Table des matières

Résumé

Dans le cadre du système MusES de représentation des connaissances musicales et du système à base de règles NéOpus, on s'intéresse à l'analyse automatique de grilles de jazz traditionnelles. Il est proposé ici un modèle temporel adapté à la représentation des séquences d'accords tels qu'on peut les trouver dans de telles grilles, ainsi qu'une grammaire de reconnaissance des formes classiques en jazz, en particulier des différentes formes de résolvantes.

Le travail effectué a donné lieu à une publication et à une communication dans une conférence internationale (ICMAI 95), et d'autres articles sont en préparation.

1. Introduction

1.1 Le cadre général

1.1.1 Le système MusES

La musique, et peut-être spécialement la musique tonale, constitue un cadre précis, dont les règles sont connues, relativement précises et consensuelles, tout en offrant un grand degré de complexité, des problèmes non triviaux de représentation des connaissances, que ce soit pour les objets manipulés par le discours musical (notes, hauteurs, accords, etc.) ou pour ses aspects temporels ([Pachet 95]). Le système MusES, écrit dans le langage Smalltalk, offre des techniques de représentation des connaissances musicales dans un contexte de musique tonale, en utilisant des techniques de programmation par objets. Il intègre des mécanismes de description des notes, des hauteurs, des intervalles, des gammes, des accords, etc. Une description plus complète du système peut-être trouvée dans [Pachet 93].

Plusieurs applications ont été construites à partir de MusES : un système de génération d'improvisation en jazz ([Ramalho & Pachet 94]), un système d'harmonisation automatique de chorals ([Pachet & Roy 95]), ainsi qu'un système d'analyse de grilles de jazz ([Mouton & Pachet 95]). Le travail effectué durant le stage avait pour objet ce dernier système.

1.1.2 Analyse automatique de grilles de jazz

Le système cherche à analyser des grilles d'accords de jazz telles que celles que l'on peut trouver dans les recueils standards, comme le Real Book [Real 81] ou le Fake Book [Fake 83 ; 91]. Il s'agit de détecter la plupart des régularités qui se trouvent dans ces recueils (voir [Pachet & al. 95]). Deux exemples de grille analysées se trouvent en annexe (pages 24 et 26).

L'analyse que l'on souhaite obtenir est un arbre dont les nœuds représentent des formes d'autant plus abstraites que leur profondeur est faible, la racine pouvant dans les cas les plus favorables décrire une forme typique telle qu'un Blues, une forme AABA, etc. Les formes les plus « profondes », celles qui sont le plus près des accords de la grille, décrivent des types de séquences d'accords standards du jazz, comme les *deux-cinq* (deuxième degré suivi du cinquième degré), les *résolutions* (ou *cadences*), les *substitutions*, etc. A chacune des formes reconnues par les procédures d'analyse est associée une tonalité. Ainsi, un même accord se trouvera-t-il analysable dans plusieurs tonalités différentes, selon la forme que l'on considère, ou, en d'autres termes, selon le degré d'abstraction que l'on souhaite avoir, de la distance avec laquelle on examine cet accord. Ainsi, dans la suite d'accords illustrée par la figure 1, l'accord [Ab min 7] peut-il être analysé dans trois tonalités différentes selon la forme considérée. C'est seulement une fois qu'un contexte plus large est connu que l'on peut affirmer que le premier deux-cinq qui était analysé en sol bémol est en réalité en deux-cinq substitué en fa, et ainsi de suite. On s'aperçoit donc qu'un accord ne peut s'analyser que dans un contexte, et que la notion même de tonalité n'a de sens que si l'on tient compte de ce contexte.

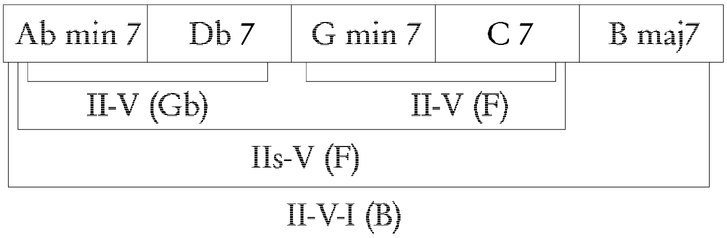


figure 1 : structure hiérarchique d'une analyse

1.2 *Présentation du travail effectué*

1.2.1 L'existant

1.2.1.1 Le système d'analyse

L'analyse repose sur un système de bases de règles, le moteur d'inférence étant le système NéOpus, qui combine règles et objets ([Pachet 92]). L'analyse se déroule en deux phases principales . Tout d'abord une phase de *reconnaissances de formes* . Dans un premier temps, à chaque accord est associé l'ensemble des tonalités dans lesquelles cet accord est analysable. Ainsi, l'accord Ré-Fa-La-Do peut-il être considéré comme un deuxième degré de Do majeur, un sixième degré de Fa majeur, un troisième degré de Si bémol majeur, etc. Cet opération est totalement déterministe. Dans un second temps, une première base de règles s'attache à reconnaître des formes relativement simples et bien connues, telles que les deux-cinq et les résolvantes . La seconde phase est une phase de *remplissage*, qui agglomère les formes reconnues suivant des règles moins absolues. Il s'agira par exemple, dans le cas d'un accord isolé précédant une forme reconnue, d'intégrer cet accord à cette forme, pour peu que certaines conditions de cohérence soient respectées, concernant la tonalité en particulier.

Le principe de l'analyse peut être séparé en deux mécanismes principaux : *agrégation* et *oubli*. Lorsque les prémisses d'une règle d'agrégation sont vérifiées, le système instancie un nouvel objet. A partir de la suite d'accords [D min 7 / G 7 / C maj7], il est d'abord créé un objet représentant un *deux-cinq*, qui comprend les deux premiers accords, puis un second représentant une *résolvante*, qui comprend le *deux-cinq* précédemment identifié et le dernier accord. Beaucoup d'objets sont ainsi créés, et certains deviennent inutiles ; ils sont alors éliminés par des règles d'*oubli*. Dans l'exemple précédant, une fois la résolvante reconnue, il est inutile que l'objet représentant le deux-cinq soit pris en compte pour la suite de l'analyse.

Le système est donc constitué de plusieurs bases de règles de niveau d'abstraction croissant, alternant des règles d'agrégation et d'oubli. C'est sans doute ici que se trouvent l'originalité et l'efficacité du système.

1.2.1.2 Les lacunes du système

Une des caractéristiques d'une grille de jazz est son apsect cyclique : non seulement la grille est répétée plusieurs fois, mais encore la dernière partie de la grille est une préparation à la reprise de la grille à son début. Certaines formes à reconnaître peuvent donc se situer à l'« intérieur » de la grille, ou bien « à cheval » sur la fin et le début. Le système tel qu'il était ne tenait que très imparfaitement compte de cette particularité. Il ne comportait pas à proprement parler de modèle temporel, et cette circularité n'était prise en compte que dans certaines règles, et ce de manière *ad hoc*. Ce manque de généralité du système entraînait un manque d'efficacité important, en particulier dans la mesure où les formes enjambant le début de la grille n'avait pas exactement le même statut que les autres.

D'autre part, comme on peut le voir sur la figure 1, il existe des aspects récursifs dans la structure harmonique d'une grille de jazz, concernant principalement les deux-cinq et les résolutions, c.-à-d. les formes prises en compte par la première base de règle. Ces aspects récursifs n'étaient pas décrits de manière absolument rigoureuse dans le système d'analyse. Par exemple, n'importe quelle forme pouvant se substituer à un deuxième degré suivie de n'importe quelle forme pouvant se substituer à un cinquième degré était considérée comme constituant un *deux-cinq*, alors que ces conditions ne sont pas assez restrictives.

1.2.2 Ce qui a été fait

La principale partie du travail effectué a consisté à concevoir puis implémenter un modèle temporel rendant compte clairement des aspects circulaires d'une grille de jazz, permettant de représenter les intervalles de temps associés aux diverses séquences d'accord, ainsi que des opérations élémentaires sur ces intervalles de temps, telles que comparaison, union, intersection. Ce modèle temporel s'appuie sur des éléments de logique temporelle.

D'autre part, la base de règles de reconnaissance des formes de base d'une grille, comme les *deux-cinq* et les *résolvantes* a été complètement réécrite. Pour pouvoir écrire cette base de règles, il a préliminairement été écrit une grammaire générative, mettant en évidence les enchaînements de formes et les aspects récursifs vus plus haut. Cette grammaire étant dépendante du contexte, elle ne pouvait par conséquent être implémentée telle quelle. Cependante, les règles d'inférence reprennent le plus exactement possible les règles cette grammaire.

Après avoir resitué l'analyse harmonique de grilles de jazz dans le cadre plus large des travaux d'intelligence artificielle se préoccupant de musique tonale, le modèle temporel proposé sera décrit en détail. La grammaire citée plus haut et d'autres modifications apportées au système, seront présentées parmi les perspectives liées à ce système d'analyse.

2. Analyse harmonique tonale

L'analyse musicale, dont l'analyse harmonique fait partie, est une discipline musicale à part entière ; elle est enseignée en tant que telle dans tout cursus de formation musicale un tant soit peu complet, et de nombreux ouvrages lui sont consacrés. L'analyse harmonique s'inscrit implicitement dans un contexte tonal, quoi que l'on puisse parler d' « harmonie » de la musique dodécaphonique ou de la musique spectrale. C'est dans le cadre strict de la musique tonale, et plus encore du jazz, que nous nous plaçons.

2.1 Informatique et musique tonale

On considère généralement que l'informatique et la musique s'associent dans deux disciplines principales : la composition assistée par ordinateur (CAO) et la synthèse du son. Cependant, d'autres activités intéressent le musicologue, le cognititien ou le spécialiste d'I.A. Ainsi, a-t-il été développé des modèles de perception [Widmer 92], des systèmes d'harmonisation automatique à quatre voix [Ebcioglu 86], d'exécution musicale interactive [Baird & al. 93]. D'autre part, la composition assistée par ordinateur concerne essentiellement la musique contemporaine, dont la musique tonale n'est plus une préoccupation. Quant à la synthèse du son, elle se trouve en amont de tout traitement réellement musical, elle fabrique en quelque sorte une matière première dont l'utilisation lui échappe le plus souvent.

La musique tonale proprement dite, c.-à-d. sans compter les héritiers des langages musicaux utilisant encore plus ou moins un matériau tonal, comme purent l'être ceux de Béla Bartók, et plus récemment d'Olivier Messiaen, la musique tonale a pratiquement cessé d'évoluer, si ce n'est peut-être encore dans le domaine du jazz.

2.2 En quoi consiste l'AHT ?

Jacques Chailley écrit dans son *Traité historique d'analyse musicale* [Chailley 77] : « l'analyse harmonique [...] ne consiste pas à insérer tous les sons entendus dans un catalogue préétabli, forcément arbitraire et incomplet, mais à comprendre *pourquoi chacun d'eux est employé et ce qu'il signifie*. [...] C'est le *rapport [d'un] accord avec ceux qui l'entourent* qui constitue l'élément essentiel de la syntaxe musicale. »

En 1983, F. Lerdahl et R. Jackendoff proposent, en s'appuyant sur les travaux de Chomsky, une « théorie formulée en termes de règles d'une grammaire musicale » [Lerdahl & Jackendoff 83]. Leur but est de donner une « desription formelle des intuitions d'un auditeur familier d'un certain idiome musical ».

Dans cette grammaire, les auteurs distinguent l'application de certaines règles à la musique classique occidentale ou à tel ou tel idiome musical. Les règles sont de deux types : les *well-formedness rules*, qui s'attachent à décrire les structures possibles, et les *preference rules* qui permettent de choisir parmi les descriptions possibles celle qui correspond à l'intuition particulière d'un morceau. Un troisième type de règles, les *transformational rules*, permet d'apporter des distorstions aux règles du premier type (l'élision, par exemple). Ces dernières règles ont un rôle relativement peu important. Pour chaque type de règles, il existe quatre domaines d'application :

- **grouping structure** : application des règles aux motifs, aux phrases et aux sections.
- **metric structure** : application des règles à la structure rythmique, en particulier au temps faibles et forts, à différents niveaux.
- **time-span reduction** : is s'agit là d'affecter aux hauteurs des notes une hiérarchie d'importance structurelle (entre notes constitutives d'un accord et notes étrangères par exemple).
- **prolongational reduction** : on cherche à élaborer une hiérarchie exprimant les tensions et relaxations harmoniques et mélodiques, continuités et progressions.

On trouvera en annexe un schéma des bases de règles utilisées (figure 9 page 13).

Dans le cadre de la musique tonale, plusieurs systèmes sont fondés sur l'utilisation de bases de règles. [Baroni & al. 88] proposent un système d'analyse musicale et poétique des airs de Giovanni Legrenzi ; [Marschen 92] se penche sur le problème de l'extraction de voix dans les fugues à quatre voix de J.S. Bach ; [Baroni & al. 92] présentent un système de production de mélodies à partir des règles régissant la théorie et l'analyse des chorals luthériens et des chansons françaises du XVIII$^{\text{ème}}$ siècle.

Plus proche du contexte qui nous intéresse, [Steedman 84] propose une grammaire générative des grilles traditionnelles de Blues. Cette grammaire n'est cependant pas directement implémentable, parce que dépendante du contexte. D'autre part, une telle grammaire ne peut s'appliquer qu'aux blues canoniques.

2.3 Utilité d'un système d'AHT

Outre son intérêt pédagogique évident, l'analyse harmonique peut aider le compositeur ou l'interprète. Il est par exemple très courant en jazz qu'une composition soit issue d'une œuvre antérieure ayant subie quelques modifications. On ne peut supposer qu'un tel travail se fasse sans que le compositeur ait fait une analyse plus ou moins fine du morceau qu'il transforme. L'improvisation est également un bon exemple de cas où les structures internes du morceau doivent être connues du musicien, ces structures ne pouvant être explicitées que par un travail d'analyse (peut-être complètement procédural, et par conséquent inconscient pour le musicien expérimenté). Quoiqu'il en soit, un travail d'analyse se fait à l'écoute d'une œuvre musicale, travail plus ou moins complexe et analyse plus ou moins fine selon le niveau de connaissance de l'auditeur, ce travail consistant essentiellement à reconnaître des formes connues. L'étude de ce processus de perception relève de la psycho-acoustique, les techniques d'intelligence artificielle pouvant aider à formaliser des modèles perceptifs tels que celui de [Widmer 92] dont il est question ci-dessous.

L'auditeur très expérimenté, lorsqu'il entend une œuvre musicale, sera capable d'analyser la structure harmonique de cette œuvre jusqu'à un niveau de détail très élevé ; il reconnaîtra immédiatement et pourra nommer les structures courtes, d'une durée de l'ordre de la note, comme les broderies, les appoggiatures, ainsi que les structures longues, couvrant une pièce entière, comme une forme sonate, une forme lied ou un rondeau. Si l'auditeur moyen n'est pas capable d'une telle finesse d'analyse, il est cependant familier de la plupart des éléments *classiques* du langage musical, et une cadence, une modulation ou une variation éveille un sentiment de *déjà entendu*. En s'intéressant à des questions d'apprentissage automatique de connaissances musicales, [Widmer 92] propose un modèle de perception dans lequel sont intégrées des fonctionnalités d'analyse harmonique. L'analyse harmonique se trouve donc être décrite comme partie intégrante d'un processus de perception, permettant de déduire des notions plus abstraites, telles que contrastes, diversité, tensions et détentes, stabilité harmonique, etc.

3. Modèle temporel

3.1 Modèles linéaire et circulaire

Une grille de jazz se présente comme une succession de cases comportant l'indication du ou des accords à jouer. Il apparaît donc immédiatement que le morceau ainsi représenté possède un début et une fin, ce qui est effectivement le cas. Cependant, il est implicite également qu'à la dernière mesure de la grille s'enchaîne à nouveau la première. Le temps dans lequel se déroule l'*exécution* du morceau présente donc un aspect *circulaire*. De plus, la grille donne les *dates* auxquelles doivent être jouer les différents accords ; de ces dates on déduit les *intervalles* de temps pendant lesquels durent les accords ou les suites d'accords.

On se trouve alors en présence de *deux* modèles temporels distincts : le premier, celui de la grille, est linéaire, fondé sur la notion d'événements discrets et ordonnés suivant une droite entière, et comportant une simple indication de répétition ; le second, celui de l'exécution du morceau, est circulaire, fondé sur la notion d'intervalles de temps, et comprend l'indication du début du morceau. Il est évident que les deux modèles contiennent exactement les mêmes informations.

Il convient d'ajouter que le processus d'exécution du morceau peut être retracé sur un mode linéaire, dans ce que l'on pourrait appeler une représentation temporelle *physique* ; il suffit pour cela de répéter la grille autant de fois qu'elle est effectivement jouée, de la *dérouler*.

C	F	F	C 7
F	F	C	A min 7
D 7	G 7	C	G7

figure 2 : exemple de grille de jazz (blues)

C'est le modèle d'exécution, c.-à-d. le modèle circulaire, qui est utilisé pour l'analyse. L'analyse est fondée sur la reconnaissance de formes correspondant à des séquences d'accords, dont l'un des éléments essentiels est l'*intervalle de temps* qu'elles occupent dans la grille. Il est primordial pendant l'analyse de pouvoir comparer d'une manière ou d'une autre ces intervalles. D'autre part, un modèle circulaire permet d'une part de reconnaître de la même façon les formes situées « au milieu » de la grille que celles se trouvant « à cheval » sur le début de la grille. Ainsi, dans l'exemple illustré par la figure 2, les résolutions C7-F (4ème et 5ème mesures) et G7-C (12ème et 1ère mesures) sont-elles reconnues par la même règle.

3.2 Modèle temporel circulaire

Il est nécessaire pour l'analyse de pouvoir *comparer* les différents groupes d'accords sur le plan *temporel* ; il faut par conséquent expliciter les relations pourvant exister entre deux intervalles de temps, dans une représentation circulaire du temps. On cherche ainsi à savoir si une séquence précède exactement une autre séquence, si elles se superposent ou sont disjointes, etc.

3.2.1 Construction et choix du modèle

Dans une représentation linéaire, James F. Allen [Allen 84] dénombre et nomme treize relations distinctes possibles entre deux intervalles de temps (illustration en annexe, figure 10 page 14).

Passer de cette représentation linéaire à une représentation circulaire revient à faire boucler la fin du second intervalle sur le début du premier. On s'aperçoit alors qu'à chacune des treize relations dans un temps linéaire correspondent deux relations dans un temps circulaire, selon que le début du premier intervalle coïncide ou non avec la fin du second[1] (figure 3). On obtient par conséquent vingt-six relations distinctes. Il faut noter que l'on ne considère pas ici les intervalles durant plus d'une période ; le temps ne serait alors plus circulaire, mais peut-être représentable par une sorte de spirale.

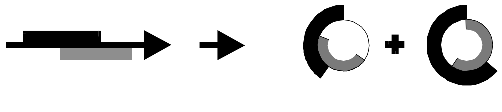


figure 3 : temps linéaire et circulaire

[1] . On verra plus loin qu'à une relation binaire dans une représentation circulaire peuvent correspondre une ou deux relations binaires dans une représentation linéaire.

3.2.1.1 Les différents modèles possibles

3.2.1.1.1 Repère avec ou sans origine

Dans l'une comme dans l'autre des représentations, on peut ajouter une origine au repère. Le nombre de relations binaires entre intervalles est alors considérablement augmenté, l'origine pouvant se situer à n'importe quel *moment particulier* de la relation, les moments particuliers étant les débuts et fins d'intervalles, ainsi que les sous-intervalles compris entre une fin (ou un début) et la fin (ou le début) suivant. Sur l'exemple de la figure 4 il y a sept moments particuliers auxquels peut se trouver l'origine.

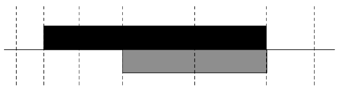


figure 4 : position de l'origine

Dans le cas linéaire, on obtient alors 101 relations, et dans le cas circulaire 150[1].

3.2.1.1.2 Intervalles couvrant une période

On peut être amené à distinguer dans certains cas les intervalles de temps couvrant une période entière, car dans ce cas la notion de début et de fin de l'intervalle n'a plus nécessairement de sens, tout au moins eu égard aux relations temporelles existant avec un autre intervalle de temps. En effet, un intervalle de temps inférieur à une période est nécessairement inclus dans un intervalle de temps couvrant toute une période. Certaines des relations binaires obtenues précédemment sont alors identiques. On trouvera en annexe le détail des relations étant ainsi regroupées (« Intervalles couvrant une période » page 20), ainsi qu'une description complète de la manière dont été énumérées et dénombrées les relations binaires entre intervalles de temps (voir « Enumération des relations binaires » page 19.

3.2.1.2 Choix du modèle

En pratique, le modèle retenu est le modèle d 'exécution[2] sans origine et sans regroupement (26 relations circulaires binaires distinctes, illustré en annexe page 17 par la figure 13). Ce choix a été dicté par un soucis d'équilibre entre simplicité et expressivité. Le modèle le plus simple, c.-à-d. le modèle comportant le moins de relations binaires est le modèle sans origine et avec regroupement ; le modèle le plus expressif, soit le modèle comportant le plus d'information, est le modèle avec origine et sans regroupement (annexes, figure 11 page 15). Ce dernier comporte 150 relations binaires distinctes, ce qui est trop important sur le plan opératoire. Le modèle le plus simple (19 relations) ne permet pas de distinguer les relations faisant intervenir une ou plusieurs formes couvrant une période entière. Ce dernier point ne semble pas, à l'heure actuelle, présenter un handicap significatif, mais le gain obtenu (7 relations) ne paraît pas justifier une perte d'information pouvant se réveler préjudiciable pour des développements utltérieurs.

Dans un soucis de clarté, les 26 relations binaires ont été nommées d'une manière systèmatique, en reprenant et combinant les noms utilisés par Allen. La liste complète de ces noms se trouve en annexe (voir « Dénomination » page 20).

La perte de l'information d'origine peut être facilement pallié dans l'écriture même des règles. Cette information est en effet toujours disponible : les intervalles sont caractérisés par une date de début, une durée, ainsi que la durée totale d'une période ; l'origine étant réprésentée ici par la date 1 (1^er^ temps de la 1^ère^ mesure), il est excessivement simple de tester si un intervalle contient ou ne contient pas l'origine.

Le regroupement des intervalles couvrant une période entière peut être obtenu pour les relations binaires en « hiérarchisant » de manière adéquate les relations binaires[3].

On constate donc que, même si l'un des modèles a été privilégié dans la réalisation informatique, on peut d'une certaine façon « naviguer » entre les différentes représentations, avec ou sans origine, avec ou sans regroupement. Il paraît donc justifié de dire que le modèle sous-jacent à la représentation temporelle des

---

[1] . Voir « Dénombrement » page 19, en annexe.
[2] . Voir « Modèles linéaire et circulaire » page 4.
[3] . Voir « Une hiérarchie de relations » page 7.

différentes suites d'accords est le modèle le plus complet, soit le modèle avec origine et sans regroupements, même si la réalisation est fondée sur un modèle plus simple (sans origine et sans regroupements).

3.2.2 Relations binaires entre intervalles circulaires

3.2.2.1.1 Une hiérarchie de relations

Les relations binaires obtenues sont spécifiques. On a besoin pour l'analyse de relations plus générales, telles que *pendant* ou *pendant ou après*, etc. La méthode de construction des relations circulaires à partir des relations d'Allen suggérait un mécanisme d'héritage. On constate que certaines des relations circulaires « héritent » directement des relations d'Allen. En réalité, mises à part *before* et *after*, à chaque relation d'Allen correspond exactement une relation circulaire.

Quant aux relations restantes, elles héritent des relations primitives (les relations héritant directement des relations d'Allen) suivant plusieurs principes :

- **Déroulement** : selon le principe vu plus haut[1], une relation circulaire donnant deux relations d'Allen héritent des deux relations primitives correspondant à ces deux relations. Ainsi, la relation *meets-is overlapped* héritent des deux relations *meets* et *is overlapped*.
- **Regroupement** : si l'on considère qu'un intervalle couvrant une période entière n'a ni début ni fin, certaines relations se confondent[2]. Ainsi les relations *starts-period*, *finishes-period*, *is overlapped-period* et *contains-period* deviennent-elles identiques, et correspondent à la notion d'inclusion stricte. Elles correspondent donc à la relation *contains*, et en hérite par conséquent.

La hiérarchie de relations ainsi obtenue est représenté en annexe (figure 15 page 18).

3.2.2.1.2 Détermination des relations

Etant donnés deux intervalles circulaires quelconques, on veut connaître la relation binaire circulaire qui les relie. La méthode consiste à utiliser à rebours le procédé utilisé pour calculer les différentes relations possibles entre deux intervalles de temps[3]. Cela revient à parcourir l'arbre ayant servi à énumérer les différentes relations (figure 16), en y plaçant un par un les instants de début et de fin de chaque intervalle. Une fois parvenu à une feuille, la relation est entièrement déterminée.

De manière pratique, on utilise un automate à états finis représentant l'arbre pour reconnaître les relations. Les états représentent les nœuds et les feuilles de l'arbre, et les transitions se font en examinant la position relative d'un nouvel instant (instant de début ou de fin de l'un des deux intervalles) par rapport aux instants déjà examinés. Le point d'entrée est l'instant de début du premier intervalle. Selon que l'instant de fin de ce même intervalle est égal à l'instant de fin ou non, l'état suivant prend une nouvelle valeur, et ainsi de suite avec les instants de début et de fin du second intervalle.

3.2.3 Algèbre

La méthode de détermination des relations binaires entre intervalles circulaires que nous venons de voir ne concerne que des couples d'intervalles. Il est cependant fréquent que l'analyse nécessite de comparer les intervalles relatifs à un plus grand nombre de formes. Pour se faire, on définit une algèbre sur les différents ensembles d'intervalles circulaires.

On note I_p l'ensemble des intervalles circulaires définis pour une période p. p correspond alors dans une grille d'accords au nombre de temps dans la grille, et l'on cherche à définir les lois de compositions internes *union*, *intersection* et *complément* sur I_p. On verra que cela nous amène à redéfinir la notion d'intervalle circulaire.

3.2.3.1 Différentes classes d'intervalles

Afin de pouvoir définir les lois de composition interne sur I_p, on est amené à généraliser la notion d'intervalle circulaire, et à définir les notions d'*intervalle vide*, et d'*intervalle non connexe*. L'intervalle vide se définit

[1] . Voir « Construction et choix du modèle » page 5.
[2] . Voir « Intervalles couvrant une période » page 6.
[3] . Voir « Algorithme » page 19.

comme l'intersection de deux intervalles circulaires liés par la relation *meets*, *is met,* ou *before-after*. Un intervalle non connexe est un ensemble d'intervalles dont l'intersection est l'intervalle vide. L'union de deux intervalles circulaires liés par la relation *before-after* ou l'intersection de deux intervalles circulaires liés par la relation *overlaps-is overlapped* donne un intervalle non connexe.

3.2.3.2 Lois de composition interne

3.2.3.2.1 Intervalles connexes

Le complément d'un intervalle connexe *i* est définit comme l'intervalle de même période que *i*, dont l'instant de début est égal à l'instant de fin de *i*, et l'instant de fin égal à l'instant de début de *i*.

Une fois connue la relation unissant deux intervalles, il est facile d'en déterminer l'union et l'intersection. On définit donc pour chaque relation binaire circulaire les fonctions représentant les lois d'union et d'intersection. L'exemple de la figure 6 illustre l'union de deux intervalles circulaires liés par la relation *overlaps*, et de deux intervalles liés par la relation *overlaps-is overlapped*. Connaissant les dates de début et de fin de chacun des intervalles, la déduction de leur union ou de leur intersection est immédiate.

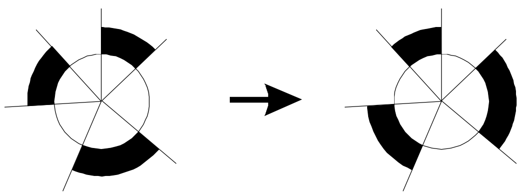
Tout cela ne concerne que les intervalles circulaires connexes. Il reste donc à définir l'union et l'intersection avec des intervalles non connexes, avec l'intervalle vide, et l'union et l'intersection entre des intervalles non connexes.

L'intervalle vide ne pose pas de problème. On a toujours, quel que soit l'intervalle *i*, et si on note Ø l'intervalle vide, $Ø \cup i = i$ et $Ø \cap i = Ø$.

Il reste maintenant à définir union et intersection des intervalles non connexes. On définit tout d'abord l'union et l'intersection d'un intervalle connexe et d'un intervalle non connexe.

3.2.3.2.2 Intervalles non connexes

Le complément d'un intervalle non connexe *i* se définit de manière aussi intuitive que le complément d'un intervalle connexe comme l'ensemble des intervalles connexes dont l'instant de début est égal à l'instant de fin d'un intervalle connexe *c* de *i*, et l'instant de fin est égal à l'instant de début de l'intervalle connexe de *i* suivant *c*.



<u>*figure 5 : complément d'un intervalle non connexe*</u>

L'intersection d'un intervalle non connexe *i* et d'un intervalle connexe *c* est définie comme l'ensemble des intersections des intervalles connexes de *i* avec *c*. Si cet ensemble est vide, l'intersection est l'intervalle vide ; si cet ensemble est réduit à un élément, l'intersection est un intervalle connexe. Si cet ensemble a plusieurs éléments, l'intersection est un intervalle non connexe.

L'union d'un intervalle non connexe *i* et d'un intervalle connexe *c* est définie comme l'ensemble des intervalles connexes de *i* ayant une intersection nulle avec *c* (ces intervalles sont laissés tels quels), ensemble auquel est ajouté l'intervalle connexe constitué par l'union globale de tous les intervalles connexes de *i* ayant une intersection non nulle avec *c* et de *c*. Cette union globale se définit de manière récursive. Soit *a* un intervalle connexe et *B* un intervalle non connexe dont tous les éléments $b_{i\,[i=1..N]}$ ont une intersection non nulle avec *a*. Soit la formule récursive suivante :

$$\boxed{\begin{aligned} u_0 &= a \\ u_{n+1} &= u_n \cup B_n \end{aligned}}$$

L'union globale de *a* et *B* est u_N, avec *N* le nombre d'intervalles connexes de *B*.

L'union et l'intersection de deux intervalles non connexes se définissent de manière récurrente, d'une façon analogue à la définition de l'union globale vue plus haut. Soient *A* et *B* deux intervalles non connexes, et *N* le nombre d'intervalles connexes contenus dans *B*. Soient les formules de récurrences suivantes :

$$\boxed{\begin{array}{l} v_0 = A \\ v_{n+1} = v_n \cup B_n \end{array}} \qquad \boxed{\begin{array}{l} w_0 = A \\ w_{n+1} = w_n \cap B_n \end{array}}$$

L'union de *A* et B est v_N et leur intersection est w_N.

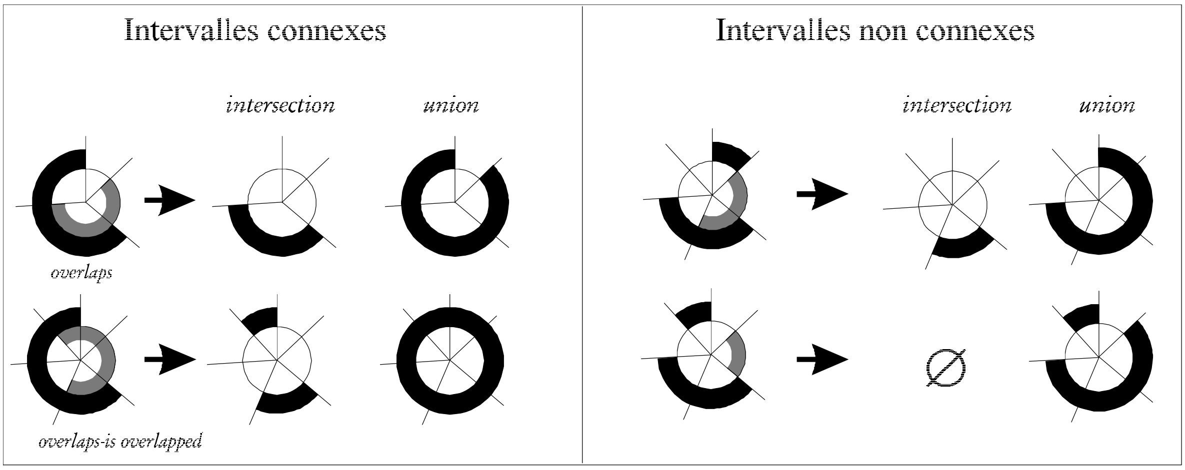


figure 6 : union et intersection d'intervalles connexes et non connexes

Ainsi sont définies sur I_p, ensemble des intervalles circulaires de période *p*, ces intervalles pouvant être l'intervalle vide, des intervalles connexes ou non connexes, les lois de composition internes de complémentation, d'union et d'intersection.

Des détails concernant l'implémentation de ce modèle temporel, des relations binaires entre intervalles et des lois de composition pourront être trouvés en annexe (« Implémentation du modèle temporel » page 22).

3.2.3.3 Un treillis booléen

On remarque ici, sans le démontrer, que l'ensemble I_p muni de la relation d'inclusion $\subseteq$ constitue en treillis booléen. La relation d'inclusion est définie par rapport aux relations circulaires déterminées plus haut[1]. On dit que $x \subseteq y$ si *x* et *y* sont mis en relation par l'une des relations : *equals-period* (1), *starts-period* (2), *finishes-period* (3), *overlaps-is overlapped-period* (4), *is overlapped-period* (5), *during-period* (6), *equal*s (8), *starts* (9), *finishes* (17)ou *during* (19)[2]. Cette relation correspond à la notion intuitive de « pendant ».

L'union et l'intersection dans I_p sont idempotentes, commutatives et associatives, et les lois d'absorption $x \cap (x \cup y)=x=x \cup (x \cap y)$ sont vérifiées ; de plus, elles sont distributives l'une par rapport à l'autre. Par théorème ([Ponnasse & Carrega 79]), on en conclue qu'on peut définir sur I_p une unique relation d'ordre $\subseteq$ telle que $(I_p, \subseteq)$ soit un treillis distributif. En outre, I_p possède un plus petit élément, l'intervalle vide, et un plus grand élément, l'intervalle de durée égale à *p*. $(I_p, \subseteq)$ est donc un treillis booléen.

4. Perspectives

Après avoir décrit quelques unes des modifications apportées au système outre les aspects temporels, nous tâcherons d'en circonscrire les limites.

---

[1] . Voir « Relations binaires entre intervalles circulaires » page 7.

[2] . Les numéros entre parenthèses renvoient à la figure 13 page 17.

4.1 Modifications apportées au système

4.1.1 Intervention double d'une même forme

Il peut arriver qu'une même forme intervienne deux fois dans la même règle harmonique. La règle de l'emprunt modal en est un bon exemple. Cette règle stipule qu'un accord situé entre deux formes analysables dans une tonalité majeure et qui est lui-même analysable dans la tonalité mineure relative peut être analysé dans cette tonalité majeure (l'exemple de la figure 7 illustre un accord de la bémol septième majeure entre deux formes en do majeur).

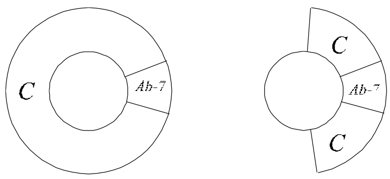


figure 7 : emprunt modal

Il faut alors écrire une règle pour chacun de ces cas.

4.1.2 Une grammaire pour les formes connues

La base de règle devant reconnaître les formes de base, telles que les deux-cinq, les résolutions, etc., a été complètement réécrite, en particulier dans le but de rendre explicite les aspects récursifs des deux-cinq (voir figure 1). Deux nouveaux types de formes ont été introduits : les cascades, correspondant aux classiques chutes de quintes, et les suites de deux-cinq. Chacune de ces formes décrit des passages modulants, ou les tonalités s'enchaînent rapidement, et qui sont en général transitoires. On trouve un exemple de ces formes dans *Blues for Alice*, dont l'analyse se trouve en annexe[1]. Cette base de règle s'appuie sur une grammaire, comparable à celle de [Steedman 84], mis à part qu'elle est plus limitée dans son objectif, qui est approximativement de reconnaître les différents types de résolution, et plus développée dans son détail.

Les règles de la grammaire de reconnaissance des résolutions, et par conséquents des formes préparatoires aux résolution telles que les deux-cinq, les chutes de quintes (cascades) et les successions de deux-cinq (two-fives fall), dans leur forme naturelle et leur forme substituée au triton (quinte diminuée) sont présentées ci-dessous.

x désigne la tonalité. 2^{nde}(x), 5^{te}(x), 5^{te} dim(x) désignent respectivement les tonalités situées à la seconde majeure, quinte et quinte diminuée supérieure de x. I(x), II(x) et V(x) désignent respectivement les accords construits sur les premier, deuxième et cinquième degrés de la tonalité x.

```
Résolvante(x)   :=  Dominante(x) I(x)
                    Dominante(5te dim(x) I(x)

Dominante(x)    :=  V(x)
                    TwoFive(x)
                    Cascade(x)
                    TwoFivesFall(x)

TwoFive(x)      :=  II(x) V(x)
                    II(x) V(5(x))
                    TwoFive(5te(x)) V(x)
                    TwoFive(5te dim(5te(x))) V(x)

Cascade(x)      :=  Dominante(5te(x)) Dominante(x)
                :=  Dominante(5te dim(5te(x))) Dominante(x)

TwoFivesFall(x) :=  TwoFive(2nde(x)) TwoFive(x)
                    TwoFivesFall(2nde(x)) TwoFive(x)
                    TwoFive(5te dim(2nde(x)) TwoFive(x)
                    TwoFiveFall(5te dim(2nde(x)) TwoFive(x)
```

[1] . Voir « Analyse de Blues for Alice (Charlie Parker) » page 24.

On remarquera que cette grammaire est dépendante du contexte, comme l'était celle de [Steedman 84]. Elle n'est donc pas implémentable directement, en tant que grammaire. Cependant, les règles de la base de règles suivent aussi fidèlement que possible les règles de la grammaire. On trouvera ci-dessous la règle de la base de règles qui reprend la première règle de la grammaire, dans le cas d'une résolution dans le mode majeur. Les éléments qui ne sont pas utiles à la compréhension ont étés supprimés de la règle.

```
resolvanteMajeur
    | Dominant c1. ChordInASequence c2 . ChordSequence cs|

      c2 isAfter: c1.
      c2 isMajor.
      c2 onStrongBeat.
      c2 root pitchEqual: c1 tonality root.

actions
      |x s|
      x := Resolvante new.
      x period: cs size.
      x beginBeat: c1 beginBeat; endBeat: c2 endBeat.
      x tonality: (c2 root majorScale); subShapes: (Array with: c1  with: c2 ).
```

4.2 Les limites du système

4.2.1 Les informations qui manquent

Plusieurs informations fort utiles au musicien lorsqu'il travaille, qu'il joue où qu'il improvise ne sont pas pris en compte par le système. Ainsi, le système n'inégrait-il pas la notion de temps forts et faibles, ou parties de mesures fortes ou faibles. Cette information a été ajoutée ; elle est en effet nécessaire, dans la mesure ou la même séquence d'accords (un deux-cinq par exemple) n'a pas le même sens selon que le premier accord est situé sur un temps fort ou un temps faible.

L'*armure*, c.-à-d. le nombre de bémols où de dièses qui se trouvent à la clef, donne une information fiable quant à la tonalité d'un morceau, au mode près : un morceau dont l'armure comporte trois dièses sera soit en la majeur, soit dans le ton relatif mineur, fa mineur. Il n'y a plus alors qu'à regarder comment commence et finit le morceau pour en déterminer la tonalité globale.

Ensuite les *reprises* et les *codas*, fort usitées dans les grilles de jazz. Un reprise indique que le musicien doit recommencer une phrase à partir d'un endroit déterminé. La coda est la ou les mesures qui permettent d'enchaîner la deuxième exécution de ce passage sur la suite du morceau. Ces informations permettent de déterminer au premier coup d'oeil certaines structures, comme la forme AABA très courante en jazz. Notons ici que prendre en compte ces informations entraînerait des modifications importantes du modèle temporel qui, dans ce cas, s'avèrerait insuffisant.

Ces éléments pourraient relativement facilement être représentés dans le système, et permettrait un gain d'efficacité pour les règles de reconnaissance des formes les plus abstraites. D'autre part, ces lacunes ne remettent pas en cause la validité des règles de plus bas niveau, c'està-dire les règles de reconnaissance des résolutions, des deux-cinq, etc.

4.2.2 Modulations et transitions

L'exemple qui suit (figure 8), tiré de [Couleau 87], illustre la difficulté de définir précisément les tonalités des passages modulants. Du strict point de vue harmonique, la tonalité de fa majeur apparaît avec l'introduction du si bémol, c.-à-d. avec l'accord Gm7. Cependant, on s'aperçoit en regardant la mélodie que les accords Gm7 et C7 de la seconde moitié de la deuxième mesure ne sont qu'un préparation à la tonalité de fa majeur, qui ne commence qu'avec la nouvelle partie de thème, soit à la troisième mesure. Le système d'analyse ne prend pas en compte de telles zones de transition. De plus, il a tendance à faire « remonter » les tonalités le plus loin possible, c.-à-d. qu'il considère qu'une forme est dans une tonalité donnée des que les accords la constituant sont, hors contexte, analysables dans cette tonalité. Ainsi, dans l'exemple cité, le système ferait commencer la tonalité de fa majeur au début de la troisième mesure. Cela est particulièrement frappant dans certains Blues simples où l'analyse décèle une forme dans la tonalité de la quarte de la tonalité principale, ce qui est souhaitable, mais qui fait commencer cette forme sur le premier temps de la grille.

figure 8 : modulation

4.2.3 Enchaînements types

Plusieurs enchaînements types, et en particulier les anatoles, manquent au système, ainsi que différentes formes de substitutions (remplacement d'un accord par un autre). Même si ces lacunes ne remettent pas fondamentalement en cause la validité des analyses effectuées, ces dernières se trouvent ne pas contenir un certains nombre d'informations importantes. Ainsi, si l'on se réfère à l'improvisation comme utilisation type d'une grille de jazz, on verra le musicien adapter une formule connue aussitôt qu'il a reconnu une anatole. Le problème réside dans la grande variété de ces formes (pour plus de détails, voir [Couleau 87]).

A un niveau un peu plus abstrait, signalons le problème du Blues : c'est une forme parfaitement connue, identifiable immédiatement pour le musicien, mais qu'il est difficile à décrire sous forme de règle. Le problème réside principalement dans les aspects *modaux*, ainsi que dans ce qu'il est convenu d'appeler la *direction harmonique* ([Berendt 83], [Couleau 87], [Joanger-Weller 86]). En ce qui concerne les aspects modaux, il faut noter une difficulté liée aux accords de septième de dominante, qui dépasse le simple cadre du Blues. Dans un cadre tonal strict, l'accord de septième de dominante est l'accord de quatre sons construits sur la dominante (c.-à-d le cinquième degré) de la gamme, soit sur Sol dans la tonalité de Do, ce qui donne accord Sol-Si-Ré-Fa. Il se trouve que cet accord est le même dans les modes majeur et mineur ; de plus, c'est un accord *attractif*, qui appelle souvent la résolution sur le degré le premier degré de la gamme. Dans beaucoup de Blues (K.C. Blues, Blue Bird de Charlie Parker, par exemple), un accord ayant la même structure que l'accord de septième de dominante (tierce majeure, quinte juste, septième mineure) est construit sur le premier degré et le quatrième de la gamme. Des notes étrangères à la gamme sont ainsi introduites (Mi bémol et Si bémol en Do majeur). Ce sont les fameuses *blue notes*, résultant du mélange des modes pentatoniques et de la musique tonale.

Une solution consiste à écrire une règle spécifique pour chacun de ces cas. Un autre à intégrer dans le système de base qu'un accord de septième de dominante est analysable dans plusieurs tonalités. C'est la première solution qui a pour l'instant été retenue. Une raison en est que ce n'est que dans le contexte de la musique de jazz que se trouvent ses exceptions, et que le système MusES se veut indépendant du style de musique. Une autre raison est que ces exceptions ne se rencontrent en général que dans certaines configurations de temps forts et faibles, ce qui peut être exprimés dans les règles d'analyse, et permet ainsi de diminuer la complexité du problème.

4.3 Conclusion

Tel qu'il est, le système d'analyse de grilles de jazz fonctionne avec une certaine efficacité. Des modifications pouvant procurer une nette amélioration des résultats pourraient être apportées sans grand effort. Dans un grand nombre de cas, les analyses fournies sont satisfaisantes, et se rapprochent suffisamment de l'analyse que pourrait faire un musicien. Cependant, un certains nombres de critiques de fond peuvent être formulées, en particulier concernant des informations importantes qui ne sont pas pris en compte. D'autre part, le mécanisme d'agrégation ne reproduit pas toujours assez fidèlement les schémas classiques d'analyse harmonique. Il conviendrait sans doute d'affiner la manière dont le système agglomère les formes pour en créer de nouvelles.

Cependant, là n'est pas le plus important. Les questions que mettent en évidence ce système sont celles de l'intégration des objets et des règles, des mécanismes du raisonnement que l'on peut représenter par des règles, et surtout de la modélisation de processus de raisonnement en général, et d'analyse en particulier. On a vu que le système intégrait différents niveaux d'abstraction, ainsi qu'un mécanisme efficace d'agrégation et d'oubli. Ce principe général, sans doute transposable à d'autres domaines que l'analyse harmonique tonale, doit pouvoir servir de base à d'autres systèmes automatiques.

5. Annexes

5.1 Grammaire générative de Lerdahl et Jackendoff

La figure suivante illustre la grammaire générative proposée par Lerdahl et Jackendoff en 1983. Les règles ou bases de règles sont encadrées, et les données d'entrée ou de sortie sont entourées par une ellipse. Cette figure est tirée de [Lerdahl & Jackendoff 83].

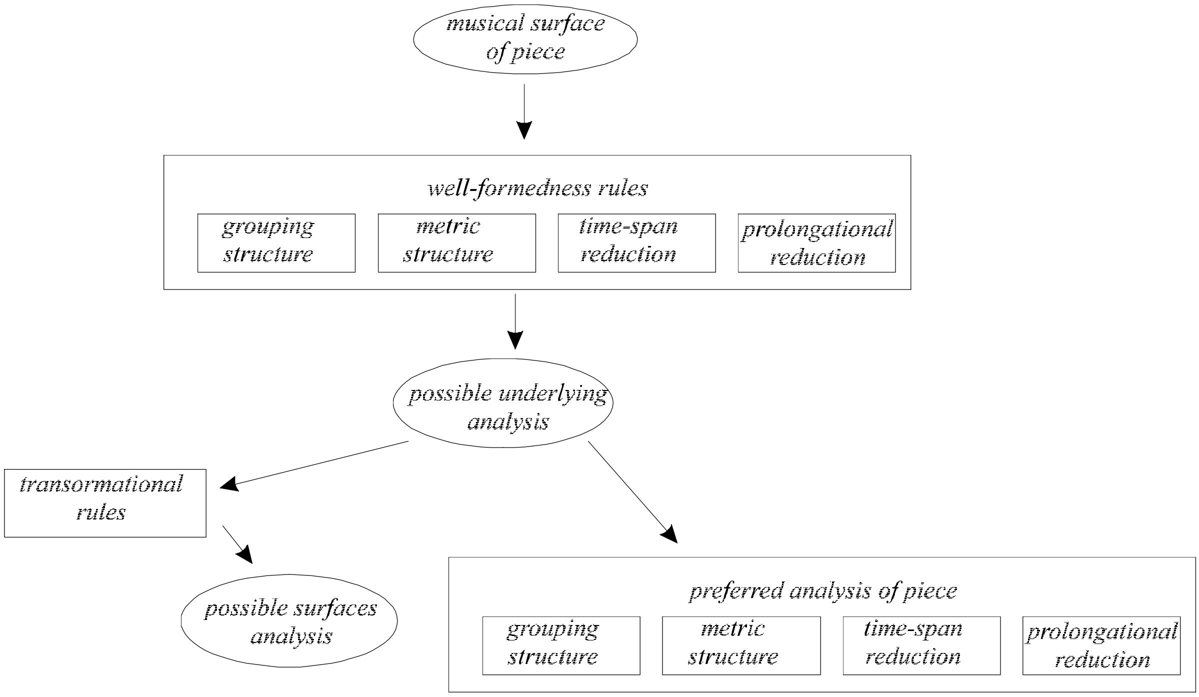
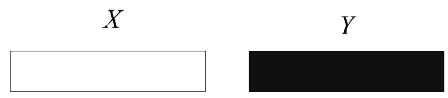
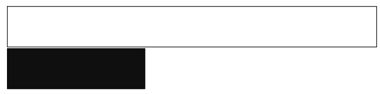
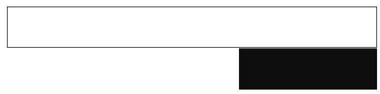


figure 9 : structure de la grammaire musicale de Lerdahl & Jackendoff

5.2 Les treize relations d'Allen

Relation	Relation réciproque	Exemple
X before Y	*Y after X*	
X equal Y	*Y equal X*	
X meets Y	*Y is met by X*	
X overlaps Y	*Y is overlapped by*	
X during Y	*Y contains X*	
X starts Y	*Y is started by X*	
X finishes Y	*Y is finished by X*	

<u>*figure 10 : les treize relations d'Allen*</u>

5.3 Relations binaires entre intervalles circulaires

figure 11 : modèle temporel avec origine et sans regroupements

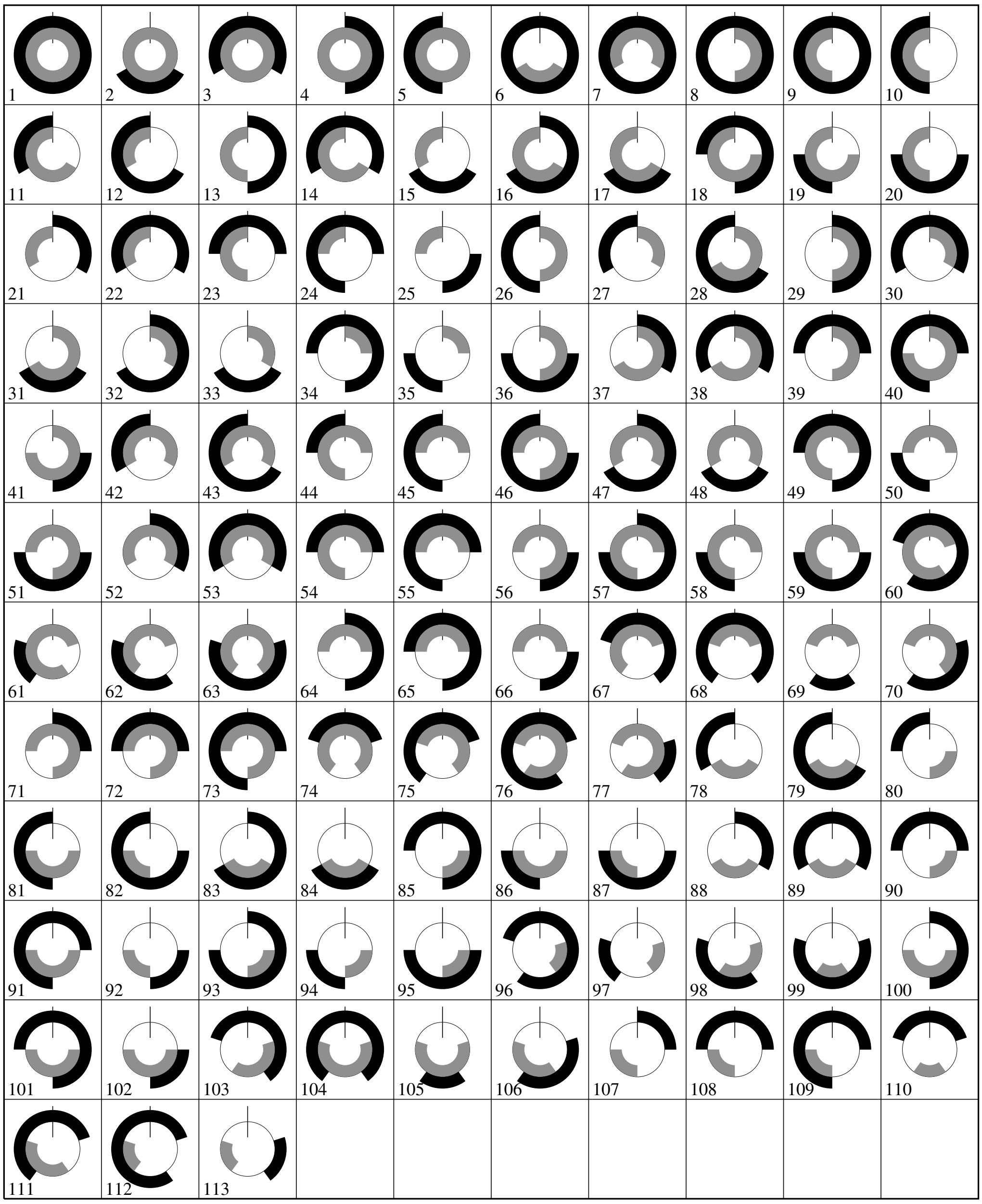


figure 12 : modèle temporel avec origine et avec regroupements

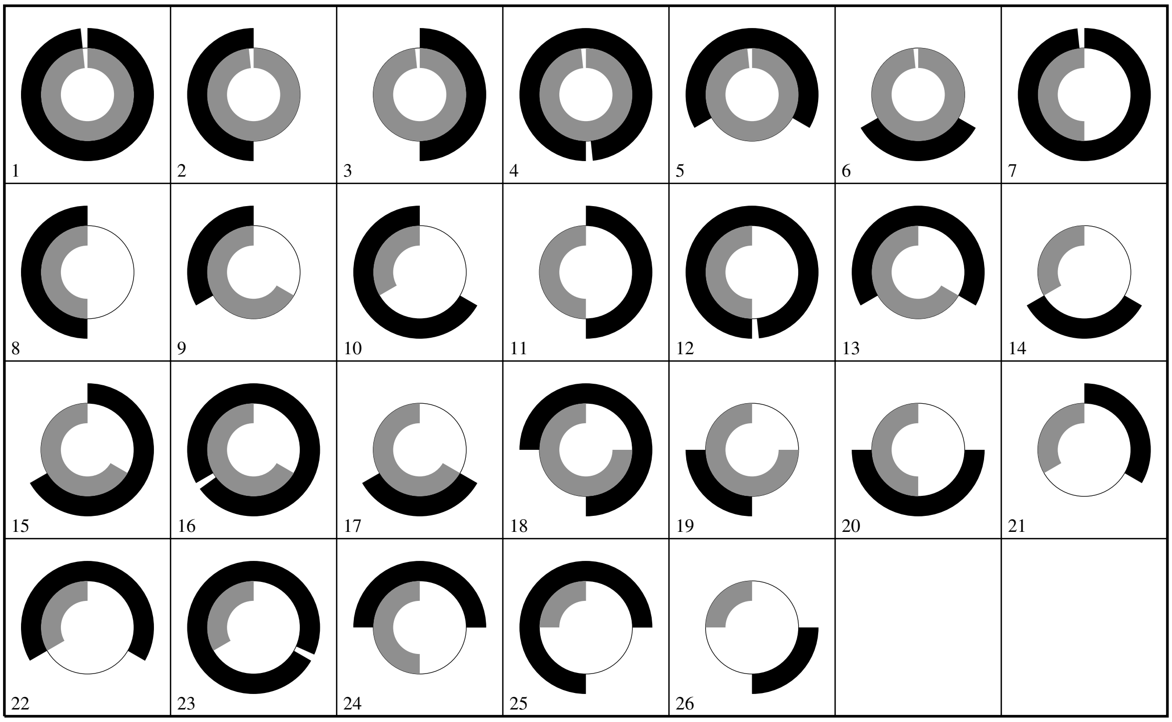


figure 13 : modèle temporel sans origine et sans regroupement

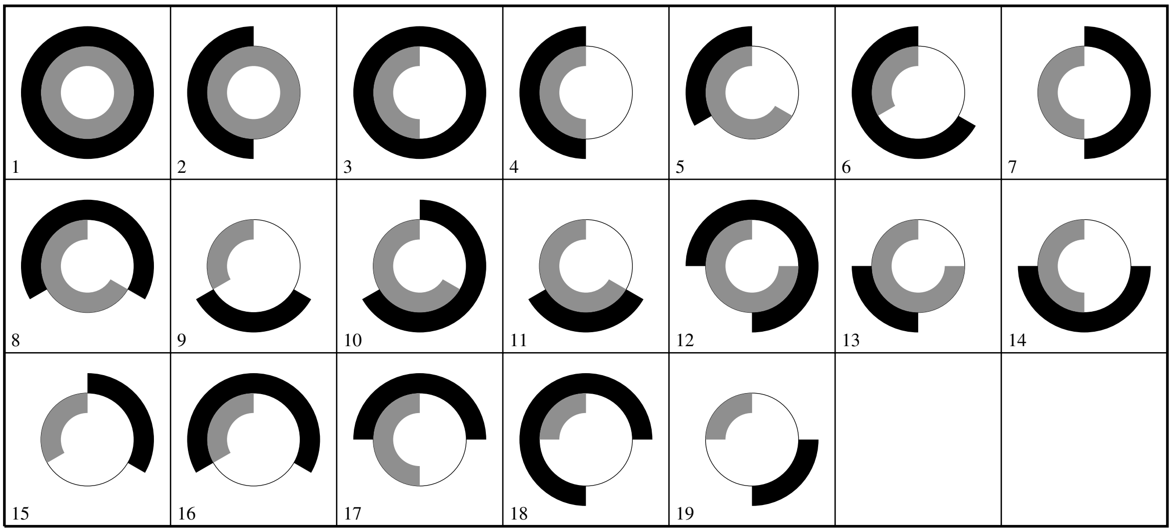
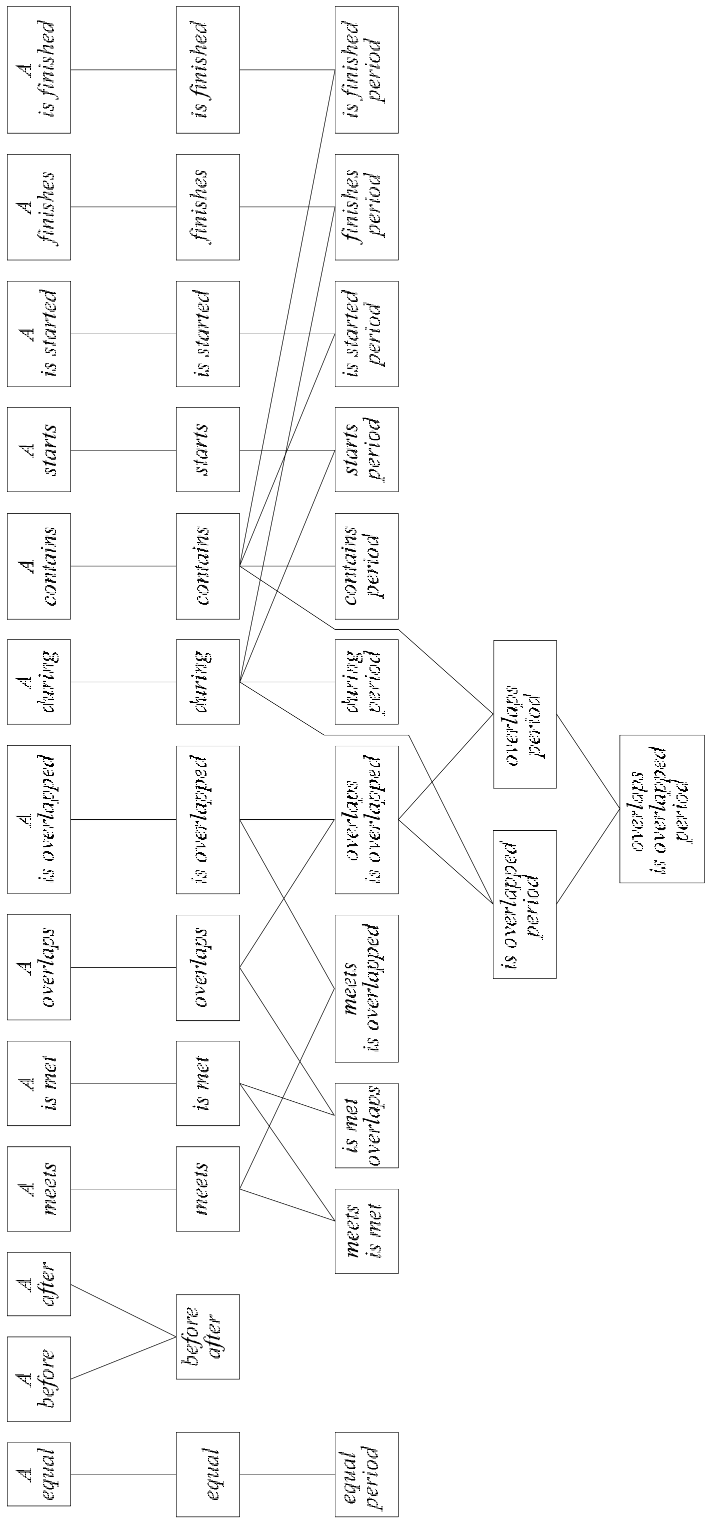


figure 14 : modèle temporel sans origine et avec regroupements

5.4 Un réseau de relations



<u>*figure 15 : hiérarchie de relations*</u>

5.5 Enumération des relations binaires

5.5.1 Algorithme d'énumération

On considère les instants de début et de fin de chaque intervalle, notés b_i et e_i pour le $i^{\text{ème}}$ intervalle, et éventuellement *O* l'origine. Soit *I* la liste constituée de ces instants ; on applique une procédure récursive sur une liste initialement vide dont chaque élément est constitué d'événements se produisant simultanément, l'ordre des éléments étant l'ordre d'occurrence de ces événements à une rotation circulaire près. La procédure consiste à insérer à *chaque* élément et *après* chaque élément d'une copie de la liste courante un nouvel élément de *I*.

La figure 16 montre le déroulement de l'algorithme avec *I* = {b1, e1, b2, e2}.

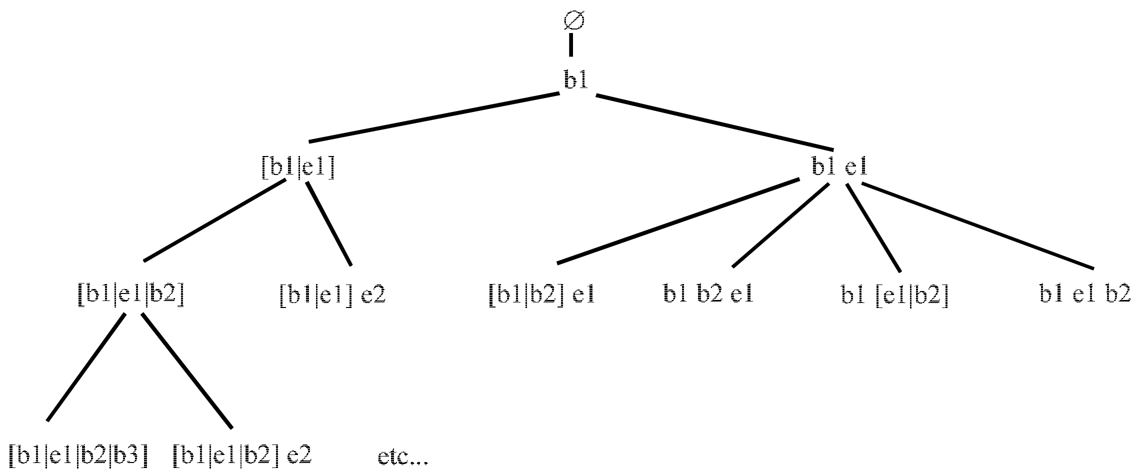


figure 16 : algorithme d'énumération

[*x*|*y*] signifie que *x* et *y* ont lieu simultanément ; *x y* signifie que *x* précède *y*.

Lorsque b_i et e_i ont lieu simultanément, cela signifie que l'intervalle *i* couvre une période complète.

La circularité est ainsi prise en compte en n'engendrant pas toutes les combinaisons possibles des divers éléments de *I*.

On peut ainsi engendrer toutes les relations binaires possibles de *n* intervalles de temps dans une représentation circulaire, avec ou sans origine.

5.5.2 Dénombrement

A partir d'un élément constitué de *n* événements (par exemple $\{[O|b_1]\ e_1\}$, *n*=2), la procédure détaillée plus haut engendre *n* éléments constitués de *n* événements ($\{[O|b_1|b_2]\ e_1\}$ et $\{[O|b_1]\ [e_1|b_2]\}$ et *n* éléments constitués de *n*+1 événements ($\{[O|b_1]\ b_2\ e_1\}$ et $\{[O|b_1]\ [e_1]\ b_2\}$.

La progression de l'algorithme peut être décrite de la manière suivante, où l'élément *j* de la ligne *i* représente le nombre d'éléments constitué de *j* événements à l'étape *i* de l'algorithme.

	1	*2*	*3*	*4*	*5*	*total*
1	1					1
2	1	1				2
3	1	3	2			6
4	1	7	12	6		26
5	1	15	50	60	24	150

figure 17 : dénombrement

Soit $N_{m,n}$ l'élément situé ligne *m* et colonne *n*.

$N_{m,n}$ est donné par la relation de récurrence suivante :

$$N_{m,1} = 1$$
$$N_{m,m+1} = 0$$
$$N_{m,n} = n.N_{m-1,n} + (n-1).N_{m-1,n-1}$$

Le nombre de relations est donné par : $\sum_{i=1}^{P} N_{m,i}$ avec P égal à deux fois le nombre d'intervalles, plus un si on veut distinguer l'origine, c.-à-d. le nombre total d'éléments à prendre à compte. On obtient par cette relation les nombres donnés par le tableau 1.

Nombre d'intervalles	*Sans origine*	*Avec origine*
2	26	150
3	1082	9366
4	94586	1091670
5	14174522	204495126

tableau 1 : nombre de relations

5.5.3 Intervalles couvrant une période

Le tableau XXX donne les numéros des

Sans origine (figure XXX)	*Avec origine (figure XXX)*
{1,4} {5,7,12,23} {2,3,5,6}	{1,5,89,95,101,107} {2,90,91,92} {3,93,99,103} {4,94,100,104,105,106} {6,96,97,98,102,108} {7,13,19,25} {27,33,39,45} {47,55,63,71,79,87} {109,117,125,133,141,149}

tableau 2 : regroupement des relations

5.5.4 Dénomination

De même qu'en « refermant » le modèle temporel linéaire sur lui-même pour obtenir un modèle temporel circulaire[1] à chaque relation d'Allen correspondait deux relations *circulaires*, en « déroulant » le modèle temporel circulaire à chaque relation *circulaire* peut correspondre une ou deux relations d'Allen. La ou les relations sont déterminées en passant du modèle d'exécution au modèle physique[2].

[1] . Voir « Construction et choix du modèle » page 5.
[2] . Voir « Modèles linéaire et circulaire » page 4.

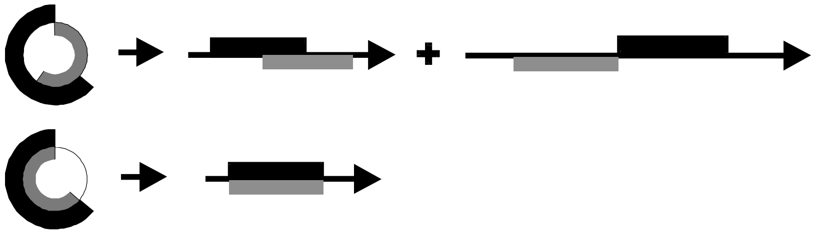


figure 18 : déroulement des relations circulaires

Pour nommer les relations binaires circulaires, on utilise les noms des relations d'Allen correspondant. Ainsi, le premier exemple de la figure 18 fait intervenir les relations *is met by* et *overlaps* ; on nommera cette relation *is met by-overlaps*. Quant aux relations qui correspondent à une seule relation d'Allen, le nom sera conservé (deuxième exemple de la figure 18 : *equal*).

Une ambiguïté se présente lorsqu'un des deux intervalles couvre une période entière. En effet, comme il a été vu plus haut[1], à une relation d'Allen correspondent deux relations circulaires, selon que l'union des deux intervalles couvre ou non la totalité de la période. Lorsqu'un des deux intervalles *contient* l'autre au sens large (ce qui est le cas dans les relations *equal*, *contains*, *starts* et *finishes*), que l'union des deux intervalles couvre la totalité de la période signifie que l'un des deux intervalles couvre la totalité de la période). Il convient donc d'en tenir compte dans la dénomination des intervalles. Cela se fera simplement en ajoutant le terme *period* au nom de la relation d'Allen.

On obtient ainsi les noms de vingt-six relations circulaires donnés par le tableau 3.

1	*equal-period*	10	*is started*	19	*during*
2	*starts-period*	11	*meets-is met*	20	*is overlapped*
3	*finishes-period*	12	*is finished-period*	21	*meets*
4	*overlaps-is overlapped-period*	13	*is met-overlaps*	22	*is finished*
5	*is overlapped-period*	14	*is met*	23	*contains-period*
6	*during-period*	15	*meets-is overlapped*	24	*overlaps*
7	*is started-period*	16	*overlaps-period*	25	*contains*
8	*equal*	17	*finishes*	26	*before-after*
9	*starts*	18	*overlaps-is overlapped*		

tableau 3 : noms des relations circulaires

Il est à noter que les noms ne sont pas ambigus, et suffisent pour retrouver la relation correspondante.

[1]. Voir « Construction et choix du modèle » page 5.

5.6 Implémentation du modèle temporel

5.6.1 Hiérarchie de classes

Les différentes classes Smalltalk représentant les intervalles de temps sont liées par la hiérarchie représentée par la figure 19.

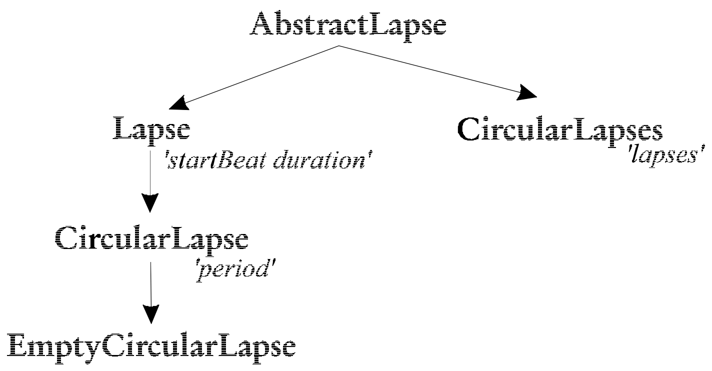


figure 19 : hiérarchie des classes d'intervalles de temps

La classe **Lapse** faisait déjà partie du système MusES. Les intervalles temporelles y sont définies par une date de début et une durée (variables d'instances *startBeat* et *duration*). Par soucis d'homogénéité, les intervalles de temps circulaires, représentés par la classe **CircularLapse**, qui hérite de la classe **Lapse**, seront également définis par les même attibuts, auxquels s'ajoutent la durée totale d'une période (variable d'instance *period*). Dans le système d'analyse néanmoins, les informations utiles sont principalement les dates de début et de fin d'un intervalle ; c'est d'ailleurs ces éléments qui ont été utilisés plus haut, que ce soit pour élaborer le modèle temporel, pour mettre au point l'algorithme de reconnaissance des relations temporelles entre intervalles circulaires ou pour définir les lois d'union, d'intersection ou de complémentation. Les primitives d'acces permettent de déduire facilement la date de fin d'après la date de début et de la durée.

La classe **CircularLapses** représente les intervalles non connexes, la variable d'instance *lapses* étant pour un objet la liste (**OrderedCollection**) des intervalles connexes le constituant.

La classe **EmptyCircularLapse** représente l'intervalle vide.

5.6.2 Relations temporelles

5.6.2.1 Description de la classe CircularLapseRelation

Afin de pouvoir déterminer la relation binaire pouvant exister entre deux intervalles de temps circulaires connexes, les relations binaires sont réifiées par l'intermédiaire de la classe **CircularLapseRelation**. Un solution élégante permettant de constituer la hiérarchie de relations vue plus haut[1] eût été d'établir une hiérarchie de classes. Malheureusement, Smalltalk ne permet pas l'héritage multiple de classes. La classe **CircularLapseRelation** est construite approximativement sur le modèle que la classe **PitchClass** du système MusES, qui représente les différentes notes possibles indépendamment de leur octave [Pachet 93]. La classe **CircularLapseRelation** possède trois variables d'instance : *parents*, *name*, et *number*, et vingt-sept variables de classe (*Contains*, *Starts*, *Finishes*, *Overlaps*, etc.), correspondant aux vingt-six relations binaires distinctes, plus une relation *undefined*, qui permet d'éviter un certain nombres d'erreurs. Les vingt-sept variables de classe sont des instances de la classe **CircularLapseRelation** et sont crées au lancement du système (méthode *initialize*). Pour une relation donnée, la variable d'instance *parents* est une collection des relations dont hérite cette relation ; la variable d'instance *number* permet de numéroter les relations (de 1 à 27), et ainsi de tester l'égalité de deux relations. La méthode permettant de savoir si une relation hérite d'une autre relation est une méthode récursive de parcours très simple :

[1] . Voir « Une hiérarchie de relations » page 7.

```
inheritsFrom: aclr
     aclr  = self  ifTrue: [^true].
     (self parents isNil) ifTrue: [^false].
     self parents do: [:each | (each inheritsFrom: aclr) ifTrue: [^true]].
     ^false.
```

5.6.2.2 Quelle relation ?

Pour déterminer la relation existant entre deux intervalles de temps circulaires connexes, on utilise un automate d'états finis[1]. Le parcours de l'automate (de l'arbre) se fait en comparant un nouvel élément (l'un des instants de début ou de fin des intervalles) avec les éléments déjà examinés, sans tenir compte des autres. Ainsi, si l'on cherche la relation qui existe entre deux intervalles de période 12, le premier commençant à l'instant 1 et finissant à l'instant 4 et le second commençant à l'instant 1 et finissant 9, on constate d'abord que les instants de début et de fin du premier intervalle sont différents, ce qui permet de s'orienter sur la première hauteur de l'arbre (à droite sur la figure 16), puis que l'instant de début du second intervalle est égal à l'instant de début du premier intervalle, ce qui permet d'emprunter une nouvelle branche (la plus à gauche à partir du nouvel état), etc. Afin de ne pas avoir à tenir compte de tous les éléments à chaque état de l'automate, et afin d'avoir une écriture homogène des méthodes constituant l'automate, ce qui permettra de les générer automatiquement, on a recours à une représentation intermédiaire des données. On stocke dans une matrice 4x4 dont les indices représentent respectivement l'instant de début, l'instant de fin du second intervalle, puis l'instant de début et l'instant de fin du premier intervalle. L'élément $(i,j)_{i>j}$ de cette matrice indique si l'élément i suit immédiatement j (valeur '<' de la matrice) ou s'ils ont lieu simultanément (valeur '=' de la matrice). Dans tous les autres cas (un ou plusieurs éléments entre i et j), la valeur est nulle.

La matrice permettant de déterminer la relation entre les intervalles de l'exemple précédant est :

nil	nil	nil	nil
<	nil	nil	nil
=	<	nil	nil
<	nil	<	nil

L'automate de reconnaissance prend comme entrée cette matrice intermédiaire. A chaque état correspond une méthode, dont un exemple peut être :

```
etat17: matrice
     (((matrice at: 4) at: 1) = #= ) ifTrue: [^self etat18: matrice].
     (((matrice at: 4) at: 2) = #= ) ifTrue: [^self etat19: matrice].
     (((matrice at: 4) at: 1) = #< ) ifTrue: [^self etat20: matrice].
     (((matrice at: 4) at: 2) = #< ) ifTrue: [^self etat21: matrice].
```

Les méthodes correspondant aux états terminaux retournent la relation binaire voulue :

```
etat18: matrice
     ^CircularLapseRelation MeetsIsMet!
```

5.6.3 Polymorphisme

Les intervalles circulaires étant représentés par plusieurs classes (**CircularLapse**, **CircularLapses**, **EmptyCircularLapse**), les méthodes d'union, d'intersection et de complément doivent être définies pour chaque classe. Les lois de composition internes des intervalles de temps ont été vues plus haut[2] ; leur implémentation ne pose pas de problème particulier. Une fois connue la relation binaire existant entre deux intervalles connexes, il est très facile de déterminer leur union et leur intersection. A chaque relation binaire correspond donc une méthode pour l'union et une méthode pour l'intersection. La méthode réalisant l'union ou

[1] . Voir « Détermination des relations » page 7.
[2] . Voir « Lois de composition interne » page 8.

l'intersection commence par déterminer quelle est la relation binaire entre les deux intervalles, puis à appeler la méthode appropriée.

5.7 Deux exemples d'analyse

Deux analyses de grilles sont présentées ici. La première, une analyse de *Blues for Alice* de Charlie Parker, est tout à fait satisfaisante. On se rend compte avec la deuxième analyse, de *All Blues*de Miles Davis, que le système a tendance à engendrer des formes trop importantes.

5.7.1 Analyse de Blues for Alice (Charlie Parker)

1 F maj7	5 E ∅ / A 7	9 D-7 / G 7	13 C-7 / F 7
17 Bb maj7	21 Bb-7 / Eb 7	25 A-7 / D 7	29 Ab-7 / Db 7
33 G-7	37 C7	41 F maj7 / D-7	45 G-7 / C 7

figure 20 : Blues for Alice (Charlie Parker)

Après avoir appliqué la première base de règles, qui s'occupe de reconnaître les résolutions, le système a déjà reconnu un certain nombre de choses importantes :

- Succession de deux-cinq, allant du temps 5 au temps 17, qui se résoud en Si bémol majeur au temps 17.
- La chute de quintes (ici substituées), allant du temps 21 au temps 41, qui se résoud en Fa majeur au temps 41.
- La résolvante (il s'agit ici d'un *turn-around*, résolvante dont la dominante est à la fin de la grille, et la résolution au début), allant du temps 45 au temps 5, la dominante étant un deux-cinq.

On constate que le seul accord n'ayant pas été analysé par cette première base de règles est l'accord de D min 7 situé sur le temps 43. Une base de règle suivante détectera cet accord comme étant un accord isolé, et, comme il est analysable dans la tonalité de Fa majeur, et qu'une forme un Fa majeur le suit, il y sera finalement intégré.

Les numéros figurent les numéros de temps, indiqués en haut à gauche de chaque case. Les noms des formes sont les noms des classes des objets reconnus par le système d'analyse.

1-1 BluesShape in F MajorScale
15-7 ExtendedShape in F MajorScale
15-5 ExtendedShape in F MajorScale
17-5 ExtendedShape in F MajorScale
21-5 ExtendedIntersectedShape in F MajorScale
21-43 ChordSubstitution in F MajorScale
21-43 Resolvante in F MajorScale
21-41 Dominant in F MajorScale
21-41 Cascade in F MajorScale
21-33 Dominant in Gb MajorScale
21-33 Cascade in Gb MajorScale
21-29 Dominant in G MajorScale

21-29 Cascade in G MajorScale
21-25 Dominant in Ab MajorScale
21-25 TwoFive in Ab MajorScale
25-29 Dominant in G MajorScale
25-29 TwoFive in G MajorScale
29-33 Dominant in Gb MajorScale
29-33 TwoFive in Gb MajorScale
33-41 Dominant in F MajorScale
33-41 TwoFive in F MajorScale
41-5 ExtendedShape in F MajorScale
43-5 ExtendedShape in F MajorScale
45-5 ChordSubstitution in F MajorScale
45-5 Resolvante in F MajorScale
45-1 Dominant in F MajorScale
45-1 TwoFive in F MajorScale
5-21 ChordSubstitution in Bb MajorScale
5-21 Resolvante in Bb MajorScale
5-17 Dominant in Bb MajorScale
5-17 TwoFivesFall in Bb MajorScale
5-13 TwoFivesFall in C MajorScale
5-9 TwoFive in D MajorScale
9-13 TwoFive in C MajorScale
13-17 TwoFive in Bb MajorScale

5.7.2 Analyse de All Blues (Miles Davis)

1 G	5 G	9 G	13 G 7
17 C	21 C 7	25 G	29 G
33 D 7	37 Eb 7 / D 7	41 G	45 G

figure 21 : All Blues (Miles Davis)

On s'aperçoit ici que le système engendre des formes trop importantes. En effet, deux formes distinctes recouvrent chacune la quasi-totalité de la grille : la première, qui va du temps 25 au temps 21, est en Sol majeur ; la seconde, qui va du temps 41 au temps 43, est en Ré majeur. Même si ces analyses sont acceptables du strict point de vue harmonique, elles ne tiennent pas compte d'éléments importants, comme la longue suite d'accords de Sol au début de la grille, qui donne une importance prépondérante à la tonalité de Sol majeur.

1-1 BluesShape in G MajorScale
  25-21 ExtendedShape in G MajorScale
    25-17 ExtendedIntersectedShape in G MajorScale
      25-33 AmorphousShape in G MajorScale
      29-17 ExtendedShape in G MajorScale
        33-17 ExtendedShape in G MajorScale
          37-17 ExtendedIntersectedShape in G MajorScale
            37-45 ChordSubstitution in G MajorScale
              37-45 Resolvante in G MajorScale
                37-41 Dominant in G MajorScale
                  37-41 Cascade in G MajorScale
                    37-39 Dominant in Ab MajorScale
                    39-41 Dominant in G MajorScale
            41-17 ExtendedShape in G MajorScale
              45-17 ExtendedIntersectedShape in G MajorScale
                45-9 AmorphousShape in G MajorScale
                1-17 ExtendedShape in G MajorScale
                  1-13 AmorphousShape in G MajorScale
  41-33 ExtendedIntersectedShape in C MajorScale
    41-13 ExtendedShape in C MajorScale
      45-13 ExtendedIntersectedShape in C MajorScale
        45-9 AmorphousShape in C MajorScale
        1-13 AmorphousShape in C MajorScale
    9-33 ExtendedShape in C MajorScale
      13-33 ExtendedIntersectedShape in C MajorScale
        13-25 ExtendedShape in C MajorScale
          13-21 ChordSubstitution in C MajorScale
            13-21 Resolvante in C MajorScale
              13-17 Dominant in C MajorScale

21-33 ExtendedShape in C MajorScale
25-33 AmorphousShape in C MajorScale

6. Bibliographie

[Allen 84] Allen J.F.*, Towards a General Theory of Action and Time*, in Artificial Intelligence, vol. 23, pp. 123-154.

[Baker 89] Baker M.*, An Artificial Intellingence Approach to Musical Grouping Analysis*, in Contemporary Music Review, Vol. 3, pp. 43-68.

[Balen 93] Balen N.*, L'Odyssée du Jazz*, Editions Liana Levi.

[Barbar & al. 95] Barbar K., Beurivé A., Desainte-Catherine M.*, Structure Hiérarchique et Relations Temporelles d'une Pièce de Musique*, in Deuxièmes Journées d'Informatique Musicale, JIM'95, Laforia, F. Pachet, F. Brown, R. Mouton, pp. 133-141.

[Baroni & al. 88] Baroni M., Dalmonte R., Jacoboni C.*, Relations entre musique et poésie dans les airs de Giovanni Legrenzi*, in La musique et les sciences cognitives, Pierre Mardaga, Ed., S. McAdams & I. Deliège.

[Baroni & al. 92] Baroni M., Dalmonte R., Jacobini C.*, Theory & Analysis of European Melody*, in Computer Representations and Models in Music, Alan Marschen & Anthony Pople, Academic Press, pp. 187-206.

[Berendt 83] Berendt J.E.*, Le Grand Livre du Jazz*, Monaco, Editions du Rocher.

[Bitsch 57] Bitsch M.*, Précis d'harmonie tonale*, Paris, Editions musicales Alphonse Leduc.

[Brinkman 80] Brinkman A.R.*, Johann Sebastian Bach's Orgelbüchlein*, in Music Theory Spectrum. 2, pp. 46-73.

[Chailley 77] Chailley J.*, Traité historique d'analyse harmonique*, Paris, Alphonse Leduc.

[Couleau 87] Couleau J.P.*, Traité d'harmonie du Jazz, vol. 1 et 2*, Salabert enseignement.

[Hodeir 63] Hodeir A.*, Le Jazz*, in Histoire de la Musique, Vol. 2, Encyclopédie de la Pléiade, Roland Manuel, pp. 1075-1090.

[Joanger-Weller 86] Joanger-Weller*, Clés pour l'harmonie*, Grenoble, distribution Michel Musique.

[Ladkin 86] Ladkin P.B.*, Time Representation : A Taxonomy of Interval Relations*, in Proceedings of AAAI-86, Morgan Kaufmann, pp. 360-366.

[Ladkin 87] Ladkin P.B.*, The Completeness of a Natural System for Reasoning with Time Intervals*, in Proceedings 10th IJCAI, pp. 462-467.

[Laske 73] Laske O.*, In Search of a Genarative Grammar for Music*, in Machine Models of Music, S.M. Schwanauer & D.A. Levitt, Ed., MIT Press, pp. 215-240.

[Lerdahl & Jackendoff 83] Lerdhal F., Jackendoff R.*, A Generative Theory of Tonal Music*, MIT Press.

[Marschen 92] Marschen A.*, Modelling the perception of Musical Voices*, in Computer Representations and Models in Music, Alan Marschen & Anthony Pople, Academic Press, pp. 239-264.

[Maxwell 92] Maxwell H.J.*, An Expert System for Harmonizing Analysis of Tonal Music*, in Understanding Music with AI : Perspectives on Music Cognition, K.E. & O.L.M. Balaban, Ed., AAAI press, pp. 335-353.

[McDermott 82] Mcdemott D.*, A Temporal Logic for Reasoning about Processes and Plans*, in Cognitive Science, vol 6, pp. 101-155.

[Meehan 80] Meehan J.*, An Artificial Approach to Tonal Music Theory*, in Computer Music Journal, Vol. 4, n 2, pp. 61-65.

[Mouton & Pachet 95] Mouton, Pachet F.*, The Symbolic vs. Numeric Controversy in Automatic Analysis of Tonal Music*, in Workshop IJCAI'95 on Music an Artificial Intelligence, Montreal, pp. 32-40.

[Pachet 94] Pachet F.*, Vers un modèle du raisonnement dans les langages à objets*, in Langages et Modèles à Objets (L.M.O.), Grenoble, pp. 111-123.

[Pachet & al. 95] Pachet F., Ramalho G., Carrive J., Cornic G.*, Representing Temporal Musical Objects and Reasoning in the MusES System.*

[Pachet & Dojat 95] Pachet F., Dojat M.*, Représentation de connaissances temporelles par objets et règles*, in Rapport interne Laforia 95/19.

[Pachet 92] Pachet F., *Représentation des connaissances par objets et règles : le système NéOpus, Ph.D. thesis*, Université Paris 6.

[Pachet 93] Pachet F., *An Object-Oriented Representation of Pitch-Classes, Intervals, Scales and Chords : The Basic MusES*, in Rapport Interne Laforia 93/38, Laforia.

[Pachet 94] Pachet F., *The MusES System : an Environment for Experimenting with Knowledge Representation Techniques in Tonal Harmony*, in Proceedings First brazilian symposium on Computer Music - SBC&M'94, Caxambu, Minas Gerais, Brazil, pp. 195-201.

[Ramalho & Pachet 94] Ramalho G., Pachet F., *From Real Book to Real Jazz Performance*, in International Conference on Music Perception an Cognition, Lièges, Belgium, pp. 349-350.

[Roads 88] Roads C., *Grammars as Representations for Music*, in Foundations of Computer Music, C.Roads & J. Strawn, Ed., pp. 403-442.

[Shoham 87] Shoham Y., *Temporal Logic in AI : Semantical and Onthological Considerations*, in Artificial Intelligence, vol 33, pp. 89-104.

[Steedman 84] Steedman M.J., *A Generative Grammar for Jazz Chord Sequences*, in Music Perception, vol. 2, pp. 52-77.

[Sundberg & al. 93] Sundberg J., Lindbloom B., *Generative Theories in Language and Music Descriptions*, in Machine Models of Music, S.M. Schwanauer & D.A. Levitt, Ed., MIT Press, pp. 263-286.

[Tsang 87] Tsang E.P., *Times Structures for Artificial Intelligence*, in Proceedings 10th IJCAI, pp. 456-461.

[Ulrich 77] Ulrich W., *The Analysis and Synthesis of Jazz by Computer*, in Proceedings 5th IJCAI, MIT, Cambridge, Ma, pp. 865-872.

[Vidyamurthy 92] Vidyamurthy G., Chakrapani J., *Cognition of Tonal Centers : a Fuzzy Approach*, in Computer Music Journal, Vol. 16, n 2, pp. 45-79.

[Widmer 92] Widmer G., *Qualitative Perception Modeling and Intelligent Musical Learning*, in Computer Music Journal, Vol. 16, n 2, MIT, pp. 51-68.

[Winograd 93] Winograd T., *Linguistic and Computer Analysis of Tonal Harmony*, in Machines Models of Music, S.M. Schwanauer & D.A. Levitt, Ed., MIT Press, pp. 113-153.

7. Tables des illustrations

Table des figures

Table des tableaux