To appear in *Soft Computing*, special issue on Formal Systems and Music, G. Assayag, V. Caffagna, M. Chemillier (eds.)

Steedman's grammar for jazz chord sequences

Marc Chemillier

Abstract This paper describes some aspects of Steedman's grammar for jazz chord sequences. We have implemented this formalism in a computer system for improvisation. Some specific problems concerning the rewrite rule mechanism arise from the real-time improvisation context. This leads to the formal study of the language of chord sequences generated by Steedman's grammar, which rely on particular cadential sequences. An improvement of the system in order to fit the real-time constraints could consist in precompiling these cadential sequences for improvisation and elaboration

Keywords Formal grammar, rewrite rule, chord sequence, harmonic substitution, jazz, improvisation, real-time

1 Introduction

Formal grammars are the simplest way to define formal systems. Recall that a formal system is a system of symbols together with rules to combine them into sequences of symbols which are considered as meaningful. In the case of formal grammars, the criterion for determining which strings of symbols are meaningful is that they can be derived from particular sequences called axioms by a specific type of derivation called "rewrite rules". For instance, the rewrite rule $aa \rightarrow a$ leads to the derivation of *abaac* into *abac*, the occurrence of *aa* being replaced by *a*.

It is easy to see that the rewrite rule mechanism is well suited for modeling the musical notion of "variation". Thus a great amount of researches has been made in the application of the grammar formalism to music (see [2, 7, 9, 10, 15, 16] for a selection of references related to this subject). One of the most relevant approaches in this direction is Steedman's study of jazz chord substitutions, which are a device used by jazz musicians to create more and more elaborate variations of the harmonic structure underlying an improvisation.

Steedman's grammar is designed to generatively specify the set of possible blues chord sequences. It is based on certain rules called "substitution rules," which allow the more elaborate sequences to be derived or generated from the most basic form by the *replacement* of the original chords by substitutes ([17] page 56).

The paper written by Steedman in 1984 has been quite influential on related works by Johnson-Laird and Pachet [8, 11, 12, 13, 14]. It has been followed by further articles by Steedman himself on the same subject [18], but these new developments are more to do with making the grammar deliver an explicit harmonic interpretation than changing the musical content of the syntactic grammar itself. In this note, we present some particular aspects of Steedman's grammar taken from his 1984 original paper, which have already been discussed in [7].

We have used Steedman's formalism as a component of a system of improvisation with the computer, designed in collaboration with Gérard Assayag and Carlos Agon. Besides the improvisation itself (the musician plays phrases which are recorded by the system, and transformed into new phrases in the same style, according to a model introduced in [1]), the system provides an accompaniment part by playing voicings based on the chord sequence of the tune on which the musician improvises [6]. This chord sequence is elaborated using Steedman's rewrite rules.

There are two ways of interacting with our improvisation system. The first way is to play musical phrases, as it is said before. The second way is to change the values of some parameters of the computation process. For instance, in the process of applying rewrite rules to the chord sequence, one can change the number of rules which are applied (this corresponds to the "deepness" of the substitution process). But there is a strong limitation in the interaction loop of our system, due to the rewrite rule mechanism. When one changes the number of rules applied to the chord sequence, this change has no effect until the current chord sequence is played entirely, and the new value is only taken into account in the computation of the next chord sequence to be played.

In the context of our improvisation system, we have to compute the substitutions of the chord sequence as a whole. We cannot make the substitutions only on a small part of it, because otherwise we would not be able to reach all the variations of the chord sequence that can be generated by the grammar. Thus when the user changes any parameter of the system (for instance the number of rules applied to the chord sequence), he has to wait that the current sequence has been played entirely before the new values he has given are taken into account.

One possible way to improve this substitution mechanism in order to make it more reactive to the inputs of the user, is to look more closely at the effects of Steedman's substitutions rules on a given chord sequence. Can we predict in some way the kind of chord sequences that can be derived from a given chord sequence by applying Steedman's grammar? To answer this question we have to study the formal properties of the "language" of chord sequences generated in this way. We present in this note a simple property, which is based on the very specific role played by the "dominant seventh chords" in Steedman's substitution rules. This property could be the first step in the direction of a deeper understanding of the kind of chord sequences that are generated by Steedman's grammar.

2
Formal grammars and real-time interaction

The following definitions are adapted from [3]. Let us recall that for every set of symbols X, we denote by X^* the set of finite sequences of elements of X.

A *grammar* $G = (s, N, T, R)$ is defined by two disjoint sets of symbols, the terminals T and the nonterminals N, a particular nonterminal $s \in N$ called the *axiom*, and a finite set R included in $(N \cup T)^* \times (N \cup T)^*$ of *rewrite rules*. A rewrite rule $(u, v) \in R$ is usually written in the form $u \to v$.

We say that a sequence f can be *derived* into g, and we write $f \to g$, iff there are factorizations $f = wuz$, $g = wvz$ with $w, u, v, z \in (N \cup T)^*$, such that $u \to v \in R$. Thus any rule $u \to v$ means that the sequence u may be replaced by v inside any sequence in which it occurs. For any integer p, define

$$f \xrightarrow{p} g$$

iff there exist $f_0, f_1, \ldots f_p$ such that $f_0 = f$, $f_p = g$, and $f_i \to f_{i+1}$ for $i = 0, \ldots p-1$. In this case, p is called the *length* of the derivation from f into g. Finally, we define

$$f \xrightarrow{*} g$$

iff $f \xrightarrow{p} g$ for some integer p.

The *language* generated by the grammar is the set of sequences of terminal symbols that can be derived from the axiom, using the rewrite rules of the grammar

$$L(G) = \{w \in T^* / s \xrightarrow{*} w\}.$$

From a musical point of view, this rewrite rule mechanism is well adapted to the description of harmonic substitutions. To illustrate this point, consider the following harmonic loop, taken from a house record entitled *Rain* by Kerri Chandler, a DJ from New York [5]. Notice that the three first bars are very similar to bars 4-6 of the well-known standard *Round Midnight* by Thelonious Monk.

Bm7 E7	Abm7	Gb7M	B7
Bbm7 Eb7	Db7	Ab7	Bb7

One can apply the following substitution rule to it

Gb7M Ab7 $\longrightarrow$ Ebm7 Ab7

which gives

Bm7 E7	Abm7	Ebm7	B7
Bbm7 Eb7	Db7	Ab7	Bb7

This short chord sequence has been for many months the working example of our experimentations on improvisation with the computer, done in collaboration with Gérard Assayag and Carlos Agon. Our system is based on the communication between two software environments developed at Ircam. The first one is Open Music, a musical Lisp environment created by Gérard Assayag and his team, which is well suited for implementing formal systems such as Steedman's grammar. The other one is Max/MSP, a system for real-time musical interaction. We use Max/MSP as an interface for our improvisation system, and it sends requests to Open Music in order to get new musical sequences to be played in real-time. Some musical examples generated by the system are available on the web (in mp3 and midi format):
www.info.unicaen.fr/~marc/publi/grammaires/grammars.html

In real-time applications where musical objects are supposed to be generated on the fly, the rewrite rule mechanism has a drastic limitation. In fact, the effect of the rewrite process on a given symbol inside a sequence may depends on other symbols of this sequence that are very far away from it. To make it more precise, consider the rewrite rule $xz \longrightarrow zz$, and the sequence *xyxxxz*. The only possible successor to y in this sequence is x

$$succ(y) = \{x\}.$$

If we apply the rewrite rule several times, does it change the number of possible successors to y? And if so, how far in the sequence are the symbols responsible for this change? In the initial sequence, one can apply the rule only once, at the end of the sequence, which gives *xyxxzz*, and it has no effect on the possible successors of y. On the contrary, if we apply the rule two more times, we obtain *xyzzzz*, which introduces a new possible successor z to y

$$succ(y) = \{x, z\}.$$

This means that in general, we cannot predict the possible successors of a given symbol until we have tried to apply the rules everywhere in the sequence, including its end.

We have tried to solve this problem by precompiling the set of chord sequences generated by the grammar. This set was represented in lexicographic order as a tree, each node corresponding to a subset of sequences sharing the same prefix. Thus the leaves coming out of one node gave the possible successors for the path going from the root of the tree to this node. From a theoretical point of view, this tree was an "automaton" recognizing the set of substituted chord sequences. It was precompiled *before* the real-time improvisation process begins. In this way, it became possible to predict in real-time the possible successors for a given chord in the sequence.

Carlos Agon designed an interface for this earlier version of the system (presented at the JIM 2001 in Bourges [6]). The reference chord sequence is shown on top of the window Fig. 1. The sequence which is currently played is shown in the middle. On the bottom of the window, there are three circles representing all the available chords (major seventh, minor seventh, and dominant seventh). The possible successors for the current chord were indicated in real-time to the user by lights on the circles (in Fig. 1 for instance, F#7M is the only possible successor to C#7). The user had the possibility to click on these circles to constrain the choice of the remaining chord sequence to be played.

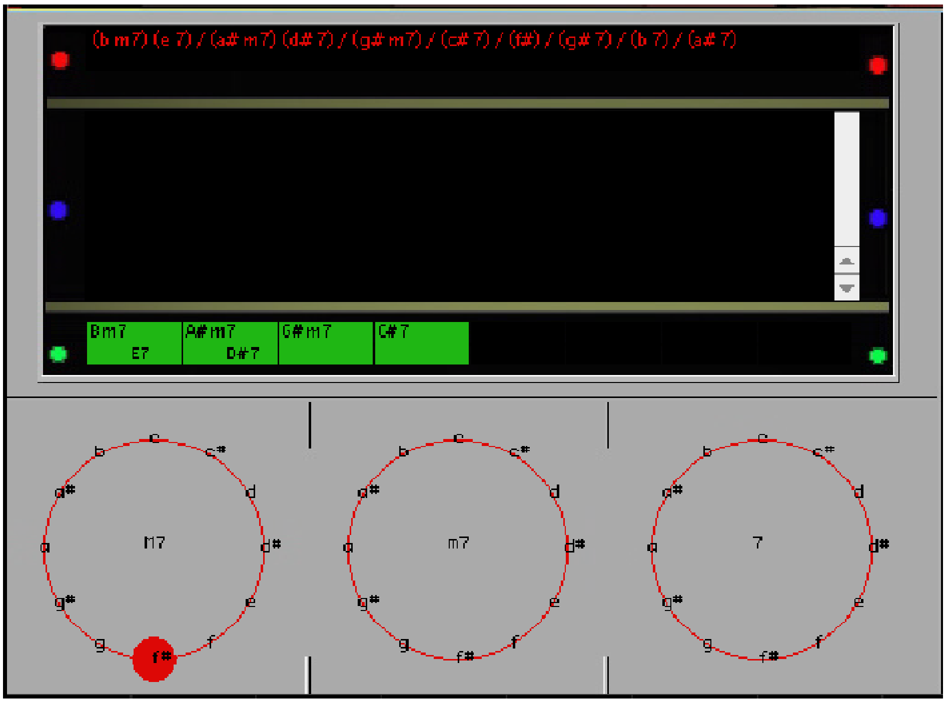

Fig. 1. Interface for choosing the next chord to be played (by Carlos Agon)

This solution was acceptable for substitutions on a short chord sequence, because in this case it was possible to compute all the sequences generated by Steedman's grammar. But such a method was no longer practicable for longer chord sequences, the number of generated sequences being too big.

In this paper, we shall see that instead of precompiling the whole set of sequences obtained by applying Steedman's grammar to a given chord sequence, one could only precompile some *particular* "cadential sequences". The idea is to look more closely at the specific form of the rewrite rules introduced by Steedman, and to make a formal study of the language of sequences generated by these rules. As we shall see, it will make more explicit the role of the cadential sequences in the substitution process.

3
Steedman's substitution rules

Steedman's grammar is obtained empirically from a set of modern jazz 12-bar chord sequences adapted from a book by Coker [4]. This set is considered as a wide and representative range of permissible variations of the blues basic form ([17] page 54). Steedman's study attempts to provide a small set of rules that characterize this musical "sublanguage", and he has founded that six rules could achieve this goal.

Some of these rules are less important than the others, and we shall not take all of them into account. The last rule, numbered "Rule 6" by Steedman, is dedicated to the introduction of a particular type of passing chord named "diminished seventh chord". We shall ignore this rule since we do not include diminished seventh chord in our system.

The rule numbered "Rule 5" provides a special way of elaborating a chord that is not part of a dominant cadence, which can be found only once in the set of blues chord sequences taken from Coker (Example (d) in Steedman's table page 54). This rule seems to be *ad hoc*, and we shall ignore it.

One more rule, numbered "Rule 2", is ignored in our system. It contains a substitution of a chord by its subdominant. Steedman points out that this rule must be used carefully. The replacement of a chord by its subdominant must occur on a right branch of the metric tree of the sequence (which means a chord which *is not* metrically stressed). Otherwise,

> "since the leftmost branches of the hierarchy are the metrically stressed ones, such a substitution [would] changes the harmonic character of the sequence" ([17] page 61).

Thus it remains three rules in Steedman's grammar (Rules 1, 3, 4) that we have implemented in our improvisation system.

Rule 1:	$x \rightarrow x\ x$
	$x7 \rightarrow x\ x7$
	$xm7 \rightarrow x\ xm7$
Rule 3a:	$w\ x7 \rightarrow Dx7\ x7$
	$w\ x7 \rightarrow Dxm7\ x7$
Rule 3b:	$w\ xm7 \rightarrow Dx7\ xm7$
Rule 4:	$Dx7\ x7 \rightarrow Stbx7\ x7$
	$Dx7\ x \rightarrow Stbx\ x$
	$Dx7\ xm7 \rightarrow Stbxm7\ xm7$

These rules are to be read as follows: *x* is a variable over the set of chord roots. We use a common notation that conveniently reflects chord relationships independently of any particular key. It is based on Roman numerals to represent chords, where the numeral identifies the degree of the chord root in the scale. The range of numerals is shown on Fig. 2. For instance, in the key of C major the chords corresponding to I, IV and V are respectively C, F and G.

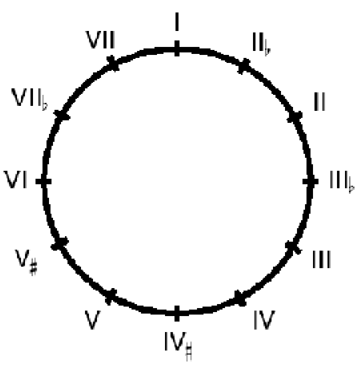

Fig. 2. Degrees of the chord root

The symbols *Dx* and *Stbx* stand for the chords with respectively the dominant and the flattened supertonic of *x* as its root (for instance *Dx* = VI and *Stbx* = IIIb when *x* = II).

In addition to the Roman numerals indicating the degree of the chord relative to the key note, the notation uses a system of suffixes, which specify certain additional notes that are played with the notes of the root chord. The suffix 7 means that the "dominant" seventh note, a tone below the root, is to be included. It is understood to be based on the major chord, which gives a "dominant seventh chord", as in V7, unless explicit indication is given that it is based on the minor by a small m immediately following the Roman numeral, as in Vm7 ([17] page 76).

Every chord is considered as nonterminal. Each rule is translated into twelve different rules by giving a degree (from the circle Fig. 2) as value to the variable *x* (and updating the corresponding values of *Dx* and *Stbx*). An axiom *s* is added to the grammar, together with a rule that instantiates the reference chord sequence on which the substitutions are applied. For instance, the axiom and the additional rule associated with the blues are defined by Steedman as follows ([17] page 61):

$s \rightarrow$ I I7 IV I V7 I

There is a convention in all Steedman's rules that the things on the right occupy the same total amount of time as the things on the left. Rules 3 and 4 have the same number of chords on each side, so that the corresponding chords have the same duration on both sides. But in Rule 1, a chord can be replaced by two copies of itself, each lasting half as long (the six basic chords of the blues above can thus be expanded into twelve chords, so that each of them occupies exactly one bar). Notice that the two copies are not identical. If the original chord was a dominant seventh chord, or a minor seventh chord, then the rightmost of its offspring is too, but not the other ([17] page 62). It is important to point out that the time division process implied by Rule 1 cannot be repeated endlessly, since chord sequences generally do not include chords with a duration shorter than a quarter of a bar.

The terminal symbols of the grammar are copies of the chords themselves (recall that the sets of terminal and nonterminal symbols are supposed to be disjoint), and there is an implicit convention that at the end of the substitution process, each chord is derived into a terminal copy of itself. Steedman introduces optional rules to add sixths, ninths, and so on to these terminal copies of the chords ([17] page 69).

The dominant seventh chords have a specific harmonic function which is central in Steedman's approach, so that his statements of the rules are very careful about these chords:

> "One of the suffixes is particularly important for the rules that follow. The suffix 7 [...] means that the note a flattened or "dominant" seventh above the root is played with the other notes of the chord. [...] This modification has the effect of making the listener expect a further chord related to the dominant seventh chord by having its root a fifth below or a fourth above" ([17] page 58).

In Rule 3 above, the symbol *w* is a variable over chords, but there are stringent restrictions upon the chord that *w* may match:

(1) *w* cannot match a dominant seventh chord,
(2) *w* cannot match a chord that has had its root changed by the previous application of a substitution rule.

As Steedman says,

> "The restriction of *w* to non-dominant seventh chords and to chords whose root has not been changed by a previous application of a substitution rule [...] prohibits a lot of ill-formed sequences that would otherwise arise" ([17] page 71).

There is a subtle distinction made by Steedman between the dominant seventh chord, and a major chord which is played with what could be called a "bluesy feeling" by adding the dominant seventh note to it. In the latter one, the seventh is just an optional note added to a terminal copy of the major chord. Steedman distinguished carefully these chords denoted by *x*7' from the real dominant seventh chord *x*7.

"There are *two* distinct harmonic functions that can be performed by the chord which on the keyboard is played with a note a semitone below the seventh of the key note. Besides the "leading" dominant seventh function just defined, it may have the function of a minor seventh chord, which does not lead anywhere in particular. [...] In the standard notation both are, therefore, written with the suffix 7. Since the present rules treat these two "homophones" differently, the nonliving chords with the minor seventh are distinguished [...] with the nonstandard suffix 7'" ([17] page 58, note 7).

This remark is quite important since the dominant seventh chords are the foundations of the substitution process described by Steedman, which can be viewed as "extending the authentic cadence". In fact, Rule 3a

"has the effect of "extending" an authentic cadence backward in the sequence" ([17] page 63).

As we shall see, these dominant seventh chords are the key of the formal property studied in the next section.

4
The language generated by Steedman's grammar

If we denote by *d* a dominant seventh chord, and by *u* a chord sequence which *does not* include any dominant seventh chord, the problem addressed in this section is to study the sequences that can be derived from *ud*. The surprising fact that we shall point out is that in some sense, *these sequences only depend on d, and do not depend on u*.

To make it clearer, we represent in Fig. 3 the derivation of the sequence *ud*. The derivation process described by Steedman as an "extension of the authentic cadence" generates a new sequence *z* which replaces a suffix of *ud* (denoted by *u"d*), and which does not depend on the specific content of *u"*. In this newly generated sequence *vz*, there may appear a surprising transition, from a musical point of view, between the end of *v* and the beginning of *z*, but this effect is attenuated by the fact that afterward, there is a logical progression through *z* leading to its final chord, which is the dominant seventh chord *d* (the sequence *z* being what we call a "cadential sequence", resulting from the extension of the authentic cadence). The fact that *z* does not depend on *u"* is related to the cumulative use of variable *w* introduced in Steedman's Rule 3.

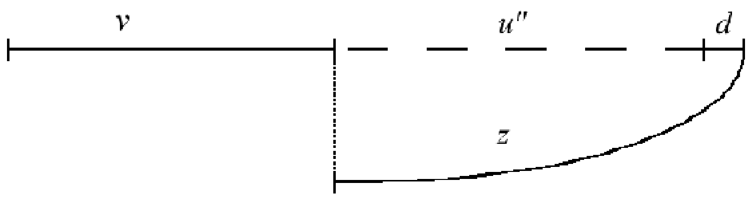

Fig. 3. The derivation from *ud* into *vz* does not depend on a suffix *u"* of *u*

We shall give a more formal statement to this simple property. But to make the things easier, we shall restrict ourselves to major chords and dominant seventh chords, and their associated rules:

Rule 1: $x \rightarrow x\ x$

$x7 \rightarrow x\ x7$

Rule 3a: $w\ x7 \rightarrow Dx7\ x7$

Rule 4: $Dx7\ x7 \rightarrow Stbx7\ x7$

$Dx7\ x \rightarrow Stbx\ x$

thus excluding the minor seventh chords $xm7$ from our statement.

We denote by A the set of all chords (x or $x7$), and by D the subset of dominant seventh chords (denoted by $x7$). The three rules of Steedman's grammar are denoted by R_1, R_3, R_4. If u is a sequence of A^*, we denote by $R(u)$ the set of sequences that can be derived from u by applying any of the three rules R_1, R_3, R_4, and we denote by $R_1(u)$ the set of sequences that can be derived from u by applying rule R_1.

Lemma 4.1 *If $y \in R(x)$ and $z \in R(y)$, then $z \in R(x)$.*

Lemma 4.2 *If a sequence contains no dominant seventh chord, the application of R_1 to it cannot introduce any new dominant seventh chord.*

Proposition 4.3 *If d is a dominant seventh chord, $d \in D$, and u a chord sequence with no dominant seventh chords, $u \in (A \backslash D)^*$, then for any sequence y derived from ud, $y \in R(ud)$, there exists a factorization $u = u'u''$ such that $y \in R_1(u')R(d)$, so that y does not depend on u''.*

Proof. By induction on the length p of the derivation of ud, assume that ud is derived into vz satisfying $v \in R_1(u')$, $z \in R(d)$. We now consider an additional rule applied to vz which gives y

$$ud \xrightarrow{p-1} vz \longrightarrow y$$

There are three possible ways of applying the last rule to vz.

(i) First, the rule is applied to the left factor v. By Lemma 4.2, $R_1(u')$ is included in $(A \backslash D)^*$, thus v contains no dominant seventh chord, which implies that R_1 is the only rule which can be applied to v. It follows that $y = v'z$ with $v' \in R_1(u')$.

(ii) Assume now that the last rule is applied to the right factor z. Then $y = vz'$ with $z' \in R(z)$, thus by Lemma 4.1 $z' \in R(d)$.

(iii) Finally, assume that $v = v'a$ and $z = bz'$, with the rule being applied to ab. This rule cannot be R_1, which has only one symbol in its left part. Since v contains no dominant seventh chord, then $a \notin D$. It follows that the rule cannot be R_4, and thus must be R_3, and we shall denote it by

$$R_3 : ab \rightarrow cb.$$

The derivation from vz into y can be written $v'abz' \rightarrow v'cbz'$.

Since $a \notin D$, then a must be a major chord, so that the application of R_1 to it is just a duplication

$$R_1 : a \rightarrow aa.$$

It follows that the last symbol of u' is a, with $u' = u'_1a$. If the derivation from u'_1a into $v'a$ does not involve the last symbol, then $v' \in R_1(u'_1)$. If it does, then a has been duplicated, so that v' ends with the symbol a, and in this case $v' \in R_1(u')$. In both cases, v' is derived from a prefix of u (whatever it be u' or u'_1).

On the other side, the symbol b can be derived

$$R_1 : b \rightarrow b'b,$$
$$R_3 : b'b \rightarrow cb,$$

where b' is the major chord with the same root as b. Thus $cbz' \in R(bz')$, and since by induction $bz' = z \in R(d)$, it follows from Lemma 4.1 that $cbz' \in R(d)$.

Finally, since y is factorized into $y = (v')(cbz')$, one has either $y \in R_1(u')R(d)$ or $y \in R_1(u'_1)R(d)$, which achieves the proof.

This Proposition 4.3 could be the basis of further investigations on the form of the chord sequences generated by Steedman's grammar. This simple proposition is in some sense general, since for any sequence u, one can factorize it into

$$u = u_1d_1u_2d_2...u_kd_ku_{k+1}$$

such that each d_i is a dominant seventh chord, and each u_i does not include any dominant seventh chord (with perhaps some u_i being empty). This means that any chord sequence can be factorized into sequences satisfying the conditions of Proposition 4.3.

The fact that the sequence z in the derivation of ud into vz only depends on d (and does not depend on the suffix u'' of u), as shown in Proposition 4.3, means that one could make a *precomputation* of such "cadential sequences" (the variable d ranging over the twelve dominant seventh chords). This could be the first step in the direction of a new implementation of the derivation process, where the chord sequence would not be treated as a whole, but substitutions rules could be applied locally, to small parts of it, in order to make the process more adapted to a real-time situation.

Aknowledgements

I thank Mark Steedman for his comments on this text.

References

1. Assayag G, Dubnov S, Delerue O (1999) Guessing the Composer's Mind: Applying Universal Prediction to Musical Style. Proc. of the ICMC 99, Beijing, China, ICMA, San-Francisco

2. Baroni M, Callegari L (1984) Musical grammars and computer analysis. Leo S. Olschki

3. Berstel J (1979) Transductions and context-free languages. Teubner

4. Coker J (1987) Improvising Jazz. 1964 1st ed. Fireside

5. Chandler K (2000) The Mood EP, Nervous Records, NY

6. Chemillier M (2001) Improviser des séquences d'accords de jazz avec des grammaires formelles. JIM 2001, 8èmes journées d'informatique musicale, IMEB, Bourges 121-126.

7. Chemillier M (2004) Grammaires, automates et musique, in: Briot J.-P., Pachet F. (eds.), Informatique musicale, Hermès, 195-230

8. Johnson-Laird P (1991) Jazz Improvisation: A Theory at the Computational Level, in: Cross I., Howell P., West R. (eds.), Representing Musical Structure, Academic Press, 291-326

9. Marsden A, Pople A (1992) Computer Representations and Models in Music. London, Academic Press

10. Nattiez J-J (1975) Fondements d'une sémiologie de la musique. 10/18, Christian Bourgois

11. Pachet F (1998) Computer Analysis of Jazz Chord Sequences: Is Solar a Blues?, in: Miranda E.R. (ed.), Readings in Music and AI, Harwood

12. Pachet F, Carrive J (1998) Intervalles temporels circulaires et application à l'analyse harmonique, in: Chemillier M., Pachet F. (eds.), Recherches et applications en informatique musicale, Hermès, 17-30

13. Pachet F (1998) Sur la structure algébrique des séquences d'accords de jazz. JIM 98, 5èmes journées d'informatique musicale, LMA Marseille, La Londe-les-Maures

14. Pachet F (1999) Surprising Harmonies. JIM 99, 6èmes journées d'informatique musicale, CNET-CEMAMu, Issy-les-Moulineaux, 187-206
15. Roads C (1979) Grammars as Representations for Music. Computer Music Journal **3:1** 48-55, reprint in: Roads C, Strawn J (eds.), Foundations of Computer Music. MIT Press, 1985
16. Schwanauer S, Levitt D (1993) Machine Models of Music. MIT Press
17. Steedman M J (1984) A Generative Grammar for Jazz Chord Sequences. Music Perception **2:1** 52-77
18. Steedman M J (1996) The Blues and the Abstract Truth: Music and Mentals Models, in: Garnham A., Oakhill J. (eds.), Mentals Models in Cognitive Science. Psychology Press, Hove, 305-318