

Sur la structure algébrique des séquences d'accords de Jazz

François Pachet
SONY CSL, 6 rue Amyot, 75005 Paris
Email: pachet@csl.sony.fr

Résumé

Dans le cadre du projet d'*écoute active* mené à Sony CSL, nous nous intéressons à la notion de règle de substitution, permettant de faire des variations de structures harmoniques de séquences d'accords de Jazz. Nous décrivons un système permettant de mettre ces règles en oeuvre de manière intuitive pour produire des séquences d'accords "intéressantes". Nous proposons une représentation récursive des règles qui permet à l'utilisateur de fabriquer ses propres règles. Enfin, nous soulevons quelques problèmes théoriques non résolus.

1. De la forme ouverte à l'écoute active

Le travail présenté ici s'inscrit dans le contexte du projet "Ecoute Active" développé au laboratoire Sony-CSL. Ce projet consiste à chercher des moyens d'enrichir la pratique de l'écoute musicale d'oeuvres existantes, en donnant à l'auditeur des moyens de modifier son écoute selon certains critères. Ces critères ou paramètres définissent des espaces dans lesquels un auditeur peut se déplacer pendant l'écoute d'un morceau de musique, en faisant varier certaines de ses caractéristiques. Les points importants de notre contexte de travail sont 1) l'exploitation de pièces musicales existantes, préenregistrées, et 2) l'identification de variations sémantiquement cohérentes sur ces pièces musicales. Ce projet d'écoute active se distingue ainsi d'autres efforts de recherche en musique et interaction: D'une part, par opposition à la notion de "forme ouverte" (Eckel, 1997), nous nous intéressons à construire des environnements dans lesquels les variations musicales préservent le sens original, pour autant que l'on puisse le définir avec précision. Par ailleurs, nous ne cherchons pas à décomposer un signal numérique musical en différentes composantes perceptives, comme proposé par Lepain (1998) dans le cadre de la consultation interactive de documents musicaux, mais tentons plutôt d'extraire des possibilités de variation d'un document donné qui "font sens" musicalement.

Dans ce contexte, nous nous sommes intéressé à une dimension de variation liée à la structure harmonique de certaines pièces de Jazz. En effet, il se trouve que pour un corpus assez large de morceaux dits "standards", tels que ceux trouvés dans le Real Book (Real, 1981) ou le Fake Book (Fake, 1983), on peut faire apparaître une structure harmonique abstraite qui comporte certaines propriétés algébriques rendant possible de telles variations, au sens où 1) des séquences existantes peuvent être modifiées selon des règles précises et 2) la sémantique d'une séquence peut être préservée, dans un certain sens, par ces modifications. Ce papier décrit ces structures algébriques de manière informelle, et propose un outil expérimental permettant de produire des variations autour de grilles d'accords.

Dans la section 2, nous introduisons la notion de règle de substitution de Jazz, en la décrivant comme une règle de réécriture, et mettons en évidence son caractère combinatoire. Dans les sections 3 et 4 nous proposons une représentation des accords et des grilles d'accord servant de support à la mise en oeuvre de ces règles. Dans la section 5 nous proposons une base de règles permettant d'engendrer des variations intéressantes à partir de grilles de base, et montrons comment implémenter un mécanisme simple pour mettre ces règles en oeuvre. Enfin nous montrons comment la représentation réflexive des règles de substitution permet de créer de nouvelles règles, elles-mêmes "licites". Nous concluons en soulevant quelques problèmes théoriques.

2. Règles de substitution harmonique

L'harmonie tonale a été depuis ses débuts, le traité de Fux par exemple, l'objet de formalisations précises. Ceci est sans doute lié au fait que la musique tonale a été pensée très tôt en termes de concepts abstraits ayant de riches propriétés algébriques, notamment ceux de note et d'accords. Sur la base de ces concepts se sont développées des règles, par exemple pour l'harmonisation (la fameuse quinte parallèle) dont l'efficacité opératoire est indéniable. A tel point que de nombreux auteurs ont voulu pousser cette formalisation à ses limites, sous la forme des fameux traités d'harmonie. La formalisation de la musique tonale "marche" tellement bien qu'elle est même devenue un style littéraire, comme le défend (Corres, 1996). Si ce phénomène est discutable d'un point de vue musical, nous pensons néanmoins que certaines formes de pensées musicales se modélisent bien sous la forme de calculs dans des algèbres non triviales. C'est en particulier le cas de l'harmonie de Jazz.

L'harmonie de Jazz hérite naturellement de l'harmonie classique et peut être considérée comme une ramification de celle-ci, développée au début du siècle, et poussée à son apogée avec le be-bop des années 40 (Charlie Parker, Miles Davis, Dizzy Gillespie). Elle se caractérise par la présence de nombreux stéréotypes harmoniques qui sont le plus souvent exprimés par des règles de substitution (e.g. Siron, 1992). Ces règles de substitution permettent de complexifier une grille donnée en remplaçant certains accords, jugés trop "plats" par d'autres, plus riches (Johnson-Laird, 1991). Elles permettent aussi d'analyser une grille, en montrant que certaines séquences peuvent être "comprises" comme des variations autour de séquences de base (voir, par exemple (Baudoin, 1990) pour une classification des formes de base du Blues).

2.1 Exemples typiques

Une des règles typiques illustrant ce phénomène est la règle de la "substitution au triton", qui dit que tout accord de septième peut être substitué par son triton¹, i.e. par un accord de septième, dont la tonique est le triton de la tonique de l'accord de départ. Cette règle se justifie en théorie par plusieurs arguments musicaux, par exemple par le fait que l'accord de septième du triton comporte la septième et la tierce de l'accord de départ, et peut donc être considéré comme "proche".

On peut représenter cette règle sur un exemple, C. Cette règle se lit comme suit: "tout accord de C 7 dans une séquence peut se réécrire en F# 7" :

(substitution au triton) $C 7 \rightarrow F\# 7$

D'autres règles permettent non pas de substituer un accord, mais de créer de nouveaux accords. Par exemple, on peut *préparer* tout accord par un accord de septième résolvant dessus. Sur un cas toujours instancié, en C, la règle peut s'écrire :

(préparation par septième) $C \rightarrow G 7 / C$

Dans ce cas, il faut noter que les deux accords de la partie droite durent chacun la moitié de l'accord original, de telle sorte que l'ensemble occupe autant de temps que l'accord initial : il y a conservation de la durée totale !

Une règle similaire indique que l'on peut préparer de la même manière un accord de septième par un accord de mineur 7 :

(préparation par min7) $G 7 \rightarrow D \text{ min } 7 / G 7$

Ces règles sont présentées dans de nombreux traités d'harmonie de Jazz, le plus souvent de manière informelle et ad hoc (Cf. par exemple Josefs, 1996). Nous en donnerons d'autres exemples plus bas (section 5.1).

¹ Le triton est un intervalle fictif qui coupe exactement la gamme diatonique en deux. Il est fictif car il ne correspond à aucun intervalle diatonique. On peut l'approximer par la quarte augmentée ou la quinte diminuée, mais ce n'est qu'une approximation : 1) quarte augmentée de quarte augmentée de do = si ##, et 2) quinte diminuée de quinte diminuée de do = ré bb.

Tous ces exemples sont donnés en considérant un accord de référence (C par exemple). Il est bien entendu que les règles sont plus générales, et s'appliquent à tout accord "similaire" à l'accord (ou les accords) de la partie gauche. Ce mot "similaire" est vague, et tout le problème consiste à lui donner un sens opérationnel intuitif, ce qui précisément est l'objet de ce papier. Ceci nécessite tout d'abord de se doter d'une représentation de base pour les accords, en particulier pour les noms d'accords. Sur cette base on construit une représentation des séquences d'accords. Nous proposerons ensuite un ensemble de règles permettant 1) d'engendrer simplement des variations intéressantes par un utilisateur non expert en harmonie et 2) d'engendrer de nouvelles règles, de manière simple et intuitive. Enfin nous montrons comment implémenter ces règles.

3. Accords

Le fondement de notre étude est la notion d'accord de Jazz, et plus précisément de nom d'accord. Nous allons dans cette section décrire cette notion en la reconstruisant à partir des objets de base du système tonal.

3.1 Reconstruction du système tonal

Donner une définition précise de la notion d'accord requiert, en principe, une reconstruction du système tonal dans un langage donné. Toute reconstruction est dépendante de l'objectif ou de la classe d'application visée, et il est donc difficile de proposer une reconstruction qui soit générale. Par exemple, Ballesta (1998) propose une reconstruction de la théorie harmonique classique en vue de construire un système d'harmonisation à base de satisfaction de contraintes. Dans le même esprit, le système *Situation* (Bonnet & Ruéda, 1998) repose sur une représentation de la notion d'accord et de séquences d'accords plus primitive, mais adaptée à la composition assistée par ordinateur. Dans un autre cadre, Hudak et al. (1996) utilisent des structures algébriques définies par la programmation fonctionnelle pour définir des objets musicaux. Enfin, le système *Elody* (Orlarey et al. 97a, 97b) permet à l'utilisateur de construire sa propre théorie, et est orienté vers la mise en oeuvre d'opérateurs d'abstraction sophistiqués pour la CAO.

Nous allons ici nous contenter d'une représentation de base des accords adaptée au contexte du Jazz, et qui nous permet de poser le problème de la substitution de manière générale.

3.2 Noms d'accords : l'accord

Les accords utilisés en Jazz possèdent certaines caractéristiques qui les rendent un peu différents des accords de la musique classique. En particulier, ils relèvent d'une nomenclature spécifique, qui permet de les nommer de manière compacte. Il est important de distinguer ici deux notions d'accords : les accords concrets ou réalisés, et les accords intentionnels.

La notion d'accord "concret" (ou réalisé en musique classique), peut être définie comme un ensemble de notes effectives. Cette notion d'accord est utile pour parler de musique harmonisée (comme dans le système *Situation* par exemple), mais son caractère extensionnel la rend peu adaptée pour parler de séquences de noms d'accords.

La notion d'accord intentionnel est à la base de la théorie des *pitch classes* (Forte, 1973). Le *pitch class* est la version intentionnelle de la note (une note privée de l'information d'octave). Mais la notion de *pitch class set* construite au dessus des *pitch class* est conçue pour la musique atonale, et rend donc très mal compte des propriétés tonales des accords de Jazz (par exemple, un accord mineur et un accord majeur sont équivalents dans cette théorie).

Dans notre contexte, nous avons besoin d'une représentation intentionnelle des accords (nous ne nous intéressons pas aux réalisations, mais uniquement aux noms d'accords), qui obéissent aux lois de la musique tonale. Nous définissons ainsi un *PitchClassChord* comme un ensemble de *pitch classes* (au sens de (Forte, 1973)), construites d'une certaine manière (par succession de tierces, voir section suivante). Cette notion va nous permettre en particulier de nommer les accords de manière non ambiguë, et conforme à la pratique du Jazz.

3.2.1 L'accord intentionnel comme une superposition de tierces

Nous proposons de représenter les *PitchClassChord* comme des successions de tierces, construites au dessus d'une note de base appelée *tonique*. De plus, nous autorisons un maximum de sept notes, avec pour chacune d'entre elle plusieurs possibilités, selon qu'elles sont absentes ou présentes avec ou sans altérations. Ceci permet de construire les accords ayant *au plus* une instance de chacune des sept pitch classes naturelles, éventuellement altérée.

Ainsi, chaque accord peut être vu, de manière interne, comme un tableau de sept indicateurs. Ces indicateurs ne sont pas tous équivalents, de manière à respecter les lois de l'harmonie diatonique (ces possibilités sont illustrées Figure 1) :

- La tonique peut être absente ou présente (2 possibilités),
- La tierce peut être absente, majeure ou mineure (3 possibilités),
- La quarte (ou quinte) peut être absente, diminuée, parfaite ou augmentée (4 possibilités),
- La septième peut être absente, diminuée, mineure, ou majeure (4 possibilités),
- La neuvième (resp. onzième et treizième) peut être absente, diminuée, majeure ou augmentée (4 possibilités).

Cette construction permet, à partir d'une tonique donnée, de construire exactement : $2*3*4*4*4*3*3 = 3456$ accords, et de les nommer de manière non ambiguë. Par exemple, un accord de mineur septième se représente par (tonique, tierce mineure, quinte parfait, septième mineure, pas de neuvième, pas de onzième, pas de treizième). Elle permet aussi de reconstituer un nom à partir d'une liste de *pitch classes*. Ceci permet de construire tous les accords courants du Jazz, ainsi que certains moins courants.

Bien sûr, tous les accords ne rentrent pas dans cette formalisation. A titre d'exemple, les accords que l'on ne peut pas engendrer sont les suivants :

- accords de plus de 7 notes de polyphonie (accord de quinzième, inexistants en Jazz d'un point de vue intentionnel).
- accords ayant plusieurs instances d'une même pitch class naturelle, comme (C C# E G). Dans certains cas, ces accords sont équivalents à des accords que nous pouvons engendrer ((C Db E G) pour l'exemple précédent).
- accords basés sur des enharmonies complexes, non réductibles à un ensemble de sept pitch classes naturelles distinctes, comme (C, D#, C##, D) ou (D, Ebb, C##). Ces derniers accords présentent peu d'intérêt pour des accords intentionnels, dans le cadre d'un système d'analyse tonale (de tels accords pourraient bien sûr être intéressants comme accords réalisés, mais pas comme noms d'accords apparaissant dans une grille d'accords).

Cette représentation par tierces successives possède de nombreux autres intérêts, comme la possibilité de faire varier la polyphonie d'un accord en fonction d'une gamme donnée. Par exemple, un accord de 1^e degré pour la gamme de C peut être indifféremment C majeur, C maj7 ou C maj7 9. D'une certaine manière, ces accords peuvent être considérés comme équivalents. Ceci permet alors d'introduire une relation de généralité entre accords, fondée simplement sur la comparaison des représentations internes :

Un accord *a* est *plus riche* qu'un accord *b*, ssi :

Toute tierce de *b* (dans sa représentation interne) est de même nature que celle de *a* correspondante, sauf si elle est absente, auquel cas la tierce de *a* peut être quelconque.

Ainsi, C maj7 est "plus riche" que C; D 9 est plus riche que D, mais D9 est incomparable à D min 7, etc.. Cette notion est utilisée plus bas pour l'algorithme d'appariement (section 5.4).

3.3 Noms d'accords : le nom

Cette construction par tierces des accords est utile pour une représentation interne, mais pas pour un nommage effectif (représentation externe). Les noms d'accords suivent en effet en Jazz des règles précises, qui permettent de les nommer de manière compacte.

Par exemple, les accords majeurs ont comme représentation interne :

(tonique, tierce majeure, quinte parfaite, pas de septième, pas de neuvième, pas de onzième, pas de treizième) et comme représentation externe ... rien du tout ! Certaines règles permettent en outre d'exprimer des notes de manière implicite. Ainsi, un accord de "C 9" (représentation externe) signifie en fait "neuvième et septième mineure", etc.

Nous ne détaillerons pas ici toutes ces règles, qui sont implémentées en MusES/Java (Pachet, 1996). Un éditeur d'accord permet de construire ces accords soit en indiquant la liste des *pitch classes* (champ "notes"), soit de spécifier leur structure interne (par les différents switch rangés en 7 colonnes). L'éditeur calcule alors automatiquement les attributs manquants (nom externe à partir de la représentation interne, ou de la liste des notes; liste des notes à partir de la représentation interne, etc.).

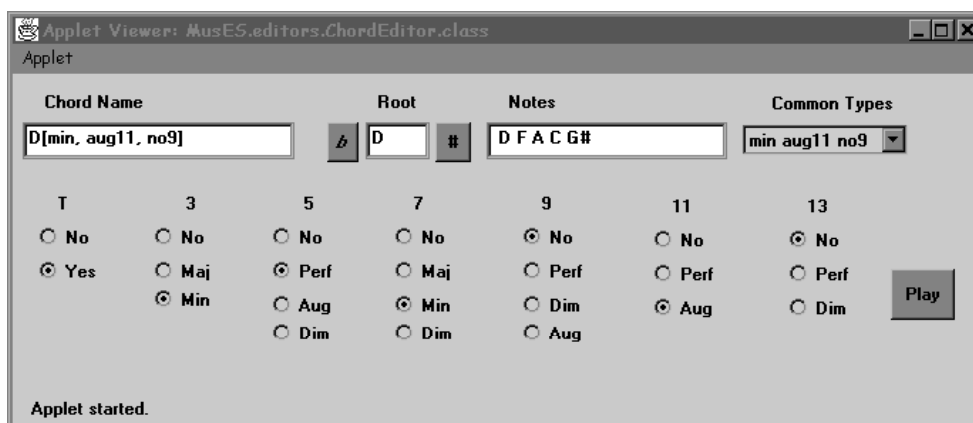


Figure 1. L'éditeur d'accords, visualisant la représentation interne sous forme de superpositions de tierces, la représentation interne du nom et le nom "commun".

4. Séquences d'accords

L'objet de notre étude est la séquence d'accord, qui se représente naturellement comme une suite ordonnée d'accords de *pitch class* tels que décrits dans la section précédente.

La représentation que nous utilisons pour les séquences d'accords est fondée sur une conception générique des objets temporels. Chaque objet temporel possède un attribut *lapse*, qui lui-même représente un intervalle de temps (temps initial + durée). Une classe particulière *TemporalPitchClassChord* est ainsi définie, qui combine à la fois les propriétés des accords de *pitch classes*, et des objets temporels. Enfin, la séquence elle-même est une liste de *TemporalPitchClassChord*, triée par date de début croissante.

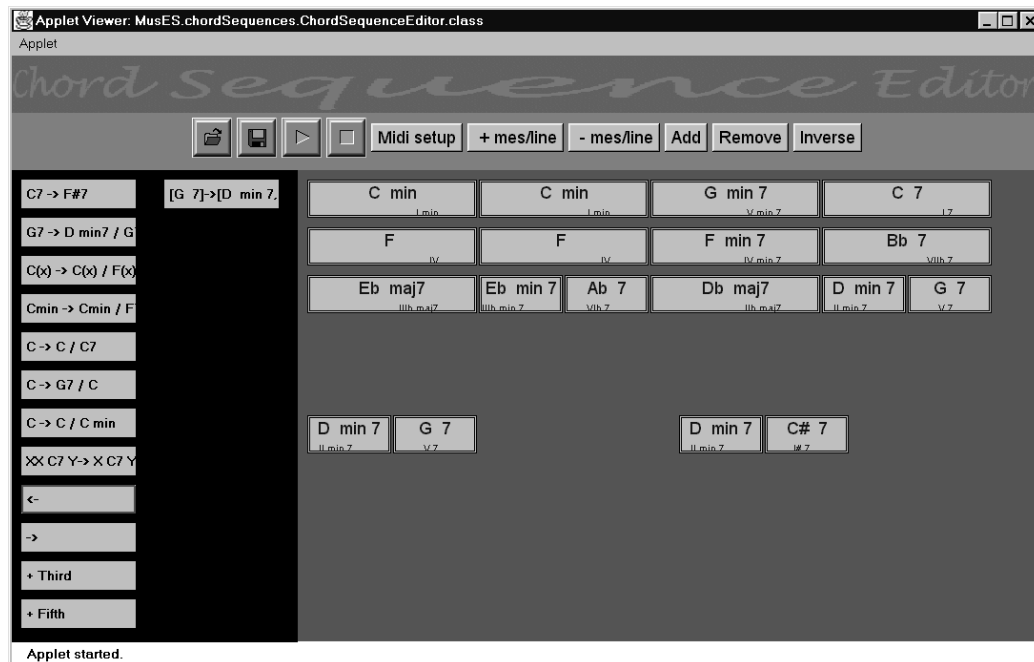


Figure 2. L'éditeur de séquences d'accords, sur la grille de *Solar* (Miles Davis). A gauche la liste des règles de substitution. En haut à droite la séquence d'accords. Au milieu, une règle considérée comme un couple de séquences d'accords.

Enfin, nous avons un format de sortie texte pour ces séquences, qui ressemble au format "standard", et qui est constitué des noms d'accords communs, et de caractères de séparation : "|" pour séparer les mesures (4 temps par défaut), "/" entre le 2^e et le 3^e temps, ";" entre le 1^{er} et le 2^e, et ":" entre le 3^e et le 4^e.

Par exemple, la séquence d'accord suivante : 'C, A min 7 / D min 7 ; G 7' représente la suite temporelle de ces quatre accords durant chacun 1 temps, et commençant respectivement sur les 4 premiers temps d'une mesure.

5. Règles de substitution et appariement des accords

Sur la base de notre représentation des séquences d'accords, nous allons maintenant décrire comment mettre en oeuvre nos règles de substitution. Ces règles de substitution peuvent être vues comme des règles de *réécriture*, au sens des systèmes de réécriture (Jouannaud & Lescanne 86). Cependant, nous n'exploitons pas les résultats théoriques concernant la réécriture, qui sont essentiellement dédiés au problème de la détection de la confluence d'une base de règles, appliquée à saturation sur une expression de départ. Or dans notre contexte, le problème de la confluence de se pose pas : les règles sont d'autant plus intéressantes qu'elles permettent un grand nombre de possibilités pour une séquence de départ. Ce qui nous intéresse ici est la manière dont on va appairer les règles avec les accords (ou les séquences d'accords).

Nous allons tout d'abord de proposer une base de règles de substitution, qui permet d'engendrer un certain nombre de variations intéressantes. Nous montrerons ensuite comment les implémenter.

5.1 Les règles de base

La base de règles "de base" que nous proposons contient les sept règles suivantes. Cette base n'est en aucun cas "minimale" (voir section 6 à ce sujet), mais est le résultat de plusieurs tentatives pour proposer un ensemble réduit de règles permettant d'obtenir des variations intéressantes.

Pachet, François, Sur la structure algébrique des séquences d'accord, Journées d'Informatique Musicale, Agelonde, May, 1998.

5.1.1 Règles simples

Ces règles permettent de produire les transformations harmoniques de base sur des sous-séquences:

Répétition

Cette règle est utilitaire, et ne possède pas de sens harmonique, elle sert simplement à préparer le matériel harmonique:

$$\text{(répétition)} \quad C \rightarrow C / C$$

Elle signifie : tout accord quelconque peut se dédoubler à l'identique.

Triton

Cette règle très importante en Jazz permet d'introduire une harmonie lointaine sur un accord de septième:

$$\text{(triton)} \quad C 7 \rightarrow F\# 7$$

Préparations

Ces deux règles permettent d'introduire des préparations harmoniques. On en distingue deux principales:

$$\text{(préparation par mineur)} \quad G 7 \rightarrow D \text{ min } 7 / G 7$$

$$\text{(préparation par septième)} \quad C \rightarrow G 7 / C$$

5.1.2 Règle sur la structure temporelle

La règle suivante permet de faire reculer dans le temps des préparations de septième. Ce phénomène est très courant en Jazz, et est à la base de la nature *context-dependent* des règles de substitution, comme analysé sommairement dans (Steedman, 1984). Elle peut s'écrire (informellement) comme suit:

$$\text{(recul de septième)} \quad XX C7 Y \rightarrow X C7 YY$$

Elle signifie qu'un accord de septième peut reculer, en prenant de la place sur l'accord précédent. Par exemple (voir la notation temporelle pour les séquences d'accords en 4) :

$$C | D 7 / G \rightarrow C / D7 | G$$

5.1.3 Règles dangereuses

Ces règles sont dangereuses au sens où leur application ne garantit plus que la sémantique originale de la séquence sera préservée. Néanmoins elles sont souvent nécessaires pour produire des variations intéressantes (voir section suivante).

Le passage à la quarte permet d'introduire un nouvel accord qui reste dans la tonalité. La règle est dangereuse car si on l'applique récursivement, on sort de la tonalité:

$$\text{(passage à la quarte)} \quad C \rightarrow C F$$

Enfin, la règle suivante permet tout simplement de supprimer des accords "à gauche":

$$\text{(destruction à gauche)} \quad X C \rightarrow C$$

Par exemple : $C / F \rightarrow F$. Cette règle est dangereuse pour la même raison : elle n'est pas contextuelle et donc elle peut détruire des accords "importants".

5.2 Exemples

Nous donnons ici deux exemples typiques de suites de transformations possibles avec notre base de règles pour illustrer ses possibilités.

Pachet, François, Sur la structure algébrique des séquences d'accord, Journées d'Informatique Musicale, Agelonde, May, 1998.

5.2.1 Constitution d'une descente chromatique à partir d'un blues de base

Cet exemple illustre comment produire des séquences harmoniques chromatiques à partir de séquence de base.

Considérons la (sous) séquence de départ (début d'un blues de base (Steedman, 1984)):

C | F | C | C7 | F ...

Nous pouvons modifier cette séquence (les 4 premiers accords), pour obtenir la séquence d'arrivée suivante :

C | B min 7 / A# 7 | A min 7 / G# 7 | G min 7 / F# 7 | F ...

Il suffit pour cela d'appliquer les règles suivantes:

C F C C7 F ...	avec règle (préparation par mineur), donne :
C F C G min 7 / C7 F ...	avec règle (triton), donne :
C F C G min 7 / F#7 F ...	avec règle (préparation par septième), donne :
C F C D 7, G min 7 / F#7 F ...	avec règle (recul de septième), donne :
C F C / D 7 G min 7 / F#7 F ...	avec règle (destruction à gauche), donne :
C F D 7 G min 7 / F#7 F ...	

On réitère avec le D7, avec les règles (préparation par mineur), (triton), (préparation par septième), (recul de septième) et (destruction à gauche), pour obtenir :

C | F / E 7 | A min 7 / G# 7 | G min 7 / F#7 | F ...

Puis on réapplique le tout sur le E7 pour donner la séquence finale.

5.2.2 Exemple #2 : fabrication de *turnarounds*

Le *turnaround* est une séquence de 4 accords qui se place traditionnellement à la fin d'un morceau, et qui est destinée à remplacer un accord de premier degré suivi d'une septième (par exemple C | G7), jugés trop ennuyeux. Un *turnaround* simple, sur deux mesures, peut être décrit par la séquence suivante:

(turnaround #1) C / A 7 | D min 7 / G 7

De nombreuses variations de *turnaround* ont été introduites par différents compositeurs de Jazz. Un bel exemple est le *turnaround* suivant, que l'on peut trouver dans plusieurs morceaux de Bill Evans :

(turnaround #2) C / Eb 7 | Ab maj7 / Db maj7

Le premier *turnaround* peut s'obtenir très simplement par l'application de 3 règles sur la séquence de base. Par exemple, partant de C | G 7 :

C G7	avec règle (préparation par mineur), donne :
C D min 7 / G7	avec règle (préparation par septième), donne :
C A7, D7 / G 7	avec règle (recul de septième), donne :
C / A 7 D min 7 / G 7	CQFD...

Nous ne sommes pas arrivés à obtenir le second *turnaround* à partir de notre base de règles. On peut néanmoins assez facilement obtenir une bonne approximation par la séquence de règles suivantes :

C G7	avec règle (préparation par septième), donne :
C D 7 / G7	avec règle (préparation par septième), donne :
C A7, D7 / G 7	avec règle (recul de septième), donne :
C / A 7 D 7 / G 7	avec règle (triton), donne :
C / D#7 D 7 / G 7	avec règle (triton), donne :
C / D#7 G# 7 / G 7	avec règle (triton), donne :
C / D#7 G# 7 / C# 7	

Il faut ensuite polir à la main en remplaçant les deux derniers accords de 7 par des accords maj7. On pourrait à ce stade décider de rajouter une règle: $C 7 \rightarrow C \text{ maj}7$, classée d'office dans la catégorie des règles "dangereuses". Mais nous verrons qu'avec la possibilité de fabriquer de nouvelles règles, ceci est inutile. Le reste est un problème d'enharmonie, que nous ne traitons pas dans le cadre de ce papier (Eb maj7 au lieu de D#7).

5.3 Interface

Pour mettre en oeuvre les règles de substitution, l'utilisateur doit choisir une règle à partir de la palette de base, puis promener la souris sur la grille. Le mécanisme d'appariement est alors déclenché de manière continue, et le résultat est visualisé par une couleur : l'accord se trouvant sous le curseur se colore en vert si la règle est applicable, et en bleu si elle ne l'est pas. Si le membre gauche de la règle comporte plusieurs accords, l'appariement est tenté en prenant l'accord sous le curseur comme premier accord. Si l'utilisateur clique sur un accord vert, la règle est déclenchée, et la substitution opérée sur la séquence. Enfin, l'utilisateur peut bien sûr "sortir" du système, et éditer un accord individuellement en utilisant l'éditeur de la Figure 1.

5.4 L'algorithme de matching

La mise en oeuvre des règles de substitution requiert une définition précise de la notion de pattern, ou d'appariement entre accords : il faut pouvoir détecter automatiquement quand 2 accords sont similaires, et il y a bien sûr de nombreuses manières de la faire. Pour des raisons de simplicité, nous avons décidé de ne garder, dans un premier temps, que l'appariement le plus simple, qui consiste à transposer la tonique en gardant la structure de l'accord constante, et à dilater/compresser la structure temporelle de manière isomorphe. Ceci nous permet d'écrire les règles en choisissant une tonalité de référence et une structure temporelle arbitraires. Ce mécanisme ne permet pas de représenter toutes les règles de notre base de départ. Mais il permet, comme nous le montrons d'offrir à l'utilisateur un moyen simple de produire de nouvelles règles, sans avoir à introduire un langage d'ordre un. Les autres règles sont implémentées "en dur" dans le système, et ne sont donc pas paramétrables.

Nous allons maintenant décrire le mécanisme de base pour l'appariement. Ensuite nous montrons comment l'utilisateur peut créer ses propres règles grâce à ce mécanisme.

5.4.1 L'algorithme générique

Pour décrire notre mécanisme d'appariement, considérons la règle suivante:

$$(1) \quad C 7 \rightarrow G \text{ min } 7 / C 7$$

Le membre gauche de cette règle (C 7) va s'apparier à tout accord ayant comme structure "7", et de durée quelconque. Une fois appliqué, le mécanisme d'appariement va transposer la partie droite de la règle de manière à respecter les rapports d'intervalles. Par exemple, la règle (1) appliquée à l'accord "G 7" durant 2 temps, va donner en partie droite: "D min 7, G 7"

De même, une structure temporelle à gauche plus complexe sera aussi considérée comme variable:

$$(2) \quad G \text{ min } 7 | C 7 \rightarrow G\# 7 | C 7$$

pourra s'appliquer à la séquence "D min 7 / G 7" pour donner "G# 7 / G 7".

Enfin, il reste un problème lié à la notion de polyphonie des accords, comme expliqué dans la section 3. En effet, dans de nombreux cas on voudrait exprimer une règle générale portant sur plusieurs structures d'accords considérées comme équivalentes; Par exemple, la règle (1) de substitution au triton devrait s'appliquer aussi à tout accord de septième "plus riche", comme un accord de neuvième. Comme décrit en section 3.2.1, on considère simplement qu'un accord X match un accord Y ssi sa structure couvre celle de X, (est plus générale), au sens de sa représentation interne. Ainsi:

$$G 7 9 \text{ match avec } C 7$$

G min 7 dim 5 ne matche pas avec C min 7, etc.

L'algorithme de matching peut donc se décrire de la manière suivante.

On note *seq* une séquence d'accords, sur laquelle on veut appliquer une règle. On note *seq_i* le *i*^e accord de la séquence, *duration(c)* la durée d'un accord *c*. La représentation interne d'un accord *c* est notée *interne(c, i)*, ou *i* est l'index de la tierce considérée (voir section 3).

On considère une règle comme une paire {*condition, action*}, ou *condition* et *action* sont elles-mêmes représentées comme des séquences d'accords. On peut alors décrire l'algorithme qui vérifie qu'une règle condition est applicable à une séquence *seq*, comme suit:

match-sequence (cond, seq)

// rend vrai si seq match avec cond

intervallePitch := intervalleBetween(cond₁, seq₁); // la transposition entre les 2 séquences

dilatationTemporelle := duration(cond₁)/duration(seq₁) // la dilatation temporelle entre les 2 séquences

for i := 1 to nbChords(cond) do // pour chaque accord on vérifie la transposition et la dilatation

if not (match-accords(cond, seq, intervallePitch)) Then return false;

if (duration(cond_i) ≠ (duration(seq_i) * dilatationTemporelle)) Then return false;

end For

return true

match-accords (a₁, a₂, i)

// a₁ et a₂ sont les deux accords, i est un intervalle diatonique

return intervalle (tonique(a₁), tonique(a₂)) = i AND

match-accords-intern(a₁, a₂)

match-accords-intern(a₁, a₂)

// retourne vrai si la représentation interne de a₂ "couvre" celle de a₁, c'est à dire a₂ est "plus riche" que a₁.

for i = 1 to 7 do

if (interne(a₁, i) ≠ interne(a₂, i) AND

interne (a₁, i) ≠ 'absente') THEN return false

endFor

return true

Figure 3. Mécanisme d'appariement des règles simples

Une fois ce mécanisme défini, il est trivial d'implémenter le mécanisme d'application d'une règle à une séquence, en suivant le même procédé (transposition et dilatation temporelle). Le résultat est alors substitué à la séquence d'origine. Ce mécanisme permet d'implanter notamment les règles (triton), (préparation par septième) et (préparation par mineur).

5.4.2 L'algorithme "en dur"

Pour les autres règles ne rentrant pas dans le cadre de notre mécanisme, nous utilisons un algorithme en dur, qui ne repose pas sur un mécanisme général. Chaque règle est implémentée par deux méthodes, au sens des langages à objets : une méthode de test, qui prend une sous séquence en argument et rend vrai si la sous séquence match avec la condition de la règle; et une méthode d'application, qui produit une nouvelle sous séquence, et qui sera substituée ensuite dans la séquence originale. C'est avec ce mécanisme que nous avons implémenté toutes les règles qui ne peuvent pas se formaliser par le mécanisme précédent, comme (répétition), (recul de septième) et (destruction à gauche).

5.5 Combinaison de règles

Un des buts de notre approche est de donner à l'utilisateur un moyen d'étendre la panoplie des règles de base. Dans cette étude, nous avons sciemment évité de proposer un véritable langage, en introduisant d'une manière ou d'une autre une notion de variable (pour représenter les accords, les toniques ou les structures des accords). Nous avons plutôt cherché un mécanisme très simple, permettant de créer de nouvelles règles, sans introduire de nouveau concept.

La solution que nous avons trouvée repose sur une idée très simple, déjà évoquée pour décrire notre algorithme d'appariement: les règles de substitutions sont elles-mêmes des séquences d'accord, plus précisément, des couples de séquences d'accords. Cette idée, suivie de son implantation (qui ne pose

Pachet, François, Sur la structure algébrique des séquences d'accord, Journées d'Informatique Musicale, Agelonde, May, 1998.

pas de problème particulier), permet alors à un utilisateur d'appliquer des règles sur les règles elles-mêmes, que ce soit leur partie gauche ou droite. Le résultat peut alors être considéré comme une nouvelle règle, et ajouté à la palette des règles, puis utilisée comme les autres. Dans la Figure 2 on voit apparaître une règle en cours d'édition, en dessous de la séquence d'accords en cours.

Cette règle est la suivante:

(règle utilisateur) $D \text{ min } 7 / G 7 \rightarrow D \text{ min } 7 / C\# 7$

Elle a été obtenue simplement en prenant la règle de base (préparation par septième), puis en appliquant la règle (triton) sur le deuxième accord de son membre droit. Le résultat est alors ajouté (bouton "Add") dans la palette.

Ce mécanisme permet alors de produire toutes sortes de nouvelles règles (par exemple les règles produisant les turnaround sophistiqués), évitant ainsi de refaire les mêmes opérations plusieurs fois, et permettant d'étendre le langage de base de manière illimitée.

Il faut noter que les règles ainsi produites sont toutes "licites", puisqu'elles peuvent être vues comme des compositions de règles existantes.

6. Problèmes ouverts

Sur la base de ce travail, nous pouvons mentionner plusieurs problèmes ouverts intéressants pour la formalisation de la musique tonale.

Le premier problème est celui de l'analyse automatique. En ce qui concerne le problème strict de l'analyse automatique, Ulrich (77) a proposé un algorithme permettant de calculer la tonalité d'une séquence simple. Steedman (84) a proposé de les formaliser les règles nécessaires à analyser les grilles de Blues, et en a identifié 6. Le but de Steedman était de proposer un ensemble minimal de règles permettant de décrire toutes les grilles de Blues. Le résultat est une grammaire *context-dependent* dont il est difficile de dire si elle est minimale, et si elle permet effectivement d'engendrer toutes les grilles de Blues, et bien que celles-ci (Steedman 95). En particulier, les règles destructrices rendent la grammaire sous-jacente dépendante du contexte, empêchant ainsi une analyse automatique simple. Nous avons vu que ces règles sont nécessaires pour obtenir un système utilisable, en pratique, ce qui est conforme à la proposition de Steedman. Nous avons proposé dans (Pachet, 1998) une solution pour l'analyse automatique, reposant sur une combinaison de règles de regroupement et de règles de destruction. Cette solution permet d'analyser correctement de nombreuses grilles, mais elle ne permet pas de caractériser formellement l'ensemble des grilles analysables, ni de prouver par exemple, l'impossibilité d'analyser une grille donnée (comme la grille *Solar* de Miles Davis).

Par ailleurs, le système proposé ici ne tient pas compte de la tonalité globale d'une séquence (ou sous séquence) d'accord, ni des degrés des accords par rapport à une tonalité de base. Ceci peut être gênant dans certains cas, ou l'on voudrait que le système choisisse une structure d'accords plus diatonique.

Enfin, notre base de départ n'est pas minimale : il faudrait pour cela pouvoir définir précisément par rapport à quoi ! Elle relève d'un choix heuristique, de même que le choix des opérateurs de base dans le système Elody. Il est pour l'instant impossible de caractériser formellement l'ensemble des séquences que l'on peut produire avec une base de règles donnée, et donc cette question est ouverte...

7. Conclusion

Nous avons montré comment on peut modéliser les règles de substitution de Jazz en les mettant en oeuvre dans une interface simple. Le système permet alors de produire des grilles d'accords "intéressantes", sans connaissances a priori sur l'harmonie de Jazz. Un des points importants est la possibilité pour l'utilisateur de créer de nouvelles règles, sans pour autant rajouter de nouveau concept au système de départ. Ce type de mécanisme permet ainsi de construire des systèmes d'exploration qui garantissent, une certaine cohérence d'un matériel musical de départ, mettant alors en lumière une dimension perceptive pertinente dans le contexte du projet "Ecoute Active". En ce sens, une version

Pachet, François, Sur la structure algébrique des séquences d'accord, Journées d'Informatique Musicale, Agelonde, May, 1998.

rythmique du système est en cours, qui permet de produire de rythmes complexes à partir d'un nombre réduit de règles simples.

Enfin, et plus fondamentalement, notre travail montre que la structure des accords de Jazz est riche et encore mal comprise d'un point de vue formel, et que de nombreux problèmes théoriques restent encore ouverts.

Références

- Ballesta P., "Contraintes et objets: clefs de voûte d'un outil d'aide à la composition", *Recherches et applications en informatique musicale*, Hermes, à paraître, 1998.
- Beaudoin P., *Jazz, mode d'emploi*, Paris, Outre Mesure, 1990.
- Bonnet A., Rueda C., "Un langage visuel basé sur les contraintes pour la composition Musicale", *Recherches et applications en informatique musicale*, Hermes, à paraître, 1998.
- Corres C., *Ecritures de la musique*, PUF, Paris, 1996.
- Eckel G., "Exploring Musical Space by Means of Virtual Architecture", Proceedings of the 8th International Symposium on Electronic Art, School of the Art Institute of Chicago, Chicago 1997.
- Fake, *The World's Greatest Fake book*, Sher Music Co, San Francisco, 1983.
- Forte A., *The Structure of Atonal Music*, Yale University Press, New Heaven, 1973.
- Hudak P., Makucevich T., Gadde S., Whong B., "Haskore music notation - an algebra of music", *Journal of Functional Programming*, 6(3), Juin 1996.
- Järinen T., "Tonal Hierarchies in Jazz Improvisation", *Music Perception*, 12(4), pp. 415-437, 1995.
- Josefs J., *Writing music for hit songs*, Writer's Digest Books, Cincinnati, MacMillan, 1996.
- Johnson-Laird P. N., "Jazz Improvisation: A Theory at the Computational level", *Representing Musical Structures*, P. Howell, R. West and I. Cross, Academic Press, pp. 291-325, 1991.
- Jouannaud J.P., Lescanne P., "La Réécriture", *Techniques et Sciences informatiques*, vol. 5(6), pp. 433-452, 1986.
- Lepain P., "Ecoute interactive des documents musicaux numériques", *Recherches et applications en informatique musicale*, Hermes, à paraître, 1998.
- Orlarey Y., Fober D., Letz S., "L'environnement de composition musicale Elody", Actes des Jim' 97, Lyon, 1997.
- Orlarey Y., Fober D., Letz S., "Elody : a Java + MidiShare Based Music Composition Environment", *Proceedings of the ICMC*, 1997b.
- Pachet F., "Computer Analysis of Jazz Chord Sequences. Is Solar a Blues ?", *Readings in Music and Artificial Intelligence*, Miranda E., Ed., Harwood Academic Publishers, To appear, 1998.
- Pachet F., Ramalho G., Carrive J., "Representing temporal musical objects and reasoning in the MusES system", *Journal of New Music Research*, vol. 25, n° 3, pp. 252-275, 1996.
- Real, *The Real Book*, The Real Book Press, 1981.
- Siron J., *La partition intérieure*, Outre Mesure, 1992.
- Steedman M.J., "A Generative Grammar for Jazz Chord Sequences." *Music Perception*, 2(1), pp. 52-77, 1984.
- Steedman M. J., "The Blues and the abstract truth: music and mental models", *draft*, 1995.
- Ulrich W., "The Analysis and Synthesis of Jazz by Computer", *Fifth International Joint Conference on Artificial Intelligence*, MIT, Cambridge, Ma, pp. 865-872, 1977.