# Time and musical structures

Bernard Bel

▶ **To cite this version:**

Bernard Bel. Time and musical structures. Interface, Journal of New Music Research, 1990, 19 (2-3), pp.107-135. hal-00134160

**HAL Id: hal-00134160**
**https://hal.science/hal-00134160**

Submitted on 1 Mar 2007

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

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

groupe
représentation
et traitement
des
connaissances

*CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE*

*31, chemin Joseph Aiguier*

*F-13402 MARSEILLE CEDEX 9 (France)*

# Time and musical structures

***Bernard Bel***

***Abstract***

A theoretical model is introduced, by the aid of which descriptions of sequential and concurrent processes may be built taking account of the sophistication and generality of contemporary musical concepts. This is achieved through an independent and unrestricted mapping between physical time and a symbolic set of dates. Sequential structures are considered first, then the nature of this mapping and its practicle implementation in a real-time synthesizer are discussed. Polymetric structures are introduced and a method is outlined for mapping events to symbolic dates when concurrent processes are incompletely described.

***Keywords***

representation of time, concurrent processes, polyrhythm, computer music.

***GRTC / 364 / Novembre 1989***

*Bernard Bel*

*Forthcoming in Interface, vol.19, N°2-3, 1990, pp.113-42 (Ed. A. Camurri)*

# Time and musical structures

Bernard Bel

ABSTRACT

A theoretical model is introduced, by the aid of which descriptions of sequential and concurrent processes may be built taking account of the sophistication and generality of contemporary musical concepts. This is achieved through an independent and unrestricted mapping between physical time and a symbolic set of dates. Sequential structures are considered first, then the nature of this mapping and its practicle implementation in a real-time synthesizer are discussed. Polymetric structures are introduced and a method is outlined for mapping events to symbolic dates when concurrent processes are incompletely described.

*Keywords:* representation of time, concurrent processes, polyrhythm, computer music.

## 1. Introduction

Time is probably the most complex dimension of music, altogether one which has been neglected in many formal models of music representation. Even so, machines playing music from scores require complete information about time in order to compute durations and delays. Such a discrepancy between the amount of information available on scores and the accuracy required in performance has led to models of timing in computer music that are more based on ad-hoc procedures than on analytical/compositional musical concepts.

Ideas introduced here evolved from the need to formalize a model of music in a domain which is highly structured yet uneasy to describe in terms of fixed categories[1]. The aim of the study was to simulate improvisation/evaluation processes in *rule-based* music composition (see Laske 1989:49-ff regarding rule-based composition). The need for efficient synthesis/recognition algorithms led us to develop an ad-hoc model of transformational grammars, the *Bol Processor*, in which constraints on duration, tempo, etc. can be comprehensively formalized (Kippen & Bel 1989). More recently, we started developing the BP as a tool for computer-aided composition[2], using a MIDI interface connected to a SYTER real-time interactive digital synthesizer[3].

The main new concepts that needed to be imbedded in the Bol Processor model were the *structure of time* (Xenakis 1963) and concurrent processes (*polyphony* in a broad meaning). The structure of time may be represented — independently of musical structures

---

[1] North Indian tabla drumming, in collaboration with ethnomusicologist Jim Kippen. Initial work was supported by *International Society for Traditional Arts Research* (ISTAR), the *National Centre for the Performing Arts* (NCPA, Bombay) and the *Ford Foundation.*

[2] Concurrent processes are still under discussion within the 'OC' study group at *Laboratoire Musique et Informatique de Marseille*. This paper should not, therefore, be considered as a final statement on the matter.

[3] SYTER was designed by J.F. Allouis at INA-GRM (Paris) and built by Digilog (Les Milles, France). This system is currently used for research in Marseille by *Laboratoire Musique et Informatique* and by the team of J.C. Risset at *Laboratoire de Mécanique et Acoustique*.

— as a mapping between *symbolic time* (similar to Jaffe's *basic time*:1985) and *physical* time. This approach is well-adapted to structural descriptions (e.g. automata or formal grammars) in which it is not desirable to encode a priori information about durations and tempo.

Changes in the structure of time do not affect coincidences stamped by symbolic dates. Therefore it is also possible to define polyrhythmic structures based on symbolic time without falling again into the complicated computation of delays. The end of this paper is a presentation of an algorithm inferring a polymetric structure from incomplete (sometimes ambiguous or contradictory) information.

Several theoretical models of concurrent processes have been applied to descriptions of polyphonic structures in music. *Trace languages* — a generalization of string languages — (Zielonka 1987, Mazurkiewicz 1984a-b) based on relations of (time-) dependency and independency defined on pairs of individual 'events' have been used by Chemillier (1987). Although this description of parallel structures bears some resemblance with trace languages and their application to music (Chemillier 1987, Chemillier & Timis 1988), a comparative study is beyond the scope of this paper (Bel 1990).

## 2. Musical events, symbols and 'meaning'

Understanding the process by which a trained listener is able to identify and relate musical events is a central problem addressing both psychoacoustics and the psychology of music. Intuitively, a musical *event* is a segment of sound to which some musical 'meaning' may be assigned. A set of interrelated musical events is called a musical *component.* Formally, given a set of events E and a set of components C, $\mathcal{P}(E) \supseteq C$. There are at least two features specific to music perception, as opposed to speech:

— several events/components may meaningfully (partially or totally) overlap each other;
— only very few events/components are prelabelled (e.g. notes, silences, chords, etc. in Western classical music).

The process by which a human listener is able to identify and follow several 'voices' on a single sound track has been simulated by models performing pattern recognition on appropriate representations of sound (supposedly similar to the listening process[4]). The next step, i.e. the segmentation of each 'voice', is a trade between 'evident' qualitative or quantitative changes, and some 'meaning' attached to particular sound segments. A typical qualitative change is a sudden change of perceived pitch, amplitude, etc. A meaning-oriented change is for instance the perception of distinct notes in a *portamento*, where the task of listening is driven either by the overall feeling of a 'tonal system' (a scale, a mode, etc.) or by the awareness of a 'rhythmic system' (e.g. a tempo)[5]. The narrow link between understanding and parametric *change* has been pointed out by Minsky (1986), and its context sensitivity by Smoliar (1989:61) in his interpretation of Ross Lee Finney's "law of the balance of parameters":

---

[4] A comprehensive survey of recent developments in psychoacoustics and sound modelling may be found in Risset (1989).

[5] Inverted commas are compulsory here, as in many parts of this paper: the denotation of musical concepts varies considerably from one musical system to the next.

> This law basically observes that the only parametric changes which matter, be they with regard to pitch, tempo, dynamics, or timbre, are those which make a difference which can be perceived in their context.

Both ethnomusicologists and contemporary musicians have contributed to build the hypothesis that few categories of musical events/components are consistently shared within a group of listeners[6]. The most significant features characterizing groups of listeners, therefore, are not so much a share of musical concepts than discriminative procedures developed by musical training. Categories and concepts, within one group, remain context-sensitive partly because several procedures may compete with each other (e.g. tonal versus rhythmic identification as mentioned above). For this reason it is unrealistic to claim the existence of abstract and universal models of music based on fixed hierarchies of musical components: the existence of clear-cut categories of musical concepts is an ethnocentric illusion imposed on Western musicians by standardized notation[7-8].

Musical 'meaning', therefore, is not a simple category assignment: sound events do not possess any potential interpretation unless they are structurally related, be it through a real, imaginery, permanent or temporary structure activated by the process of listening. Any labelling of musical components/symbols, therefore, reflects the (real, imaginery, potential, expected etc.) relations these components display in mutual interactions[9].

A musical component $x \in C$ has a finite number of descriptive features. The variables attached to features may be nominal, ordinal, preordinal, graduated, metric, statistical or probabilistic (Vecchione 1989:9-10). Let $B_1$, $B_2$, ... $B_n$ be the sets of possible values for variables $y_1$, $y_2$, ... $y_n$, and $f$ an injective mapping such that $f(x) = (y_1, y_2, \ldots y_n)$, the values taken by all variables relative to $x$. In this viewpoint (*concept elaboration*), musical 'meaning' is less a hierarchical description of C than any *interesting* structure of f(C), 'interest' being the matter of producing an explanation under some particular viewpoint.

## 3. Representing sequences of events

Throughout this paper we will use lower-case characters for labelling discrete events although in real music descriptions it is more convenient to deal with strings or iconic representations.

One of the essential descriptive features of musical events/components is their time-span interval. The following three categories may be distinguished:

— (1) events with fixed duration, e.g. saying the sentence: "Hello, my name is John" at given speed;
— (2) events with undetermined duration, e.g. a stroke on a drum, a *fermata*;
— (3) events with no intrinsic duration, e.g. depressing a key on an electronic organ keyboard;

---

6 The contribution of musical semiotics to this awareness should not be underestimated, see for instance Molino (1988).

7 Indeed there are many other art-music systems giving importance to notation, but most of them do not claim consistency nor completeness.

8 This argument comes in support to Smoliar's (1989) provokative paper on music modeling.

9 This issue is crucial because of new developments of formal music theory adressing several branches of AI based on different approaches to 'symbols'. See for instance Newell (1980) and Leman (1989).

Events belonging to the first two categories are entirely put in time when the on-setting date is known, but for any event of the third category it is necessary to know two parameters: its on-setting and off-setting dates (or any of the two plus its duration).

Intuitively, a string of symbols is meant to represent a sequence of events. For instance, consider the sketch 'abcd' where:

a = 'John opens the door and comes in'
b = 'John says: "Hi, my name is John"'
c = 'Bill gets up'
d = 'John and Bill shake hands'

'abcd' may be thought of as a strict sequence, but in a real situation overlappings or gaps between two (or more) events would possibly take place. The reason for calling 'abcd' a sequence is the awareness of a strict ordering of *on-setting* dates: we know for sure that Bill did not get up before John started talking. If we consider all possible relative positions of the time-span intervals of two events, say X and Y, we obtain the following set mentioned by Allen & Kautz (1985), Van Benthem (1983:58-79) and Vecchione (1984:150ff). Following Vecchione's notation:

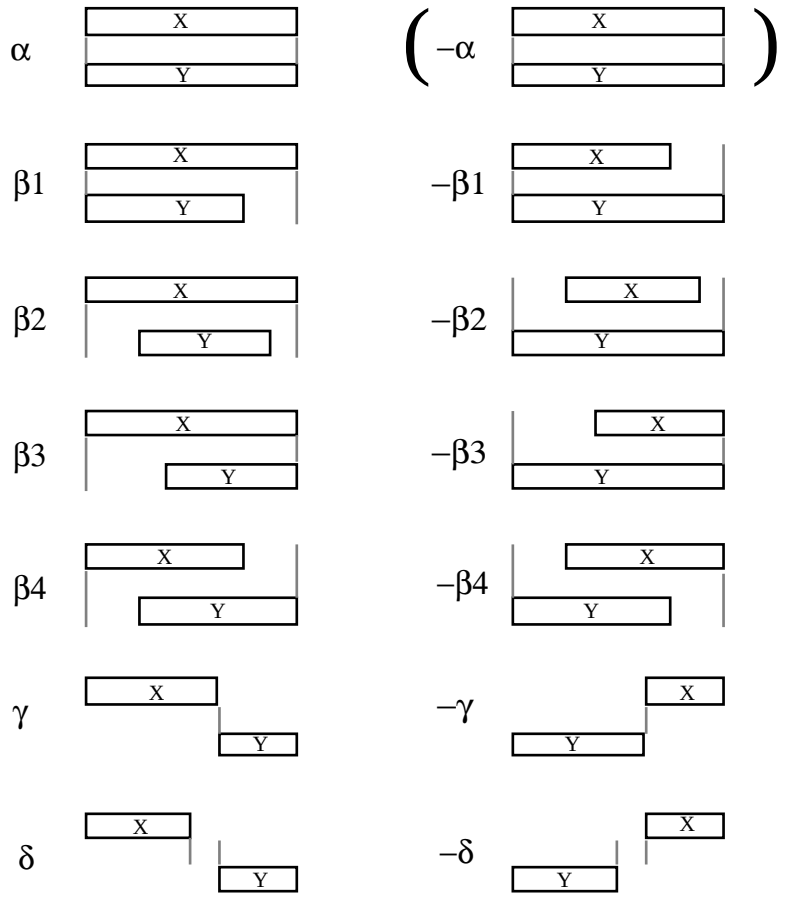



**Fig. 1**

If 'XY' is meant to be a *sequence*, acceptable configurations are $\alpha$, $\beta 1$, $\beta 2$, $\beta 3$, $\beta 4$, $\gamma$ or $\delta$. Configurations $\alpha$ and $\beta 1$ may also be excluded as they denote *simultaneous* events in a strict or broad sense[10].

If we consider events belonging to the second category, whose off-setting date is irrelevant, $\beta 2$, $\beta 3$, $\beta 4$, $\gamma$ and $\delta$ may be viewed as equivalent. With events of the third category we should call 'sequential' only pairs of events that verify $\gamma$ or $\delta$.

For all categories it may be argued that $\delta$ is not a strict sequence: it is always possible to define an 'empty' event, a silence filling the gap, if any, between the off-setting date of X and the on-setting date of Y.

Events of categories (1) and (2) have either predetermined or imprecise durations, therefore no off-setting date needs to be computed. As to events of category (3), we will consider that the off-setting date of each event coincides with the on-setting date of the next event. For consistency we may append a 'NIL' event to the end of every sequence.

In this view, the *duration* of an event in a sequence is the time-span interval between its on-setting date and the on-setting date of the next event in the sequence. We use the same word 'duration' whatever category the event belongs to. In category (3) this concept is consistent with common-sense 'duration'. In category (2), we may consider that a 'punctual' event extends its 'effect' until the next event in the sequence starts, as it may be observed in homophonic drumming sequences. In category (1), however, using a word like 'duration' is arguable because it relates to the time elapsed till the next event takes place, not the actual duration of the current event. Nevertheless it is acceptable in sound sequences to the extent that the on-setting of a new event pulls attention away from the remaining part of the previous event.

There are two reasons for using the same definition of 'duration' in the three categories. One is that it allows synchronizing sequences containing events irrespective of their categories. The second one is that in computer music there is rarely a test verifying that an event of category (1) has come to its end. If the duration of the event, or its maximum possible value, is known, then a machine or a human composer is able to fix on-setting dates depending on which configurations ($\beta 1$, $\beta 2$, $\beta 3$, $\beta 4$ or $\gamma$) are acceptable in a sequence. A test on the completion of an event would be part of a *handshake* procedure synchronizing processes. If several sequences are competing with each other, handshake interruptions may refer to events occurring in either sequence. Models for the management of asynchronous/synchronous parallel processes (see the literature relative to automated machinery: Petri networks, etc.) are beyond the scope of this paper: we will only envisage sequences synchronized by the internal clock of the system: a typical situation is a sequencer hooked to a synthesizer via its MIDI interface.

## 4. Symbolic dates

The relevant time features of events in a sequence are their mappings to an index (their position in the string) and to on-setting dates:

---

[10] The use of these time operators/predicates in formal representations of music has been discussed by Vecchione (1984-5), Risch (1988) and Bel (1989c). A similar approach has been proposed by Oppo (1984).

$$S \text{—}\theta\text{—>} \mathcal{D}$$
$$S \text{—}\rho\text{—>} \mathcal{N}$$

where S denotes a sequence, $\mathcal{D}$ a strictly ordered (enumerable) set of (on-setting) *symbolic* dates, $\mathcal{N}$ the set of positive integers, and $\rho(x)$ the position of event x in the string describing S.

Intuitively, a symbolic date is a reference such as "the third quaver after the 27th bar" where *physical* date might be "4137 milliseconds after the violin started". This may also be related to what Boulez calls "*temps strié*" as opposed to "*temps lisse*" (Boulez 1963:104-ff).

Both $\theta$ and $\rho$ are uni-valued functions. $\rho$ is a bijective mapping, $\theta$ is injective: given two distinct events X and Y, the meaning of property $\theta(X) = \theta(Y)$ would be that X and Y have identical on-setting dates, i.e. either configuration $\alpha(X,Y)$ or $\beta1(X,Y)$ is verified, which we assumed is not acceptable in a sequence . Consequently, X and Y must belong to distinct sequences $S_1$ and $S_2$ such that $\theta_1(X) = \theta_2(Y)$ (see for instance events *c* and *e* on fig.2 below).

Simultaneity (of on-setting dates), i.e. *synchronization*, is the main feature to consider when associating several sequences in a concurrent process. In this model, therefore, we envisage simultaneity as a high-level property imbedded in the mappings $\theta$ of sequences to the set of symbolic dates.

The set of dates $\mathcal{D}$ is not necessarily unique: in fact, each musical component may have its own *local* symbolic time..

## 5. Symbolic versus physical time

The set(s) of symbolic dates $\mathcal{D}$ is (are) mapped to physical time, i.e. the set of rational numbers $\mathcal{Q}$. Each such mapping $\phi$ is a restriction to $\mathcal{D}$ of a more general mapping that we call the *structure of time*. The following is a mapping of two sequences represented by strings 'abcd' and 'efg':

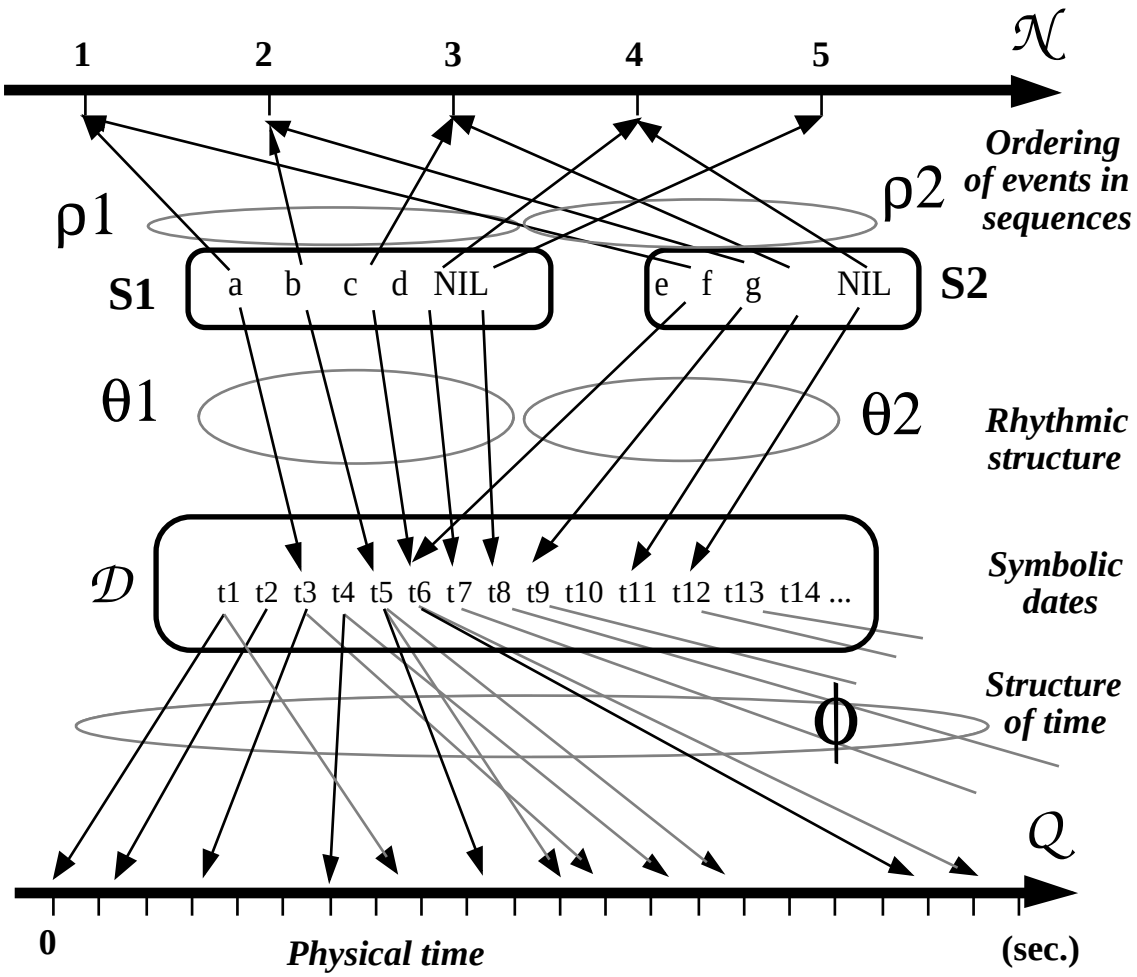



**Fig.2**

In many common-sense descriptions, ϕ is a bijective mapping with additional property: $d_i < d_j <=> \phi(d_i) < \phi(d_j)$. If we consider $\Delta(d_i,d_j) = | \phi(d_j) - \phi(d_i) |$, the absolute value of the difference, Δ is a distance on $\mathcal{D}$ and it is easy to prove that $(\mathcal{D},\Delta)$ is a *metric* space[11]. $(\mathcal{D},\Delta)$ is also *Euclidian* if the following property holds:

$$j - i = l - k \Rightarrow \phi(d_j) - \phi(d_i) = \phi(d_l) - \phi(d_k)$$

To simplify we will talk about *metric time* and *Euclidian time*. Informally, when $(\mathcal{D},\Delta)$ is Euclidian it contains explicit information about the durations of events. On the other hand, metric (non-Euclidian) time may be used for expressing *accelerando* and *ritardando*.

In general, ϕ is a multi-valued injective mapping with no restriction on ordering: events mapped to the same symbolic date may be performed more than one single time; events in sequences may also come in an order varying from one occurrence to the next, etc. An essential constraint remains that events with identical symbolic dates must be performed together regardless of the structure of time. Conversely, an *injective* mapping prevents two

---

[11] i.e. $\forall i,j,k,\ D(d_i,d_j) + D(d_j,d_k) \geq D(d_i,d_k)$

events, X and Y, from being performed simultaneously unless their symbolic dates are identical: $\phi(d_i) = \phi(d_j) \Rightarrow d_i = d_j$. Several models of structure of time are discussed in §8.

The composition of the two mappings $(\phi \cdot \theta)$ is the *in-time structure* of the sequence, i.e. the mapping that permits its actual performance. Structure of time and in-time structures are two concepts borrowed from Xenakis (1963, 1971, 1972:57), which we find essential as they deal with a set of dates not necessarily structured as a Euclidian space[12].

## 6. Rhythmic structure

The *rhythmic structure* of any sequence S is function $\theta$ itself.

In the same way $\rho_1$ and $\rho_2$ represent the bijective mappings ordering events in sequences $S_1$ and $S_2$, we may call $\rho_T$ the bijective mapping ordering the set of dates $\mathcal{D}$. For any sequence $S_i$, $(\rho_i^{-1} \cdot \theta_i \cdot \rho_T)$ is a monotonously increasing function. This is equivalent to saying that any two events having consecutive labels in a string must be in relative position $\beta 2$, $\beta 3$, $\beta 4$, or $\gamma$. This could be expected since the on-setting symbolic dates reflect the left-to-right ordering of the string.

To compare rhythmic patterns in two sequences one needs to eliminate their (symbolic) time offset. Let $O_i = \rho_T(\theta_i(\rho_i^{-1}(1)))$ be the origin of sequence $S_i$. For the two examples on fig.2, $O_1 = 3$ ('a' is mapped to $t_3$) and $O_2 = 6$ ('e' is mapped to $t_6$). Let us compare the image sets of functions:

$$R_i(k) = \rho_T(\theta_i(\rho_i^{-1}(k))) - O_i, \text{ where } k \in \mathcal{N}$$

Each image set of a $R_i$ function characterizes a rhythmic pattern independently on the symbolic date when the sequence started (and, needless to say, on the physical durations of events when non-Euclidian time is used). For example we get the following sets of values:

Sequence $S_1$ = 'abcd':

$$\rho_T(\theta_1(\rho_1^{-1}(k))) = \{3,5,6,7,8\} \Rightarrow R_1(k) = \rho_T(\theta_1(\rho_1^{-1}(k))) - O_1 = \{0,2,3,4,5\}$$

Sequence $S_2$ = 'efg':

$$\rho_T(\theta_2(\rho_2^{-1}(k))) = \{6,9,11,12\} \Rightarrow R_2(k) = \rho_T(\theta_2(\rho_2^{-1}(k))) - O_1 = \{0,3,5,6\}$$

Trying to match $R_1(k)$ with $R_2(k)$ does not yield any information except strict identity or difference. Approximate matching is more conveniently performed on lists of *symbolic durations* $I_i(k)$ defined as follows:

> Given $n_i$, the length of the string representing a sequence $S_i$,
> for any k such that $0 < k < n_i$,
> $I_i(k) = R_i(k+1) - R_i(k)$.

---

[12] We do not claim that the terminology in this paper is faithful to Xenakis. Interestingly, although Xenakis had introduced three structures in his early work (1963:190-1,200, 1971): in-time (*structure en-temps*), out-time structures (*structure hors-temps*) and structure of time (*structure temporelle*), in 1972 he did not seem to be concerned any more with the structure of time.

In the example above, $I_1(k) = (2,1,1,1)$ and $I_2(k) = (3,2,1)$. We call these durations 'symbolic' because they are built on the set of symbolic dates. Although $I_1(1) = I_2(2) = 2$ in this example, point to events ('a' in 'abcd' and 'f' in 'efg') with identical symbolic durations, their physical durations may be unrelated.

The meaning of two sequences with identical lists of symbolic durations is that *all the (on-setting) dates of the corresponding events would be identical if the sequences were performed from the same time origin*.

Rhythmic patterns, therefore, are invariant with respect to translations on symbolic time[13]. For example, if the first sequence whose symbolic dates were (3,5,6,7,8) is shifted to the left by two 'units', we obtain the list of dates (1,3,4,5,6) which has exactly the same pattern. Unless the set of dates is Euclidian, this does not apply to changes of the physical starting date: if a sequence is delayed or repeated it must be 'adapted' to the local time structure, using the $\phi$ function.

## 7. Notating symbolic durations

If we consider the Cartesian product $S \times \mathcal{N}$ we can represent events together with their symbolic durations considering the function:

$$S \text{ —}\psi\text{—> } S \times \mathcal{N} \text{ such that } \psi(x) = (x, I(\rho(x)))$$

For instance, $\psi(S_1) = ((a,2),(b,1),(c,1),(d,1))$ and $\psi(S_2) = ((e,3),(f,2),(g,1))$.

Since in each couple the first argument is an alphanumeric string and the second one a positive integer, every couple may be unambiguously represented as an (unordered) set. In addition, we may also leave out commas wherever they indicate concatenation, plus the pair of brackets bordering the list. The notation obtained:

$$\psi(S_1) = \{a,2\}\ \{b,1\}\ \{c,1\}\ \{d,1\} = \{2,a\}\ \{b,1\}\ \{1,c\}\ \{d,1\} = \text{... etc.}$$

and

$$\psi(S_2) = \{e,3\}\ \{f,2\}\ \{g,1\} = \{3,e\}\ \{2,f\}\ \{g,1\} = \text{... etc.}$$

will be useful in polymetric structures (see infra).

Another notation system, designed by North Indian drummers (Kippen 1988:xvi-xxiii), is based on the idea that no symbolic duration is ever smaller than 1. Default duration is conventionally '1'. An event X with symbolic duration *n* may be represented by appending (n-1) prolongational gaps to the right of its label. In a drumming sequence prolongational gaps are equivalent to silences: the time-span between two strokes is filled with the resonance of the first stroke, if any. However, if we deal with events of category (3) (see §3), it is also necessary to dispose of an empty event (e.g. a 'note-off' in MIDI terminology) distinct from the prolongational silence. Silences may be notated with hyphens, and prolongational gaps with underline characters. Using this system,

---

[13] A similarity function is easily defined for comparing the rhythmic patterns of two sequences: given $I_1(k)$ and $I_2(k)$, find the maximum common subsequence and calculate its length. If pattern expansions are considered relevant, search for common subsequences with identical duration ratios.

$$\psi(S_1) = \{a,2\}\ \{b,1\}\ \{c,1\}\ \{d,1\} = a\ \_\ b\ c\ d$$
and
$$\psi(S_2) = \{e,3\}\ \{f,2\}\ \{g,1\} = e\ \_\ \_\ f\ \_\ g$$

The advantage of this system is twofold[14]:

— it facilitates the pronunciation of rhythmic sentences: speak symbols or clap '_ ' on metronome beats;

— when durations are 1 the representation is simple string notation, i.e. $\psi(S) = S$.

We may notate:

$$\psi(S_1) = \{a,2\}\ \{b\ c\ d,3\} = \{a,2\}\ b\ c\ d = \{\{a,2\}\ \{b\ c\ d,3\},5\} = ...etc.$$

We will see later how an expression like {b c d,5} may be notated without brackets. A useful minor extension of the system is the use of integers in replacement of strings of '_ '. The following notations are equivalent:

$$a\ b\ \_\ \_\ \_\ c\ = a\ b\ \{\_\ \_\ \_,3\}\ c = a\ b\ \{3\}\ c = a\ b\ 3\ c$$

Each of the last two transformations illustrates extensions of the system: the set represented in curled brackets may have less (or more) than two elements, and brackets may be omitted for sets with only one element. Conversely,

$$a\ b\ c\ d = \{a\}\ \{b\}\ \{c\ d\} = \{a,1\}\ \{b,1\}\ \{c\ d,2\}$$

demonstrates that duration may be omitted in a bracket when it is exactly the length of the string.

Also note identical following expressions:

$$a = \{a\} = \{a,1\} = \{a\ \_,1\} = \{a\ \_\ \_\ \_\ \_,1\} = ...etc.$$

$$a\ b\ c = \{a\ b\ c\} = \{a\ b\ c,3\} = \{a\ \_\ b\ \_\ c\ \_,3\} = \{a\ \_\ \_\ b\ \_\ \_\ c\ \_\ \_,3\} = ...etc.$$

### 7.1. Tempo indications

The reader may feel impatient to know how {b c d,5} can be interpreted. In the absence of explicit information the symbolic durations of b, c and d should be identical. In terms of physical or Euclidian durations this is easy to achieve by stating that the duration of each event shall be 5/3. The next paragraph will indicate necessary changes in the structure of the set of dates $\mathcal{D}$. Let us, for the time being, indicate this transformation with '/3'. We obtain:

$$\{b\ c\ d,5\} = /3\ \{b\ c\ d,15\} = /3\ \{b,5\}\ \{c,5\}\ \{d,5\} = /3\ b\_\ \_\ \_\ \_\ c\_\ \_\ \_\ \_\ d\_\ \_\ \_\ \_$$

in which '/3' indicates a change in the division of each *beat* (this concept will be introduced in §7.2). A *change* presupposes of course that the default division was 1. Another method for solving that problem is based on the last remark of the preceding paragraph:

$$\{b\ c\ d,5\} = \{b\_\ \_\ \_\ \_\ c\_\ \_\ \_\ \_\ d\_\ \_\ \_\ \_,5\} = /3\{b\_\ \_\ \_\ \_\ c\_\ \_\ \_\ \_\ d\_\ \_\ \_\ \_,15\}$$
$$= /3\{b\_\ \_\ \_\ \_\ c\_\ \_\ \_\ \_\ d\_\ \_\ \_\ \_\ \} = /3\ b\_\ \_\ \_\ \_\ c\_\ \_\ \_\ \_\ d\_\ \_\ \_\ \_$$

The following three examples indicate transformations dealing with multiple changes of tempo. (The same process is illustrated by Boulez, 1963:56, example 14):

---

[14] Both prolongational-gap and curled-bracket systems have been implemented in the most recent version of the Bol Processor.

{a b,3} {c d,2} = /2 {a b,6} /1 {c d,2} = /2 {a,3} {b,3} /1 {c,1} {d,1}
= /2 a_ _ b_ _ /1 c d = /2 a_ _ b_ _ /2 c_ d_ = /2 a_ _ b_ _ c_ d_ = /2 a_ _ b_ _ c_ d_

{a b c,3} /2 {d e,2} /1{f,1} = {a b c} /2 {d e} /1{f} = a b c /2 d e /1 f
= /2 a_ b_ c_ def_ = /2 a_ b_ c_ de f_

{a b c,2} {d e,5} = /3 {a b c,6} /2 {d e,10} = /6 {a b c,12} /6 {d e,30}
= /6 {a,4} {b,4} {c,4} {d,15} {e,15} =
/6 a_ _ _ b_ _ _ c_ _ _ d_ _ _ _ _ _ _ _ _ _ _ _ _ _ e_ _ _ _ _ _ _ _ _ _ _ _ _ _
= /6 a_ _ _ b_ _ _ c_ _ _ d_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ e_ _ _ _ _ _ _ _ _ _ _ _ _ _
= /6 a_ _ _ b_ _ _ c_ _ _ d_ _ _ _ _ 6 _ _ _ e_ _ 6 6

In the final expressions the tempo was unique along the whole sequence. Consequently, tabulations or spaces could be used to indicate beats. This makes it clear (first example) that the on-setting date of *b* is exactly the off-beat of the second beat. The same with *e* in the last example.

A minor syntax modification is necessary to introduce fractional gaps. We notate:

{a b c,3} /2{_ ,3} /1 {d e,2} = a b c /2 _ _ _ /1 {d e,2} = a b c /2 _ _ _ /1 d e
= /2 a _ b _ c _ _ _ _ d _ e _ = /2 a _ b _ c _ 3 d _ e _

but an equivalent description is:

a b c /2 _ _ _ /1 d e = (a b c /2 **3** /1 d e =) a b c **3/2** /1 d e = a b c **3/2** d e

The second expression is illegal because spaces have no syntactic value in our system: '/2 **3**' would be wrongly interpreted as '/23'. Correct expressions are the last two ones: '3' has become the numerator of the ratio; consequently, the tempo after '3/2' remains the same one as in 'abc'.

In a fractional gap the denominator of the ratio may be interpreted a *relative* change of tempo. The following transformations:

a b /2 c d e f 4/3 g h = a b /2 c d e f {_ ,4/3} g h = a b /2 c d e f /2 {_ ,4/3} /2 g h
= a b /2 c d e f {_ ,4/6} /2 g h = a b /2 c d e f /6 {_ ,4} /2 g h
= a b /6 c _ _ d _ _ e _ _ f _ _ /6 {_ ,4} /6 g _ _ h _ _
= a b /6 c _ _ d _ _ e _ _ f _ _ {_ ,4} g _ _ h _ _
= /6 a _ _ _ _ _ b _ _ _ _ _ c _ _ d _ _ e _ _ f _ _ 4 g _ _ h _ _

are correct, whereas the following one

a b /2 c d e f 4/3 g h =? a b /2 c d e f /3 {_ ,4} /2 g h
=? a b /6 c _ _ d _ _ e _ _ f _ _ /6 {_ ,8} /6 g _ _ h _ _
=? /6 a _ _ _ _ _ b _ _ _ _ _ c _ _ d _ _ e _ _ f _ _ 8 g _ _ h _ _

is wrong.

### 7.2. Fractional symbolic dates and durations

Fractional gaps point to the idea that time could be measured with integer ratios instead of integers. Consequently, /3 {a b c,2} = /1 {a b c, 2/3}. In other words, the set of symbolic dates should now be ordered with a mapping to the set of positive rational numbers $\mathcal{Q}^+$. Any symbolic date is a ratio p/q, or equivalently a pair of integers (p,q). To compare ratios we use the equivalence class '=' building $\mathcal{Q}^+$ on $\mathcal{N} \times \mathcal{N}^*$:

$$\forall\ p \in \mathcal{N},\ \forall\ k,q \in \mathcal{N}^*,\ \ (p,q) = (k.p,k.q)$$

Strict ordering '<' is defined as follows:

$$\forall\ p,r \in \mathcal{N},\ \forall\ q \in \mathcal{N}^*,\ (p,q) < (r,q) \iff p < r$$

$$\forall\ p,r \in \mathcal{N},\ \forall\ q,s \in \mathcal{N}^*,\ (p,q) < (r,s)$$
$$\iff \exists\ p',r' \in \mathcal{N},\ \exists\ q',s' \in \mathcal{N}^*,\ (p,q) = (p',q') \text{ and } (r,s) = (r',s') \text{ and } (p',q') < (r',s')$$

Symbolic durations are computed from symbolic dates using the classical subtraction operation, a partial mapping of ($\mathcal{Q}^+$ x $\mathcal{Q}^+$) to $\mathcal{Q}^+$:

$$\forall\ p,r \in \mathcal{N},\ \forall\ q \in \mathcal{N}^* \text{ such that } p \le r,$$
$$(r,q) - (p,q) = (r-p,q)$$

with the additional property:

$$\forall\ A,B,C \in \mathcal{Q}^+,\ B < A,\ C < A,\ C < B,\ A - B = (A - C) - (B - C)$$

The rest of the theory (durations, rhythmic patterns, etc.) is unchanged. Evidently the size of the set of dates $\mathcal{D}$ may grow considerably larger than it was the case with integer dates. One may argue that, since for every sequence it is possible to have a unique tempo (e.g. the lowest common multiple — LCM — of all tempos appearing in the sequence), it is possible to use a unique tempo for all sequences (the LCM of all individual tempos) and report the change of scale into the structure of time $\phi$. The method is acceptable but it has two drawbacks:

— the global tempo may be a very large number resulting in cumbersome notation of sequences;

— the system is not incremental: what if a new sequence is introduced, the tempo of which is not a divider of the global tempo?

In order to deal with $\mathcal{D}$ it is helpful to structure it with an equivalence relation. A very simple structure is the following: every rational number A = (p,q) has a unique representation

$$\xi(A) = i + \frac{n}{d}$$

such that

$$i,n \in \mathcal{N},\ d \in \mathcal{N}^*,\ (i.d+n,d) = (p,q),\ n < d \text{ and } n/d \text{ is a ratio in its lowest terms.}$$

Let $\mu$ be the mapping of $\mathcal{Q}^+$ to itself yielding (n,d) from (p,q) in the above relation. The equivalence relation '$\approx$' on $\mathcal{D}$:

$$\forall\ t_1,t_2 \in \mathcal{D},\ t_1 \approx t_2 \iff \mu(\rho_T(t_1)) = \mu(\rho_T(t_2))$$

may be interpreted as a (not necessarily Euclidian) *pulse*. Rational number (n,d) = $\xi$(t) is a *label* for each class. For instance, (0,1) is the label for dates falling exactly on *beats*, (1,2) may be called the *off-beat*, (3,4) the third quarter, etc. These class labels are assigned automatically when a new sequence is analysed: if N is the global tempo of the sequence, the classes through which the symbolic dates of events in S may be related are those with labels (p,q) such that (p,q) = (n,M) for every positive integer n < M. The following example will clarify this process. Consider the sequence:

{a b c,2} {d e,5} =

/6 a_ _ _ b_ _ _ c_ _ _ d_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ e_ _ _ _ _ _ _ _ _ _ _ _ _ _ _

= /6 a_ _ _ b_ _ _ c_ _ _ d_ _ _ _ _ _ _ _ _ _ _ e_ _ _ _ _ _ _ _ _ _ _ _ _ _ _

in which M = 6. The possible values for (n,M) are:

(0,6) = (0,1)
(1,6)
(2,6) = (1,3)
(3,6) = (1,2)
(4,6) = (2,3)
(5,6)

Every event in the sequence is assigned a class:

| Event | Symbolic date t | $\xi(t)$ | Class label $\mu(t)$ |
|---|---|---|---|
| a | (0,6) = (0,1) | 0 + 0/1 | **(0,1)** |
| - | (1,6) | 0 + 1/6 | (1,6) |
| - | (2,6) = (1,3) | 0 + 1/3 | (1,3) |
| - | (3,6) = (1,2) | 0 + 1/2 | (1,2) |
| b | (4,6) = (2,3) | 0 + 2/3 | **(2,3)** |
| - | (5,6) | 0 + 5/6 | (5,6) |
| | | … | |
| c | (8,6) = (4,3) | 1 + 1/3 | **(1,3)** |
| | | … | |
| d | (12,6) = (2,1) | 2 + 0/1 | **(0,1)** |
| | | … | |
| e | (27,6) = (9,2) | 4 + 1/2 | **(1,2)** |
| | | … | |

Only four classes are needed for structuring $\mathcal{D}$: (0,1), (1,3), (1,2) and (2,3). Other classes such as (1,6) and (5,6) may be introduced later. A structural representation of this sequence is the following:

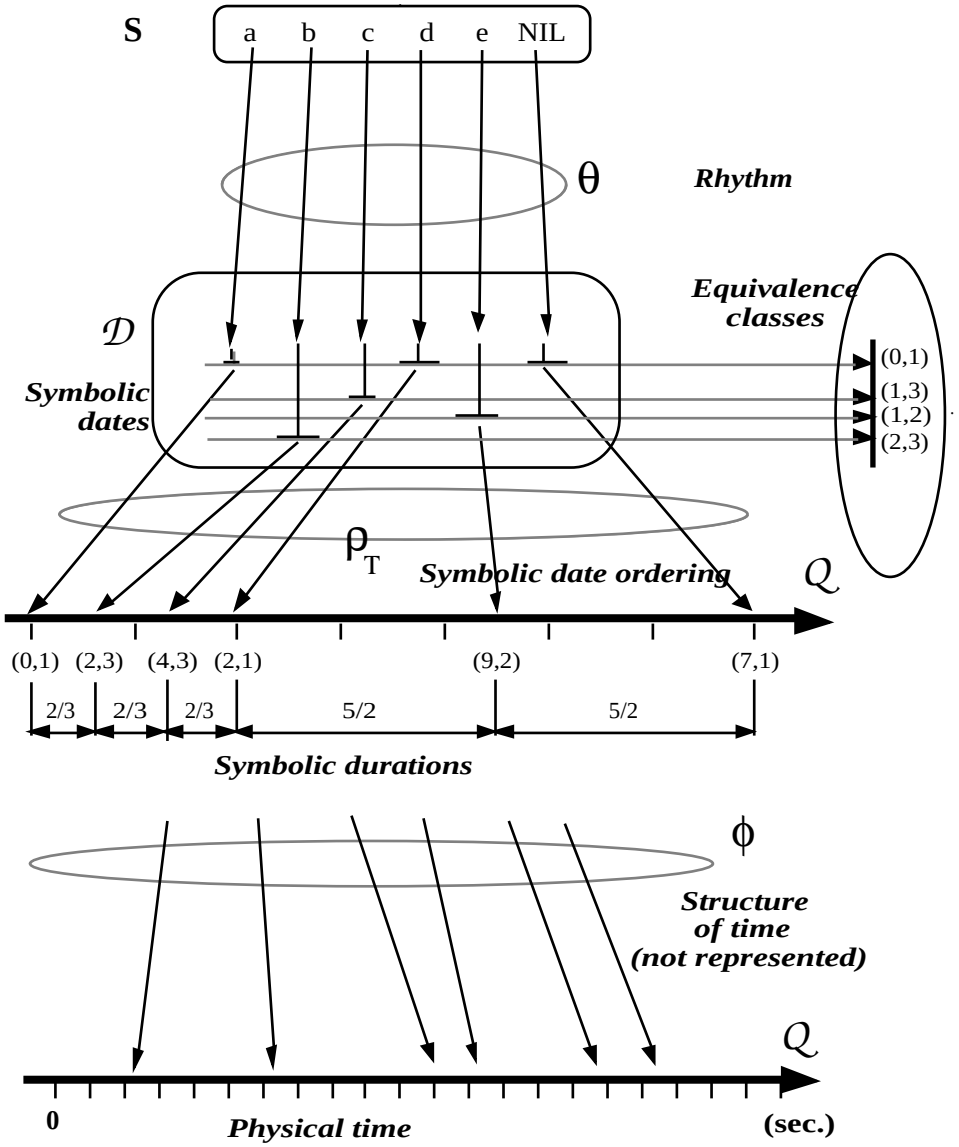



**Fig.3**

This method for structuring the set of symbolic dates is not unique but it is general enough for representing and matching rhythmic patterns in sequences whatever the structure of time.

## 8. Structure of time: a computational approach

In a digital synthesizer, 'physical time' is either the direct output of a sawtooth generator, or an output adjusted manually or automatically (in response to instructions contained in a symbolic sequence S itself).

The structure of time $\phi$ may be represented with any algebraic formula, or explicitly. Explicit representations are implemented in *tables* whereas formulae refer to *procedures*. In real-time synthesis, $\phi^{-1}$ is required for converting physical time to symbolic dates: one single table or procedure is needed if $\phi$ is injective (see §5).

Several 'cascade' mappings are almost necessarily involved, i.e. $\phi = \phi_1 \cdot \phi_2 \cdot \ldots \phi_n$. Jaffe (1985:40) only considered strictly increasing functions, but he introduced multiple

mappings for different voices and methods with view of merging them in the actual performance (ibid:44-ff). In SYTER implementations, any $\phi_i^{-1}$ is not necessarily monotonous nor even continuous:

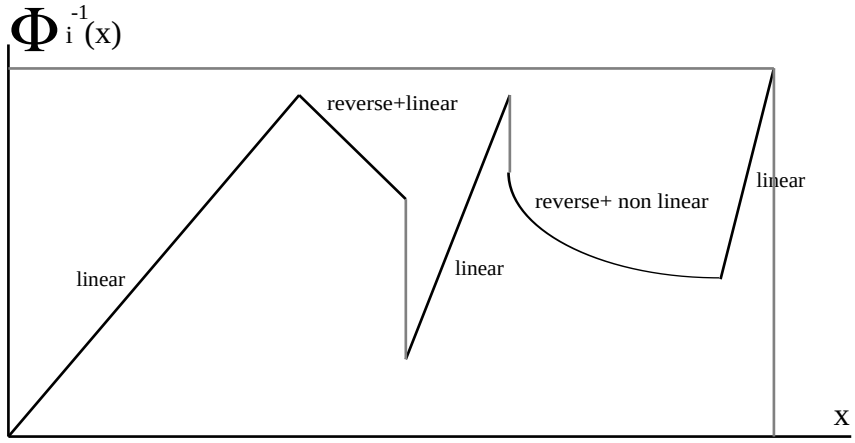



**Fig.4**

In this example, the same component is (partly) performed several times; in each occurrence the flow of symbolic time is either 'normal' (linear and 'clockwise') or 'modified' ('anticlockwise', non linear, etc.). Using several tables addressed by the same variable linked to physical time, it is possible to split physical time to several local sets of symbolic dates. Example below shows a split to three sets of symbolic dates:

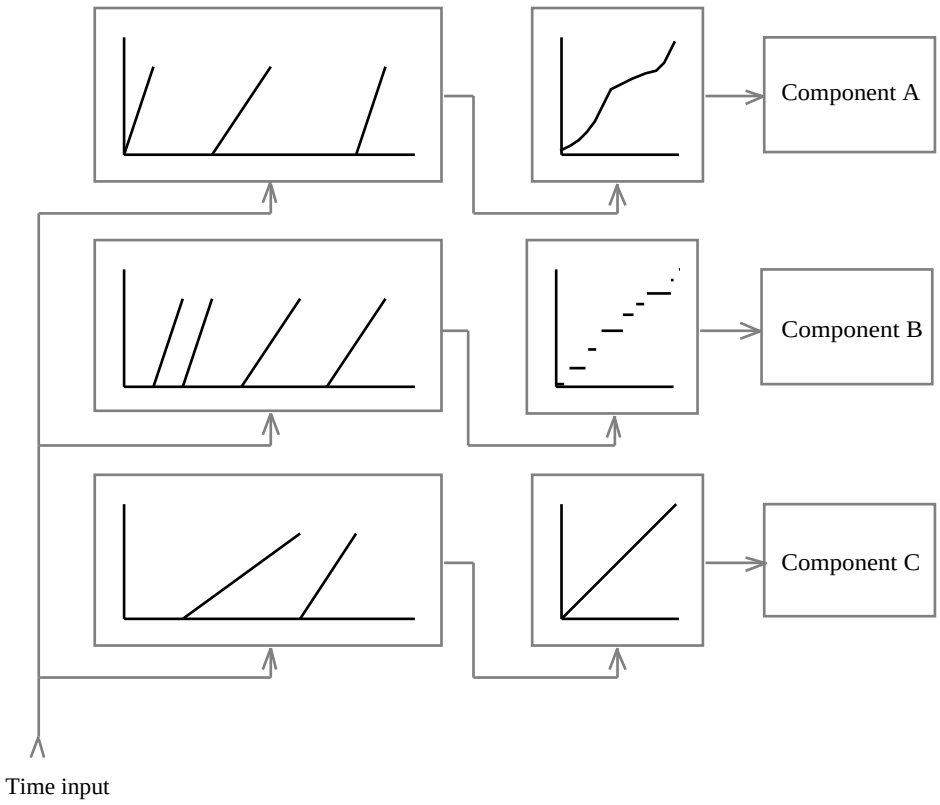



**Fig.5**

Each table in the left part is a score or 'history' of component A, B or C. This set-up produces an effect similar to that of a conditional branching triggering A, B and C. For each component, one or more table(s) is (are) needed to define the structure of its 'local time': component A has a continuous yet *rubato*-like overall structure, B has a discrete 'tempo' structure, while C is performed in 'linear' time. In the left part, straight segments have been used to provoke repetitive performances of each component at different speeds. Indeed more complicated functions may be used to account for context-sensitive distortions of each component's local time.

Each musical component may be in turn a set of components and/or musical events. The split to subcomponents may be the result of more conditional branchings, as above, or sudden changes of the value or range of any sound parameter, as evidenced in the local time structure of component B. Every parameter may have its own 'private' local time structure, and possible interactions with the musical output of the synthesizer itself or with other human/mechanical performers[15].

Time mappings and the manual/automatic control of the flow of physical time are highly dependant on technical features of sound-making machines, and will not be further discussed here. An important problem, however, is the structure of local time in each component.

Time structures, like pitch structures in many traditional music systems, are often felt intuitively as a trade between strict periodicity (a boring order) and chaos (often a structure that is not immediately perceived). One way of producing such structures is to rely on *automatic sequences*, i.e. sequences generated by automata, or equivalently formal grammars (Bel 1989d), constant length substitutions (Allouche & Mouret 1988), etc. Since it deals with nominative variables, this first method does not take directly into account additive properties of time/pitch intervals. Another approach is to build strictly periodic sets of integers and combine them using set operations. Since finite sets are needed, a prior scaling (an arbitrary origin and a unitary vector) of the domain is necessary: for example, use 440Hz as an origin and 1 cent (frequency ratio $2^{1/1200}$) as an elementary interval for pitch scaling; similarly, define a time origin and an elementary pulse interval for time scaling. Once the scale has been fixed (although independantly of it), sets of values may be built by means of *sieves*.

### 8.1. Sieves

Scales offer a wide scope for intuitively grasping the idea of musical structure because of the predominance and relative complexity of pitch relations in many traditional music systems. Partly for this reason, sieves were first introduced by Xenakis (1963) as a structural model of musical scales. The concept we introduce here has been used for a practical implementation in SYTER.

An *elementary sieve* is a congruence class on relative integers. For example, *class 3 modulo 5* is the set of relative integers *x* verifying property P(x):

> There exists a relative integer *k* such that x = 5k+3
> i.e. $x \equiv 3\ [5]$ in standard mathematical notation.

The sieve generated by this formula is the following set:

$$5_3 = \{... -12, -7, -2, 3, 8, 13, 18, ...\}$$

---

[15] A stochastic control of on-setting times taking into account the density of events is found in Xenakis' *Achorripsis*, 1956).

where '$5_3$' is the notation used by Xenakis. P(x) is a boolean expression which we may call 'atomic' as it relates to an elementary sieve.

Although the structure of an elementary sieve is infinite and strictly periodic, in practice the domain of P(x) must be restricted: musical parameters such as pitch, time, intensity, etc. are necessarily bound to limits. Consequently, four parameters are necessary to define an elementary sieve: its modulo and origin (class label or remainder of the Euclidian division), and the two limits of its definition domain.

It is possible to manipulate these parameters to show more or less of the periodicity of the infinite structure. For instance, taking a period larger than the domain may lead to a structure containing one single value — or no value at all... This makes it possible to introduce breaks into the apparent periodicity of a compound sieve.

A *compound sieve* is a set of integers built by union, intersection and complementation of elementary sieves. It may also be defined as a Boolean expression built on atomic expressions with the aid of the three operators: *and, or, not*., e.g.:

$$\overline{3_2} \wedge 4_1$$

that is, *no* ($3_2$) *or* $4_1$.

This set is calculated as follows: first build $4_1$ = {... -11, -7, -3, 1, 5, 9, 13, ...} and $3_2$ = {... -10, -7, -4, -1, 2, 5, 8, 11, 14, ...}, then take only those elements that belong to $4_1$ and not to $3_2$: {... -11, -3, 1, 9, 13, ...}

It is possible to approach the problem of sieves from two practicle viewpoints: (1) generate a sieve given a Boolean formula, or (2) characterize a given sequence of integers. The latter is a rather difficult, although rich in potential applications, rule-discovery problem: any finite sequence of relative integers may be considered in a great number of ways as a non-periodical part of a periodical structure.

### 8.2. Tabulated sieves

A first implementation of sieves in SYTER (Bel 1989a) has been realised with the aid of tables that are precomputed and stored on disk. Numeric tables are the main support for structures in SYTER. Instead of producing a particular sound event by calling its attached elementary procedure, it is necessary to define all its parameters (a characteristic set of metric variables that allow its generation by a general-purpose hook-up of modules) (Bel 1989b). Discrete events, i.e. time structures, are produced if at least one of the parameters has been changed significantly. Tabulated sieves come in response to such a need.

Given I, an interval of real numbers, and a finite increasing sequence G defined on I, a *tabulated sieve* is any mapping f of I to itself such that:

> For any x belonging to I, f(x) belongs to G
> For any pair x1, x2 belonging to I, $x1 < x2 \Rightarrow f(x1) \leq f(x2)$

A typical exemple of set-up of two tabulated sieves, one governing pitch and the other one durations, is the following:

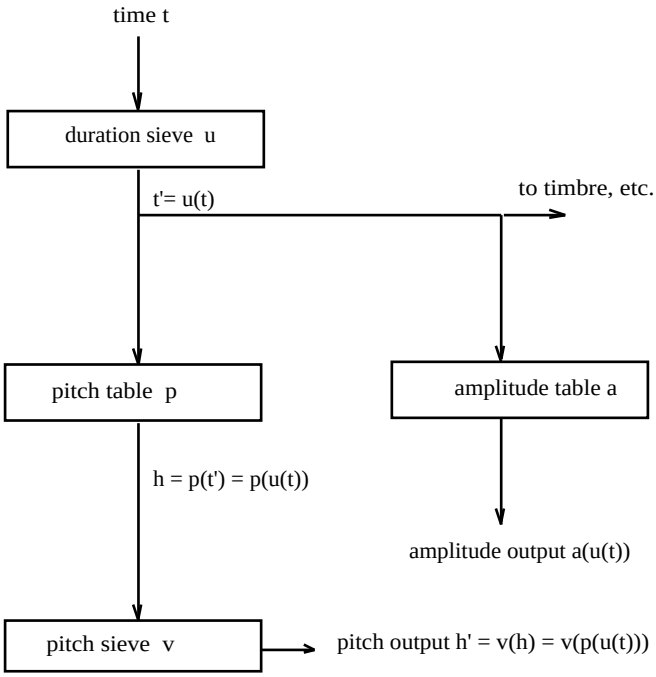



**Fig. 6**

In this set-up, the tabulated duration sieve may be for instance:

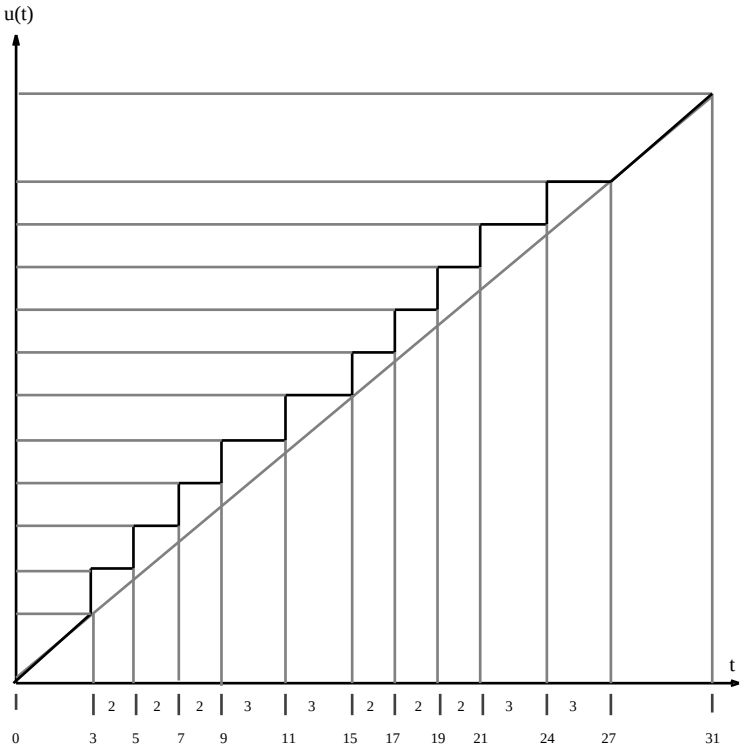



**Fig.7**

and the pitch tabulated sieve:

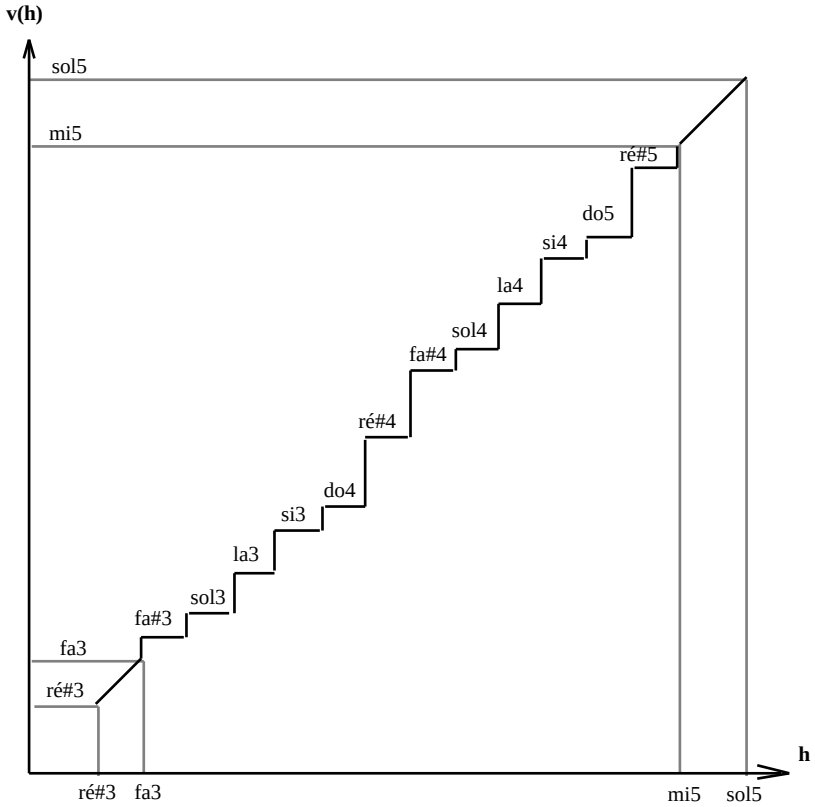



**Fig.8**

(We used simple units of duration and a conventional semitonic system in this example. Indeed, any general system, e.g. microtonal, may be described in the same way.)

Both tabulated sieves have been built from compound sieves: divisions on the horizontal axis of the time sieve, and of the vertical axis of the pitch sieve, respectively, belong to pseudo-periodiocal sequences, while divisons of the remaining axis are equal. This is only one among infinitely many ways of imbedding a compound sieve into a tabulated sieve (Bel 1989a).

# 9. Polymetric structures

## 9.1. Representation

A polymetric structure may be defined recursively as follows:

***Syntax***

— any sequence is a polymetric structure;

— if A and B are polymetric structures, {A,B} is a polymetric structure;

— if A, B and C are polymetric structures, {{A,B},C,...} may be notated {A,{B,C},...} or {A,B,C,...}.

***Semantic***

{A,B} is meaningful if and only if that A and B have identical on-setting and off-setting dates.

The use of curled brackets is consistent with the one we introduced in sequences:

— for any sequence S, {S} = S is a polymetric structure;

— for any sequence S of duration d ∈ $Q^+$, {S,d} may be interpreted as the simultaneity of sequence S with a silence whose duration is exactly d.

In other words the following notations are equivalent:

a b c d = {a b c d,4} = {a b c d, _ _ _ _}

where the latter is a polymetric structure.

The semantic definition suggests that there may be meaningless polymetric structures. This is obviously the case with {3,5} or {a b c, 5/4, 9/8}; but it is less evident whether or not it is possible to assign a meaning to structures like {a b c, d e}, {/3 a b, d e f}, {/2 a b, /3 c d e, 1}, {/3 a b c, /5 d e}, etc. For this we need conventional constraints allowing a machine to assign the 'simplest' meaning to structures which are not entirely described. We first decide that 'simple' transformations are those that do not distort rhythmic patterns. This rules out the transformation called *modification mobile* by Boulez (1963:57-8), not that we believe such a transformation should be ignored, but because it requires explicit parameters in a symbolic representation. Another principle is that a *simple* interpretation is one in which many events have identical on-setting symbolic dates.

The following conventions are based on these principles:

— if a sequence does not contain any tempo indication, its tempo may be attributed an arbitrary value;
— if A, B and C an arbitrary polymetric structures, in the structure 'A {B} C' the tempos of A and C are identical unless explicitly indicated;
— if two sequences A and B containing tempo indications appear in the same polymetric structure, any change of the tempo of A must be reflected proportionally on the tempo of B;
— the default tempo value is 1.

For example,

S = a b c /2 de /3 f g h = /1 a b c /2 d e /3 f g h (default value)

S = /3 a b c d = /1 /3 a b c d

(this sequence starts with default tempo 1 but this value is immediately changed.)

Let us now consider {a b c, d e}. The tempos of 'a b c' and that of 'd e' may be given arbitrary values although there is only a set of values that will assign a meaning to the structure, e.g. {/3 a b c, /2 d e}, {/15 a b c, /10 d e}, ...etc.

Consequently, if {$A_1$, ..., $A_n$} is a polymetric structure, the structure is always meaningful when less than two substructures contain tempo indications. If there are indications in several substructures it is necessary to check consistency. If there are no tempo indications the global tempo of the structure is ambiguous. To resolve such an ambiguity we introduce two additional rules:

### 9.2. Homogeneity rules

— if A, B and C an arbitrary polymetric structures, in the structure 'A {B} C' the global tempo of B is the same as those of A and C unless explicitly indicated;
— the global tempo of {/$p_1$ $A_1$, ..., /$p_n$ $A_n$} is the largest $p_i$.

Using these rules we may for instance interpret the structure:

S = a b {c d, e f g} h i j

in which the global tempo is arbitrary. Let us find the lowest possible tempo. We first consider {c d, e f g} which has infinitely many interpretations:

{c d, e f g} = {/2 c d, /3 e f g} or {/4 c d, /6 e f g} or {/6 c d, /9 e f g} or ...etc.

If we keep the first interpretation, the global tempo of S may be either 2 or 3. The second rule imposes 3:

S = /3 a b {/2 c d, /3 e f g} /3 h i j = /3 a b {/2 c d, /3 e f g} h i j
= /6 a _ b _ {/6 c _ _ d _ _, /6 e _ f _ g _} h _ i _ j _
= /6 a _ b _ {c _ _ d _ _, e _ f _ g _} h _ i _ j _

The second homogeneity rule allows the choice of an arbitrarily large integer, e.g. all multiples of 3 in the example shown. This means that the global tempo of {/$p_1$ $A_1$, ..., /$p_n$ $A_n$}, and consequently that of 'A {B} C', may be adjusted to match, whenever possible, another tempo imposed by constraints in a larger structure.

## 10. An algorithm for interpreting polymetric structures

The following is the main part of an algorithm implemented in HyperBP[16] for producing polymetric sound structures.

Let S be a polymetric structure.

The aims of the algorithm are:

- — determine whether or not the structure may be assigned a meaning;
- — if yes, calculate its symbolic fractional length;
- — determine whether or not the structure contains explicit tempo indications;
- — rewrite the structure with a unique global tempo.

S may be interpreted as a sequence of structures:

$$S = S_1 \ldots S_m$$

in which each $S_i$ is either a sequence of terminal symbols, or a polymetric structure {$A_1$, ..., $A_n$}.

The main procedure POLY calls two subroutines: SEQU for the evaluation of strict sequences, and SIMUL for the evaluation of simultaneous structures {$A_1$, ..., $A_n$}.

POLY, SEQU and SIMUL have identical outputs:

- — 'ok' which remains true as long as the structure is consistent with constraints;
- — 'P' and 'Q' such that (P,Q) is the fractional length of the structure;
- — 'scale' which is the needed increase of the global tempo;
- — 'length', the symbolic duration of the substructure;
- — 'globaltempo', the global tempo of the substructure;
- — 'OUT', an array which contains the substructure rewritten with a unique global tempo.

The structure is parsed from left to right. Imbricated structures may cause recursive calls to POLY, SEQU or SIMUL, but the pointer to the symbol currently read is a global variable.

---

[16] An HyperCard™ and C-language version of the Bol Processor (Kippen & Bel 1989) with MIDI interface implemented on the Apple™ Macintosh. HyperBP is available as shareware from the author.

The original structure is not copied in recursive calls. Variables ‘scale’ and ‘fixtempo’ are also global.

POLY starts assuming that the global tempo is 1. This tempo is explicitly mentioned in the output array ‘OUT’.

SEQU reads the next symbol and increments the current fractional duration on the basis of the current tempo. If SEQU reads a tempo indication, the current tempo is adjusted and ‘fixtempo’ is set to true.

SIMUL works in several stages. Suppose that the substructure is $\{A_1, ..., A_n\}$.

— call POLY for the evaluation of each $A_i$ ; if ‘ok’ turns to false, abort; if ‘scale’ is increased by a factor *s* while parsing $A_i$, multiply by *s* all tempo indicators in ‘OUT’ and the fractional lengths of $A_j$ for all $j < i$ ; let fixtempo[i], p[i], q[i], scale[i] and globaltempo[i] be the resulting outputs;

— determine *ii* such that either fixtempo[ii] is true or, if fixtempo[i] is false for all *i* then (p[ii],q[ii]) is maximum; in this process the check for consistency is performed: if both fixtempo[i] and fixtempo[j] are true, then it is necessary that (p[i],q[i]) = (p[j],q[j]) otherwise ‘ok’ is set to false and POLY is aborted;

— determine a set of integer coefficients: (r[1],..., r[n]) such that:

$$\forall\, i \in \{1,n\},\ (p[i],\ r[i].q[i]) = (p[ii],\ s.q[ii])$$
$$= \text{fractional length of the substructure}$$
$$= (P,\ Q)$$

This last part of procedure SIMUL is explained now.

The relation above may be written as follows:

$$\forall i \in \{1,n\},\ \frac{r[i].q[i]}{p[i]} = \frac{s.q[ii]}{p[ii]}$$

Let *L* be the lowest common multiple (LCM) of p[i] for all $i \in \{1,n\}$, and let p'[i] = L / p[i]. If we multiply by *L* both sides of the equation we obtain:

$$r[i].q[i].p'[i] = s.q[ii].p'[ii]$$

Let *M* be the LCM of (q[i].p'[i]) for all $i \in \{1,n\}$, and let q'[i] = M / (q[i].p'[i]). If we divide by *M* both sides of the equation , we obtain:

$$\forall i \in \{1,n\},\ \frac{r[i]}{q'[i]} = \frac{s}{q'[ii]}$$

The lowest values of r[i] and *s* satisfying this relation are:

$$r[i] = q'[i]$$
$$s = q'[ii] = M / (q[ii].p'[ii]) = M.L / (p[ii].q[ii])$$

The value of *s* is also the rate by which variable ‘scale’ is increased. The output value of ‘globaltempo’ is (M * p[ii] / L).

A feature which is not presented here is the handling of undetermined gaps. For example, if X = ‘a b c d e’ and Y = ‘f g h’, and if the configuration of time-span intervals between X and Y is $\beta 4$ (see §3), the corresponding structure is notated:

{ a b c d e ?, ? f g h}

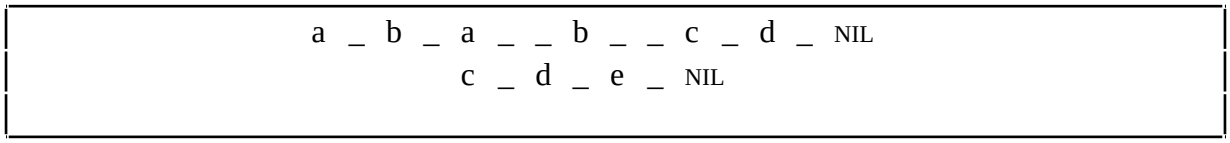
in which '?' denotes undetermined gaps. HyperBP finds the smallest gaps compatible with tempo constraints for which the maximum number of on-setting dates match with each other in both strings, e.g. {a b c d e _, _ _ _ f g h}. Other solutions may be found if more explicit information is given.

## 11. Examples of interpretations

**ab{ab,cde}cd**

= /3 ab { /2 ab, /3 cde} /3 cd = /6 a _ b _ {a _ _ b _ _, c _ d _ e _} c _ d _

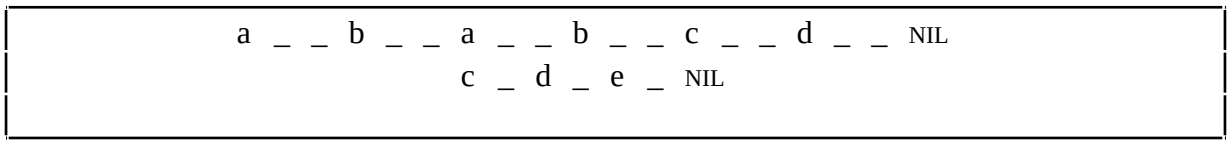
Duration 7/1 Scale 3




**ab{/1 ab,cde}cd**

= /2 ab { /2 ab, /3 cde} /2 cd = /6 a _ _ b _ _ {a _ _ b _ _, c _ d _ e _} c _ d _

Duration 6/1 Scale 2

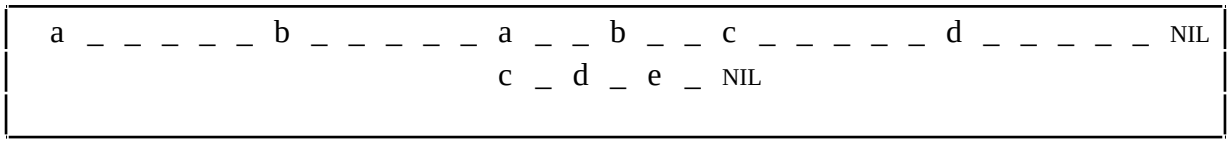



**ab{/2 ab,/3 cde}cd**

= /2 ab { /4 ab, /6 cde} /2 cd

= /12 a _ _ _ _ _ b _ _ _ _ _ {a _ _ b _ _ , c _ d _ e _ } c _ _ _ _ _ d _ _ _ _ _

Duration 5/1 Scale 1

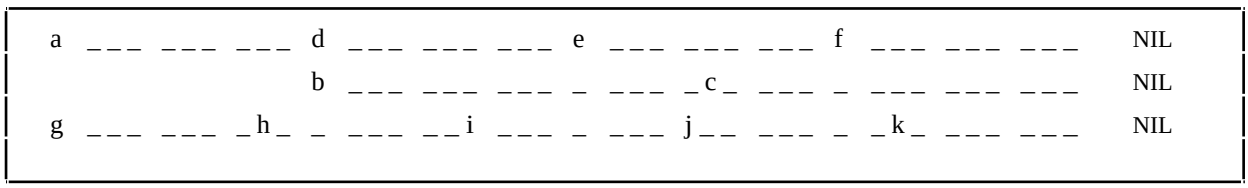



**{a {bc,def},ghijk}**

= { /12 a { /8 bc, /12 def}, /15 ghijk}

= /120 {a_ _ _ _ _ _ _ _ _ {b_ _ _ _ _ _ _ _ _ _ _ _ _ _ _c ...etc.

Duration 5/1 Scale 15




**/3 ab {ab,cde} cd**

= /9 ab { /6 ab, /9 cde} /9 cd = /18 a_b_{a_ _ b_ _, c_d_e_} c_d_

Duration 7/3 Scale 3

a _ b _ a _ _ b _ _ c _ d _ NIL
c _ d _ e _ NIL

## ACKNOWLEDGEMENTS

I feel indebted to Marcel Frémiot and Bernard Vecchione for their advisory contributions to this work, Jim Kippen who proofed several versions of this paper, Otto Laske who encouraged me to submit it, and Antonio Camurri who did the editorial work.

## REFERENCES

Allouche, J.P., & A. Mouret (1988). Libertés non anarchiques, automates finis et champs matriciels. *Proceedings of Colloque International "Structures Musicales et Assistance Informatique"*. Marseille (France): Laboratoire Musique et Informatique, 45-50.

Allen, J. F. & H. Kautz (1985). A Model of Naive Temporal Reasoning. In *Formal Theories of the Commonsense World*, J.R. Hobbs & R.C. Moore, Eds. Norwood NJ (USA): Ablex, 251-68.

Bel, B. (1989a). Logiciels pour SYG. Marseille (France): Laboratoire Musique et Informatique.

(1989b). SYG: théorie. Marseille (France): Laboratoire Musique et Informatique.

(1989c). Pattern grammars in formal representations of musical structures. Workshop on AI and Music, *11th Intern. Joint Conf. on Artificial Intelligence*. Detroit MI (USA),118-45.

(1989d, forthcoming). Langages formels et musique. *Rapport d'activités, groupe 'OC'*. Marseille (France): Laboratoire Musique et Informatique.

(1990, forthcoming). *Représentation, acquisition et production de connaissances en musique*, phD dissertation.

Boulez, P. (1963). *Penser la musique aujourd'hui*. Paris (France): Gonthier.

Chemillier, M. (1987). *Solfège, commutation partielle et automate minimal d'intersection.* Unpublished. Paris (France): Université Paris VII, UER de Mathématiques.

Chemillier, M., & D. Timis (1988). Toward a theory of formal musical languages. *14th International Computer Music Conference*, Cologne (FRG),175-83.

Jaffe, D. (1985). Ensemble timing in computer music. *Computer Music Journal*, vol.9, N°4, 38-48.

Kippen, J. (1988). *The Tabla of Lucknow: a Cultural Analysis of a Musical Tradition.* Cambridge (UK): Cambridge University Press.

Kippen, J. & B. Bel (1989, forthcoming). Modelling music with grammars: formal language representation in the Bol Processor. *Computer Representations and Models in Music*., A. Marsden and A. Pople, Eds., London (UK): Academic Press.

Laske, O. (1989). Composition Theory: An Enrichment of Music Theory. *Interface*, Vol.18, 45-59.

Leman, M. (1989). Emerging Properties of Tonality in a Self-Organizing System. (Abstract) Workshop on AI and Music, *11th Intern. Joint Conf. on Artificial Intelligence*. Detroit MI (USA), 102-4.

Mazurkiewicz, A. (1984a). Semantics of concurrent systems: a modular fixed-point trace approach. *5th European Workshop on Applications and Theory of Petri Nets*, 353-71.

(1984b). Traces, histories, graphs: instances of a process monoid. *Lecture Notes in Computer Science* 176, 115-33.

Minsky, M. (1986). *The Society of Mind*. New York (USA): Simon & Schuster.

Oppo, F. (1984). Per una teoria generale del languaggio musicale. *Musical Grammars and Computer Analysis*, M. Baroni & L. Callegari, Eds. Firenze (Italy): Olschki, 115-30.

Risch, V. (1988). PROLOG et les grammaires temporelles. *Proceedings of Colloque International "Structures Musicales et Assistance Informatique"*. Marseille (France): Laboratoire Musique et Informatique, 77-86.

Risset, J.C. (1989). Recent developments in musical data-processing and psychoacoustics. *Etat de la Recherche Musicale (au 1er janvier 1989)*. Aix-en-Provence (France): Agence Régionale pour la Coordination des Activités Musicales et Chorégraphiques.

Smoliar, S. (1989). Music Notation: Cognitive Red Herring?. Workshop on AI and Music, *11th Intern. Joint Conf. on Artificial Intelligence*. Detroit MI (USA), 53-62.

Van Benthem, J.F.A.K. (1983). *The Logic of Time*. Dordrecht (NL): Reidel.

Vecchione, B. (1984). *Pour une science de la réalité musicale*. Thèse de Doctorat de IIIème cycle, Université de Provence Aix-Marseille I.

(1985). *La réalité musicale: éléments d'épistémologie musicologique.* Thèse de Doctorat d'Etat, Université Paris VIII.

(1989). Vers une méthodologie de l'assistance informatique en analyse musicale et en composition (II): rappels de théorie algébrique des questionnements. Marseille (France): Laboratoire Musique et Informatique (MIM 52).

Xenakis, I. (1963). *Musiques formelles*. Paris (France): La Revue Musicale.

Augmented translation in (1971) *Formalized music*, Bloomington (USA): Indiana University Press.

(1972). Vers une métamusique. In *Architecture et Musique*, Paris (France): Casterman, 38-70.

Zielonka, W. (1987). Notes on finite asynchronous automata. *RAIRO: Informatique théorique et Applications*, vol. 21, n°2, 99-135.

Bernard Bel, GRTC
Centre National de la Recherche Scientifique
31 chemin Joseph Aiguier
F-13402 Marseille Cedex 09
*E-mail:* bel@frmop11.bitnet

Bernard Bel is a computer scientist with background in electronics. Since 1979 he has been collaborating with anthropologists, musicologists and musicians on a scientific study of North Indian melodic and rhythmic systems. In 1981 he built a sophisticated real-time *Melodic Movement Analyser* (MMA) with the aid of which he developed automatic transcription and analysis of *raga* music (in collaboration with Jim Arnold, Joep Bor and Wim van der Meer). In 1986 he joined GRTC, an AI laboratory of the *Centre National de la Recherche Scientifique* in Marseille. At present he is working on algorithms for automatic rule generation derived from hypotheses on training methods in traditional *tabla* (in collaboration with ethnomusicologist Jim Kippen). He is also currently involved in research on computer music in collaboration with the *Laboratoire Musique et Informatique de Marseille* (MIM).