Toward a theory of formal musical languages

Marc Chemillier
LITP
Université de Paris 7
4, Place Jussieu,
75005 Paris FRANCE

Dan Timis
Computer Music Studio
Department of Music
University of California
Santa Barbara CA 93106 USA

Abstract

The idea that algebraic operations are involved in musical combinatorics is generally accepted. While many interesting computer programs has been developed lately in the fields of computer aided composition, musical analysis, musical pedagogy, or score processing there is no general mathematical model able to deal with the complex interaction between the vertical and the horizontal aspects of the musical syntax. In this paper we are proposing an extension of the traditional theory of formal languages by using sequences in which the usual letters are replaced by ***sets of musical events****. Besides the* ***concatenation****, two new operations are introduced: the* ***superimposition*** *and the* ***interimposition****. The resulting mathematical structure is then studied. All the advances in the domains of finite automata, generative grammars and other computational models may be applied in a natural and consistent manner and new solutions to the problems that appear in the previously mentioned fields could be explored.*

Introduction

"Can techniques from linguistics be usefully applied to the study of music?" is one of the questions Curtis Roads poses in his article "Grammars as Representation for Music" [Roads 1979]. One might indeed explore the parallels between music and the spoken language; then, linguistic theories could be applied to music and vice versa, but how to use music theories to explain linguistic phenomena has, to our knowledge, never been systematically investigated. A possible explanation of this one-way situation, would be that linguistics is a broader science which covers totally or partially music phenomena. We believe that this is not the case.

If one wishes to develop new music theories using formal techniques borrowed from other domains, one could seek assistance from mathematics, specifically from the theory of formal languages which is, in our opinion, able to embrace among other subjects, both languages, the spoken one and the musical one. This theory which took rapid advances during the last thirty years, has strong foundations and has shown interesting mathematical properties. Some important aspects of the theory, generative grammars, developed around the work of linguists such as Chomsky, were introduced by the necessity to effectively explain linguistic phenomena, and are now extensively used in the field of programming languages. While in the musical field, no new theory of equal originality and power has been developed lately, musicians are naturally attracted by the powerful linguistic model of generative grammars.

Formal techniques applied to music are not really new. Early in the sixties, Xenakis proposed an important distinction between two aspects of music: *in-time* and *outside-time* [Xenakis 1971]. Pitches and durations are well suited for outside-time formal manipulations. One of the contributions in this domain is the well established and widely accepted pitch-class sets theory proposed by Milton Babit, and later developed by Forte [Forte 1973] and Rahn [Rahn 1980]. On the other hand, in the in-time domain, no theory of comparable importance has been introduced since Schenker. Several researchers attempted the use of linguistic theories in the musical field. We refer the reader to [Roads 1979] for details about the work of Ruwet, Nattiez, Laske, Smoliar, Moorer, Winograd, Roads and Lerdahl-Jackendoff concerning generative grammars and music.

We believe that between the spoken and the musical languages there are enough differences to make the application of linguistic theories to the musical domain difficult and unnatural. On the other hand, the same abstract theories that were partially used in the study of natural languages, could be extended, adapted, and applied to the musical field. Paraphrasing Curtis Roads' question, one might ask: "Can formal mathematical techniques be usefully applied to the study of music, in a similar way they were applied to linguistics?". We would like to bring a positive answer to this question.

Spoken Language and Musical Language

There are many similarities between speech and music but numerous differences too. We will briefly mention only a few of them. One could notice that both languages have a sonic and a written form, and for both, the order in which sonic events follow one another in time is essential. Among the dissimilarities we may look at the way meanings are linked to the elements of the discourse. Most of the time each word of the vocabulary of a spoken language is linked to a particular concept (although sometimes the meaning may change in a special context). Musical relationships (intervals, chords, etc...) among different elements (pitches, durations, etc.) are in western music, more important than the elements themselves. This explains why so many theorists studied the outside-time organization of the musical material.

The two aspects that differentiate music and spoken languages we would like to emphasize in this article are simultaneousness and parallelism. Elementary objects of the spoken language follow one after another in time, *successor* and *predecessor* being the basic relationships between words. Unlike them, musical sounds may be *simultaneous* or *overlap*, musical sequences may be superimposed or evolve independently. Especially in our western tradition, where the harmonic and polyphonic sides of music are essential, a theory based mostly or exclusively on relationships of the successor/predecessor type, is inadequate. Unfortunately, linguistics does not cover any simultaneous or parallel phenomena.

Researchers applying generative grammars to music, took more or less simultaneousness and parallelism into account. Holtzman in his description of the Trio from Schoenberg's Suite für Klavier op. 25 [Holtzman 1981], produces each of the two voices of the canon separately using two different grammars, one for the sequence of pitches, and one for the sequence of durations.

Lerdahl and Jackendoff believe that "although it is possible in principle to extend the theory to simultaneous multiple descriptions, the formal complications would be so enormous that they would obscure the presentation of other, perhaps more fundamental aspects of musical structure" [Lerdahl 1983, page 116]. First, we would like to argue that simultaneousness and parallelism are two of the most fundamental aspects of western music; and second, we believe that the role of a formal theory is to simplify our understanding of the studied phenomenon and not to complicate it.

Juxtaposition and Superimposition

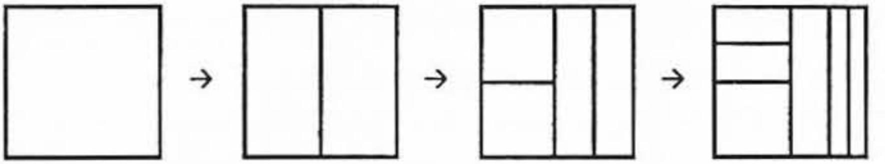
In an article concerned mostly with stochastic processes, Kevin Jones introduces what he calls *space grammars* [Jones 1981]. The usual production rules are used to divide the *time-space*. But *multi-dimensional spaces* may be divided using special production rules for each dimension, using the symbol $/_n$, indicating a split in the nth dimension. The following example is given for a two-dimensional space grammar:

$$A \rightarrow A /_1 A \quad (1) \qquad A \rightarrow A /_2 A \quad (2) \qquad A \rightarrow a \quad (3)$$

A possible derivation is presented:

$$A \quad \rightarrow \quad A\ A \quad (1) \quad \rightarrow \quad \begin{bmatrix} A \\ A \end{bmatrix} A\ A \quad (2),(1) \quad \rightarrow \quad \begin{bmatrix} \begin{bmatrix} A \\ A \end{bmatrix} \\ a \end{bmatrix} a\ A\ A \quad (2),(3),(3),(1) \quad \rightarrow \text{etc.}$$

While extending grammars to the vertical domain is an original and very valuable idea, we do not agree completely with Kevin Jones' geometrical interpretation of the space-grammars in which time-space and pitch-space are treated as if they were equivalent. The first four steps of the previous example would give:



where the horizontal axis represents time-space and the vertical axis pitch-space. While time may be considered as a continuous space which can be divided into what we usually call durations, pitches belong to a discrete space where sets theory type manipulations are more appropriate.

It seems to us that an extension of the theory of formal languages to include simultaneous and parallel phenomena, could significantly advance our insights into the in-time functioning of the musical language. We consider it important that the horizontal and the vertical aspects of music be seen as related, although each of them obey to different laws. Traditionally the theory of formal languages manipulates sequences of symbols that can be juxtaposed using the concatenation operation. What is needed to include the vertical aspect, is a new operation that will allow us to superimpose musical events, or any other complex musical structures.

Formal Events

Any simple or complex musical phenomenon may be considered as an event. Notes are the most usual example, but parts of a note (like the attack, the decay, or the beginning of a vibrato), chords, melodic fragments, or modulations may also be treated as musical events. When the only way to distinguish two events is through the time when they occur, we will use the same symbol for both events (they are identical without regard to time). Let a and b be two distinguished events, a occurring before b may be expressed by *concatenating* (juxtaposing) the two symbols, leading to the sequence ab. It is obvious that the sequence ba has a different meaning (b occurs before a). It is also obvious that a occurring before b and both occurring before c, is equivalent to b occurring before c and a occurring before both of them. We say that the concatenation is associative but it is not commutative.

On the other hand, a occurring at the same time as b is equivalent to b occurring at the same time as a. Moreover, a occurring at the same time as b and both occurring at the same time as c is equivalent to b occurring at the same time as c and both occurring at the same time as a, while two or more a events occurring at the same time are indistinguishable by definition. We will call the operation by which we make two or more events occur simultaneously *superimposition*. One may say that the superimposition is commutative, associative, and has the idempotent property like the union operation in the sets theory. It is thus convenient to represent events that occur simultaneously as a set. Superimposing two sets of events is equivalent to making the union of the two sets, while juxtaposing sets of events may be done by concatenating sets of events instead of elementary symbols. We form then, sequences of sets of events.

Moreover we are able to concatenate sequences of sets of events to form other sequences. But how do we superimpose two sequences? Under the assumption that the sequence of times where each event of a sequence occurs, is the same for both of the sequences of sets of events, we may simply superimpose one by one the sets from one sequence with the corresponding sets of the second one, and concatenate the resulting sets. Then the new sequence may be again concatenated with, and superimposed on other sequences resulting in more and more complex structures, all represented in the same simple manner, that is, by a sequence of sets of events.

The superimposition will be denoted $\|$ and the concatenation $\cdot$. If we superimpose a set of two events $\{a, b\}$ with another set of two events $\{a, c\}$ we will obtain the set $\{a, b, c\}$:

$$\{a, b\} \| \{a, c\} = \{a, b, c\}$$

Concatenating the two sets will result in:

$$\{a, b\} \cdot \{a, c\} = \{a, b\}\{a, c\}$$

the superimposition of two sequences of sets of events:

$$\{a, b\}\{a, c\}\{b\} \| \{a, c\}\{a\}\{c\} = (\{a, b\} \| \{a, c\}) \cdot (\{a, c\} \| \{a\}) \cdot (\{b\} \| \{c\}) = \{a, b, c\}\{a, c\}\{b, c\}$$

Properties of operations dealing with the horizontal and vertical aspects of music, similar to the superimposition and the concatenation described here, were already mentioned by other researchers. Xenakis [Xenakis 1971] remarked the idempotence property, the commutativity, and the associativity of what he calls *juxtaposition in the outside-time domain*. He also noticed that *juxtaposition in the in-time domain* is not commutative.

Mira Balaban [Balaban 1984], talks about *musical concatenation* (denoted $\|_q$ where q is a time unit). Two particular cases are called the *vertical concatenation* (denoted |) which is equivalent to what we call superimposition, and the *horizontal concatenation* (denoted —) which is similar and has identical properties with what we call concatenation (juxtaposition). Unlike her, we only deal with time simultaneity and succession without considering the duration of the events (which however, may be expressed, as we will see in a further section). This enables us to build a system which is more coherent according to a mathematical point of view, and leads to a consistent algebraic structure in which every application of the superimposition or of the concatenation operations, always produces the same kind of object, that is, a sequence of sets of events.

Formal Musical Languages

The theory of formal languages manipulates symbols called ***letters***, out of a set called ***alphabet***. Let A be an alphabet; a sequence of letters from A is called a ***word*** over A. Two words may be juxtaposed to form a new word; this operation is called ***concatenation***. The number of letters in a word u, denoted $|u|$, is called its ***length***. A special word of length 0 and denoted ε, is called the ***empty word***. The set of all words over A, including ε, is denoted A^*. Thus, if $A = \{a, b\}$, then $A^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \ldots\}$. The length of the concatenation of two words u and v, is: $|uv| = |u| + |v|$. The concatenation is associative (1) and ε is neutral to the operation of concatenation (2):

$$(uv)w = u(vw) \qquad (1) \qquad\qquad \varepsilon u = u\varepsilon = u \qquad (2)$$

We may remark that it is not commutative ($uv \neq vu$).

The concatenation UV of two sets of words U and V, of A^*, is the set of all words obtained by concatenating a word from U with a word from V. Thus, if $U = \{aa, bb\}$ and $V = \{ab, ba\}$, then $UV = \{aaab, aaba, bbab, bbba\}$. The same set U, concatenated with itself n times, is denoted U^n, where $U^0 = \{\varepsilon\}$. The ***closure*** of a set U, denoted U^*, is the union of the sets $U^0, U^1, U^2, U^3 \ldots$ Thus, A^* may be defined also as follows: $A^* = A^0 \cup A^1 \cup A^2 \cup A^3 \cup \ldots$ A subset of words from A^*, denoted L, is called a ***language*** over A^*. We say that $(A^*, \cdot)$, that is, the set A^* together with the operation of concatenation, form a mathematical structure called *free monoid*.

In the theory of formal musical languages that we are trying to define here, instead of concatenating letters, we concatenate *sets of events*. Thus, if we have a set of two *elementary events* $A = \{a, b\}$, let $P(A)$ be the set of all events that will possibly occur at the same time, that is, all the subsets of A. $P(A) = \{\phi, \{a\}, \{b\}, \{a, b\}\}$ where ϕ is the empty set $\{\}$, that is, the set with no elements. For convenience we will sometimes use a different symbol for each element of $P(A)$. The set of all *compound events* in our case is $E = P(A) = \{\phi, e_1, e_2, e_3\}$ where $e_1 = \{a\}$, $e_2 = \{b\}$, and $e_3 = \{a, b\}$. The set E may be considered an alphabet and then, $E^* = \{\varepsilon, \phi, e_1, e_2, e_3, \phi\phi, \phi e_1, \phi e_2, \phi e_3, e_1\phi, e_1e_1, e_1e_2 \ldots\}$. One should notice that the empty word ε, and the empty set ϕ, are not equivalent The first has length 0 and is neutral for the concatenation, the second has 0 elements, but length 1, and is neutral for the union operation in E. We will denote words of E^*, the same way as words of A^*, that is $u, v, w, \ldots$, but call them *musical sequences*. Musical sequences may be concatenated to form other musical sequences, and it is obvious that $(E^*, \cdot)$ is a free monoid.

We define the *superimposition* denoted $\|$, of two musical sequences, u and v, as follows:

1. $u \| \varepsilon = \varepsilon \| u = u$ (the empty word is neutral for the superimposition)
2. if $u = e_iu'$ and $v = e_jv'$ $(e_i, e_j \in E)$ then $u \| v = e_iu' \| e_jv' = (e_i \cup e_j) \cdot (u' \| v')$

For example:

$$e_1\phi e_2e_3 \| e_2e_1e_2 = (e_1 \cup e_2) \cdot (\phi e_2e_3 \| e_1e_2) = e_3 \cdot (\phi \cup e_1) \cdot (e_2e_3 \| e_2)$$
$$= e_3 \cdot e_1 \cdot (e_2 \cup e_2) \cdot (e_3 \| \varepsilon) = e_3e_1e_2e_3$$

that is: $\{a\}\{\}\{b\}\{a, b\} \| \{b\}\{a\}\{b\} = \{a, b\}\{a\}\{b\}\{a, b\}$

A symmetrical operation, denoted $\perp$, which we will call *interimposition*, may be obtained by replacing the union with the intersection in 2.

1. $u \perp \varepsilon = \varepsilon \perp u = \varepsilon$ (the empty word is absorbing for the interimposition)
2. if $u = e_i u'$ and $v = e_j v'$ $(e_i, e_j \in E)$ then $u \perp v = e_i u' \perp e_j v' = (e_i \cap e_j) \cdot (u' \perp v')$

For example:

$$e_3 \phi e_2 e_3 \perp e_1 e_1 e_2 = (e_3 \cap e_1) \cdot (\phi e_2 e_3 \perp e_1 e_2) = e_1 \cdot (\phi \cap e_1) \cdot (e_2 e_3 \perp e_2)$$
$$= e_1 \cdot \phi \cdot (e_2 \cap e_2) \cdot (e_3 \perp \varepsilon) = e_1 \phi e_2$$

that is: $\{a, b\}\{\}\{b\}\{a, b\} \perp \{a\}\{a\}\{b\} = \{a\}\{\}\{b\}$

One could notice that $|(u \parallel v)| = \max(|u|, |v|)$ and $|(u \perp v)| = \min(|u|, |v|)$.

Let M_1 and M_2 be two subsets of E^*, called *musical languages*. As with any two sets, we may apply the union and the intersection operation on M_1 and M_2, obtaining a new musical language. Like any language M_1 and M_2 may be concatenated:

$$M_1 M_2 = \{ w \mid w = uv, u \in M_1, v \in M_2 \}$$

We define the superimposition of two musical languages as follows:

$$M_1 \parallel M_2 = \{ w \mid w = u \parallel v, u \in M_1, v \in M_2 \}$$

The interimposition of two musical languages is:

$$M_1 \perp M_2 = \{ w \mid w = u \perp v, u \in M_1, v \in M_2 \}$$

The mathematical structure of solfege

We should mention that $(E^*, \parallel, \perp)$, is a distributive lattice. It is easy to verify the following properties:

1. $\parallel$ and $\perp$ are commutative:

   $u \parallel v = v \parallel u$ $\qquad$ $u \perp v = v \perp u$

2. $\parallel$ and $\perp$ are associative:

   $(u \parallel v) \parallel w = u \parallel (v \parallel w)$ $\qquad$ $(u \perp v) \perp w = u \perp (v \perp w)$

3. $\parallel$ and $\perp$ have the idempotent property:

   $u \parallel u = u$ $\qquad$ $u \perp u = u$

4. $\parallel$ and $\perp$ have the absorption property:

   $u \parallel (u \perp v) = u$ $\qquad$ $u \perp (u \parallel v) = u$

5. $\parallel$ and $\perp$ are distributive:

   $u \parallel (v \perp w) = (u \parallel v) \perp (u \parallel w)$ $\qquad$ $u \perp (v \parallel w) = (u \perp v) \parallel (u \perp w)$

Furthermore:

6. concatenation is left-distributive to $\parallel$ and $\perp$:

   $u \cdot (v \parallel w) = (u \cdot v) \parallel (u \cdot w)$ $\qquad$ $u \cdot (v \perp w) = (u \cdot v) \perp (u \cdot w)$

As already shown:

7. ε is neutral for $\parallel$:

   $u \parallel \varepsilon = \varepsilon \parallel u = u$

And according to the free monoid properties:

8. $\cdot$ is associative $\qquad$ 9. ε is neutral for $\cdot$

Let S be any set on which we define three operations: $\cdot$, $\|$, and $\perp$. Then, from the definition proposed in [Chemillier 1987], $(S, \cdot, \|, \perp)$ is a *solfege* if it verifies the following properties:

1. $(S, \cdot)$ is a monoid with a neutral element ε
2. $(S, \|, \perp)$ is a distributive lattice
3. $\cdot$ is left-distributive over $\|$ and $\perp$
4. ε is neutral for $\|$

$(E^*, \cdot, \|, \perp)$ has then, the mathematical structure of a solfege.

Homomorphisms

Let τ be an application from E^* to E^*. We may say that τ is an *homomorphism* of solfege if:

$\tau(u \cdot v) = \tau(u) \cdot \tau(v)$, $\quad \tau(u \| v) = \tau(u) \| \tau(v)$, $\quad \tau(u \perp v) = \tau(u) \perp \tau(v)$

Some of the usual transformations that musicians apply to musical sequences like transposition, rhythmical augmentation, etc. are homomorphisms. For instance, the inversion on $A = \{a, b\}$, could be defined as follows: $\tau(a) = b$ and $\tau(b) = a$. Then τ may be extended to be a homomorphism from E^* to E^*, in the following way (recall that $e_1 = \{a\}$, $e_2 = \{b\}$, and $e_3 = \{a, b\}$):

$\tau(\phi) = \phi$ because we need $\tau(e_1 \cap e_2) = \tau(e_1) \cap \tau(e_2) = e_2 \cap e_1 = \phi$

$\tau(e_1) = e_2$, $\tau(e_2) = e_1$, and

$\tau(e_3) = e_3$ because we need $\tau(e_1 \cup e_2) = \tau(e_1) \cup \tau(e_2) = e_2 \cup e_1 = e_3$

τ is then a homomorphism for $(E^*, \cdot, \|, \perp)$. For example:

$\tau(e_2 e_1 \cdot \phi e_3) = \tau(e_2 e_1) \cdot \tau(\phi e_3) = e_1 e_2 \phi e_3$

$\tau(e_2 e_1 \| \phi e_3) = \tau(e_2 e_1) \| \tau(\phi e_3) = e_1 e_3$

$\tau(e_2 e_1 \perp \phi e_3) = \tau(e_2 e_1) \perp \tau(\phi e_3) = \phi e_2$

Let δ be the retrograde function which could be defined in the following manner:

1. $\delta(\varepsilon) = \varepsilon$
2. if $u = e_i u'$ then $\delta(u) = \delta(u') e_i$

For instance:

$\delta(e_1 e_2 e_3) = \delta(e_2 e_3) e_1 = \delta(e_3) e_2 e_1 = \delta(\varepsilon) e_3 e_2 e_1 = e_3 e_2 e_1$

δ is not an homomorphism:

$\delta(e_1 e_2 \cdot e_3) \neq \delta(e_1 e_2) \cdot \delta(e_3)$ $\quad (\delta(e_1 e_2) \cdot \delta(e_3) = e_2 e_1 e_3)$

$\delta(e_1 e_2 \| e_3) \neq \delta(e_1 e_2) \| \delta(e_3)$ $\quad (\delta(e_1 e_2 \| e_3) = e_2 e_3$ and $\delta(e_1 e_2) \| \delta(e_3) = e_3 e_1)$

$\delta(e_1 e_2 \perp e_3) \neq \delta(e_1 e_2) \perp \delta(e_3)$ $\quad (\delta(e_1 e_2 \perp e_3) = e_1$ and $\delta(e_1 e_2) \perp \delta(e_3) = e_2)$

Not every mapping defined on A with values in E^* can be extended to a homomorphism from E^* to E^*. We need some additional conditions:

THEOREM: *For every mapping λ from A to E^*, when card A ≥ 2, there is an unique homomorphism from E^* to E^* which is an extension of λ,* ***if and only if****:*

(i) $\forall\, a, b \in A, \quad |\lambda(a)| = |\lambda(b)| = n$

(ii) $\exists\, u \in E^n, \quad \forall\, a, b \in A, \quad a \neq b \Rightarrow \lambda(a) \perp \lambda(b) = u.$

Let λ be the following mapping defined on $A = \{a, b, c\}$:

$\lambda(a) = bb, \quad \lambda(b) = cb, \quad \lambda(c) = ab \qquad$ (ab stands for $\{a\}\{b\}$)

We need:

$\lambda(\{a, b\}) = \lambda(a) \parallel \lambda(b) = \{b, c\}b, \qquad \lambda(\{a, c\}) = \lambda(a) \parallel \lambda(c) = \{a, b\}b,$

$\lambda(\{b, c\}) = \lambda(b) \parallel \lambda(c) = \{a, c\}b, \qquad \lambda(\{a, b, c\}) = \lambda(a) \parallel \lambda(b) \parallel \lambda(c) = \{a, b, c\}b$

We have:

$u = \lambda(a) \perp \lambda(b) = \lambda(a) \perp \lambda(c) = \lambda(b) \perp \lambda(c) = \phi b$

and we will pose:

$\lambda(\phi) = u = \phi b$

For instance we have:

$\{a, b\}\phi b = a\phi b \parallel b\phi b = \{a, b, c\}bb \perp \{a, b\}cb$

and we can verify that:

$\lambda(\{a, b\}\phi b) = \lambda(a\phi b) \parallel \lambda(b\phi b) = \lambda(\{a, b, c\}bb) \perp \lambda(\{a, b\}cb) = \{b, c\}b\phi bcb$

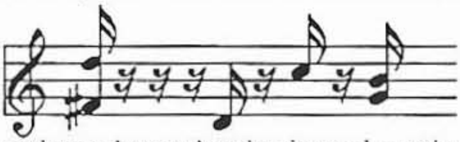
Time mapping of formal events

The beauty of any formal approach is that the same theory may be applied to different domains or in different manners to the same field. The more different ways to practically use a formal theory, the more powerful the theory is. Detached from any meaning, symbols may be manipulated much more easily, and interesting properties and relationships may appear. We would like to suggest that abstract manipulations of symbols, before any attempt to link them to real musical parameters, are more likely to guarantee the development of a consistent theory. These are the reasons why we chose to start with a pure formal approach, and only later to try to apply it to music.

When a formal theory is applied to phenomena from the real world, many ambiguities may occur. All we have now, is a set of sequences, that we call musical sequences, and several operations that may be applied to them. But how can a formal musical sequence, be related to a musical work or fragment? First we need to associate to each element of the sequence a time, when all the events in the set occur. Second, we need to define the exact meaning of each event.

Let u be a sequence of sets of events $s_1s_2s_3\ldots s_n$, and t be a sequence of time-increasing moments $t_1t_2t_3\ldots t_n$, ($t_i < t_{i+1}$). The application, associating each set s_i, to a moment t_i, is called *time-mapping*. It should be mentioned that u and t are discrete sequences and by consequence the mapping has not the same meaning as the one described in [Jaffe 1985]. However the elements of t come from a continuous domain that can be assimilated either with what David Jaffe is calling the *basic time* (the time of the score) or the *clock time* (the time of a real performance). In the first case, a second mapping from the basic time to the clock time is possible.

Thus, if $u = \{f\#, d5\}\{d\}\{c5\}\{b, g\}$, $t = (0, 1, 1.5, 2)$ (in seconds), we consider the events being 0.25 seconds long, and the duration of a quarter note is one second, we have the following fragment:



Now, of course, events do not have only very short durations and sometimes they overlap. One event may have a begin time, an ending time, and a duration. Any of these three parameters may be deduced from the other two. The *begin time - duration* combination is used in many computer-music programs derived from MUSIC V. While theoretically valid, the *end time - duration* approach is unnatural and though never or very rarely used. The MIDI specification with its *Note On / Note Off* messages, belongs to the *begin time - end time* category. To implement this last approach, we need to add to our set of events, a new event a', for each event a (a' meaning end of the event a). Let A' be the new set with twice the number of elements of A and E'^* the set of all sequences on $P(A')$. If now we have $u = \{f\#, d5\}\{f\#', d\}\{d5', c5\}\{c5', d', b, g\}\{b', g'\}$ and $t = (0, 1, 1.5, 2, 3)$, the previous example becomes:

It is sometimes convenient to consider t as a sequence of equally spaced moments called *ticks*. Now, we can see the usefulness of the empty set ϕ, with the meaning: no event occurs at the current tick. If the tick is equivalent to a sixteenth note, then for our example, we will have: $u = \{f\#, d5\}\phi\phi\phi\{f\#', d\}\phi\{d5', c5\}\phi\{c5', d', b, g\}\phi\phi\phi\{b', g'\}$.

The *begin time - duration* approach could be implemented by attaching to the symbol a', the meaning: the event a continues. Our example will be then: $u = \{f\#, d5\}\{f\#', d5'\}\{f\#', d5'\}\{f\#', d5'\}\{d, d5'\}\{d', d5'\}\{d', c5\}\{d', c5'\}\{b, g\}\{b', g'\}\{b', g'\}\{b', g'\}$. This solution is not so unnatural and useless as it seems to be. A way of thinking about how we perceive music at the fifth tick, could be that, we hear the new $d4$ in the same time noticing that the $d5$ continues and the $f\#$ from the previous tick is no longer present. We may write the example as follows:

Let $B = \{NoteIsOn, NoteIsOff\}$. A mapping from A to B is called the *state of a tick*. A function taking as arguments the state of the previous tick and the set of events at the current tick and returning the state of the current tick is called a *semantics*. Let $A = \{a, b, c\}$, $A' = \{a, b, c, a', b', c'\}$, δ be the semantics of the *begin time - end time* type, and γ be the one of the *begin time - duration* type. We may have then:

$\delta(\{(a, NoteIsOn), (b, NoteIsOn), (c, NoteIsOff)\}, \{a', c\}) = \{(a, NoteIsOff), (b, NoteIsOn), (c, NoteIsOn)\}$

that is, a' turns a to *Off*, b does not change, and c is turned to *On*

$\gamma(\{(a, NoteIsOn), (b, NoteIsOn), (c, NoteIsOff)\}, \{a', c\}) = \{(a, NoteIsOn), (b, NoteIsOff), (c, NoteIsOn)\}$

that is, a' continues a (stays *On*), b is not present (turns *Off*), and c is turned to *On*

Applications of the theory of formal musical languages

Chomsky's hierarchy of languages is applicable to formal musical languages too. One may speak then of *context-sensitive, context-free*, and *regular musical languages*. We have already proven some interesting properties of the regular musical languages that are partially exposed in [Chemillier 1987] and will be developed in future articles. The basic result concerning the regular musical languages is:

THEOREM: *The superimposition of two regular musical languages is a regular musical language. The same holds for the interimposition of two regular musical languages.*

We have developed an algorithm allowing the construction of a finite automaton, recognizing a regular musical language, resulting from the superimposition (interimposition) of the behaviors of two other finite automata. Simple musical rules in the metrical, rhythmical, melodic, or harmonic domains may be described by using regular expressions and finite automata. Complex systems for generating, recognizing, or transforming musical structures may be implemented by superimposing, interimposing, concatenating, or using the union, the intersection or the inversion of regular musical languages. Some of the concerned fields are: sophisticated sequencers, event list editors, automated musical analysis, or computer assisted composition.

The next step is to systematically explore the superimposition and interimposition of context-free and context-sensitive musical languages. We think that by including the vertical aspect, a new hierarchy of languages may appear. Further developments will include algorithms for complex pattern recognition, rewriting rules, combinatorics on words, and codes theory.

References and Bibliography

[Balaban 1980]	Balaban Mira, *Toward a general computer study of western tonal music*, State University of New-York at Albany, 1984.
	Balzano G. J., The group theoretical description of 12-fold and microtonal pitch systems, *Computer Music Journal*, vol 4, no. 4, 1980.
	Barbaud P., *La musique discipline scientifique*, Dunod, 1968.
	Chemillier M., Monoïde libre et musique (première partie): les musiciens ont-ils besoin des mathématiques? *RAIRO*, vol. 21, no. 3, mai 1987.
[Chemillier 1987]	Chemillier M., Monoïde libre et musique (seconde partie) *RAIRO*, vol. 21, no. 4, decembre 1987.
	Chemillier M., Timis D., *Toward a theory of formal musical languages*, Rapport LITP 88-29, Paris 1988
	Chemillier M., Solfege, commutation partielle et automate minimal d'intersection, to appear in *Mathématiques, informatique, et sciences humaines*, 1989
	Chomsky N., *L'analyse formelle des langues naturelles*, trad. Ph. Richard, N. Ruwet, Gauthier-Villars, 1968.
[Forte 1973]	Forte A., *The structure of atonal music*, Yale University, 1973.
	Greussay P., *Modèles de descriptions symboliques en analyse musicale*, Thèse, Université de Paris 8, 1973.
[Holzman 1981]	Holtzman S. R., Using generative grammars for composition, *Computer Music Journal*, vol. 5, no. 1, 1981.
[Jaffe 1985]	Jaffe D., Ensemble timing in computer music, *Computer Music Journal*, vol 9, no. 4, 1985.
[Jones 1981]	Jones K., Compositional applications of stochastic processes, *Computer Music Journal*, vol. 5, no. 2, 1881.
[Lerdahl 1983]	Lerdahl F., Jackendoff R., *A generative theory of tonal music*, MIT, 1983.
[Rahn 1980]	Rahn J., *Basic atonal theory*, Longman, 1980.
	Riotte A., *Formalisation des structures musicales*, Université de Paris 8, 1979.
	Roads C., *Composing grammars*, presented at the 1977 International Computer Music Conference, USCD, La Jolla.
[Roads 1979]	Roads C., Grammars as representations for music, *Computer Music Journal*, vol. 3, no. 1, 1979.
	Timis D., *Théorie des codes et analyse musicale*, Mémoire de DEA, Université de Paris 4, 1983.
[Xenakis 1971]	Xenakis I., *Formalized music*, Bloomington, Indiana University Press, 1971.