrightarrow{r_4} \quad (6.13)$$

$$\overrightarrow{\gamma_5} = \overrightarrow{x_5x_5} \quad (6.14)$$

$$\overrightarrow{x_0x_6} = \overrightarrow{r_2} \oplus \overrightarrow{r_4} \quad (6.15)$$

$$\overrightarrow{\gamma_6} = \overrightarrow{x_6x_6} \quad (6.16)$$

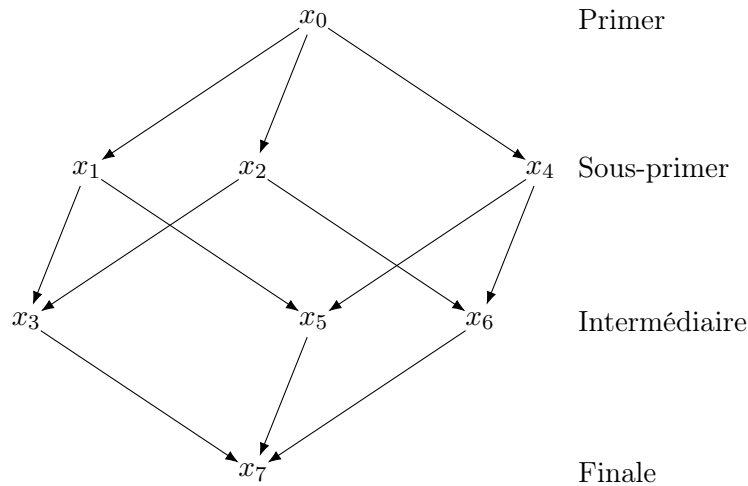


FIGURE 6.4 – Les types de positions sur un patron cubique. On distingue x_0 qui conserve la qualité de primer dans l'ensemble du système, trois sous-primers x_1 , x_2 et x_4 (au lieu de deux dans le carré). Les éléments x_3 , x_5 et x_6 sont en position contrastive intermédiaire et le dernier élément x_7 est en position qualifiée de finale.

On note que la position finale x_7 n'a pas d'équivalent dans le système carré. Elle se retrouve au point de convergence de trois sous-systèmes carrés qui peuvent créer un conflit d'implication si les analogies correspondantes ne concordent pas. C'est sur la gestion de cette situation que se "joue" l'essentiel de l'adaptation du modèle Système & Contraste au système cubique.

Définition 10 (Représentation systémique du système cubique)

La représentation systémique d'un système cubique est l'octuplet $(x_0, \overrightarrow{r_1}, \overrightarrow{r_2}, \overrightarrow{r_4}, \overrightarrow{\gamma_3}, \overrightarrow{\gamma_5}, \overrightarrow{\gamma_6}, \overrightarrow{\gamma_7})$, composé du primer, des trois relations primordiales, des trois contrastes intermédiaires et du contraste final.

6.1.3.2 Projections en position finale

Plusieurs situations peuvent se présenter, qui impactent la façon de définir l'élément projeté $\widehat{x_7}$.

Propagation du système porteur : une première façon est de considérer le système cubique comme étant défini entièrement par ses relations primordiales, c'est-à-dire que

$$\overrightarrow{x_0x_1} = \overrightarrow{x_2x_3} = \overrightarrow{x_4x_5} = \overrightarrow{x_6x_7} \quad (6.17)$$

$$\overrightarrow{x_0x_2} = \overrightarrow{x_1x_3} = \overrightarrow{x_4x_6} = \overrightarrow{x_5x_7} \quad (6.18)$$

$$\overrightarrow{x_0x_4} = \overrightarrow{x_1x_5} = \overrightarrow{x_2x_6} = \overrightarrow{x_3x_7}. \quad (6.19)$$

Dans ce cas, l'élément projeté $\widehat{x_7}$ est unique. On le notera $\widehat{x_7}^\oplus$ et on écrira :

$$\widehat{x_7}^\oplus = x_0 + (\vec{r_1} \oplus \vec{r_2} \oplus \vec{r_4}). \quad (6.20)$$

Cependant, s'il existe des contrastes non nuls sur les positions intermédiaires, l'hypothèse que le système est entièrement défini par ses relations primordiales est contredite : au moins une des équations n'est pas vérifiée. Nous avons donc préféré explorer une deuxième façon de procéder, qui part d'hypothèses moins fortes, mais qui permet de construire un élément projeté sans contradiction dans un plus grand nombre de cas.

Analogie entre faces parallèles : Cette méthode consiste à rechercher une paire de faces parallèles dans le cube qui présente un fort degré d'analogie et dont une des deux contient l'élément final en position contrastive. Au préalable, il convient de définir un concept supplémentaire, celui d'analogie de sous-systèmes carrés.

Définition 11 (Analogie de sous-systèmes carrés)

Deux sous-systèmes carrés (x_i, x_j, x_k, x_m) et $(x_{i'}, x_{j'}, x_{k'}, x_{m'})$ issus d'un même système principal sont dits analogues si ils satisfont les deux propriétés suivantes :

1. ils correspondent à des faces parallèles du système principal dont ils sont issus ;
2. les relations homologues entre éléments dans chacun des 2 systèmes sont égales 2 à 2 :

$$\begin{aligned} \overrightarrow{x_ix_j} &= \overrightarrow{x_{i'}x_{j'}} \\ \overrightarrow{x_ix_k} &= \overrightarrow{x_{i'}x_{k'}} \\ \overrightarrow{x_jx_m} &= \overrightarrow{x_{j'}x_{m'}} \\ \overrightarrow{x_kx_m} &= \overrightarrow{x_{k'}x_{m'}} \end{aligned}$$

Une formulation équivalente de l'analogie de deux sous-systèmes carrés est l'existence d'égalités des relations entre éléments homologues dans les deux systèmes :

$$\overrightarrow{x_ix_{i'}} = \overrightarrow{x_jx_{j'}} = \overrightarrow{x_kx_{k'}} = \overrightarrow{x_mx_{m'}} \quad (6.21)$$

On notera aussi qu'une conséquence de l'analogie est :

$$\overrightarrow{x_ix_m} = \overrightarrow{x_{i'}x_{m'}} \quad (6.22)$$

ce qui revient à dire que les éléments $x_i, x_m, x_{i'}$ et $x_{m'}$ forment un système non contrastif qui est situé sur un plan diagonal, transversal au cube et qui peut constituer un système porteur.

Examinons maintenant plus en détail comment exploiter une analogie entre sous-systèmes carrés pour définir un élément projeté pour x_7 :

Considérons par exemple l'hypothèse que le sous-système $[x_4 \diamond \widehat{x}_7]$ est analogue au sous-système $[x_0 \diamond x_3]$. Dans ce cas, il est nécessaire que $\overrightarrow{x_4 x_5} = \overrightarrow{x_0 x_1}$ et que $\overrightarrow{x_4 x_6} = \overrightarrow{x_0 x_2}$. Conformément à la relation 6.22, on aurait alors :

$$\overrightarrow{x_4 \widehat{x}_7} = \overrightarrow{x_0 x_3} \quad (6.23)$$

Dans la pratique, l'élément x_7 effectivement observé peut dévier de cette attente, mais si le reste du système possède les propriétés d'analogie, cette implication peut se traduire sous la forme

$$\widehat{x}_7 = x_4 + \overrightarrow{r_3} \quad (6.24)$$

ou de façon équivalente

$$\widehat{x}_7 = x_3 + \overrightarrow{r_4} \quad (6.25)$$

Or $x_3 = x_0 + \overrightarrow{r_1} \oplus \overrightarrow{r_2} \oplus \overrightarrow{\gamma_3}$ et donc

$$\widehat{x}_7 = x_0 + \overrightarrow{r_1} \oplus \overrightarrow{r_2} \oplus \overrightarrow{r_4} \oplus \overrightarrow{\gamma_3} \quad (6.26)$$

En d'autres termes, l'élément projeté s'obtient en appliquant la combinaison de l'ensemble des relations primordiales au primer, et en y ajoutant le contraste observé en position 3, dont on s'attend qu'il se produise également en position 7 au sein de $[x_4 \diamond \widehat{x}_7]$ (dans le cas non-contrastif).

Cette hypothèse prend en compte la possible présence d'un contraste en x_3 . Toutefois, si on observe un contraste en x_5 ou en x_6 , ceci contredit l'analogie entre les deux sous-systèmes², et vient invalider la projection \widehat{x}_7 , ci-dessus définie.

Définition 12 (Implication valide)

On dit qu'une implication \widehat{x}_7 est valide si l'hypothèse d'analogie sur laquelle elle est basée n'est pas contredite par les prédécesseurs de x_7 dans $[x_0 \diamond x_7]$.

Alternativement, on peut considérer deux autres hypothèses d'analogie entre sous-systèmes : d'une part entre $[x_0 \diamond x_5]$ et $[x_2 \diamond x_7]$ et d'autre part entre $[x_0 \diamond x_6]$ et $[x_1 \diamond x_7]$. Les définitions correspondantes des éléments projetés deviennent respectivement :

$$\widehat{x}_7 = x_5 + \overrightarrow{r_2} = x_0 + \overrightarrow{r_1} \oplus \overrightarrow{r_2} \oplus \overrightarrow{r_4} \oplus \overrightarrow{\gamma_5} \quad (6.27)$$

$$\widehat{x}_7 = x_6 + \overrightarrow{r_1} = x_0 + \overrightarrow{r_1} \oplus \overrightarrow{r_2} \oplus \overrightarrow{r_4} \oplus \overrightarrow{\gamma_6} \quad (6.28)$$

On constate alors que les formules des équations 6.25 à 6.28 correspondent à trois possibilités d'implications, éventuellement différentes, et il convient d'utiliser une notation permettant de les distinguer. En concordance avec la notation utilisée préalablement pour le système carré, on notera donc

$$\widehat{x}_7^k = x_{7-k} + \overrightarrow{r}_k \quad (6.29)$$

$$= x_0 + \overrightarrow{r}_1 \oplus \overrightarrow{r}_2 \oplus \overrightarrow{r}_4 \oplus \overrightarrow{\gamma}_k \quad (6.30)$$

Dans les trois cas, l'obtention de l'élément projeté repose implicitement sur un système transversal constitué des 4 éléments x_0 , x_k , x_{7-k} et \widehat{x}_7^k . Ceux-ci ne forment pas un sous-système car les relations ne sont pas toutes portées par des arêtes dans le patron tensoriel mais ils peuvent servir pour construire des implications dans le cas d'analogies entre sous-systèmes. On appelle *éléments pivots* de ce système transversal les éléments x_k et x_{7-k} .

2. En effet, si $\overrightarrow{\gamma}_5 \neq \overrightarrow{0}$ alors $\overrightarrow{x_4 x_5} = \overrightarrow{r}_1 \oplus \overrightarrow{\gamma}_5 \neq \overrightarrow{x_0 x_1}$. De même, si $\overrightarrow{\gamma}_6 \neq \overrightarrow{0}$ alors $\overrightarrow{x_4 x_6} = \overrightarrow{r}_2 \oplus \overrightarrow{\gamma}_6 \neq \overrightarrow{x_0 x_2}$.

On remarquera que dans la formule 6.29, k et $7 - k$ jouent des rôles interchangeables et correspondent à des positions symétriques par rapport au centre du cube³. Plus généralement, on notera que tout couple d'éléments (x_k, x_{p-k}) occupe des positions symétriques dans $[x_0 \diamond x_p]$.

Dans le système cubique, les paires d'éléments symétriques sont (x_0, x_7) , (x_1, x_6) , (x_2, x_5) et (x_3, x_4) , comme illustré en figure 6.5. On notera $\overline{x_k}$ le symétrique de x_k . La figure 6.6 illustre les 3 constructions par analogie possibles dans un cube, que nous venons de décrire.

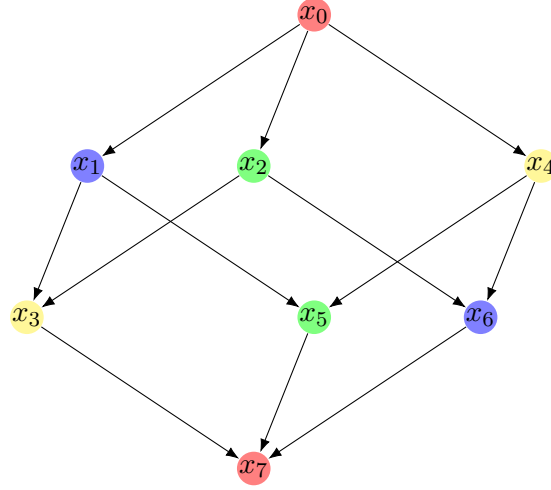


FIGURE 6.5 – Les paires d'éléments symétriques dans le système cubique

6.1.3.3 Schémas implicatifs

On peut donc distinguer 3 cas de figure pour les systèmes cubiques :

- Si tous les éléments en position intermédiaire sont non-contrastifs ($\vec{\gamma}_3 = \vec{\gamma}_5 = \vec{\gamma}_6 = \vec{0}$), alors toutes les implications sont valides et concordantes et tous les systèmes transversaux permettent de construire \widehat{x}_7 de façon unique :

$$\widehat{x}_7 = \widehat{x}_7^\oplus = \widehat{x}_7^3 = \widehat{x}_7^5 = \widehat{x}_7^6 = x_0 + (\vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4). \quad (6.31)$$

La figure 6.7a illustre ce cas.

- Si exactement un seul élément en position intermédiaire x_k est contrastif ($\vec{\gamma}_k \neq \vec{0}$), la seule implication valide consiste à construire \widehat{x}_7 à partir du système transversal utilisant le couple $(x_k, \overline{x_k})$ comme pivots :

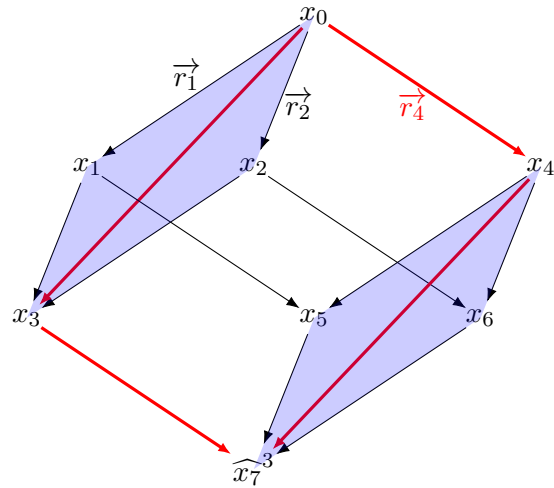
$$\widehat{x}_7^k = \overline{x_k} + \vec{r}_k \quad (6.32)$$

$$= x_0 + \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{\gamma}_k \quad (6.33)$$

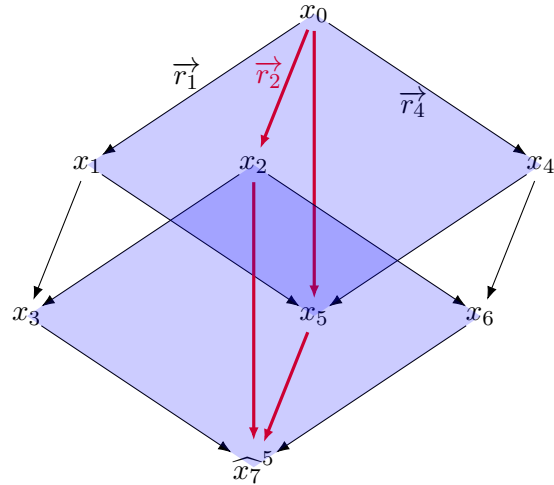
Ce cas est illustré figure 6.7b.

- Si plusieurs éléments en position intermédiaire sont contrastifs, ni l'implication par propagation des relations primordiales, ni aucune des implications par analogie de sous-systèmes ne sont valides. Aucun système transversal ne permet de construire \widehat{x}_7 dans le cadre d'un paradigme d'analogie (cf exemple en figure 6.7c). Nous verrons dans la sous-section suivante comment cette difficulté peut être contournée.

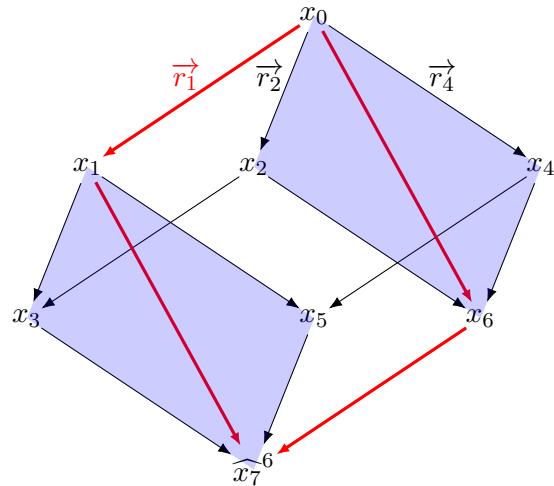
3. On notera que pour $k = 7$ ou $k = 0$, cette formule n'apporte aucune capacité d'implication ($\widehat{x}_7^7 = \widehat{x}_7^0 = x_7$)



(a) Implication via x_3 (eq. 6.25)



(b) Implication via x_5 (eq. 6.27)



(c) Implication via x_6 (eq. 6.28)

FIGURE 6.6 – Visualisation des implications de \widehat{x}_7 pour des systèmes comportant des sous-contrastes. Les systèmes transversaux des implications sont mis en évidence.

A	E	F#m	C#m
A	E	F#m	C#m

(a) Système cubique sans contraste intermédiaire

A	E	F#m	C#m
A	A	F#m	F#m

(b) Système cubique avec contraste intermédiaire

A	E	F#m	D
A	A	F#m	?

(c) Système cubique sans implication valide sur le dernier accord

FIGURE 6.7 – Exemples d’implications (en italique) sur l’élément final de systèmes cubiques à base d’accords

On peut remarquer que l’implication par composition \widehat{x}_7^\oplus n’est valide que si toutes les implications par analogie de sous-systèmes sont également valides et concordantes. Il n’existe donc aucun cas où cette implication est à la fois valide et différente des autres. En ce sens, l’implication par propagation du système porteur est d’une puissance moindre pour construire les implications dans le système cubique.

6.1.3.4 Constitution du contraste

Puisque le système cubique ne produit pas toujours un élément impliqué \widehat{x}_7 , on distingue deux cas pour la définition du contraste $\vec{\gamma}_7$ et du caractère contrastif de x_7 .

- Si une ou plusieurs implications sont valides, on procède, comme pour les éléments intermédiaires, c’est-à-dire qu’on rapporte l’élément observé x_7 à l’élément impliqué \widehat{x}_7 pour déterminer le contraste. On a ainsi

$$\vec{\gamma}_7 = \overrightarrow{\widehat{x}_7 x_7}. \quad (6.34)$$

x_7 est contrastif si $\vec{\gamma}_7 \neq \vec{0}$, et non-contrastif sinon.

- Si aucune implication n’est valide, l’élément impliqué \widehat{x}_7 n’est pas définissable par un système d’analogie. On sort du cadre conceptuel du modèle Système & Contraste tel que défini dans son contexte originel. On peut toutefois définir $\vec{\gamma}_7$ en rapportant x_7 à un élément virtuel défini par extension comme :

$$\widehat{x}_7 = x_0 + \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{\gamma}_3 \oplus \vec{\gamma}_5 \oplus \vec{\gamma}_6 \quad (6.35)$$

soit

$$\vec{\gamma}_7 = \overrightarrow{\widehat{x}_7 x_7} \quad (6.36)$$

Dans cette situation, $\vec{\gamma}_7$ n’est pas construit à partir d’une implication par analogie valide, la question se pose de considérer x_7 comme contrastif même lorsque $\vec{\gamma}_7 = \vec{0}$ ⁴.

Afin de distinguer les contrastes des deux situations ci-dessus, on utilisera lorsque cela est nécessaire les notions de contraste *simple* et de contraste *recomposé*. Ainsi, dans un système cubique, tous les contrastes situés en position intermédiaire sont simples, mais le contraste $\vec{\gamma}_7$ peut être recomposé.

6.1.3.5 Illustrations

La figure 6.8 montre la forme générale des représentations *in extenso* et systémiques pour un système cubique, tandis que la figure 6.9 montre des exemples concrets. Comme pour le système carré, on peut constater que les séquences plus simples ont des représentations systémiques comportant moins de relations non-nulles.

4. Pour des raisons liées à notre implémentation, c’est cette option que nous avons retenu. Il est également envisageable de considérer \widehat{x}_7 comme une implication valide, mais cette situation ne concerne qu’un faible nombre de cas.

x_0	x_1	x_4	x_5
x_2	x_3	x_6	x_7
x_0	\vec{r}_1	\vec{r}_4	$\vec{\gamma}_5$
\vec{r}_2	$\vec{\gamma}_3$	$\vec{\gamma}_6$	$\vec{\gamma}_7$

FIGURE 6.8 – Représentation *in extenso* (haut) et systémique (bas) pour un système cubique.

Dm	Am	Dm	Am	Ab	Ab	E	E	Abm	$C\#m$	A	B
Bb	F	Bb	C	$F\#$	$F\#$	Eb	Eb	D	Ab	Ab	$C\#m$
Dm	-2	$\vec{0}$	$\vec{0}$	Ab	$\vec{0}$	-8	$\vec{0}$	Abm	-2	-3	+6
-1	$\vec{0}$	$\vec{0}$	+2	-4	$\vec{0}$	-10	$\vec{0}$	-5	-10	-9	-3

FIGURE 6.9 – Exemples de systèmes cubiques et leurs représentations systémiques

Une propriété intéressante de la représentation systémique du cube est que la représentation d’une séquence de type **aabbccdd** où tous les éléments ont été dupliqués a le même nombre de valeurs non nulles que la représentation systémique sur un système carré de la séquence sans duplication **abcd**. Cette propriété est particulièrement avantageuse dans le cas où le tempo une séquence a été surestimé ou sous-estimé d’un facteur 2 comme cela peut-être le cas. De la même façon, les représentations systémiques d’une séquence et de cette même séquence répétée deux fois ont le même nombre de valeur non-nulles. La figure 6.10 illustre ces propriétés.

A	A	A	A	E	E	A	A
E	D	A	A	D	D	E	D
A	$\vec{0}$	A	$\vec{0}$	+2	$\vec{0}$	A	$\vec{0}$
+2	-4	$\vec{0}$	$\vec{0}$	-4	$\vec{0}$	+2	-4
		$\vec{0}$	$\vec{0}$			$\vec{0}$	$\vec{0}$

FIGURE 6.10 – Illustration de l’effet d’une duplication et d’une répétition

6.1.4 Cas tesseractique

L’étape suivante vers une formulation générale est l’extension du modèle sur un hypercube de dimension 4, ou système tesseractique.

La principale différence réside dans les multiples relations d’analogie qui permettent de calculer une projection en position finale, qui est désormais l’élément x_{15} . Pour commencer, nous précisons brièvement les éléments constitutifs d’un système tesseractique.

6.1.4.1 Éléments constitutifs du système tesseractique

Dans le système tesseractique, on distingue 5 classes d’éléments selon leur niveau de profondeur par rapport au primum :

- le primum x_0 est l’unique élément de niveau 0,
- les sous-primums, x_1, x_2, x_4 et x_8 sont des éléments de niveau 1,
- les éléments en position intermédiaire agissant comme les éléments finaux de systèmes carrés, $x_3, x_5, x_6, x_9, x_{10}$ et x_{12} sont des éléments de niveau 2,
- les éléments en position intermédiaire agissant comme les éléments finaux de systèmes cubiques, x_7, x_{11}, x_{13} et x_{14} sont des éléments de niveau 3,

— enfin, l'élément final x_{15} du système tesseractique est de niveau 4.

La figure 6.11 représente les relations d'un système tesseractique, regroupant horizontalement les éléments d'un même niveau.

Définition 13 (Niveau d'un élément)

On définit $N(x_i)$ le niveau de l'élément x_i dans le système P comme le nombre d'arêtes le séparant du primer x_0 dans le patron tensoriel.

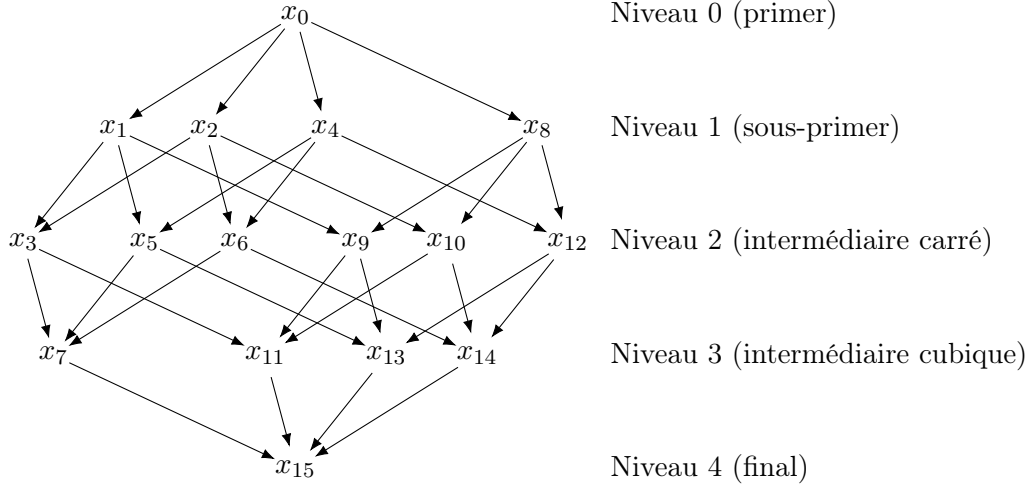


FIGURE 6.11 – Système tesseractique et niveau des positions de ses éléments

Définition 14 (Représentation systémique du tesseract)

On notera $(x_0, \vec{r}_1, \vec{r}_2, \vec{r}_4, \vec{r}_8, \vec{\gamma}_3, \vec{\gamma}_5, \vec{\gamma}_6, \vec{\gamma}_7, \vec{\gamma}_9, \vec{\gamma}_{10}, \vec{\gamma}_{11}, \vec{\gamma}_{12}, \vec{\gamma}_{13}, \vec{\gamma}_{14}, \vec{\gamma}_{15})$ la représentation systémique d'un tesseract, composée du primer, des 4 relations primordiales vers les sous-primers, des 6 contrastes intermédiaires carrés (niveau 2), des 4 contrastes intermédiaires cubiques (niveau 3) et du contraste final (niveau 4).

On peut alors transposer les définitions du système cubique :

— les sous-primers x_1, x_2, x_4 et le nouveau sous-primer x_8 définissent les relations primordiales $\vec{r}_1, \vec{r}_2, \vec{r}_4$ et \vec{r}_8 :

$$\vec{r}_1 = \overrightarrow{x_0x_1} \tag{6.37}$$

$$\vec{r}_2 = \overrightarrow{x_0x_2} \tag{6.38}$$

$$\vec{r}_4 = \overrightarrow{x_0x_4} \tag{6.39}$$

$$\vec{r}_8 = \overrightarrow{x_0x_8} \tag{6.40}$$

— on construit pour les éléments de niveau 2 les sous-systèmes carrés correspondants :

$$\overrightarrow{x_0x_3} = \vec{r}_1 \oplus \vec{r}_2 \quad \vec{\gamma}_3 = \overrightarrow{x_3x_3} \tag{6.41}$$

$$\overrightarrow{x_0x_5} = \vec{r}_1 \oplus \vec{r}_4 \quad \vec{\gamma}_5 = \overrightarrow{x_5x_5} \tag{6.42}$$

$$\overrightarrow{x_0x_6} = \vec{r}_2 \oplus \vec{r}_4 \quad \vec{\gamma}_6 = \overrightarrow{x_6x_6} \tag{6.43}$$

$$\overrightarrow{x_0x_9} = \vec{r}_1 \oplus \vec{r}_8 \quad \vec{\gamma}_9 = \overrightarrow{x_9x_9} \tag{6.44}$$

$$\overrightarrow{x_0x_{10}} = \vec{r}_2 \oplus \vec{r}_8 \quad \vec{\gamma}_{10} = \overrightarrow{x_{10}x_{10}} \tag{6.45}$$

$$\overrightarrow{x_0x_{12}} = \vec{r}_4 \oplus \vec{r}_8 \quad \vec{\gamma}_{12} = \overrightarrow{x_{12}x_{12}} \tag{6.46}$$

L'adaptation du niveau 3 nécessite d'introduire une notation spécifique pour désigner la symétrie d'un élément dans un sous-système du tesseract.

Définition 15 (Élément symétrique dans un sous-système)

Soit x_k un élément d'un sous-système $[x_m \diamond x_n]$. On désignera par ${}^m\overline{x_k}^n$ son élément symétrique par rapport au centre de ce sous-système, c'est-à-dire le sommet opposé.

Si x_m est le primum ($m = 0$) et si x_n est l'élément final ($n = 15$ pour le tesseract), on notera simplement $\overline{x_k}$. Afin d'alléger la notation, on ne note m et/ou n que si respectivement x_m n'est pas le primum du système principal et/ou x_n n'est pas l'élément final du système principal.

Muni du concept d'élément symétrique dans un sous-système, l'examen de la figure 6.11 permet de généraliser la définition de l'implication valide établie dans le cadre des projections en position finale de système cubique :

Définition 16 (Implication valide sur un sous-système cubique)

L'implication \widehat{x}_i^h , basée sur l'analogie entre les sous-systèmes carrés $[x_0 \diamond x_h]$ et $[\overline{x_h}^i \diamond x_i]$, est dite valide si l'hypothèse d'analogie n'est pas contredite par les prédécesseurs de x_i dans le sous-système cubique $[x_0 \diamond x_i]$, lequel est constitué des deux systèmes carrés précédents.

Dans le cas du tesseract, cette définition s'applique aux valeurs de i égales à 7, 11, 13 et 14, et aux valeurs de h appartenant respectivement à $\{3, 5, 6\}$, $\{3, 9, 10\}$, $\{5, 9, 12\}$ et $\{6, 10, 12\}$ (les trois prédécesseurs directs de chacun des éléments de niveau 3).

Ainsi peut-on écrire l'ensemble des implications valides sur les éléments de niveau 3 et les contrastes correspondant sous la forme :

$$\widehat{x}_7 = \begin{cases} x_6 + \vec{r}_1 & \text{si } \vec{\gamma}_3 = \vec{\gamma}_5 = \vec{0} \\ x_5 + \vec{r}_2 & \text{si } \vec{\gamma}_3 = \vec{\gamma}_6 = \vec{0} \\ x_3 + \vec{r}_4 & \text{si } \vec{\gamma}_5 = \vec{\gamma}_6 = \vec{0} \end{cases} \quad \vec{\gamma}_7 = \overrightarrow{\widehat{x}_7 x_7} \quad (6.47)$$

$$\widehat{x}_{11} = \begin{cases} x_{10} + \vec{r}_1 & \text{si } \vec{\gamma}_3 = \vec{\gamma}_9 = \vec{0} \\ x_9 + \vec{r}_2 & \text{si } \vec{\gamma}_3 = \vec{\gamma}_{10} = \vec{0} \\ x_3 + \vec{r}_8 & \text{si } \vec{\gamma}_9 = \vec{\gamma}_{10} = \vec{0} \end{cases} \quad \vec{\gamma}_{11} = \overrightarrow{\widehat{x}_{11} x_{11}} \quad (6.48)$$

$$\widehat{x}_{13} = \begin{cases} x_{12} + \vec{r}_1 & \text{si } \vec{\gamma}_5 = \vec{\gamma}_9 = \vec{0} \\ x_9 + \vec{r}_4 & \text{si } \vec{\gamma}_5 = \vec{\gamma}_{12} = \vec{0} \\ x_5 + \vec{r}_8 & \text{si } \vec{\gamma}_9 = \vec{\gamma}_{12} = \vec{0} \end{cases} \quad \vec{\gamma}_{13} = \overrightarrow{\widehat{x}_{13} x_{13}} \quad (6.49)$$

$$\widehat{x}_{14} = \begin{cases} x_{12} + \vec{r}_2 & \text{si } \vec{\gamma}_6 = \vec{\gamma}_{10} = \vec{0} \\ x_{10} + \vec{r}_4 & \text{si } \vec{\gamma}_6 = \vec{\gamma}_{12} = \vec{0} \\ x_6 + \vec{r}_8 & \text{si } \vec{\gamma}_{10} = \vec{\gamma}_{12} = \vec{0} \end{cases} \quad \vec{\gamma}_{14} = \overrightarrow{\widehat{x}_{14} x_{14}} \quad (6.50)$$

La figure 6.12 illustre un cas de système tesseractique sur une séquence de 16 accords.

Reste à étudier la projection \widehat{x}_{15} pour le dernier élément x_{15} qui fait spécifiquement l'objet de la sous-section suivante.

F	F	Bb	Bb
C	C	F	F
F	Am	Bb	Dm
Dm	F	Gm	Bb
F	$\vec{0}$	-2	$\vec{0}$
+2	$\vec{0}$	$\vec{0}$	$\vec{0}$
$\vec{0}$	+3	$\vec{0}$	$\vec{0}$
-3	-2	$\vec{0}$	$\vec{0}$

FIGURE 6.12 – Exemple d’un système d’accords sur un tesseract et sa représentation systémique.

6.1.4.2 Projection en position finale du système tesseractique

Dans le même esprit que ce qui a été fait pour le système cubique, la construction de l’élément projeté en position finale du système tesseractique peut s’appuyer sur la propagation du système porteur :

$$\widehat{x}_{15}^{\oplus} = x_0 + (\vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{r}_8). \quad (6.51)$$

Néanmoins, pour les mêmes raisons que précédemment, cette méthode n’est pas la plus intéressante.

L’autre famille de solutions consiste à baser la projection \widehat{x}_{15} sur l’hypothèse d’analogie entre sous-systèmes. Toutefois, la situation s’avère plus complexe (mais plus riche de possibilités), pour construire un schéma implicatif. Il existe en effet deux types d’options possibles : soit construire une analogie entre deux sous-systèmes cubiques, soit une analogie entre quatre sous-systèmes carrés.

Analogie entre sous-systèmes cubiques Étudions d’abord le cas d’une analogie entre systèmes cubiques. On peut aisément reprendre la définition d’une analogie entre sous-systèmes carrés (dans le cube) pour l’étendre à des analogies entre sous-systèmes cubiques (dans le tesseract). En effet, dans un tesseract, il existe 4 paires d’hypercubes homologues entre lesquels on peut considérer des analogies : $[x_0 \diamond x_7]$ et $[x_8 \diamond x_{15}]$, $[x_0 \diamond x_{11}]$ et $[x_4 \diamond x_{15}]$, $[x_0 \diamond x_{12}]$ et $[x_2 \diamond x_{15}]$ ou $[x_0 \diamond x_{14}]$ et $[x_1 \diamond x_{15}]$ (voir figure 6.13).

Ainsi, si l’on émet l’hypothèse que les sous-systèmes cubiques $[x_0 \diamond x_7]$ et $[x_8 \diamond x_{15}]$ sont analogues, ceci se traduit par des égalités entre les relations correspondantes des deux sous-systèmes. On a notamment : $\overrightarrow{x_8 x_{8+i}} = \overrightarrow{x_0 x_i}$ pour $1 \leq i \leq 6$ ainsi que :

$$\overrightarrow{x_8 \widehat{x}_{15}^8} = \overrightarrow{x_0 x_7} \quad (6.52)$$

Or, $\overrightarrow{x_0 x_7} = \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{\gamma}_3 \oplus \vec{\gamma}_5 \oplus \vec{\gamma}_6 \oplus \vec{\gamma}_7$. Donc, en faisant intervenir \vec{r}_8 , on peut écrire :

$$\widehat{x}_{15}^8 = x_0 + \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{r}_8 \oplus \vec{\gamma}_3 \oplus \vec{\gamma}_5 \oplus \vec{\gamma}_6 \oplus \vec{\gamma}_7 \quad (6.53)$$

Comme dans le cas du cube, il faut que toutes les équations d’analogie soient vérifiées pour que la projection \widehat{x}_{15}^8 soit valide. Dans le cas présent, cela n’est pas incompatible avec la présence de contrastes en x_3 , x_5 , x_6 et/ou x_7 , c’est-à-dire en position de niveau 2 et de niveau 3 dans le *premier* sous-système cubique. Par contre, pour que l’implication sur \widehat{x}_{15}^8 soit valide, il est

nécessaire que les éléments de même niveau situés dans le *deuxième* sous-système cubique⁵ respectent les équations d'analogie, ce qui entraîne qu'ils ne soient pas contrastifs dans le système tesseractique dans son ensemble⁶.

Des formules comparables peuvent être écrites dans le cas des trois autres analogies entre paires d'hypercubes, l'ensemble se résumant comme dans le cas de l'équation 6.29 à la formule générale pour $k \in \{1, 2, 4, 8\}$:

$$\widehat{x}_{15}^k = x_{15-k} + \vec{r}_k = \overline{x}_k + \vec{r}_k \quad (6.54)$$

On rappelle que ceci signifie que l'élément projeté selon l'hypothèse d'analogie entre $[x_0 \diamond \overline{x}_k]$ et $[x_k \diamond x_{15}]$ peut s'obtenir de façon équivalente à partir d'un système carré transversal constitué des éléments x_0, x_k, \overline{x}_k dont \widehat{x}_{15}^k est l'élément projeté.

Analogie entre sous-systèmes carrés Dans le cas du tesseract, un second type de construction d'élément projeté est possible en s'appuyant sur des analogies mettant en jeu 4 sous-systèmes carrés homologues.

Prenons par exemple les 4 sous-systèmes carrés $[x_0 \diamond x_3], [x_4 \diamond x_7], [x_8 \diamond x_{11}]$ et $[x_{12} \diamond x_{15}]$, tels que représentés sur la figure 6.14a. Les propriétés suivantes découlent directement de l'hypothèse d'analogie :

$$\overline{x_0x_1} \rightarrow = \overline{x_4x_5} \rightarrow = \overline{x_8x_9} \rightarrow = \overline{x_{12}x_{13}} \rightarrow \quad (6.55)$$

$$\overline{x_0x_2} \rightarrow = \overline{x_4x_6} \rightarrow = \overline{x_8x_{10}} \rightarrow = \overline{x_{12}x_{14}} \rightarrow \quad (6.56)$$

ainsi que :

$$\overline{x_0x_3} \rightarrow = \overline{x_4x_7} \rightarrow = \overline{x_8x_{11}} \rightarrow = \overline{x_{12}\widehat{x}_{15}^3} \rightarrow \quad (6.57)$$

La dernière équation fait apparaître un système transversal $(x_0, x_3, x_{12}, \widehat{x}_{15}^3)$ qui peut donc servir de support, comme pour les cas précédents, au calcul d'un élément projeté (si l'hypothèse d'analogie considérée est valide) :

$$\widehat{x}_{15}^3 = x_3 + \vec{r}_{12} \quad (6.58)$$

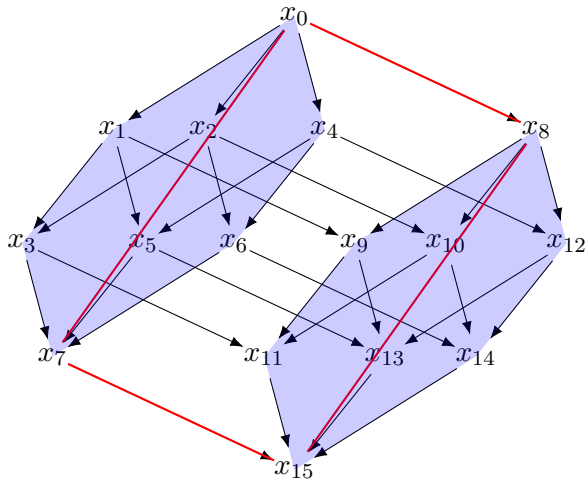
et la décomposition de x_3 et de \vec{r}_{12} en éléments systémiques permet d'obtenir :

$$\widehat{x}_{15}^3 = x_0 + \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{r}_8 \oplus \vec{\gamma}_3 \oplus \vec{\gamma}_{12} \quad (6.59)$$

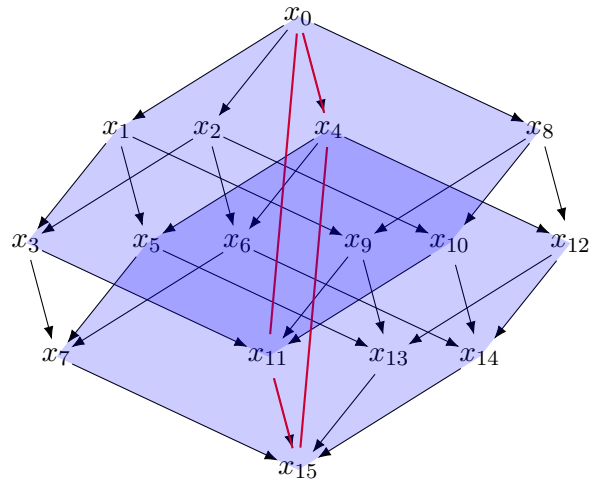
Ici, l'implication reste valide s'il existe des contrastes en x_3 ou en x_{12} . En revanche, toutes les autres positions de niveau 2 et 3 doivent être non-contrastives dans le tesseract pour que l'hypothèse d'analogie soit valide.

5. x_9, x_{10}, x_{12} pour le niveau 2 et x_{11}, x_{13}, x_{14} pour le niveau 3.

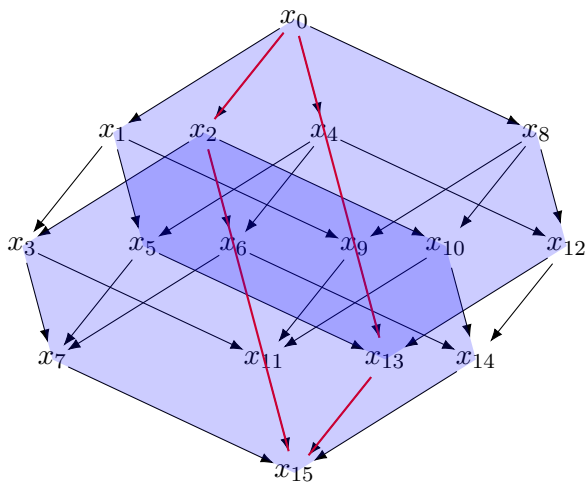
6. En l'espèce, \widehat{x}_{15}^8 est valide dès lors que : $\vec{\gamma}_9 = \vec{\gamma}_{10} = \vec{\gamma}_{12} = \vec{\gamma}_{11} = \vec{\gamma}_{13} = \vec{\gamma}_{14} = \vec{0}$



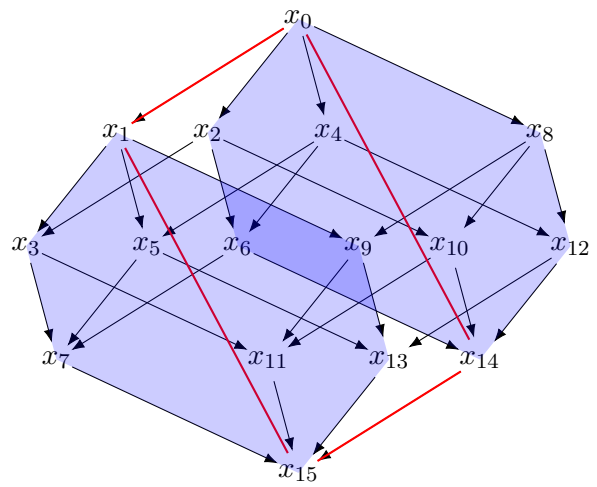
(a) Analogie entre $[x_0, x_7]$ et $[x_8, x_{15}]$



(b) Analogie entre $[x_0, x_{11}]$ et $[x_4, x_{15}]$



(c) Analogie entre $[x_0, x_{13}]$ et $[x_2, x_{15}]$



(d) Analogie entre $[x_0, x_{14}]$ et $[x_1, x_{15}]$

FIGURE 6.13 – Mise en évidence des différentes analogies entre sous-systèmes cubiques utilisables pour l'implication de \widehat{x}_{15} .

L'inventaire des possibilités d'analogies entre sous-systèmes carrés permet de recenser 6 possibilités, qui s'avèrent équivalentes 2 à 2, comme le suggère d'ailleurs la symétrie des positions des deux termes de contraste dans l'équation 6.59. On a donc 3 hypothèses d'analogie possibles entre sous-systèmes carrés qui conduisent toutes les 3 à la formulation désormais familière :

$$\widehat{x_{15}}^k = x_k + \overrightarrow{x_0 x_k}. \quad (6.60)$$

qui s'étend donc aux valeurs de $k \in \{3, 5, 6\}$.

Ainsi, l'élément projeté $\widehat{x_{15}}^k$ s'obtient à nouveau par implication en s'appuyant sur le système carré transversal correspondant aux positions de x_0 , x_k , $\overline{x_k}$ et x_{15} .

On précise également que l'implication $\widehat{x_{15}}^k$ est valide si et seulement si les éléments contrastifs du système (à l'exception de $\overrightarrow{\gamma_{15}}$) sont limités aux sous-systèmes $[x_0 \diamond x_k]$ et $[x_0, \overline{x_k}]$.

6.1.4.3 Schémas implicatifs par éléments pivots

On s'intéresse maintenant à l'inventaire des implications valides selon les configurations de contrastes observés, et on établit un principe général qui les encapsule toutes.

Dans la sous-section précédente, on a défini 7 schémas d'implication possibles sur x_{15} : 4 par analogie de sous-systèmes cubiques et 3 par analogie de sous-systèmes carrés. En examinant chacun d'eux successivement, on peut établir les conditions de validité selon les contrastes observés dans les éléments antérieurs du système tesseractique :

— la projection $\widehat{x_{15}}^1$ (équivalente à la projection $\widehat{x_{15}}^{14}$) est valide si et seulement si

$$\overrightarrow{\gamma_3} = \overrightarrow{\gamma_5} = \overrightarrow{\gamma_7} = \overrightarrow{\gamma_9} = \overrightarrow{\gamma_{11}} = \overrightarrow{\gamma_{13}} = \overrightarrow{0}$$

— la projection $\widehat{x_{15}}^2$ (équivalente à la projection $\widehat{x_{15}}^{13}$) est valide si et seulement si

$$\overrightarrow{\gamma_3} = \overrightarrow{\gamma_6} = \overrightarrow{\gamma_7} = \overrightarrow{\gamma_{10}} = \overrightarrow{\gamma_{11}} = \overrightarrow{\gamma_{14}} = \overrightarrow{0}$$

— la projection $\widehat{x_{15}}^4$ (équivalente à la projection $\widehat{x_{15}}^{11}$) est valide si et seulement si

$$\overrightarrow{\gamma_5} = \overrightarrow{\gamma_6} = \overrightarrow{\gamma_7} = \overrightarrow{\gamma_{12}} = \overrightarrow{\gamma_{13}} = \overrightarrow{\gamma_{14}} = \overrightarrow{0}$$

— la projection $\widehat{x_{15}}^7$ (équivalente à la projection $\widehat{x_{15}}^8$) est valide si et seulement si

$$\overrightarrow{\gamma_9} = \overrightarrow{\gamma_{10}} = \overrightarrow{\gamma_{11}} = \overrightarrow{\gamma_{12}} = \overrightarrow{\gamma_{13}} = \overrightarrow{\gamma_{14}} = \overrightarrow{0}$$

— la projection $\widehat{x_{15}}^3$ (équivalente à la projection $\widehat{x_{15}}^{12}$) est valide si et seulement si

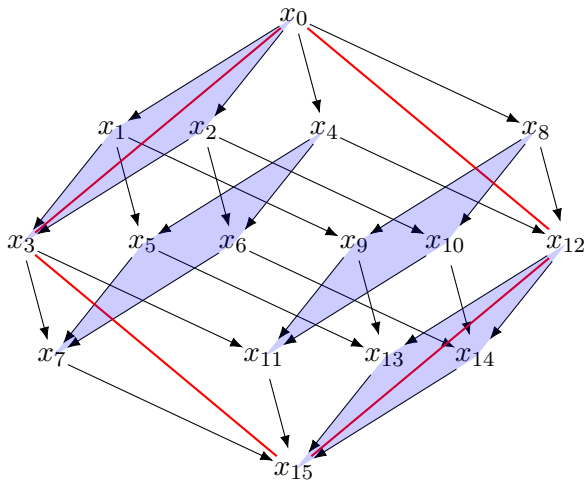
$$\overrightarrow{\gamma_5} = \overrightarrow{\gamma_6} = \overrightarrow{\gamma_7} = \overrightarrow{\gamma_9} = \overrightarrow{\gamma_{10}} = \overrightarrow{\gamma_{11}} = \overrightarrow{\gamma_{13}} = \overrightarrow{\gamma_{14}} = \overrightarrow{0}$$

— la projection $\widehat{x_{15}}^5$ (équivalente à la projection $\widehat{x_{15}}^{10}$) est valide si et seulement si

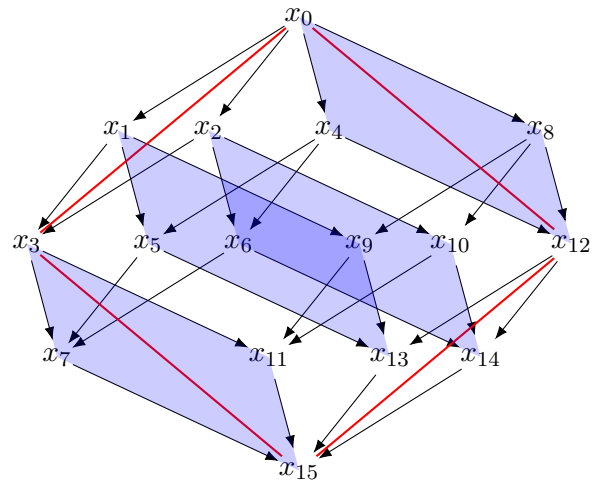
$$\overrightarrow{\gamma_3} = \overrightarrow{\gamma_6} = \overrightarrow{\gamma_7} = \overrightarrow{\gamma_9} = \overrightarrow{\gamma_{11}} = \overrightarrow{\gamma_{12}} = \overrightarrow{\gamma_{13}} = \overrightarrow{\gamma_{14}} = \overrightarrow{0}$$

— la projection $\widehat{x_{15}}^6$ (équivalente à la projection $\widehat{x_{15}}^9$) est valide si et seulement si

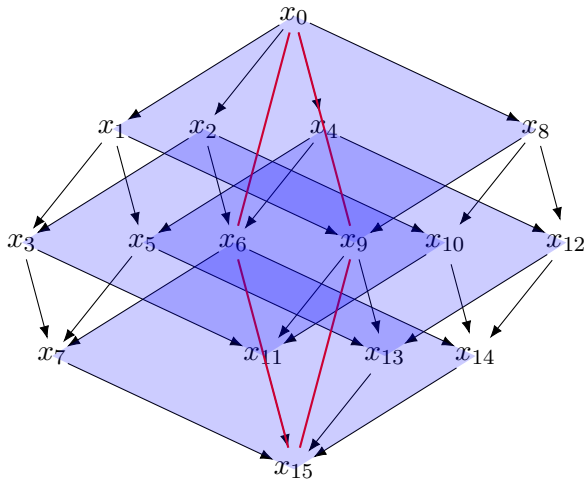
$$\overrightarrow{\gamma_3} = \overrightarrow{\gamma_5} = \overrightarrow{\gamma_7} = \overrightarrow{\gamma_{10}} = \overrightarrow{\gamma_{11}} = \overrightarrow{\gamma_{12}} = \overrightarrow{\gamma_{13}} = \overrightarrow{\gamma_{14}} = \overrightarrow{0}$$



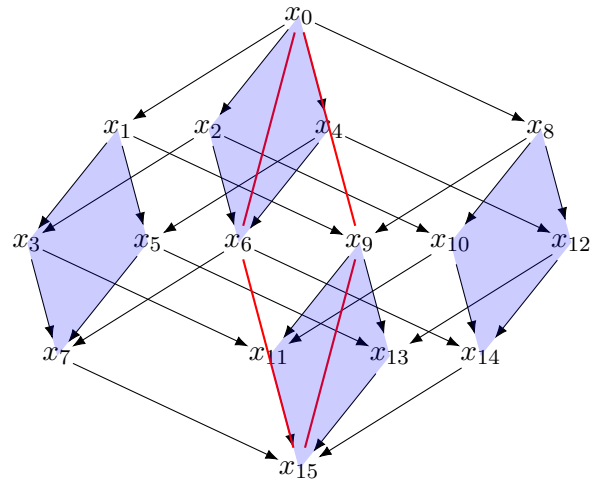
(a) Analogie entre $[x_0 \diamond x_3]$, $[x_4 \diamond x_7]$, $[x_8 \diamond x_{11}]$ et $[x_{12} \diamond x_{15}]$



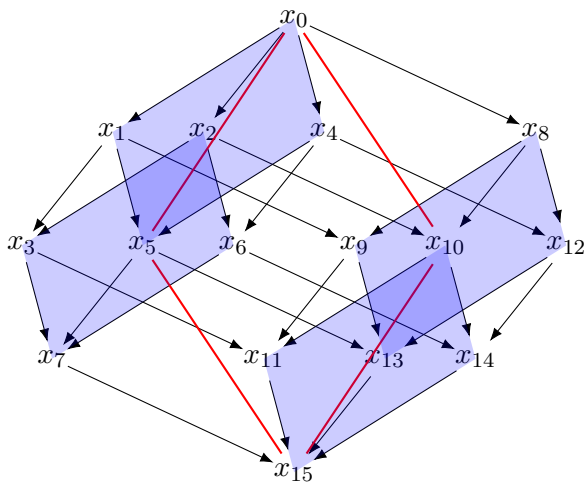
(b) Analogie entre $[x_0 \diamond x_{12}]$, $[x_1 \diamond x_{13}]$, $[x_2 \diamond x_{14}]$ et $[x_3 \diamond x_{15}]$



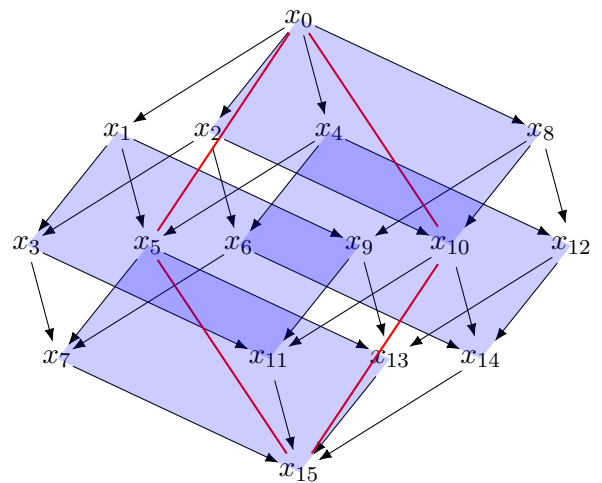
(c) Analogie entre $[x_0 \diamond x_9]$, $[x_2 \diamond x_{11}]$, $[x_4 \diamond x_{13}]$ et $[x_6 \diamond x_{15}]$



(d) Analogie entre $[x_0 \diamond x_6]$, $[x_1 \diamond x_7]$, $[x_8 \diamond x_{14}]$ et $[x_9 \diamond x_{15}]$



(e) Analogie entre $[x_0 \diamond x_5]$, $[x_2 \diamond x_6]$, $[x_8 \diamond x_{12}]$ et $[x_{10} \diamond x_{15}]$



(f) Analogie entre $[x_0 \diamond x_{10}]$, $[x_1 \diamond x_{11}]$, $[x_4 \diamond x_{14}]$ et $[x_5 \diamond x_{15}]$

FIGURE 6.14 – Mise en évidence des différentes analogies entre sous-systèmes carrés utilisables pour l'implication de \widehat{x}_{15} .

On peut alors distinguer un nombre limité de cas de figure qui identifient les implications valides selon les contrastes observés préalablement :

- Si aucun élément n'est contrastif, alors toutes les implications sont valides. De la même façon que pour le cas cubique, on définit donc :

$$\widehat{x}_{15} = x_0 = \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{r}_8 \quad (6.61)$$

- Si exactement un élément x_i de niveau 3 est contrastif et si tous les éléments contrastifs de niveau 2 sont des prédécesseurs de x_i , alors seule la projection \widehat{x}_{15}^i est valide. On définit alors :

$$\widehat{x}_{15} = \widehat{x}_{15}^i \quad (6.62)$$

- Si aucun élément de niveau 3 n'est contrastif et si précisément deux éléments symétriques de niveau 2, x_j et \overline{x}_j , sont contrastifs, alors seule la projection \widehat{x}_{15}^j est valide. On définit là encore :

$$\widehat{x}_{15} = \widehat{x}_{15}^j \quad (6.63)$$

- Si aucun élément de niveau 3 n'est contrastif et si les (au moins 2) éléments contrastifs de niveau 2 ont un successeur direct commun (donc de niveau 3), x_k , alors seule la projection \widehat{x}_{15}^k est valide. On définit :

$$\widehat{x}_{15} = \widehat{x}_{15}^k \quad (6.64)$$

- Si aucun élément de niveau 3 n'est contrastif et si exactement un élément de niveau 2 x_ℓ est contrastif, alors 3 projections sont simultanément valides : \widehat{x}_{15}^ℓ , \widehat{x}_{15}^m et \widehat{x}_{15}^n où x_m et x_n sont les successeurs directs de x_ℓ . Ces trois projections sont concordantes, et on définit :

$$\widehat{x}_{15} = \widehat{x}_{15}^\ell = \widehat{x}_{15}^m = \widehat{x}_{15}^n = x_0 + \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{r}_8 \oplus \vec{\gamma}_\ell \quad (6.65)$$

- Sinon, aucune implication n'est valide. On peut toutefois définir \widehat{x}_{15} par extension :

$$\widehat{x}_{15} = x_0 + \vec{r}_1 \oplus \vec{r}_2 \oplus \vec{r}_4 \oplus \vec{r}_8 \oplus \vec{\gamma}_3 \oplus \vec{\gamma}_5 \oplus \vec{\gamma}_6 \oplus \vec{\gamma}_7 \oplus \vec{\gamma}_9 \oplus \vec{\gamma}_{10} \oplus \vec{\gamma}_{11} \oplus \vec{\gamma}_{12} \oplus \vec{\gamma}_{13} \oplus \vec{\gamma}_{14} \quad (6.66)$$

Dans tous les cas, on définit comme d'habitude le contraste :

$$\vec{\gamma}_{15} = \overrightarrow{\widehat{x}_{15}x_{15}} \quad (6.67)$$

L'énumération des équations 6.61 à 6.66 peut paraître quelque peu laborieuse. Toutefois, elle a le mérite de mettre en évidence l'existence de quelques tests simples et systématiques pour déterminer si une implication est valide. Alors qu'un examen exhaustif nécessiterait l'examen de 2^{10} possibilités !

En outre, un examen minutieux de ces conditions de validité permet d'établir que si un élément x_i est contrastif, une condition nécessaire (mais non suffisante) pour que l'implication \widehat{x}_{15}^j soit valide est que $j = i$ ou que $x_i \prec x_j$, l'exposant j indiquant alors le couple d'éléments pivots (x_j, \overline{x}_j) devant être utilisé pour construire la projection.

Cette ultime propriété est essentielle : elle généralise à la compression tensorielle par analogie la contrainte d'antériorité définie dans le cadre de la compression par réplification en fournissant un test simple qui permet d'éliminer progressivement les sous-systèmes qui débouchent sur des projections discordantes (constituant des implications non valides) au fur et à mesure que surviennent des contrastes dans les systèmes successifs.

6.1.5 Cas général du système hypercubique

On dispose maintenant de l'essentiel des notions nécessaires à l'approche du cas général des principes de compression tensorielle par analogie dans un système hypercubique de dimension quelconque d .

Comme on l'a vu pour les systèmes cubiques et tesseractiques, chacun des éléments en position intermédiaire peut être décrit selon un sous-système de dimension inférieure. On se focalise donc sur l'élément final $x_f = x_{2^d-1}$.

On peut alors considérer toutes les analogies entre sous-systèmes de dimension d' ($1 < d' < d$) comme étant le support potentiel des multiples implications sur l'élément x_f et de conjecturer que chacune de ces analogies peut se résumer à un schéma d'implication selon un système transversal (certaines analogies distinctes conduisant éventuellement à un même système transversal).

Sous cette hypothèse, on peut alors définir pour toute position k dans l'hypercube l'implication

$$\widehat{x}_f^k = x_k + \overrightarrow{x_0 x_k} \quad (6.68)$$

Chaque analogie permet également de poser des équations qui sont éventuellement contredites si des éléments de $[x_0 \diamond x_k]$ ou de $[x_0 \diamond \overline{x_k}]$ sont contrastifs.

Plus généralement, on parvient à exprimer les conditions nécessaires à une implication valide dans n'importe quelle position x_i sous la forme d'une contrainte d'antériorité généralisée, à savoir que : $\forall x_h \prec x_i$, si $\overrightarrow{\gamma}_h \neq \overrightarrow{0}$ alors \widehat{x}_i^k est valide seulement si une des conditions suivantes est remplie :

$$\begin{aligned} &k = h \\ &\text{ou} \\ &x_h \prec x_k \\ &\text{ou} \\ &k = i - h \\ &\text{ou} \\ &x_h \prec \overline{x_k}^i \end{aligned}$$

On rappelle que x_k et $\overline{x_k}^i$ sont alors les éléments pivots de l'implication \widehat{x}_i^k .

On dispose donc là d'un principe systématique de compression tensorielle multiéchelle, présentant une combinatoire maîtrisée, du fait qu'elle est proportionnelle au nombre de systèmes carrés transversaux dont le nombre croît beaucoup moins vite que la combinatoire des positions contrastives. C'est d'ailleurs sur la base de ce procédé appliqué en dimension $d = 5$ (systèmes de 32 éléments) qu'ont été réalisées l'ensemble des expériences décrites plus loin dans ce chapitre.

6.2 Comparaison de paires d'accords

Pour présenter un intérêt autre que théorique, la compression tensorielle par analogie nécessite la définition de relations entre éléments, au-delà de leur simple identité ou dissemblance. Nous rappelons que dans le cadre de ce travail, nous nous intéressons à des séquences d'accords réduits à des triades majeures et mineures.

Une façon d'arranger les 24 triades majeures et mineures est le cercle des tierces présenté en figure 6.15. Sur ce cercle, un déplacement d'un degré correspond à un glissement des notes présentes dans la triade, comme le montre le tableau 6.1. Ainsi, deux triades voisines sur le cercle comportent 2 notes en commun, quelle que soit leur position.

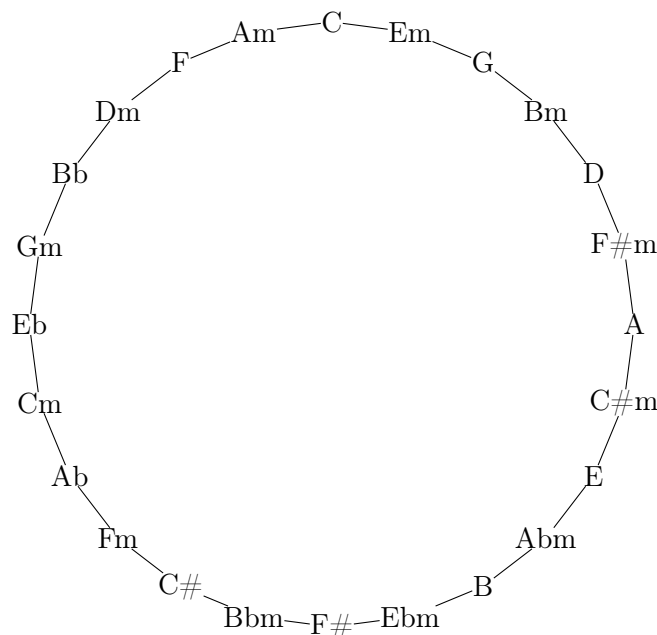


FIGURE 6.15 – Le cercle utilisé pour représenter l'espace des accords

Voisin anti-horaire	Accord	Voisin horaire
Fondamentale	(sixte)	
Tierce	Fondamentale	(sixte)
Quinte	Tierce	Fondamentale
(septième)	Quinte	Tierce
	(septième)	Quinte

TABLE 6.1 – Composition des accords voisins sur le cercle des tierces de la figure 6.15. Les tierces, sixtes et septièmes sont majeures pour les accords majeurs et mineures pour les accords mineurs.

On peut représenter la relation entre deux accords par le nombre de degrés qui les séparent sur le cercle et le sens de rotation. Les relations entre deux paires d'accords séparés (dans le même sens) par le même nombre de degrés sur le cercle peuvent être considérées comme identiques. Par exemple, l'accord de G_M est à C_M ce que $C_m^\#$ est à $F_m^\#$.

Ces espaces de représentation des accords et de leurs relations vérifient toutes les propriétés requises dans la section 6.1.

Variante dégénérée On peut noter qu'en définissant l'espace des relations de telle façon que seule l'identité est transposable à un autre accord, le modèle se dégénère en un modèle équivalent à celui du chapitre précédent. Cette définition est telle que la composition n'est pas

toujours définie, toutefois la commutativité et l’associativité sont vérifiées lorsque les deux sens de l’opération sont définis.

Le principe de l’implication unique peut également s’appliquer à ce cas dégénéré : les positions où aucune implication n’est définie ou valide peuvent ainsi être traitées de la même façon que les positions où toutes les implications sont invalides.

Modèles avancés Le cercle des tierces que nous utilisons dans ces travaux est une représentation pratique mais très sommaire de la tonalité en musique. Le développement récent d’espaces algébriques de modélisation de la musique [ANDREATTA, 2009 ; BIGO et al., 2014 ; ANDREATTA, 2016] offre une excellente perspective d’évolution sur ce plan. Toutefois, ces modèles ne respectent pas nécessairement l’ensemble des hypothèses que nous avons émises afin de définir l’élément projeté de façon unique. Afin de pouvoir les utiliser, des adaptations seront nécessaires pour utiliser la méthode avec des implications floues, avec plusieurs éléments virtuels en superposition.

6.3 Complexité d’un graphe de redondance

Dans le chapitre 5, nous avons proposé un schéma d’encodage d’une séquence selon un patron tensoriel. Dans ce schéma, chaque élément de la séquence est encodé successivement, en tenant compte de ses prédécesseurs :

$$\mathcal{C}(X|P) = \sum_{x_i \in X} \mathcal{C}(x_i|\pi^*(x_i)) \quad (6.69)$$

Les mécanismes d’implication que nous avons décrit dans la première partie du présent chapitre permettent une nouvelle définition de $\mathcal{C}(x_i|\pi^*(x_i))$. Pour chaque élément, on indique s’il est contrastif ou non. Pour les éléments non-contrastifs, cette indication est suffisante pour les reconstruire sans avoir besoin de spécifier davantage d’information. Les éléments contrastifs, en revanche, requièrent un supplément d’information pour être spécifiés, que l’on suppose être de l’ordre de l’information nécessaire à la spécification d’une valeur sans a priori.

On peut distinguer trois cas, selon le niveau de l’élément considéré :

- le primer ($C_\Delta(x_i) = 0$), pour lequel aucun contexte n’est disponible : cet élément doit être spécifié ;
- les sous-primers ($C_\Delta(x_i) = 1$) n’ont pas d’élément impliqué correspondant, toutefois on peut considérer par défaut un codage par réplication : on peut alors considérer un sous-primer différent du primer comme nécessitant une spécification supplémentaire, c’est-à-dire discordant, alors que son identité peut être considérée comme implicite ;
- les autres éléments ($C_\Delta(x_i) \geq 2$), pour lesquels on observe ou non un contraste.

Ainsi, à l’exception du primer, on peut encoder tous les éléments par les réponses aux questions suivantes :

1. x_i est il discordant / contrastif ? (1 bit d’information)
2. Si oui, quel est la différence / le contraste ? (environ $c = \log_2(|\mathcal{P}|)$ bits d’information)
Sinon, aucune information supplémentaire requise (0 bit d’information)

Dans le cadre de la comparaison de segmentations, les réponses à la première question représentent un terme constant sur l'ensemble du morceau. Comme pour les premières formulations du critère de complexité, il est possible de simplifier le critère en éliminant ce terme constant et en prenant c comme unité de mesure de l'information. On obtient alors $\mathcal{C}(X|P)$ simplement en comptant le nombre d'éléments non-nuls dans la représentation systématique de X .

6.3.1 Gestion des déformations

Le modèle décrit en section 6.1 permet de décrire les séquences pouvant être projetées sur un patron hypercubique.

Afin de pouvoir modéliser des segments de longueurs différentes, nous avons introduit en section 5.4 deux types de déformation sur les patrons tensoriels permettant de traiter des segments de longueurs n'étant pas des puissances de 2 : les patrons avec omission et les patrons avec insertion.

Ces déformations consistent à supprimer ou à dupliquer une partie contiguë et convexe de l'hypercube, définie par un élément et l'ensemble de ses successeurs.

Si les patrons avec omission ne demandent pas de traitement supplémentaire, dans le cas des patrons avec insertion, les éléments insérés doivent également être codés. Une façon de procéder consiste à considérer les éléments insérés comme formant un sous-système avec les éléments correspondants du système principal. Pour ce sous-système, seules les spécifications des éléments insérés sont comptabilisées, les autres éléments étant déjà définis par le système principal.

6.3.2 Algorithme

Comme on l'a vu en section 6.1, pour les éléments x_i de niveau supérieur ou égal à 2, il existe plusieurs implications basées sur des analogies permettant de construire un élément projeté \hat{x}_i^h servant de référence pour l'encodage de x_i . Selon les éléments de $[x_0 \diamond x_i]$ qui sont contrastifs, les hypothèses d'analogie sur lesquelles sont basées certaines de ces implications sont contredites.

L'observation d'un contraste dans une position donnée invalide certaines implications sur ses successeurs. En se basant sur la contrainte d'antériorité généralisée, on peut définir un algorithme qui permet d'implémenter efficacement la détermination des implications valides et des projections correspondantes.

En pratique, on peut maintenir pour chaque élément x_i l'ensemble V_i des éléments pivots des systèmes transversaux correspondants aux implications valides. En l'absence de tout contraste, cet ensemble comprend tous les éléments de $[x_0 \diamond x_i]$ à l'exception de x_0 et x_i .

Avec un parcours topologiquement trié selon \prec , comme le parcours chronologique, on peut examiner chacun des éléments de la séquence, afin de déterminer $\mathcal{C}(x_i|\pi^*(x_i))$, en mettant à jour les ensembles V_j des pivots valides pour chaque élément à chaque fois que l'on rencontre un contraste. L'algorithme 11 détaille ce processus sous forme de pseudo-code.

De la même façon qu'au chapitre 5, cette fonction de score est injectée dans un algorithme de plus court chemin (algorithme 7) afin d'obtenir une séquence de tenseurs de complexité minimale.

Algorithme 11 Calcul de la complexité avec un encodage à historique tensoriel par analogie

Entrée: Une séquence X sur un patron P

pour tout x_i élément de X **faire**

$$V_i = \pi^*(x_i) \setminus \{x_0\}$$

fin pour

score := 0

pour tout x_i élément de X **faire**

si $N(x_i) = 0$ **then**

▷ x_i est le primer

score := score + 1

sinon si $N(x_i) = 1$ **then**

▷ x est un sous-primer

si $x_i = x_0$ **then**

score := score + 1

sinon

score := score + 0

fin si

sinon

si $V_i = \emptyset$ **then**

▷ x_i n'a pas d'implication valide

score := score + 1

pour tout $x_j \in \sigma^*(x_i)$ **faire**

$$V_j := V_j \cap \left(\{x_i, \overline{x_i^j}\} \cup \sigma^*(x_i) \cup \overline{\sigma^*(x_i)^j} \right)$$

fin pour

sinon

si $\exists x_h \in V_i : \overline{x_0 x_h} = \overline{x_h x_i x}$ **then**

▷ x_i est non-contrastif

score := score + 0

sinon

▷ x est contrastif

score := score + 1

pour tout $x_j \in \sigma^*(x)$ **faire**

$$V_j := V_j \cap \left(\{x_i, \overline{x_i^j}\} \cup \sigma^*(x_i) \cup \overline{\sigma^*(x_i)^j} \right)$$

fin pour

fin si

fin si

fin si

fin pour

Sortie: score, une estimation de la complexité de X selon P .

6.4 Évaluation des méthodes à encodage tensoriel

Les expériences de ce chapitre réutilisent les bases AUTO, CORRECTED et MANUAL, décrites en section 3.3.1, qui correspondent à différents niveaux de fiabilité de l’annotation. On réutilise également la méthodologie détaillée pour la segmentation linéaire en section 3.3.5.2, notamment l’emploi de la F-mesure, de la précision et du rappel avec tolérance nulle et de ± 3 temps comme métriques d’évaluation. On peut toutefois noter que la phase de conversion en segmentation linéaire n’est pas nécessaire ici, car la sortie de l’algorithme est directement assimilable à une segmentation.

6.4.1 Influence du modèle de relations

Dans ce chapitre et le précédent, nous avons décrit trois méthodes d’estimation de la complexité d’une séquence selon un patron tensoriel. Toutes trois sont basées sur l’encodage successif des éléments du système d’après un historique tensoriel.

1. Le premier est un encodage par réplication simple, décrit en section 5.5.2, qui considère exclusivement les éléments précédents qui sont en voisinage direct sur le patron tensoriel pour trouver un élément de référence à répliquer. Dans ce modèle, un élément ne contribue pas à la complexité du système s’il est égal à au moins un de ses prédécesseurs directs.
2. Le deuxième est un encodage par réplication avec contrainte d’antériorité, décrit en section 5.5.3, qui est plus restrictif et considère tout le sous-système aboutissant à l’élément sous considération pour déterminer un élément de référence à répliquer parmi les prédécesseurs directs. Dans ce modèle, un élément ne contribue pas à la complexité du système s’il est égal à un de ses prédécesseurs directs, à la condition supplémentaire que toutes les paires homologues du sous-système sont équivalentes.
3. Le troisième est une compression tensorielle par analogie, décrit en section 6.3, et s’appuie sur le sous-système pour construire un élément virtuel par analogie. Dans ce modèle, un élément ne contribue pas à la complexité du système s’il est équivalent à l’élément virtuel.

La table 6.2 montre la F-mesure moyenne en utilisant chacune de ces estimations de complexité comme critère à minimiser pour la segmentation structurale. Le critère avec encodage par réplication simple atteint des performances relativement faibles, ce qui est cohérent avec l’observation que la complexité de certains motifs est fortement sous-évaluée. L’ajout de la contrainte d’antériorité améliore sensiblement la F-mesure, ce qui permet d’atteindre des performances similaires à celles atteintes par les algorithmes à base de grammaire étudiés dans la première partie de cette thèse.

Le critère avec compression tensorielle par analogie domine nettement les deux autres, avec une F-mesure supérieure de près de 10 points de pourcentage au critère avec contrainte de précédent.

Type d’encodage	F-mesure (sans tolérance)
par réplication simple	43.9%
par réplication avec contrainte d’antériorité	50.3%
par analogie	59.6%

TABLE 6.2 – F-mesure moyenne selon le critère de complexité utilisé sur la base MANUAL. Les performances rapportées correspondent aux valeurs optimales des coûts de déformation (voir section 6.4.2.1).

6.4.2 Sensibilité aux hyper-paramètres

6.4.2.1 Pénalités de déformation

Dans le chapitre 5, nous avons décrit deux types de déformations sur les patrons hypercubiques, afin de pouvoir modéliser des segments d'une plus grande variété. Pour refléter la complexité plus élevée introduite par ces déformations, nous avons incorporé un terme $\mathcal{C}(P)$ au critère de complexité :

$$\mathcal{C}(S) = \min_P \mathcal{C}(S|P) + \mathcal{P} \quad (6.70)$$

avec

$$\mathcal{C}(P) = \begin{cases} p_0 & \text{si } P \text{ est hypercubique} \\ p_0 + p_+ & \text{si } P \text{ est avec insertions} \\ p_0 + p_- & \text{si } P \text{ est avec omission} \\ p_0 + p_+ + p_- & \text{si } P \text{ est mixte} \end{cases} \quad (6.71)$$

où p_0 est une constante s'annulant dans les comparaisons, et p_+ et p_- sont les paramètres du critère de complexité, pénalisant les déformations, que nous étudions ici.

La table 6.3 montre l'influence de ces paramètres sur la F-mesure obtenue avec l'encodage par analogie.

TABLE 6.3 – F-mesure moyenne (sans tolérance, en pourcentage) selon les valeurs des pénalités appliquées aux déformations du patron hypercubique. En raison de la faible longueur des séquences, une pénalité de 0.01 est équivalente à considérer le nombre de déformations uniquement afin de départager les égalités. Inversement, une pénalité de 1000 signifie que cette déformation n'est utilisée que si aucune segmentation n'est possible en utilisant un nombre inférieur de déformations.

$p_+ \backslash p_-$	0	0.01	1	2	3	4	1000
0	43.7	48.6	47.9	48.4	48.0	47.6	47.2
0.01	48.4	49.8	52.7	54.1	53.1	52.1	51.4
1	47.8	52.3	55.5	55.6	55.6	55.2	53.4
2	50.2	54.1	56.4	59.5	58.7	56.9	54.8
3	50.2	54.6	56.7	59.3	59.6	58.6	55.1
4	49.7	54.6	56.9	58.4	57.5	57.9	53.1
1000	49.7	53.9	54.9	56.3	56.7	54.6	38.1

Dans les cas où l'une des pénalités est neutralisée, on observe une baisse marquée de la performance de segmentation, y compris par rapport à une pénalité extrêmement faible, qui ne prend en compte le nombre de déformations correspondantes que pour départager des égalités. La baisse de F-mesure associée à l'utilisation d'une pénalité prohibitivement élevée, interdisant en pratique ce type de déformation, est également notable bien que plus faible⁷.

En dehors de ces valeurs extrêmes, l'impact des pénalités de déformation est modéré : les valeurs évoluent de façon assez lisse sur une plage d'environ 5 points de pourcentage.

7. Le cas où les deux pénalités sont très élevées a un comportement très irrégulier dû à la flexibilité très limitée dans les tailles de segment, qui ne sera pas détaillé.

6.4.2.2 Régularisation

Les annotations de structure que nous utilisons sont produites explicitement avec une échelle de segmentation cible, qui est de l'ordre de 32 temps. Afin de cibler cette échelle pour les résultats de l'inférence, nous avons expérimenté plusieurs méthodes de régularisation des scores de complexité des segments.

La méthode retenue, similaire à celle utilisée avec efficacité par [SARGENT, BIMBOT et VINCENT, 2011], consiste à définir une taille cible pour chaque segment (ici 32 temps, la même que pour l'annotation). Pour chaque segment, une pénalité est ajoutée à son score de complexité, proportionnelle à la différence entre sa longueur et la longueur cible. L'amplitude des déviations attendues étant différente selon que le segment soit plus ou moins long que la cible, le coefficient a été dissocié selon ces deux cas. Ainsi, si $S(X)$ est le score de complexité de la séquence X , son score régularisé $S_r(X)$ est

$$S_r(X) = \begin{cases} S(X) + r_+ (|X| - l_r) & \text{si } |X| \geq l_r \\ S(X) + r_- (l_r - |X|) & \text{si } |X| < l_r \end{cases} \quad (6.72)$$

où $|X|$ est la taille de la séquence, l_r est la taille cible des segments et r_+ et r_- sont les coefficients de régularisation.

TABLE 6.4 – F-mesure (sans tolérance, en pourcentage) selon les paramètres de régularisation. On utilise $p_+ = p_- = 2.25$, la paire de valeurs offrant les meilleurs résultats sans régularisation.

$r_- \backslash r_+$	0	0.0001	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2
0	61.1	61.4	61.4	60.8	60.9	60.5	60.2	60.4	60.4	60.1
0.0001	62.8	62.8	62.5	61.9	61.7	61.2	61.3	61.1	61.1	60.7
0.025	62.2	62.2	62.4	62.4	62.6	62.7	61.7	61.7	61.2	60.5
0.05	65.5	65.5	65.3	65.5	66.2	66.1	65.9	64.8	64.6	64.6
0.075	66.2	66.2	66.7	67.0	66.9	66.7	66.6	66.3	66.0	65.5
0.1	67.8	67.9	67.0	66.9	66.7	66.7	66.8	66.5	66.7	66.0
0.125	67.8	67.9	67.1	66.5	66.4	66.1	66.4	66.3	66.1	66.1
0.15	68.0	67.8	67.5	67.5	66.5	66.1	65.9	66.0	66.0	65.8
0.175	67.4	67.4	67.6	67.2	66.8	66.5	65.9	65.7	65.7	65.7
0.2	65.9	65.9	66.1	66.5	66.6	65.8	65.3	65.0	65.0	64.7

La table 6.4 détaille les résultats obtenus en faisant varier les coefficients de régularisation, en fixant les pénalités de déformation à la paire de valeurs optimale obtenue sans régularisation⁸. On remarque que le coefficient r_+ a une influence plus faible que r_- sur la performance finale. Contrairement aux autres paramètres, on n'observe pas d'écart marqué pour r_+ entre un coefficient nul et non-nul, et la valeur maximale de F-mesure est atteinte avec $r_+ = 0$. Il est probable que la tendance de segments plus longs à obtenir des scores de complexité plus élevés suffise à prévenir une surabondance des segments longs.

8. Une deuxième expérience raffinant les résultats de la table 6.3 a permis d'identifier une meilleure paire optimale.

TABLE 6.5 – Performances maximales (et paramètres de déformation correspondants) du modèle complet à système tensoriel, avec régularisation, pour chacun des ensembles d’annotations.

Ensemble d’annotation	F-mesure stricte	F-mesure ± 3 temps	p_+	p_-
AUTO	37.4%	55.2%	2.5	2.5
CORRECTED	63.1%	64.8%	2.5	3.5
MANUAL	69.0%	70.0%	2.25	3

En fixant les paramètres de régularisation aux valeurs ($r_+ = \epsilon, r_- = 0.125$)⁹ pour faire varier les pénalités de déformation, on observe les mêmes tendances que celles de la table 6.3. La nouvelle F-mesure maximale atteinte est de 69.0%, pour une pénalité d’insertion de 2.25 à nouveau et une pénalité d’omission plus élevée, égale à 3.

6.4.3 Seuil de tolérance et base d’annotation

Dans les expériences du chapitre 3, nous observions un écart de 22 points de pourcentage entre la F-mesure sur la base AUTO et sur la base MANUAL, traduisant une forte sensibilité à la présence d’erreurs dans la séquence d’entrée. Parallèlement, la formulation de la F-mesure incluant une marge de tolérance donnait une valeur supérieure de 9 à 15 points de pourcentage selon la base d’annotations, un écart relativement faible comparé à ceux observés pour la plupart des algorithmes utilisant des données audio.

La table 6.5 donne les résultats moyens obtenus avec les paramètres optimaux sur chacun des ensembles d’annotations et la figure 6.16 montre la répartition pour les morceaux individuels.

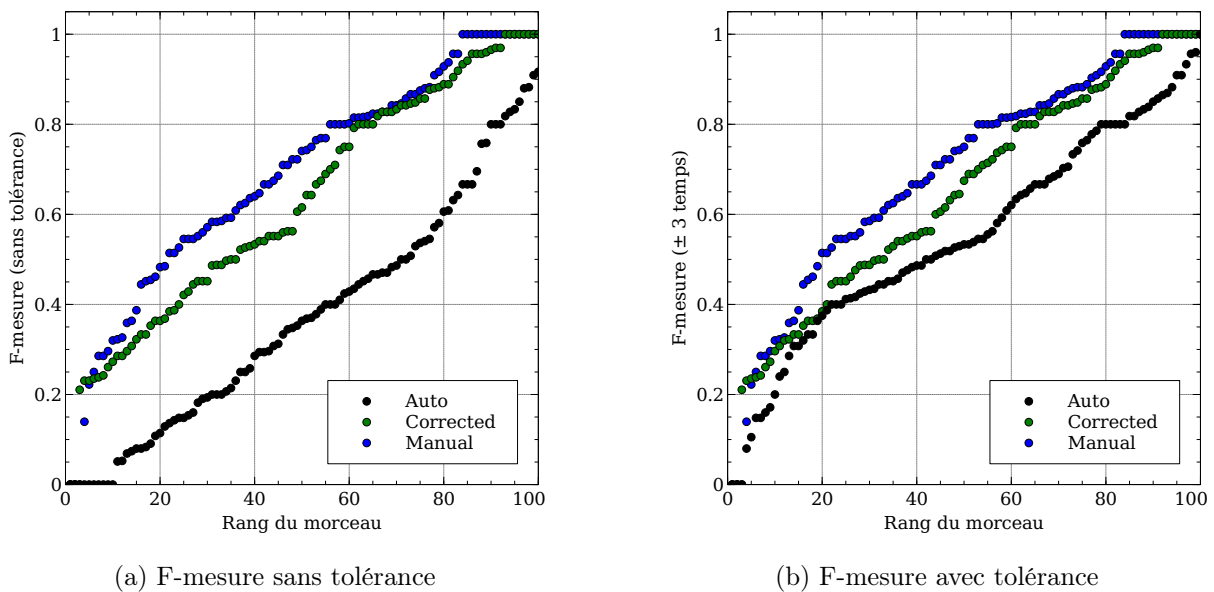


FIGURE 6.16 – F-mesure sur chacun des 100 morceaux de RWC-Pop selon l’annotation utilisée pour l’encodage tensoriel par analogie avec régularisation. Pour plus de lisibilité, les morceaux ont été ré-ordonnés par F-mesure croissante selon chaque annotation

9. Les expériences avec $r_+ = 0$ ayant été effectuées ultérieurement, ces valeurs diffèrent légèrement de l’optimum trouvé dans la section précédente.

Comme pour les méthodes à base de GDU, les révisions successives des annotations d'accords améliorent fortement les performances de l'algorithme d'inférence. Le passage de la base CORRECTED à la base MANUAL a ici un effet plus faible que pour les GDU, ce qui est attendu car un de ses impacts principaux attendus de la révision de la base de données est l'amélioration de la cohérence à long terme dans les séquences, qui n'est pas pertinente pour l'évaluation de la complexité interne des segments.

Un point remarquable est la faible variation des valeurs optimales pour les paramètres de déformation : pour chacun des ensembles, la paire de valeurs maximisant la F-mesure sans tolérance est la même que celle maximisant la F-mesure avec tolérance. En outre, les paramètres optimaux varient peu d'un ensemble d'annotations à l'autre.

Enfin, la différence entre les deux formulations de la F-mesure est extrêmement faible sur les ensembles corrigés manuellement.

6.4.4 Exemples de segmentation

Pour illustrer cette section expérimentale, nous examinons le résultat de quelques segmentations, représentatives des différents niveaux de performance de l'inférence, que nous comparons aux segmentations de référence, évaluées selon le même critère.

RWC-Pop 001 : Performance faible

La figure 6.17 montre la segmentation de référence et la segmentation inférée pour le premier morceau de l'annotation MANUAL. Pour ce morceau, la segmentation inférée ne correspond que partiellement ($F=0.39$) à la segmentation de référence.

Les segments de ce morceau sont particulièrement irréguliers : en plus des segments réguliers de 16 ou 32 temps, on y trouve des segments de 24, 28, 34, 36 et 44 temps (comme mentionné précédemment, les segments de 44 temps ne peuvent pas être modélisés par des patrons hypercubiques et leurs déformations). Un autre facteur d'erreur est que les premières moitiés de segment ont tendance à finir avec les mêmes accords le segment précédent, créant ainsi un système moins contrastif mais décalé.

On peut toutefois noter que certaines formes fortement répétitives qui tranchent avec les segments précédents, comme celles au début du 4^e segment ou du segment de 44 éléments, permettent ponctuellement de resynchroniser la segmentation.

RWC-Pop 022 : Performance élevée

Sur le morceau numéro 22 de la base MANUAL, la segmentation inférée est presque exactement la segmentation de référence, comme représenté en figure 6.18. La seule différence est que le dernier segment a été fusionné avec l'avant-dernier, principalement en raison de la régularisation qui pénalise fortement le segment final de 8 éléments.

Un relâchement de la régularisation est envisageable sur les premiers et derniers segments, que nous avons déjà identifiés comme étant fréquemment très courts. Il s'agit toutefois d'un compromis, car il est également fréquent que les segments en fin de morceau soient des formes étendues (par exemple en allongeant les dernières notes) qui risqueraient alors d'être scindés en deux.

Abm Abm Abm Abm F# F# F# F# E E E E F# F# F# F#
 F# F# F# F# Eb Eb Eb Eb
 10 (24)
 Abm Abm Abm Abm E E E E B B B B Eb Eb Eb Eb Abm
 Abm Abm Abm E E E E F# F# Eb Eb
 9.5 (28)
 Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E F#
 F# Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E
 F# F#
 5 (32)
 Cm Cm Cm Cm Cm Cm F Cm Cm Cm Cm Cm Cm Cm Bb
 Cm Cm Cm Cm Cm Cm F Ab Ab Bb Bb Cm Cm Cm Cm
 8 (32)
 Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm Cm Cm Cm Ab
 Ab Ab Ab Eb Eb Eb Eb G G G G Eb Eb Eb Eb
 5 (32)
 Abm Abm Abm Abm E E E E B B B B Eb Eb Eb Eb Abm
 Abm Abm Abm E E E E F# F# Eb Eb
 9.5 (28)
 Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E F#
 F#
 7 (16)
 Cm Cm Cm Cm Cm Cm F Cm Cm Cm Cm Cm Cm Cm Bb
 Cm Cm Cm Cm Cm Cm F Ab Ab Bb Bb Cm Cm Cm Cm
 8 (32)
 Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm Cm Cm Cm Ab
 Ab Ab Ab Eb Eb Eb Eb G G G G Eb Eb Eb Eb
 5 (32)
 Abm Abm Abm Abm E E E E B B B B Eb Eb Eb Eb Abm
 Abm Abm Abm E E E E F# F# Eb Eb Abm Abm Abm Abm
 Abm Abm E Eb
 12.25 (36)
 Ebm Ebm Ebm Ebm Ebm Ebm Ebm Ab Ebm Ebm Ebm Ebm
 Ebm Ebm Ebm C# Ebm Ebm Ebm Ebm Ebm Ebm Ebm Ab
 C# C# F# F# B B E E A A D D Eb Eb Eb Eb Eb Eb
 Eb Eb
 N/A (44)
 Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm Cm Cm Cm Ab
 Ab Ab Ab Eb Eb Eb Eb G G G G Eb Eb Eb Eb
 5 (32)
 Abm Abm Abm Abm E E E E B B B B Eb Eb Eb Eb Abm
 Abm Abm Abm E E E E F# F# Eb Eb
 9.5 (28)
 Abm Abm Abm Abm E E E E B B B B Eb Eb Eb Eb Abm
 Abm Abm Abm E E E E F# F# Eb Eb
 9.5 (28)
 Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E F#
 F# Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E
 F# F# F# F#
 7.25 (34)
 Abm N N N
 7.5 (4)

Abm Abm Abm Abm F# F# F# F# E E E E F# F# F# F#
 F# F# F# F#
 7.75 (20)
 Eb Eb Eb Eb Abm Abm Abm Abm E E E E B B B B Eb
 Eb Eb Eb Abm Abm Abm Abm E E E E F# F# Eb Eb
 6 (32)
 Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E F#
 F# Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E
 F# F#
 5 (32)
 Cm Cm Cm Cm Cm Cm F Cm Cm Cm Cm Cm Cm Cm Bb
 Cm Cm Cm Cm Cm Cm F Ab Ab Bb Bb Cm Cm Cm Cm
 8.5 (28)
 Cm Cm Cm Cm Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm
 Cm Cm Ab Ab Ab Ab Eb Eb Eb Eb G G G G
 4 (32)
 Eb Eb Eb Eb Abm Abm Abm Abm E E E E B B B B Eb
 Eb Eb Eb Abm Abm Abm Abm E E E E F# F# Eb Eb
 6 (32)
 Abm Abm Abm Abm F# F# F# F# C# C# C# C# E E F#
 F#
 7 (16)
 Cm Cm Cm Cm Cm Cm F Cm Cm Cm Cm Cm Cm Cm Bb
 Cm Cm Cm Cm Cm Cm F Ab Ab Bb Bb
 8.5 (28)
 Cm Cm Cm Cm Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm
 Cm Cm Ab Ab Ab Ab Eb Eb Eb Eb G G G G
 4 (32)
 Eb Eb Eb Eb Abm Abm Abm Abm E E E E B B B B Eb
 Eb Eb Eb Abm Abm Abm Abm E E E E F# F# Eb Eb
 Abm Abm Abm Abm Abm Abm E Eb
 12.25 (40)
 Ebm Ebm Ebm Ebm Ebm Ebm Ab Ebm Ebm Ebm Ebm
 Ebm Ebm Ebm C# Ebm Ebm Ebm Ebm Ebm Ebm Ab
 C# C# F# F# B B E E A A D D Eb Eb Eb Eb Eb Eb
 13.25 (40)
 Eb Eb Eb Eb Ab Ab Ab Ab Eb Eb Eb Eb G G G G Cm
 Cm Cm Ab Ab Ab Ab Eb Eb Eb Eb G G G G
 7 (32)
 Eb Eb Eb Eb Abm Abm Abm Abm E E E E B B B B Eb
 Eb Eb Eb Abm Abm Abm Abm E E E E F# F# Eb Eb
 6 (32)
 Abm Abm Abm Abm E E E E B B B B Eb Eb Eb Eb Abm
 Abm Abm Abm E E E E F# F# Eb Eb Abm Abm Abm Abm
 8 (32)
 F# F# F# F# C# C# C# C# E E F# F# Abm Abm Abm
 Abm F# F# F# F# C# C# C# C# E E F# F# F# F# Abm
 N N N
 12.25 (34)

(b) Segmentation inférée. Score: 115.5

(a) Segmentation de référence. Score : 118 + un segment non parsable

FIGURE 6.17 – Segmentation de référence et segmentation inférée pour le morceau RWC-Pop 001. Après chaque segment est indiqué son score de complexité selon un encodage par analogie avec les paramètres retenus comme optimaux sur le corpus ($r_+ = 0$, $r_- = 0.125$, $p_+ = 2.25$ et $p_- = 3$) et sa taille.

Bb Bb F F Bb Bb N N
6 (8)
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Ab Ab Eb Eb
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Ab Ab Eb Eb
3 (32)
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Ab Ab Eb Eb
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Bb Bb Bb Bb
5 (32)
Gm Gm Gm Gm Gm Gm Gm C C C C C C C Eb Eb
Eb Eb Eb Eb Eb Eb C# C# C# C# F F F F
5 (32)
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
6 (32)
Gm Gm F F Eb Eb Gm Gm Cm Cm Cm Cm F F F F Bbm N
N N N N N N
17 (24)
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Ab Ab Eb Eb
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Bb Bb Bb Bb
5 (32)
Gm Gm Gm Gm Gm Gm Gm Gm C C C C C C C Eb Eb
Eb Eb Eb Eb Eb Eb C# C# C# C# F F F F N N
8.25 (34)
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
6 (32)
Gm Gm F F Eb Eb Gm Gm Cm Cm Cm Cm F F F F
10 (16)
C# C# C# C# Ab Ab Ab Ab B B B B F# F# F# F# A A
A A A A A Ab Ab Ab Ab Ab Ab Ab Ab
7 (32)
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
N N N N
11.25 (36)
B B B B G G G G E E E E D D F#m F# B B B B G G
G G E E E E D D F#m F#
6 (32)
B B B B G G G G E E E E D D F#m F# B B B B G G
G G E E E E D D F#m F#
6 (32)
Abm Abm F# F# E E Abm Abm C#m C#m C#m C#m F# F#
F# F#
10 (16)
B B B B A A A A B B B B A A E E B B B B A A A A
B B B B A A E E
3 (32)
B B F# F# N N N N
7 (8)

(a) Segmentation de référence. Score : 121.5

Bb Bb F F Bb Bb N N
6 (8)
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Ab Ab Eb Eb
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Ab Ab Eb Eb
3 (32)
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Ab Ab Eb Eb
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Bb Bb Bb Bb
5 (32)
Gm Gm Gm Gm Gm Gm Gm Gm C C C C C C C Eb Eb
Eb Eb Eb Eb Eb Eb C# C# C# C# F F F F
5 (32)
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
6 (32)
Gm Gm F F Eb Eb Gm Gm Cm Cm Cm Cm F F F F Bbm N
N N N N N N
17 (24)
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Ab Ab Eb Eb
Bb Bb Bb Bb Ab Ab Ab Ab Bb Bb Bb Bb Bb Bb Bb Bb
5 (32)
Gm Gm Gm Gm Gm Gm Gm Gm C C C C C C C Eb Eb
Eb Eb Eb Eb Eb Eb C# C# C# C# F F F F N N
8.25 (34)
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
6 (32)
Gm Gm F F Eb Eb Gm Gm Cm Cm Cm Cm F F F F
10 (16)
C# C# C# C# Ab Ab Ab Ab B B B B F# F# F# F# A A
A A A A A Ab Ab Ab Ab Ab Ab Ab Ab
7 (32)
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
Bb Bb Bb Bb F# F# F# F# Eb Eb Eb Eb C# C# Fm F
N N N N
11.25 (36)
B B B B G G G G E E E E D D F#m F# B B B B G G
G G E E E E D D F#m F#
6 (32)
B B B B G G G G E E E E D D F#m F# B B B B G G
G G E E E E D D F#m F#
6 (32)
Abm Abm F# F# E E Abm Abm C#m C#m C#m C#m F# F#
F# F#
10 (16)
B B B B A A A A B B B B A A E E B B B B A A A A
B B B B A A E E
3 (32)
B B F# F# N N N N
8.25 (40)

(b) Segmentation inférée. Score: 119.75

FIGURE 6.18 – Segmentation de référence et segmentation inférée pour le morceau RWC-Pop 022. Après chaque segment est indiqué son score de complexité selon un encodage par analogie avec les paramètres retenus comme optimaux sur le corpus ($r_+ = 0$, $r_- = 0.125$, $p_+ = 2.25$ et $p_- = 3$) et sa taille.

Ab Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Ab
 Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm Fm
 Fm Fm Bb Bb N N
 8.25 (40)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm
 4 (16)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm F# F# F# F# Ab Ab
 Ab Ab
 5 (32)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm F# F# F# F# Ab Ab
 Ab Ab Eb Eb Eb Eb Eb Eb Eb
 9.25 (40)
 Ab Ab Ab Ab F# F# F# F# Eb Eb Eb Eb Eb Eb Eb
 Ab Ab Ab Ab F# F# F# F# Ab Ab Ab Ab N N N
 9 (32)
 Ab Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Ab
 Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm Fm
 Fm Fm Bb Bb N N
 8.25 (40)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm
 4 (16)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm F# F# F# F# Ab Ab
 Ab Ab Eb Eb Eb Eb Eb Eb Eb
 9.25 (40)
 Ab Ab Ab Ab F# F# F# F# Eb Eb Eb Eb Eb Eb Eb
 Ab Ab Ab Ab F# F# F# F# Ab Ab Ab Ab N N N
 9 (32)
 Ab Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Ab
 Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm Fm
 Fm Fm Bb Bb N N
 8.25 (40)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb
 Ebm Ebm
 2 (32)
 Ab Ab Ab Ab F# F# F# F# Eb Eb Eb Eb Eb Eb Eb
 Ab Ab Ab Ab F# F# F# F# Ab Ab Ab Ab N N N
 9 (32)
 Ab Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Ab
 Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm Fm
 Fm Fm Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm Fm
 Fm Bb Bb N N
 12 (56)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb
 Ebm Ebm
 2 (32)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb N N N
 N
 4 (32)

(a) Segmentation de référence. Score : 103.25

Ab Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Ab
 Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm Fm
 Fm Fm Bb Bb N N
 8.25 (40)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb
 Ebm Ebm
 2 (32)
 Eb Eb Eb Eb Eb Eb Ebm Ebm F# F# F# F# Ab Ab Ab
 Ab
 7 (16)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm F# F# F# F# Ab Ab
 Ab Ab
 5 (32)
 Eb Eb Eb Eb Eb Eb Eb Eb Ab Ab Ab Ab F# F# F# F#
 Eb Eb Eb Eb Eb Eb Eb Eb Ab Ab Ab Ab F# F# F# F#
 Ab Ab Ab Ab N N N
 9.25 (40)
 Ab Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Ab
 Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm Fm
 Fm Fm Bb Bb N N
 8.25 (40)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb
 Ebm Ebm
 2 (32)
 Eb Eb Eb Eb Eb Eb Ebm Ebm F# F# F# F# Ab Ab Ab
 Ab
 7 (16)
 Eb Eb Eb Eb Eb Eb Eb Eb Ab Ab Ab Ab F# F# F# F#
 Eb Eb Eb Eb Eb Eb Eb Eb Ab Ab Ab Ab F# F# F# F#
 Ab Ab Ab Ab N N N
 9.25 (40)
 Ab Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Ab
 Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm Fm
 Fm Fm Bb Bb N N
 8.25 (40)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb
 Ebm Ebm
 2 (32)
 Ab Ab Ab Ab F# F# F# F# Eb Eb Eb Eb Eb Eb Eb
 Ab Ab Ab Ab F# F# F# F# Ab Ab Ab Ab N N N
 9 (32)
 Ab Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C Ab
 Ab Ab Ab Bb Bb Bb Bb Gm Gm Gm Gm C C C C
 4 (32)
 Fm Fm Fm Fm Bb Bb Bb Bb Gm Gm Gm Gm C C C C Fm
 Fm Fm Fm Bb Bb N N
 8 (24)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb
 Ebm Ebm
 2 (32)
 Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb Eb Eb Ebm
 Ebm Eb Eb Eb Eb Eb Eb Ebm Ebm Eb Eb Eb N N N
 N
 4 (32)

(b) Segmentation inférée. Score: 95.25

FIGURE 6.19 – Segmentation de référence et segmentation inférée pour le morceau RWC-Pop 024. Après chaque segment est indiqué son score de complexité selon un encodage par analogie avec les paramètres retenus comme optimaux sur le corpus ($r_+ = 0$, $r_- = 0.125$, $p_+ = 2.25$ et $p_- = 3$) et sa taille.

RWC-Pop 024 : Performance moyenne

Le morceau numéro 24 de la base MANUAL est un morceau pour lequel la F-mesure entre la segmentation inférée est légèrement au dessus de la moyenne ($F=0.71$). La figure 6.19 montre la segmentation de référence et la segmentation obtenue par l'algorithme d'inférence.

On peut voir que les segmentations diffèrent en quelques endroits :

- La frontière entre le deuxième segment et le troisième segment (0 :19 à 0 :41 dans l'audio) est placée 16 temps (un demi-segment) plus tard dans la segmentation inférée. En effet, sur la séquence d'accords, la première moitié du 3^e segment forme un système non contrastif avec le deuxième segment (il s'agit d'une répétition à l'identique). L'indice principal qui permet de distinguer une frontière à cet endroit est le retour du chant dans l'instrumentation, mais cet indice n'est pas présent dans la séquence d'accords. On retrouve également cette configuration entre le 6^e et le 7^e segment.
- La frontière entre le 4^e et le 5^e segment est placée 8 temps plus tôt dans la segmentation inférée. Ces 8 temps en Eb (0 :56 à 1 :00) sont répétés dans le segment suivant et permettent de constituer un système relativement régulier avec le segment suivant. Dans la version audio du morceau, la dynamique et la présence d'un fill de batterie permettent d'écarter cette hypothèse, mais ces indices ne sont encore une fois pas présents dans la séquence d'accords.
- Le troisième segment en partant de la fin (3 :01 à 3 :28) a été divisé en deux sous-segments. On peut remarquer d'une part que le segment de la référence obtient le même score que les deux segments réunis, et d'autre part que cette sous-division du segment est également plausible d'après l'audio.

Conclusion du chapitre

Le schéma de compression tensorielle par analogie présenté dans ce chapitre a permis d'apporter à la fois une extension formelle au principe d'encodage par réplication présenté dans le chapitre précédent et un gain de performances dans le cadre de notre protocole expérimental de segmentation structurelle de séquences d'accords sur le corpus RWC-Pop.

En exploitant les systèmes d'analogie dans les segments musicaux et les schémas implicatifs qu'ils induisent, nous avons étudié des cas de dimension croissante, qui généralisent le modèle Système & Contraste (patron carré) à des patrons cubiques, tesseractiques voire hypercubiques de dimension quelconque.

Cette étude approfondie a permis de concevoir un algorithme pour déterminer le schéma de projection optimal permettant de minimiser la complexité de la description de la séquence par implications successives dans des sous-systèmes d'analogie.

L'algorithme proposé simplifie la combinatoire inhérente au critère considéré car il ramène son calcul à un problème (au pire) quadratique en fonction du nombre d'éléments dans l'hypercube.

L'application de ces méthodes à la segmentation structurelle de séquences d'accords produit des résultats pertinents, qui atteignent 70% de F-mesure. Ces résultats sont d'autant plus intéressants qu'ils sont peu sensibles au réglage des hyper-paramètres et robustes à la finesse de

la segmentation recherchée. Par ailleurs ils n'exploitent aucune connaissance musicologique élaborée (uniquement le cercle des tierces), ni aucune statistique inter-opus sur les propriétés des segments structurels.

L'ensemble de ces résultats, formels et expérimentaux, conforte donc l'intérêt à incorporer les schémas de compression tensorielle par analogie dans les paradigmes d'inférence de la structure musicale, de certains aspects de laquelle ils semblent permettre de rendre compte pertinemment.

Chapitre 7

Conclusions et perspectives

Dans cette thèse, nous nous sommes intéressés au problème de l'inférence de la structure musicale à partir de séquences symboliques discrètes, et plus particulièrement des séquences d'accords. La structure musicale acceptant de nombreuses définitions, nous nous sommes focalisés sur un point de vue particulier consistant à aborder la structure musicale comme étant la façon d'organiser les données observées de sorte à permettre une description simple du morceau de musique considéré au sens d'une famille de modèles prédéfinie.

Afin de transcrire cette définition en critères applicables par des algorithmes, nous nous sommes penchés sur les concepts clés de la théorie de l'information que sont l'entropie de Shannon et la complexité de Kolmogorov. Sans constituer une application exacte de ces concepts, les méthodes d'inférence que nous avons étudiées sont guidées par des principes sous-jacents de compression d'information et peuvent être assimilés à méthodes d'encodage sans perte des séquences musicales.

Dans les chapitres 3 et 4, nous avons étudié une première famille de méthodes d'inférence de structure, à base de Grammaires à Dérivation Unique, qui est en lignée directe avec les méthodes générales de compression sans perte, initialement utilisées pour la découverte de structure dans l'ADN [GALLÉ, 2011]. Ces approches utilisent une classe restreinte de grammaires hors-contexte pour procéder à la factorisation récursive de séquences répétées.

Nos expériences d'applications de cette méthode à des séquences d'accords ont montré que leur capacité à se transposer à ce type de données musicales est problématique. Si la première phase de l'algorithme, qui sélectionne itérativement les factorisations à effectuer, donne des résultats encourageants, la seconde phase, qui ré-écrit les règles sélectionnées de façon optimale les unes en fonction des autres, ne semble avoir pratiquement aucun effet sur le processus d'inférence pour les séquences musicales.

À partir de ces expériences, nous avons identifié plusieurs tendances systématiques dans les différences entre la structure inférée et les annotations de référence. Une des tendances principales est l'incapacité du modèle à représenter les variantes et les terminaisons sur des sections dont la répétition n'est pas exacte.

Pour remédier à cette limitation, nous avons proposé une extension du modèle de grammaire utilisé afin de permettre une meilleure prise en compte des répétitions inexactes, ainsi que des

modifications sur l'algorithme lui permettant d'exploiter de nouvelles capacités. Face aux autres erreurs systématiques, nous avons décrit un cadre pour la définition d'un critère de complexité plus détaillé, en s'appuyant sur des principes pouvant s'interpréter en termes de complexité de Kolmogorov et d'entropie de Shannon. L'incorporation de ces modifications dans l'algorithme IRRMGP n'a toutefois pas permis de mettre en évidence une différence significative des performances d'inférence de la structure musicale.

En termes de perspectives sur cette partie du travail concernant les GDU, il s'avère nécessaire de remédier aux difficultés d'optimisation qui ont été mises en évidence, afin de pouvoir évaluer la valeur réelle des extensions proposées sur le modèle proprement dit. Une intégration des éditions dans la phase gloutonne en permettrait une utilisation plus efficace dans la grammaire finale. L'utilisation d'une stratégie de remplacement de répétitions à plus long terme pourrait également permettre à l'algorithme d'éviter certaines de ses erreurs courantes. Enfin, la substitution de l'algorithme IRR par un algorithme différent remplissant un rôle similaire, mais évitant ses faiblesses, ne remettrait pas en cause les extensions proposées pour le modèle de représentation et pour le critère de complexité.

Si ces difficultés sont surmontées, le choix des types d'éditions et l'instanciation des différentes parties du critère de complexité offre un large champ d'exploration dont la surface a seulement été effleuré : les types d'éditions que nous avons retenues sont très génériques et ne sont pas toujours en adéquation avec une utilisation sur des séquences musicales ; de même, l'utilisation de critères plus musicalement informés serait probablement bénéfique.

Sur la deuxième partie de cette thèse (chapitres 5 et 6), nous avons défini une méthode d'estimation de la complexité d'une séquence musicale à l'échelle du segment structurel reposant sur un principe de compression tensorielle. On considère dans ce cadre que la segmentation macroscopique d'un morceau repose principalement sur l'optimisation et l'économie des propriétés internes des segments qui la constituent.

Fondée sur une extension du modèle Système & Contraste [BIMBOT et al., 2012], la modélisation tensorielle permet de délinéariser la séquence des éléments musicaux en les répartissant dans des patrons à base hypercubique selon des systèmes de similarité et/ou d'analogie multiéchelle. Ces patrons sont éventuellement altérés par des déformations qui permettent de rendre compte d'une plus grande variété de longueurs et d'organisations de segments.

L'estimation de la complexité du segment s'opère par la construction d'implications dans le système, qui permettent de définir des éléments virtuels qui constituent à une projection logique des éléments précédents. Chaque déviation de l'élément observé par rapport à l'élément virtuel constitue un contraste local, qui contribue la complexité totale du système, de même que les déformations du patron tensoriel par rapport au patron de référence hypercubique.

Les développements formels et algorithmiques présentés dans les chapitres 5 et 6 établissent des procédés de compression permettant de grouper les éléments musicaux selon des patrons tensoriels et des critères de complexité de sophistication croissante, mais présentant le point commun de construire puis d'utiliser des schémas implicatifs qui se formulent à partir de propriétés simples sur la position des contrastes antérieurs et s'appuyant sur des systèmes carrés

transversaux résidant dans l'hypercube support. Il en découle un algorithme qui permet d'optimiser le critère de complexité avec une combinatoire maîtrisée, même pour des hypercubes de dimension conséquente.

L'application du modèle de compression tensorielle pour la segmentation de séquences d'accords vient confirmer sa pertinence. Couplée à une régularisation permettant de fixer l'échelle de segmentation autour d'une valeur cible, la méthode à base de compression tensorielle permet d'obtenir, sur la seule base des accords, une F-mesure atteignant 70% sur les frontières de segments sans exploiter aucune connaissance musicologique sophistiquée ni aucune propriété statistique sur les segments structurels.

Une première piste d'extension de ce modèle est d'améliorer l'éventail des patrons tensoriels au-delà des hypercubes et de leurs déformations simples. En effet, la possibilité de rendre compte de structures de segments non-hypercubiques s'avère essentielle pour traiter la diversité des formes de segments rencontrés en musique (y compris en musique pop, pourtant considérée injustement comme assez pauvre à cet égard). Cette thèse a permis d'y remédier en partie, mais il reste tout un pan à explorer dans la recherche des familles de patrons tensoriels nécessaires et suffisantes pour généraliser efficacement le paradigme proposé.

Une deuxième marge d'amélioration réside dans la nature et les propriétés de l'espace des relations entre éléments servant à construire les systèmes d'analogie. Dans ce travail, nous avons considéré de façon très sommaire le cercle des tierces, qui a le mérite de garantir l'existence et l'unicité de tout élément impliqué. Il est toutefois envisageable, et même assurément souhaitable, d'adapter la méthode d'estimation de complexité afin de gérer la co-existence de plusieurs éléments virtuels possibles. Cette adaptation permettrait de faciliter l'application de la méthode à des types de données sur lesquelles il est difficile de définir un espace de relations simple et pertinent, vérifiant qui plus est toutes les hypothèses adéquates, par exemple si l'on veut modéliser des systèmes de motifs rythmiques.

Une des limitations de la méthode par encodage tensoriel telle que nous l'avons appliquée est qu'elle ignore complètement les répétitions à long terme dans un morceau, qui est pourtant généralement considérée comme une des caractéristiques principales de la structure musicale. Le cœur de l'algorithme consistant à effectuer l'optimisation de la segmentation comme une simple recherche de plus court chemin, l'intégration d'un critère de répétition semble être une piste réalisable et à très fort potentiel.

Au cours de cette thèse, nous nous sommes exclusivement intéressés au cas où l'on dispose d'une unique séquence symbolique décrivant le morceau, correspondant en général à une seule dimension musicale. La musique étant un signal très riche, il s'agit d'une représentation très réductrice : pour un seul morceau, on pourrait ainsi avoir simultanément à disposition des données symboliques multiflux, sur l'harmonie, l'instrumentation, les motifs rythmiques, la mélodie, la dynamique et encore bien d'autres. On pourrait envisager de traiter individuellement chacune de ces strates d'informations, puis d'effectuer une synthèse des segmentations obtenues, ou, mieux encore, une compression conjointe

Une application à des données audio semble constituer une piste extrêmement stimulante, dans l'optique de procéder à une segmentation structurelle concomitante à l'extraction de descripteurs musicaux par échange réciproque d'informations entre les algorithmes de d'estimation

de structure et d'extraction des descripteurs, ou en guidant l'un par l'autre, comme peuvent le faire les modèles de langage en reconnaissance automatique de la parole.

L'ensemble des perspectives esquissées pourront trouver dans les aspects développés dans la présente thèse quelques éléments sur lesquels s'appuyer.

Bibliographie

- ABDALLAH, Samer A., Henrik EKEUS, Peter FOSTER, Andrew ROBERTSON et Mark D. PLUMBLEY (2012). « Cognitive Music Modelling : An Information Dynamics Approach ». In : *3rd International Workshop on Cognitive Information Processing (CIP)* (cf. page 28).
- ABDALLAH, Samer A. et Mark D. PLUMBLEY (2009). « Information Dynamics : Patterns of expectation and surprise in the perception of music ». In : *Connection Science* 21.2-3, pages 89–117 (cf. page 28).
- ABOUElhODA, Mohamed Ibrahim, Stefan KURTZ et Enno OHLEBUSCH (2004). « Replacing suffix trees with enhanced suffix arrays ». In : *Journal of Discrete Algorithms* 2.1, pages 53–86 (cf. page 35).
- ADLER, Amir, Valentin EMIYA, Maria G. JAFARI, Michael ELAD, Rémi GRIBONVAL et Mark D. PLUMBLEY (2012). « Audio inpainting ». In : *IEEE Transactions on Audio, Speech and Language Processing* 20.3, pages 922–932 (cf. page 13).
- ALLALI, Julien, Pascal FERRARO, Pierre HANNA et Costas ILIOPOULOS (2007). « Local Transpositions in Alignment of Polyphonic Musical Sequences ». In : *String Processing and Information Retrieval (SPIRE)*, pages 26–38 (cf. page 64).
- ALLISON, Lloyd, Timothy EDGOOSE et Trevor I. DIX (1998). « Compression of strings with approximate repeats ». In : *Intelligent Systems for Molecular Biology (ISMB)*, pages 8–16 (cf. page 34).
- ALTSCHUL, Stephen F., Warren GISH, Webb MILLER, Eugene W. MYERS et David J. LIPMAN (1990). « Basic Local Alignment Search Tool ». In : *Journal of Molecular Biology* 215.3, pages 403–10 (cf. page 66).
- ANDREATTA, Moreno (2009). « Calcul algébrique et calcul catégoriel en musique ». In : *Le calcul de la musique : Composition, modèles & outils*. Sous la direction de Laurent POTIER. Publications de l’université de Saint-Etienne (cf. page 121).
- (2016). « Topological structures in computer-aided music analysis ». In : *Computational Music Analysis*, pages 57–80 (cf. page 121).
- APOSTOLICO, Alberto et Stefano LONARDI (2000). « Off-line Compression by Greedy Textual Substitution ». In : *Proceedings of the IEEE* 88.11, pages 1733–1744 (cf. page 34).
- AUCOUTURIER, Jean-Julien et Mark SANDLER (2002). « Finding repeating patterns in acoustic musical signals : Applications for audio thumbnailing ». In : *Audio Engineering Society Conference : 22nd International Conference : Virtual, Synthetic, and Entertainment Audio* (cf. page 12).
- BARBAR, Kablan, Myriam DESAINTE-CATHERINE et Alain MINIUSI (1993). « The Semantics of Musical Hierarchies ». In : *Computer Music Journal* 17.4, pages 30–37 (cf. page 15).
- BARONI, Mario, Simon MAGUIRE et William DRABKIN (1983). « The Concept of Musical Grammar ». In : *Music Analysis* 2.2, pages 175–208 (cf. page 29).
- BENT, Ian et William DRABKIN (1987). *The New Grove Handbook in Music Analysis*. Macmillan Press. London (cf. page 16).

- BEURIVÉ, Anthony et Myriam DESAINTE-CATHERINE (2001). « Representing Musical Hierarchies with Constraints ». In : *In Seventh International Conference on Principles and Practice of Constraint Programming, Musical Constraints Workshop*. Paphos, pages 23–33 (cf. page 15).
- BIGO, Louis, Daniele GHISI, Antoine SPICHER et Moreno ANDREATTA (2014). « Spatial Transformations in Simplicial Chord Spaces ». In : *Proceedings ICMC/SMC*. September, pages 1112–1119 (cf. page 121).
- BILLE, Philip (2005). « A Survey on Tree Edit Distance and Related Problems ». In : *Theoretical Computer Science* 337.1-3, pages 217–239 (cf. page 56).
- BIMBOT, Frédéric (2016). « Towards an Information-Theoretic Framework for Music Structure ». In : *Dagstuhl Seminar on Computational Music Structure Analysis*. Dagstuhl, Germany, pp. 167–168 (cf. pages 5, 28, 99).
- BIMBOT, Frédéric, Emmanuel DERUTY, Gabriel SARGENT et Emmanuel VINCENT (2011). « Methodology and Resources for the Structural Segmentation of Music Pieces into Autonomous and Comparable Blocks ». In : *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)* Ismir, pages 287–292 (cf. page 14).
- (2012). « Semiotic structure labeling of music pieces : concepts, methods and annotation conventions ». In : *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*. Ismir. Porto, Portugal (cf. pages 14, 19, 21, 81, 136).
- (2016). « System & Contrast : a Polymorphous Model of the Inner Organization of Structural Segments Within Music Pieces. » In : *Music Perception* 33.5, pages 631–661 (cf. page 82).
- BIMBOT, Frédéric, O LE BLOUCH, Gabriel SARGENT et Emmanuel VINCENT (2010). « Decomposition into Autonomous and Comparable Blocks : a Structural Description of Music Pieces ». In : *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 189–194 (cf. page 14).
- BIMBOT, Frédéric, Gabriel SARGENT, Emmanuel DERUTY, Corentin GUICHAOUA et Emmanuel VINCENT (2014). « Semiotic Description of Music Structure : An Introduction to the Quaero/Metiss Structural Annotations ». In : *Audio Engineering Society Conference : 53rd International Conference : Semantic Audio*. London, United Kingdom (cf. page 52).
- BRUDERER, Michael J., Martin MCKINNEY et Armin KOHLRAUSCH (2006). « Structural Boundary Perception in Popular Music ». In : *Proceedings of the 7th International Society of Music Information Retrieval Conference (ISMIR)*. Tome 4, pages 198–201 (cf. page 14).
- BURROWS, Michael et David J. WHEELER (1994). *A Block-sorting Lossless Data Compression Algorithm*. Rapport technique 124. Systems Research Center, page 24 (cf. page 40).
- CAMBOUROPOULOS, Emiliios, Maxime CROCHEMORE, Costas ILIOPOULOS, Laurent MOUCHARD et Yoan PINZON (2002). « Algorithms For Computing Approximate Repetitions In Musical Sequences ». In : *International Journal of Computer Mathematics* 79.11, pages 1135–1148 (cf. page 66).
- CARRASCOSA, Rafael, François COSTE, Matthias GALLÉ et Gabriel INFANTE-LOPEZ (2010). « Choosing Word Occurrences for the Smallest Grammar Problem ». In : *International Conference on Language and Automata Theory and Applications*. Berlin, Heidelberg, pages 154–165 (cf. page 35).
- (2011a). « Searching for Smallest Grammars on Large Sequences and Application to DNA ». In : *Journal of Discrete Algorithms* 11.1, pages 62–72 (cf. page 35).
- (2011b). « The Smallest Grammar Problem as Constituents Choice and Minimal Grammar Parsing ». In : *Algorithms* 4.4, pages 262–284 (cf. page 35).
- CERRA, Daniele et Mihai DATCU (2010). « A Similarity Measure using Smallest Context-free Grammars ». In : *Data Compression Conference*, pages 346–355 (cf. page 33).

- CHAITIN, Gregory J. (1966). « On the Length of Programs for Computing Finite Binary Sequences ». In : *Journal of the ACM* 13.4, pages 547–569 (cf. page 26).
- CHARIKAR, Moses, Eric LEHMAN et Ding LIU (2002). « Approximating the Smallest Grammar : Kolmogorov Complexity in Natural Models ». In : *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*. ACM, pages 792–801 (cf. page 34).
- CHARIKAR, Moses, Eric LEHMAN, Ding LIU et al. (2005). « The Smallest Grammar Problem ». In : *IEEE Transactions on Information Theory* 51.7, pages 2554–2576 (cf. page 34).
- CHATER, Nick (1999). « The Search for Simplicity : A Fundamental Cognitive Principle ? » In : *The Quarterly Journal of Experimental Psychology* 52A.2, pages 273–302 (cf. page 28).
- CHO, Taemin (2011). *Manually Annotated Chord Data Set of US Pop Songs and Popular Music Collection of RWC Music Database*. URL : <https://github.com/tmc323/Chord-Annotations> (cf. page 41).
- CHOMSKY, Noam (1956). « Three Models for the Description of Language ». In : *IEEE Transactions on Information Theory* 2.3, pages 113–124 (cf. page 29).
- (1957). *Syntactic Structures*. Walter de Gruyter (cf. page 29).
- CILIBRASI, Rudi et Paul M.B. VITÁNYI (2005). « Clustering by Compression ». In : *IEEE Transactions on Information Theory* 51.4, pages 1523–1545 (cf. page 26).
- CLIFFORD, Raphaël et Costas ILIOPOULOS (2004). « Approximate String Matching for Music Analysis ». In : *Soft Computing* 8.9, pages 597–603 (cf. page 66).
- CONT, Arshia (2008). « ANTESCOFO : Anticipatory Synchronization and Control of Interactive Parameters in Computer Music ». In : *International Computer Music Conference (ICMC)* (cf. page 13).
- DANNENBERG, Roger B. (2005). « Toward automated holistic beat tracking, music analysis, and understanding ». In : *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 366–373 (cf. page 13).
- DAVIES, Matthew E. P., Philippe HAMEL, Kazuyoshi YOSHII et Masataka GOTO (2013). « AutoMashUpper : An Automatic Multi-Song Mashup System ». In : *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*. Sous la direction d’Alceu DE SOUZA BRITTO JR., Fabien GOUYON et Simon DIXON. Curitiba, Brazil, pages 575–580 (cf. page 12).
- DE HAAS, W. B. (2012). « Music Information Retrieval Based on Tonal Harmony ». Thèse de doctorat. Utrecht University (cf. page 15).
- DERUTY, Emmanuel, Frédéric BIMBOT et Brigitte VAN WYMEERSCH (2013). *Methodological and Musicological Investigation of the System & Contrast Model for Musical Form Description*. Research Report RR-8510. INRIA, page 117 (cf. pages 13, 82).
- DOMINGOS, Pedro (1999). « The Role of Occam’s Razor in Knowledge Discovery ». In : *Data Mining and Knowledge Discovery* 3, pages 409–425 (cf. page 28).
- EHMANN, Andreas F., Mert BAY, J. Stephen DOWNIE, Ichiro FUJINAGA et David DE ROURE (2011). « Music Structure Segmentation Algorithm Evaluation : Expanding on MIREX 2010 Analyses and Datasets ». In : *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. November 2014, pages 561–566 (cf. page 40).
- GALLÉ, Matthias (2010). « A New Tree Distance Metric for Structural Comparison of Sequences ». In : *Dagstuhl Seminar : Structure Discovery in Biology : Motifs, Networks & Phylogenies* (cf. page 56).
- (2011). « Searching for Compact Hierarchical Structures in DNA by Means of the Smallest Grammar Problem ». Thèse de doctorat. Université de Rennes 1 (cf. pages 5, 19, 29, 45, 48, 135).

- GALLÉ, Matthias, Pierre PETERLONGO et François COSTE (2008). « In-place Update of Suffix Array while Recoding Words ». In : *International Journal of Foundations of Computer Science* 20.6, pages 54–67 (cf. page 35).
- GIRAUD, Mathieu et Sławek STAWORKO (2015). « Modeling Musical Structure with Parametric Grammars ». In : *Mathematics and Computation in Music*, pages 85–96 (cf. page 64).
- GOTO, Masataka (2006). « AIST Annotation for RWC Music Database ». In : *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, pages 359–360 (cf. page 42).
- GOTO, Masataka, Hiroki HASHIGUCHI, Takuichi NISHIMURA et Ryuichi OKA (2002). « RWC Music Database : Popular, Classical and Jazz Music Databases ». In : *Proceedings of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*. Tome 2. October, pages 287–288 (cf. page 40).
- GROSCHE, Peter, Joan SERRÀ, Meinard MÜLLER et Josep Lluís ARCOS (2012). « Structure-Based Audio Fingerprinting for Music Retrieval ». In : *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*. Ismir. Porto, Portugal, pages 55–60 (cf. page 12).
- GRÜNWARD, Peter D. (2007). *The Minimum Description Length Principle*. MIT press, page 736 (cf. page 28).
- GRÜNWARD, Peter D. et Paul M.B. VITÁNYI (2003). « Kolmogorov Complexity and Information Theory : with an Interpretation in terms of Questions and Answers ». In : *Journal of Logic, Language and Information* 12, pages 497–529 (cf. page 26).
- HIGUERA, Colin de la (2009). *Grammatical Inference : Learning Automata and Grammars*. Cambridge University Press (cf. page 31).
- KILIAN, Jurgen et Holger H. HOOS (2004). « MusicBLAST—Gapped Sequence Alignment for MIR ». In : *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 2–5 (cf. page 66).
- KLAVANS, Judith et al. (1991). « Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars ». In : *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, pages 306–311 (cf. page 56).
- KOLMOGOROV, Andrey (1965). « Three Approaches to the quantitative definition of Information ». In : *Problems of Information Transmission* 1.1, pages 3–11 (cf. page 26).
- (1968). « Three Approaches to the Quantitative Definition of Information * ». In : *International Journal of Computer Mathematics* 2.1-4, pages 157–168 (cf. page 26).
- LAMERE, Paul (2012). *The Infinite Jukebox*. URL : <https://musicmachinery.com/2012/11/12/the-infinite-jukebox/> (visité le 2017) (cf. page 12).
- LARSSON, N. Jesper et Alistair MOFFAT (2000). « Off-line Dictionary-based Compression ». In : *Proceedings of the IEEE* 88.11, pages 1722–1732 (cf. pages 34, 35).
- LEHMAN, Eric et A. SHELAT (2002). « Approximation Algorithms for Grammar-based Compression ». In : *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 205–212 (cf. page 34).
- LERDAHL, Fred et Ray JACKENDOFF (1985). *A Generative Theory of Tonal Music*. MIT press (cf. page 15).
- LEVENSHTAIN, Vladimir I. (1966). « Binary Codes Capable of Correcting Deletions, Insertions, and Reversals ». In : *Soviet Physics Doklady*. Tome 10. 8, pages 707–710 (cf. page 64).
- LÉVY, Fabien (2004). « Complexité grammatologique et complexité apercéptive en musique : étude esthétique et scientifique du décalage entre la pensée de l’écriture et la perception cognitive des processus musicaux sous l’angle des théories de l’information et de la complexité ». Thèse de doctorat, 1 vol. (387 p.) (Cf. page 16).

- LI, Ming, Xin CHEN, Xin LI, Bin MA et Paul M.B. VITÁNYI (2004). « The Similarity Metric ». In : *IEEE Transactions on Information Theory* 50.12, pages 3250–3264 (cf. page 26).
- LI, Ming et Paul M.B. VITÁNYI (1991). « Combinatorics and Kolmogorov complexity ». In : *1991 Proceedings of the Sixth Annual Structure in Complexity Theory Conference* (cf. page 26).
- LIUTKUS, Antoine, Zafar RAFII, Roland BADEAU, Bryan PARDO et Gaël RICHARD (2012). « Adaptive Filtering for Music/Voice Separation Exploiting the Repeating Musical Structure ». In : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pages 53–56 (cf. page 12).
- LOGAN, Beth et Stephen CHU (2000). « Music Summarization using Key Phrases ». In : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Tome 2, pages 749–752 (cf. page 12).
- LOUBOUTIN, Corentin et Frédéric BIMBOT (2016). « Description of Chord Progressions by Minimal Transport Graphs Using the System & Contrast Model ». In : *42nd International Computer Music Conference (ICMC)*. Utrecht, Netherlands (cf. pages 82, 103).
- (2017a). « Modeling the Multiscale Structure of Chord Sequences using Polytopic Graphs ». In : *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*. Suzhou, China (cf. page 82).
- (2017b). « Polytopic Graph of Latent Relations : A Multiscale Structure Model for Music Segments ». In : *6th International Conference of the Society of Mathematics and Computation in Music (MCM)*. Mexico City, Mexico (cf. page 82).
- LOUBOUTIN, Corentin et David MEREDITH (2016). « Using General-purpose Compression Algorithms for Music Analysis ». In : *Journal of New Music Research* 45.1, pages 1–16 (cf. page 40).
- MARSDEN, Alan (2010). « Schenkerian Analysis by Computer : A Proof of Concept ». In : *Journal of New Music Research* 39.3, pages 269–289 (cf. page 15).
- MARTIN, Benjamin (2012). « Analyse de structures répétitives dans les séquences musicales ». Thèse de doctorat. Université Bordeaux 1, page 168 (cf. pages 13, 15, 56).
- MARTIN, Benjamin, Daniel G BROWN, Pierre HANNA et Pascal FERRARO (2012). « BLAST for Audio Sequences Alignment : A Fast Scalable Cover Identification Tool ». In : *Ismir'12 Ismir*, pages 529–534 (cf. page 68).
- MAUCH, Matthias, Katy NOLAND et Simon DIXON (2009). « Using Musical Structure to Enhance Automatic Chord Transcription ». In : *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*. Kobe, Japan, pages 231–236 (cf. page 13).
- MCFEE, Brian et Daniel P.W. ELLIS (2014). « Analyzing Song Structure with Spectral Clustering ». In : *Proceedings of the 15th International Society for Music Information Retrieval Conference Ismir*, pages 405–410 (cf. page 15).
- MCFEE, Brian, Oriol NIETO et Juan P. BELLO (2015). « Hierarchical Evaluation of Segment Boundary Detection ». In : *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 406–412 (cf. page 57).
- MCVICAR, Matt et Daniel P.W. ELLIS (2014). « Leveraging Repetition for Improved Automatic Lyric Transcription in Popular Music ». In : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3117–3121 (cf. page 12).
- MEREDITH, David (2012). « A Geometric Language for Representing Structure in Polyphonic Music ». In : *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*. Porto, Portugal, pages 133–8 (cf. page 28).
- MONGEAU, Marcel et David SANKOFF (1990). « Comparison of Musical Sequences ». In : *Computers and the Humanities* 24.3, pages 161–175 (cf. page 64).

- MÜLLER, Meinard et Frank KURTH (2007). « Towards Structural Analysis of Audio Recordings in the Presence of Musical Variations ». In : *Eurasip Journal on Advances in Signal Processing* 2007 (cf. page 18).
- MUSICBRAINZ (2017). *Database Statistics*. URL : <https://musicbrainz.org/statistics> (cf. page 10).
- NARMOUR, Eugene (1990). « The Analysis and Cognition of Basic Melodic Structures : The implication-Realization Model ». In : (cf. page 15).
- (1992). *The Analysis and Cognition of Melodic Complexity : The Implication-Realization Model*. University of Chicago Press (cf. pages 15, 99).
- (2000). « Music Expectation By Cognitive Rule-Mapping ». In : *Music Perception* 17.3, pages 329–398 (cf. page 99).
- NEVILL-MANNING, Craig G. (1996). « Inferring Sequential Structure ». Thèse de doctorat (cf. pages 32, 40).
- NEVILL-MANNING, Craig G. et Ian H. WITTEN (1997). « Identifying Hierarchical Structure in Sequences : A Linear-time Algorithm ». In : *Journal of Artificial Intelligence Research* 7, pages 67–82 (cf. page 34).
- PACHET, François (1999). « Surprising Harmonies ». In : *International Journal of Computing Anticipatory Systems* 4.2, pages 1–20 (cf. page 40).
- (2003). « The Continuator : Musical Interaction With Style ». In : *Journal of New Music Research* 32.3, pages 333–341 (cf. page 13).
- PAULUS, Jouni, Meinard MÜLLER et Anssi P. K LAPURI (2010). « Audio-Based Music Structure Analysis ». In : *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*. Ismir, pages 625–636 (cf. page 15).
- PAUWELS, Johan et Geoffroy PEETERS (2013). « Segmenting Music through the Joint Estimation of Keys, Chords and Structural Boundaries ». In : *Proceedings of the 21st ACM international conference on Multimedia*, pages 741–744 (cf. page 13).
- PEARCE, Marcus T. (2005). « The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition ». Thèse de doctorat. City University London (cf. page 28).
- PEARCE, Marcus T. et Geraint A. WIGGINS (2006). « Expextation in Melody : The Influence of Context and Learning ». In : *Music Perception* 23.4, pages 377–405 (cf. page 28).
- PEETERS, Geoffroy et Emmanuel DERUTY (2009). « Is Music Structure Annotation Multi-Dimensional? A Proposal for Robust Local Music Annotation ». In : *Proceedings of the 3rd International Worskhop on Learning Semantics of Audio Signals* part 2, pages 75–90 (cf. page 14).
- REHBEIN, Ines et Josef Van GENABITH (2007). « Treebank Annotation Schemes and Parser Evaluation for German ». In : *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. June, pages 630–639 (cf. page 56).
- RISSANEN, Jorma (1978). « Modelling by the Shortest Data Description ». In : *Automatica* 14, pages 465–471 (cf. page 28).
- ROADS, Curtis (1979). « Grammars as Representation for Music ». In : *Computer Music Journal* 3.1, pages 48–55 (cf. page 29).
- SAKAMOTO, Hiroshi (2005). « A Fully Linear-time Approximation Algorithm for Grammar-based Compression ». In : *Journal of Discrete Algorithms* 3.2-4, pages 416–430 (cf. page 32).
- SARGENT, Gabriel, Frédéric BIMBOT et Emmanuel VINCENT (2011). « A Regularity-constrained Viterbi Algorithm and its Application to the Structural Segmentation of Songs ». In : *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. Miami, Florida, pages 483–488 (cf. page 126).

- SCHENKER, H (1935). « Der freie Satz ». In : *Vienna : Universal Edition. (translated 1979)* (cf. page 15).
- SHANNON, Claude E. (1948). « A Mathematical Theory of Communication ». In : *The Bell System Technical Journal* 27.3, pages 379–423 (cf. page 27).
- SMITH, Jordan B.L. (2014). « Explaining Listener Differences in the Perception of Musical Structure ». Thèse de doctorat. Queen Mary, University of London, UK (cf. page 13).
- SMITH, Jordan B.L., John Ashley BURGOYNE, Ichiro FUJINAGA, David DE ROURE et J. Stephen DOWNIE (2011). « Design and Creation of a Large-Scale Database of Structural Annotations ». In : *International Conference on Music Information Retrieval (ISMIR)*, pages 555–560 (cf. pages 13, 56).
- SOLOMONOFF, R.J. (1964). « A Formal Theory of Inductive Inference, Part I ». In : *Information and Control* 7.1, pages 1–22 (cf. page 26).
- STORER, James A. et Thomas G. SZYMANSKI (1982). « Data Compression via Textural Substitution ». In : *Journal of the ACM* 29.4, pages 928–951 (cf. page 34).
- WIERING, Frans, Justin de NOOIJER, Anja VOLK et Hermi TABACHNECK-SCHIJF (2009). « Cognition-based Segmentation for Music Information Retrieval Systems ». In : *Journal of New Music Research* 38.2, pages 139–154 (cf. page 14).
- WIGGINS, Geraint A., Daniel MÜLLENSIEFEN et Marcus T. PEARCE (2010). « On the Non-existence of Music : Why Music Theory is a Figment of the Imagination ». In : *Musicae Scientiae* 14.1_suppl, pages 231–255 (cf. page 28).
- YANG, En hui et Da K. HE (2003). « Efficient Universal Lossless Data Compression Algorithms Based on a Greedy Sequential Grammar Transform - Part Two : With Context Models ». In : *IEEE Transactions on Information Theory* 49.11, pages 2874–2894 (cf. page 32).
- ZIV, Jacob et Abraham LEMPEL (1977). « A Universal Algorithm for Sequential Data Compression ». In : *IEEE Transactions on Information Theory* 23.3, pages 337–343 (cf. page 34).
- (1978). « Compression of Individual Sequences via Variable-rate Coding ». In : *IEEE Transactions on Information Theory* 24.5, pages 530–536 (cf. page 34).