DESCRIZIONE FORMALE DI PROCESSI MUSICALI

Goffredo HAUS

Istituto di Cibernetica - Milano -

Se si tiene presente che il termine "processo" può essere interpretato come l'evoluzione (od il comportamento) di un sistema e che la natura di questo sistema (fisica o logica) determina la natura del processo (fisica o logica), è possibile fare alcune considerazioni.

Uno strumento musicale che suona produce un processo fisico.

Una partitura di un brano musicale è la descrizione di un processo logico avvenuto nella mente del compositore; quindi una partitura è un meta processo.

La comunicazione tra compositore ed ascoltatore si può, in generale, schematizzare con il seguente concatenarsi di processi:

composizione	→	testo	→	interpretazione	→	produzione
pr. log.		metaprocesso		pr. log.		pr. fis.

del suono	→	percezione del suono	→	interpretazione
		pr. fis.		pr. log.

Più strumenti musicali che suonano contemporaneamente possono essere considerati i processori di più processi logici.

La percezione dei suoni da essi prodotti è un unico processo fisico dipendente da tutti i processi fisici prodotti dagli strumenti.

La mente può riconoscere i processi logici che il compositore vuole comunicare, dalla percezione per mezzo di processori quali la eduazione, la concentrazione, etc.; inoltre può crearne di nuovi.

Il nostro interesse sarà rivolto ai metaprocessi che descrivono i processi musicali logici.

Un processo può essere visto come la sequenza di un certo tipo di eventi.

Siano dati un generico alfabeto ed un'operazione di composizione interna +; allora sequenza si definisce ricorsivamente con:

a) $\sigma \in \Sigma$ è una sequenza;

b) se x è una sequenza, $x\sigma$ è una sequenza.

Con Σ^* si indica l'insieme delle sequenze su Σ.

Con $\wedge$ si indica la sequenza vuota.

Con Σ^* si indica l'insieme delle sequenze su Σ non vuota.

Si dice monoide libero la tripla $< \Sigma^*, \bullet , \wedge >$ in cui Σ^* è l'insieme delle sequenze su Σ , $\circ$ è la operazione di giustapposizione, $\wedge$ è la sequenza (o stringa) vuota.

Dati due monoidi liberi $< M, \bullet, 1 >$ e $< M', \bullet', 1' >$ si dice morfismo di monoidi una applicazione

$$\phi : < M, \circ , 1 > \to < M', \circ', 1' >$$

tale che

a) $\phi(1) = 1'$

b) $\phi(x\ y) = \phi(x) \quad ' \phi(y)$

Si dice isomorfismo una applicazione che gode delle proprietà di un morfismo e di una corrispondenza biunivoca.

Le strutture algebriche definite possono essere interpretate in senso musicale.

Si possono definire alcuni operatori su Σ^* in Σ^* particolarmente utili al fine di produrre evoluzioni di stringhe.

Σ è un sottoinsieme dell'insieme N dei numeri interi in corrispondenza biunivoca con i diversi simboli dei valori di uno dei parametri sonori.

Ad esempio, le dodici note del sistema temperato possono essere rappresentate dall'insieme $\{0,1,2,3,4,5,6,7,8,9,10,11\}$.

Vogliamo quindi definire un insieme di operatori a partire dal gruppo abeliano $< \Sigma,+,\circ >$.

L'operazione di composizione interna $+$ è definita da:

a) $$\sigma_1 + \sigma_2 = \begin{cases} \sigma_1 + \sigma_2 & \text{se} \quad \sigma_1 + \sigma_2 < 12 \\ \sigma_1 + \sigma_2 - 12 & \text{se} \quad \sigma_1 + \sigma_2 \geq 12 \end{cases}$$

con $\sigma_1, \sigma_2 \in \Sigma$

b) $$\sigma_0 + x = (\sigma_0 + \sigma_1)(\sigma_0 + \sigma_2) \,\ldots\ldots\, (\sigma_0 + \sigma_n)$$

con $\sigma_0 \in \Sigma$ e $x = \sigma_1 \sigma_2 \ldots\ldots \sigma_n \in \Sigma^*$.

Si dice <u>operatore di trasposizione</u> su Σ^+ di grado t e si indica con $T(t)$ un operatore così definito:

$$T(t) : \Sigma^+ \to \Sigma^+ \quad \text{con} \quad t \in \Sigma$$

tale che

a) $T(t)(\sigma) = t + \sigma$

b) $T(t)(x\sigma) = T(t)(x)\ T(t)(\sigma)$.

L'operatore di trasposizione $T(t)$ è un isomorfismo di Σ^+ in sé stesso.

Ad esempio: siano $1,2,3,4,5 \in \Sigma$; allora

$$2345 = T(1)(1234) = T(1)(12)\ T(1)(34) = 2345.$$

Si dice classe di operatori $\mathcal{T}$ la classe degli operatori $T(t)$ tali che $T(t) \in \mathcal{T}$ e $t \in \Sigma$.

La cardinalità dell'insieme $\mathcal{T}$ è uguale alla cardinalità di Σ.

Un particolare operatore della classe è l'operatore $T(0)$ che traspone $\sigma \in \Sigma$ in sé stesso; cioé

$$T(0)(\sigma) = 0 + \sigma = \sigma$$

$T(0)$ si può chiamare operatore di identità I definito da:

a) $I(\sigma) = \sigma$

b) $I(x\sigma) = I(x)\ I(\sigma)$.

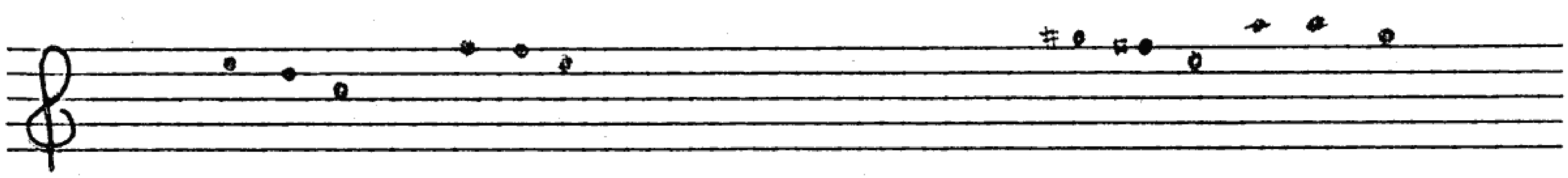
L'operatore di trasposizione è particolarmente utile se applicato alle altezze dei suoni; serve per compiere funzioni della musica tradizionale quali le progressioni o le modulazioni.

Ad esempio, le progressioni possono essere di due tipi (Schoenberg,1967):

a) progressioni modulanti;

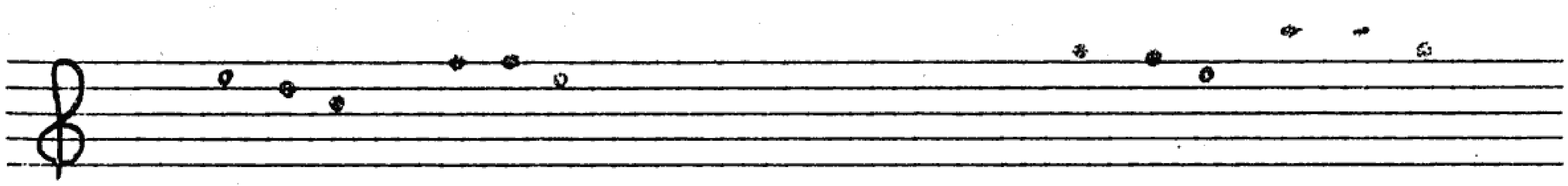
b) progressioni tonali.

Se si assume come alfabeto l'insieme $\Sigma = \{0,1,...,11\}$ l'operatore di trasposizione $T(t)$ produce una modulazione di t semitoni su una sequenza $x \in \Sigma^+$.



x=420554 T(4)(420554)=864998

Se si assume come alfabeto l'insieme $\Sigma = \{0,1,2,3,4,5,6\}$, in cui i sette numeri rappresentano i sette gradi di una tonalità maggiore (ad esempio il "do maggiore"), l'operatore di trasposizione $T(t)$ produce una progressione tonale di grado t-esimo su una sequenza $x \in \Sigma^+$.



x=210332 T(2)(210332)=432554

Si dice <u>operatore di inversione speculare</u> su Σ^+ rispetto ad i e si indica con $S(i)$ un operatore così definito:

$$S(i) : \Sigma^+ \to \Sigma^+ \quad \text{con} \quad i \in \Sigma$$

tale che

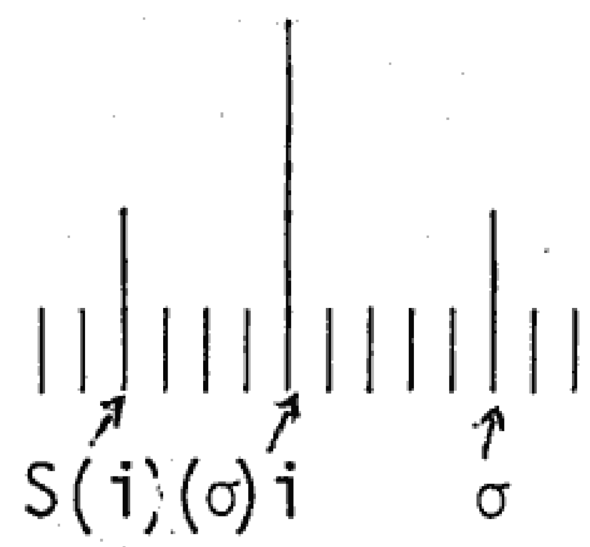
a) $S(i)(\sigma) = 2i - \sigma$

b) $S(i)(x\sigma) = S(i)(x)\ S(i)(\sigma)$.

L'operatore di inversione speculare $S(i)$ è un isomorfismo di Σ^+ in sé stesso.

Infatti: sia $\Sigma = \{0,1,\dots,11\}$; allora

$2345 = S(1)(0\ 11\ 10\ 9) = S(1)(0\ 11)\ S(1)(10\ 9) = 2345$



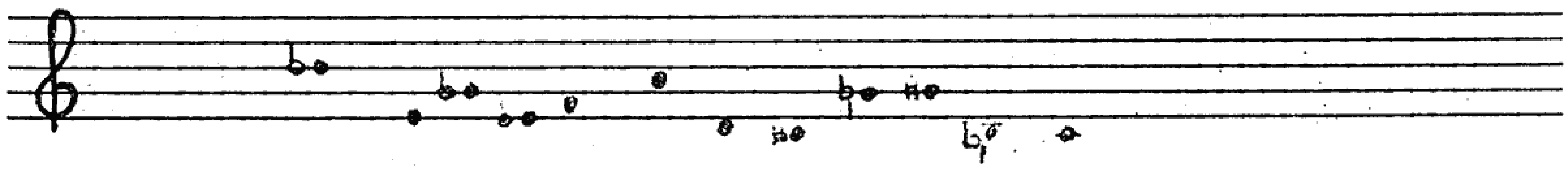
Si dice classe di operatori $\mathcal{S}$ la classe degli operatori $S(i)$ tali che $S(i) \in \mathcal{S}$ e $i \in \Sigma$.

La cardinalità dell'insieme $\mathcal{S}$ è uguale alla cardinalità di Σ.

L'operatore di inversione speculare applicato alle sequenze di altezze compie la funzione di invertire tutti gli intervalli di una sequenza data.

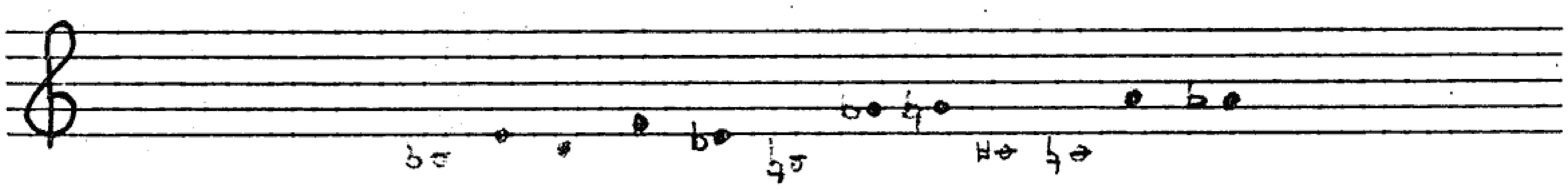
Questa funzione è molto importante nella musica degli ultimi secoli; J.S. Bach utilizzò molto spesso questa funzione, soprattutto nelle ultime opere quali l'"Arte della fuga" e l'"Offerta musicale"; anche Beethoven utilizzò spesso la inversione nelle sue ultime composizioni quale il "Quartetto per archi in fa maggiore, Op. 135; l'inversione è diventata anche una delle forme di manipolazione del materiale seriale sonoro fondamentali; i musicisti della scuola dodecafonica di Vienna, infatti, hanno assunto que sto tipo di variazione come strumento tra i basilari per la composizione (Rufer, 1954).

Ad esempio, nelle "Variazioni per orchestra, Op. 31" di Arnold Schoenberg la serie dodecafonica di base è



x=10 4 6 3 5 9 2 1 7 8 11 0

e viene spesso usata la sua inversione speculare rispetto ad $i = 4$ cioè rispetto al mi :



S(4)(x)=10 4 2 5 3 11 6 7 1 0 9 8

Gli operatori di inversione speculare su alfabeti di cardinalità pari sono uguali a due a due.

Infatti se si prende $\Sigma = \{0,1,\ldots,11\}$, si ha che

$$S(i)(\sigma) = S(i+6)(\sigma)$$

Si dice <u>operatore di retrogradazione</u> su Σ^* e si indica con R un operatore così definito:

$$R : \Sigma^* \to \Sigma^*$$

tale che

a) $R(\wedge) = \wedge$

b) $R(\sigma) = \sigma$

c) $R(x\sigma) = \sigma R(x)$

L'operatore di retrogradazione R è una corrispondenza biunivoca tra coppie di elementi di Σ^*, ma non è un isomorfismo di Σ^* in sé stesso.

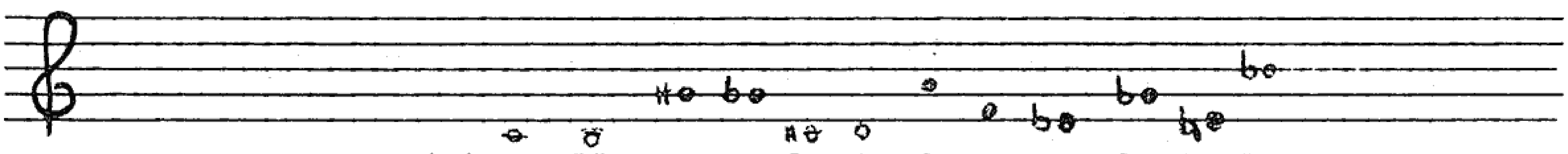
Infatti: sia $\Sigma = \{0,1,\ldots,11\}$; allora

$4321 = R(1234) \neq R(12)\,R(34) = 2143$.

L'operatore di retrogradazione applicato ad uno qualunque dei parametri del suono compie la funzione di rovesciare l'ordine delle sequenze di simboli relative ai diversi parametri.

Applicato alle altezze dei suoni ha la stessa importanza storica dell'operatore di inversione speculare, essendo stato ampiamente usato da compositori come Bach, Beethoven, Schoenberg, Berg, Webern e perfino dai compositori delle correnti seriali d'avanguardia (Boulez, Stockhausen, etc.).

Ad esempio, nelle "Variazioni per orchestra, Op. 31" di Arnold Schoenberg la serie dodecafonica di base riportata precedentemente viene spesso utilizzata nella sua forma retrogradata.

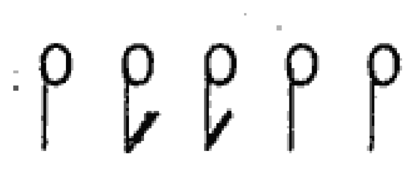


R(x)=0 11 8 7 1 2 9 5 3 6 4 10

Applicato alla sequenze di durate, l'operatore di retrogradazione R nella manipolazione di strutture ritmiche è di particolare utilità.

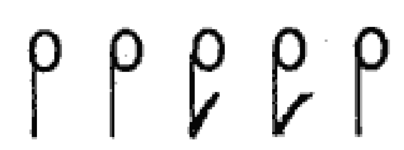
Ad esempio, se si prende $\Sigma = \{0,1,2,3,4,5\}$ dove $0 \leftrightarrow \rho$, $1 \leftrightarrow \rho$, $2 \leftrightarrow \rho$, $3 \leftrightarrow \rho$, $4 \leftrightarrow \rho$, $5 \leftrightarrow \rho$, si possono descrivere facilmente alcune figure ritmiche.

Nel "Canone perpetuo" dalla "Offerta musicale" di J.S.Bach la struttura ritmica di alcune battute è data dalla sequenza:



$x = 34433$

Nel corso della composizione viene spesso utilizzata la sequenza:

 $R(x) = 33443$

Considerazioni analoghe valgono per le sequenze che descrivono l'evoluzione del volume del suono.

Si indica con $t_i(s,m)$ un operatore che compie un'azione di trasposizione sulla sottosequenza che va dallo s-esimo elemento allo m-esimo elemento della sequenza $x \in \Sigma^+$ e che giustappone la stringa trasposta alla sequenza data x.

Ad esempio, se $x = 1\ 2\ 3\ 4\ 5$, $s = 2$, $m = 3$, $i = 1$, con $s, m \in N$

allora

$$xt_i(s,m) = 1\ 2\ 3\ 4\ 5\ \ t_1(2,3) = 1\ 2\ 3\ 4$$

Si indica con $s_i(s,m)$ un operatore che compie un'azione di inversione speculare rispetto ad i sulla sottosequenza che va dallo s-esimo elemento allo m-esimo elemento della sequenza $x \in \Sigma^+$ e che giustappone la stringa invertita specularmente alla sequenza data x.

Ad esempio, se $x = 2\ 5\ 7\ 8$, $s = 1$, $m = 3$, $i = 4$, con $s,m \in N$ allora

$$xs_i(s,m) = 2\ 5\ 7\ 8\ \ s_4(1,3)) = 2\ 5\ 7\ 8\ 6\ 3\ 1\ .$$

Si indica con $r(s,m)$ un operatore che compie un'azione di retrogradazione sulla sottosequenza che va dallo s-esimo elemento allo m-esimo elemento della sequenza $x \in \Sigma^*$ e che giustappone la stringa retrogradata alla sequenza data x.

Ad esempio, se $x = 3\ 4\ 5\ 6\ 7\ 8\ 9$, $s = 2$, $M = 6$ con $s,m \in N$

allora

$$xr(s,m) = 3\ 4\ 5\ 6\ 7\ 8\ 9\ \ r(2,6) = 3\ 4\ 5\ 6\ 7\ 8\ 9\ 7\ 6\ 5\ 4\ .$$

Gli operatori t_i, s_i, r possono essere composti tra loro, cioé possono essere applicati più operatori alla stessa sottosequenza di una sequenza data x.

Si dice classe di operatori Π la classe degli operatori $\pi(s,m)$ tali che $\pi(s,m)$ sia scomponibile in uno o più operatori t_i, s_i, r con $s,m \in N$.

La cardinalità dell'insieme Π è uguale a quattro volte la cardinalità di Σ.

Infatti, dalle possibili combinazioni degli operatori che sono stati definiti si possono ottenere solo quattro sottoclassi della classe Π ciascuna avente cardinalità uguale alla cardinalità di Σ.

Se si indica con t_i^n, s_i^n , r^n la applicazione per n volte degli operatori t_i, s_i, r rispettivamente, gli operatori $\pi \in \Pi$ si possono indicare con i seguenti quattro tipi:

a) <u>operatori t_{i+}</u> : si ottengono dalla combinazione di un numero pari o dispari di operatori t_j con un numero pari di operatori a_k, in un ordine qualunque;

b) <u>operatori t_{i-}</u> : si ottengono dalla combinazione di un numero pari o dispari di operatori t_j con un numero dispari di operatori s_k, in un ordine qualunque;

c) <u>operatori rt_{i+}</u> : si ottengono dalla combinazione di un numero dispari di operatori r con un operatore t_{i+}, in un ordine qualunque;

d) <u>operatori rt_{i-}</u> : si ottengono dalla combinazione di un numero dispari di operatori r con un operatore t_{i-}, in un ordine qualunque.

E' opportuno ricordare che:

a) $r^n(s,m) = t_0(s,m)$ se n è un numero pari;

ad esempio, se $x = 2\ 3\ 4\ 5$, $s = 1$, $m = 4$, $n = 2$ si ha

$2\ 3\ 4\ 5\ r(r(1,4)) = 2\ 3\ 4\ 5\ r(5\ 4\ 3\ 2) = 2\ 3\ 4\ 5\ 3\ 4\ 5 = 2\ 3\ 4\ 5\ t_0(1,4)$

b) $t_i(t_k(s,m)) = t_{i+k}(s,m) = t_k(t_i(s,m)) = t_{(i+k)+}(s,m)$

c) $t_i(s_k(s,m)) \neq s_k(t_i(s,m))$

d) $s_i(s_k(s,m)) \neq s_k(s_i(s,m))$

e) $r(s_i(s,m)) = s_i(r(s,m))$

f) $r(t_i(s,m)) = t_i(r(s,m))$

g) $t_i(s_k(s,m)) = t_{(i+2k)-}(s,m)$

h) $s_k(t_i(s,m)) = t_{(2k-i)-}(s,m)$

i) $s_i(s_k(s,m)) = t_{(2i-2k)+}(s,m)$

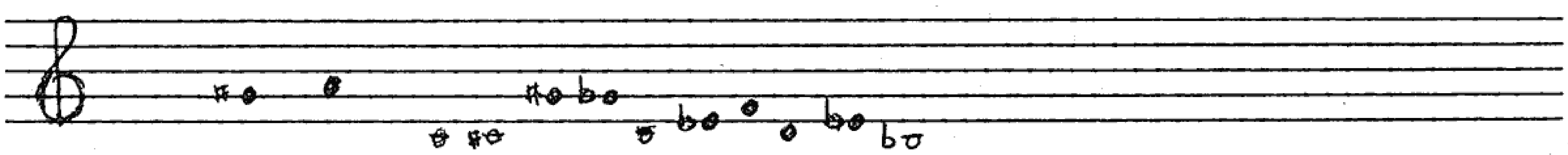
l) $s_k(s_i(s,m)) = t_{(2k-2i)+}(s,m)$

Gli operatori π così costruiti permettono di compiere funzioni più complesse sulle sequenze; nella musica si è fatto spesso riscontro in funzioni (soprattutto melodiche ed armoniche) di complessità di questo tipo.

La forma di composizione più usata è la retrogradazione di sottosequenze che sono la inversione speculare di sottosequenze di sequenze date (o, che è lo stesso, la inversione speculare di sottosequenze che sono la retrogradazione di sotto sequenze di sequenze date).

Questa funzione complessa è stata molto usata da J.S.Bach e da tutti i suoi epigoni, e da tutti i compositori che abbiano fatto uso delle tecniche compositive dodecafoniche o, più in generale, seriali.

Ad esempio, nelle "Variazioni per orchestra, Op. 31" di Arnold Schoenberg la serie dodecafonica di base (che è, come abbiamo già ricordato,

$x = 4\ 6\ 3\ 5\ 9\ 2\ 1\ 7\ 8\ 11\ 0$) viene spesso utilizzata retrogradata ed invertita specularmente rispetto al mi; cioè:

$r(s_4(s,s+11))=s_4(r(s,s+11))=rt_{8-}(s,s+11)) =$

$=8\ 9\ 0\ 1\ 7\ 6\ 11\ 3\ 5\ 2\ 4\ 10$

essendo la sottosequenza che va dallo elemento s-esimo allo elemento (s+11)-esimo la serie x.

Possiamo dare ora una sintessi che generi delle liste finite di elementi di Σ che chiameremo processori $\rho \in \mathcal{P}$.

A questa sintassi assoceremo poi una interpretazione dei processori così definiti su Σ.

Sintassi

a) $\sigma \in \Sigma \Longrightarrow \sigma \in \mathcal{P}$

b) $x \in \mathcal{P} \Longrightarrow$ b1) $x\sigma \in \mathcal{P}$

b2) $\omega(x) \geq m \Longrightarrow x\pi(s,m) \in \mathcal{P} \quad \forall\ \pi \in \Pi$

dove ω è una misura a valori in N e definita su $\mathcal{P}$ che indica la lunghezza della sequenza che il processore x realizza; ω è tale che:

a) $\omega(\sigma) = 1$

b) $\omega(x\sigma) = \omega(x)+1$

c) $\omega(xt_i(s,m)) = \omega(x)+m-s+1$

d) $\omega(xs_i(s,m)) = \omega(x)+m-s+1$

e) $\omega(xr(s,m)) = \omega(x)+m-s+1$

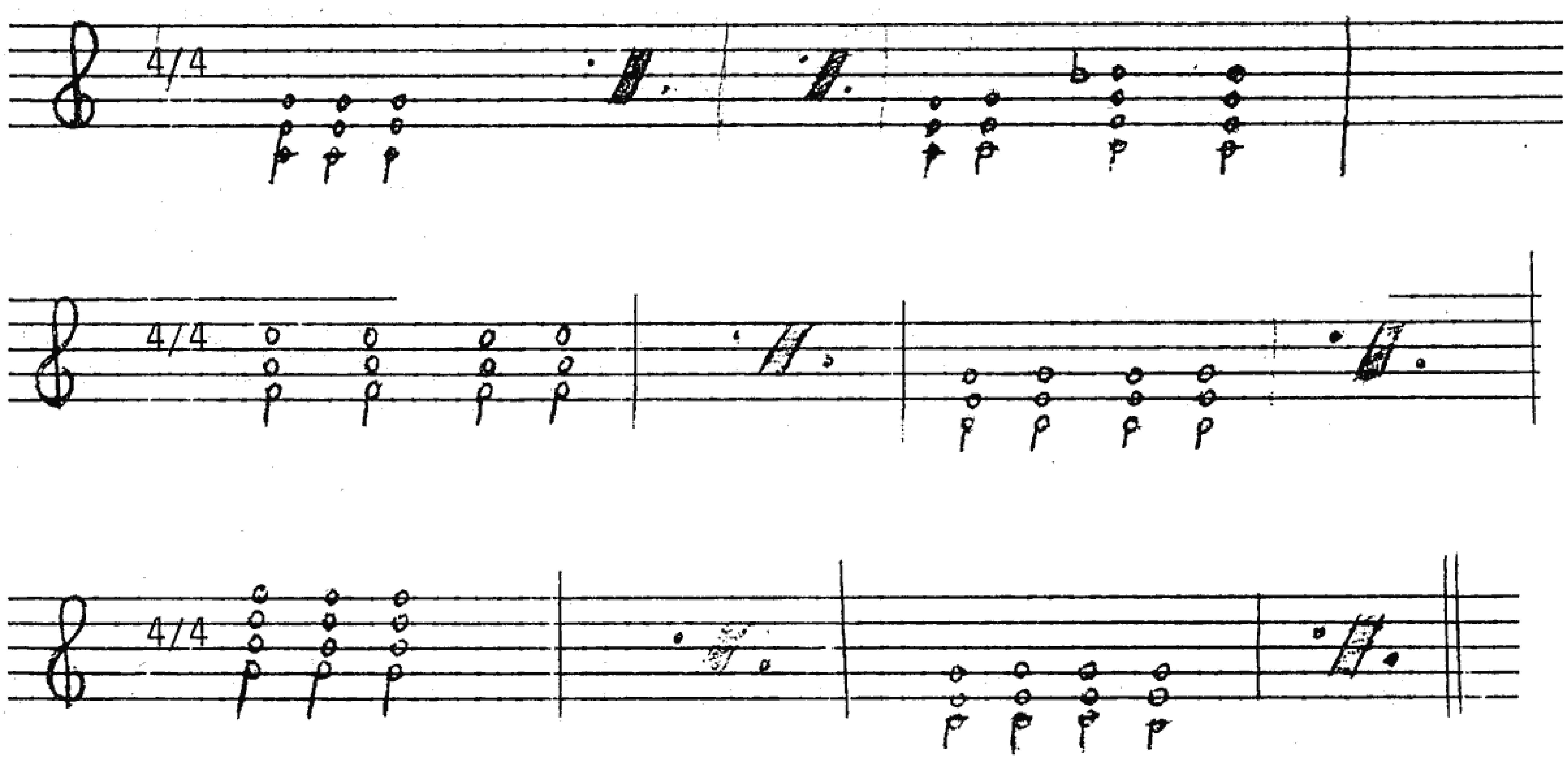
f) $\omega(x\ (x,m)) = \omega(x)+m-s+1$

Si definisce $\ell(x)$ una misura a valori in N che indica il numero di elementi di un processore (o, più in generale, di una sequenza).

Vediamo due esempi di processori.

1) Sia $\Sigma = \{0,1,2,3\}$ l'alfabeto in cui $0 \leftrightarrow$ accordo di do maggiore $1 \leftrightarrow$ accordo di do maggiore con la settima diminuita, $2 \leftrightarrow$ accordo di fa maggiore, $3 \leftrightarrow$ accordo di sol maggiore settima.

Allora, il semplice "giro" di blues



può essere realizzato dal processore

$$P = 0\ t_0(1,1)\ t_0(1,2)\ t_0(1,4)\ t_0(1,6)\ 1\ t_0(15,15)\ 2\ t_0(17,17)\ t_0(17,18)\ t_0(17,20)\ t_0(1,8)\ 3\ t_0(33,33)\ t_0(33,34)\ t_0(33,36)\ t_0(1,8).$$

Il processore P ha $\ell(P) = 17$ ed $\omega(P) = 48$.

2) Lo stesso "giro" di blues può essere realizzato dal processore

$P' = 0\ t_0(1,1)\ t_0(1,2)\ t_0(1,4)\ t_0(1,6)\ t_1(1,2)\ t_2(1,8)\ t_0(1,8)$
$t_3(1,8)\ t_0(1,8)$.

Il processore P' ha $\ell(P') = 10$ ed $\omega(P') = 48$.

Sia $\Sigma = \{0,1,......,11\}$.

Il seguente esempio è una stringa che non è un processore:

$x = 2\ 4\ 3\ 5\ r(3,8)\ 5\ 8$ ....

Infatti, $\omega(x') < m$ essendo $x' = 2\ 4\ 3\ 5$, poiché $\omega(x') = 4$ ed $m = 8$.

Per verificare che una stringa sia un processore, è sufficiente controllare, procedendo da sinistra a destra, che m non sia superiore ad $\omega(x)$ ogni volta che si incontra un operatore nella sottostringa x del processore P, di cui è già stata controllata la correttezza della sottostringa y tale che $\omega(y) + 1 = \omega(x)$.

Semantica

Si dice semantica l'interpretazione dei processi sullo insieme delle sequenze di elementi di Σ.

La semantica che definiremo sarà ottenuta assegnando delle regole che associano sequenze ad ogni processore.

Le regole sono:

$W : P \to \Sigma^+$ è definita da:

a) $W(\sigma) = \sigma$

b) $W(x\sigma) = W(x)\sigma$

c) $W(x\pi(x,m)) = W(x)y$ dove y è la stringa generata dall'applicazione dello operatore $\pi \in \Pi$ alla sottosequenza di $W(x)$ che va dallo s-esimo allo m-esimo elemento.

La definizione del semplice linguaggio P dei processori che sono generati dalla sintassi ed interpretati dalla semantica suddette, permette di formulare alcuni teoremi.

Teorema (1) - Ad ogni processore $P \in \mathcal{P}$ è associata una e una sola sequenza $W(P) \in \Sigma^+$, ed inoltre $\omega(P) = \ell(W(P))$.

Dim. - (a) Ai processori di tipo $\sigma \in \Sigma$ è associata l'unica sequenza σ, cioé $W(\sigma) = \sigma$ (regola a), semantica), ed inoltre $\omega(\sigma) = 1 = \ell(\sigma) = \ell(W(\sigma))$

(b) Supponiamo, per ipotesi di induzione, che il teorema sia verificato per il processore x: allora al processore x è associata la sequenza $W(x)$ ed inoltre $\omega(x) = \ell(W(x))$.

Si hanno due casi:

i) sia $x\sigma$ un processore: allora $W(x\sigma) = W(x)\sigma$ (regola b), semantica); poiché $W(x)$ è unica, allora anche $W(x\sigma)$ è unico, ed inoltre:

$$\omega(x\sigma) = \omega(x)+1 = \ell(W(x))+1 = \ell(W(x)\sigma) = \ell(W(x\sigma)).$$

ii) sia $x\pi(s,m)$ un processore, allora $m \leq \omega(x)$ (regola b), sintassi); esiste allora una unica sottosequenza tra il posto s ed il posto m; diciamo y tale sottosequenza; allora $W(x\pi(s,m)) = W(x)y$ (regola c),semantica); quindi, poiché $W(x)$ è unica, anche $W(x\pi(x,m))$ è unica ed inoltre:

$$\omega(x\pi(s,m)) = \omega(x) + m - s+1 = \ell(W(x)) + m-s+1 =$$
$$= \ell(W(x)y) = \ell(W(x\pi(s,m))) .$$

c.v.d.

Teorema (2) - Se x è un processore, si ha $\omega(x) \geq \ell(x)$.

Dim. - (a) Per i processori di tipo $\sigma \in \Sigma$ vale:

$$\omega(\sigma) = 1 = \ell(\sigma)$$

(b) Ammettendo per ipotesi di induzione che il teorema sia verificato per il processore x, cioé che $(x) \geq \ell(x)$, allora:

i) se x è un processore:

$$\omega(x\sigma) = \omega(x)+1 \geq \ell(x) + 1 = \ell(x\sigma)$$

ii) se $x\pi(s,m)$ è un processore:

$$\omega(x\pi(s,m)) = \omega(x)+m-s+1 > \ell(x)+m-s+1 \geq \ell(x)+1 =$$
$$= \ell(x\pi(s,m)).$$

c.v.d.

Il teorema (1) mette in evidenza che la definizione data di semantica è adatta ai nostri scopi; infatti dalla definizione data risulta che ogni processore è una particolare descrizione di una sequenza di elementi dello alfabeto Σ.

Il teorema (2) dimostra che la descrizione di sequenze per mezzo dei processori che sono stati definiti produce una economia nella descrizione stessa.

La presenza di questa economia induce a cercare di dare una valutazione delle grandezza che essa può assumere.

Introduciamo perciò una misura della complessità di descrizione di una sequenza di elementi di Σ rispetto al linguaggio dei processori P.

Si dice <u>complessità descrittiva</u> della sequenza $S \in \Sigma^+$ rispetto al linguaggio dei processori P il numero reale $\Phi(S)$ dove

$$\Phi : \Sigma^+ \to (0,1|$$
$$\Phi(S) = \operatorname*{Min}_{\{P|W(P)=S\}} \frac{\ell(P)}{\omega(P)}$$

$\Phi(S)$ ha le seguenti proprietà:

a) se $S = \sigma$ allora $\Phi(\sigma) = 1$

b) $\Phi(S) \leq 1$ per il teorema (2)

c) $\Phi(S) \geq \frac{\log_2 \ell(S)}{\ell(S)}$ poiché se si prende la sequenza S che ha intuitivamente la complessità descrittiva minima, cioé σ ripetuto per P^n volte, si può vedere che il processore di lunghezza minima che la produce ha lunghezza n+1;

$\sigma\ t_{o+}(1,1)\ t_{o+}(1,2)\ t_{o+}(1,4) \ldots t_{o+}(1,2^{n-1})$

(P)	1	2	3	4	n+1
(P)	1	2	4	8	2^n

posto $2^n = \ell(S)$, per $n \to \infty$ si ha la c).

$\Phi(S)$ può essere considerato anche una misura del tasso di informazione della stringa S di lunghezza $\ell(S)$.

Per dare una valutazione più approssimata dello estremo inferiore di $\Phi(S)$ è utile definire una nuova variabile.

Si dice <u>complessità descrittiva minima sulle sequenze di lunghezza n</u> il numero:

$$\psi(n) = n \inf_{\ell(S)=n} \Phi(S)$$

Allora vale il seguente teorema.

Teorema (3) - $\psi(n) = k$ è tale che si ha sempre

$$2^{k-2} + 1 \leq n \leq 2^{k-1}$$

Dim. - E' immediato verificare l'enunciato se si considera il seguente schema:

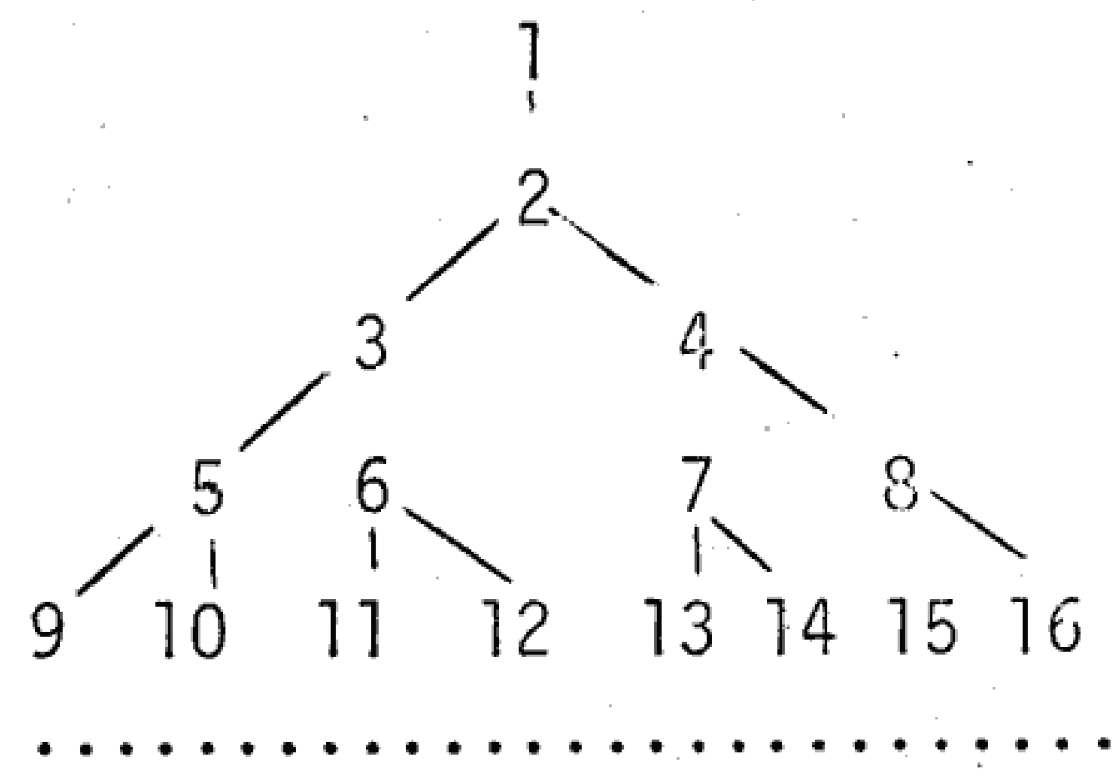
$k = 1 \quad n = 1$

$K = 2 \quad n = 2$

$k = 3 \quad 3 \leq n \leq 4$

$k = 4 \quad 5 \leq n \leq 8$

$k = 5 \quad 9$ _ n _ 16



Corollario (1) - $\psi(n)$ è asintotico a $\log_2 n$.

Corollario (2) - $\psi(n) > \log_2 n$

Infatti, per il teorema (3), si ha:

$$n \leq 2^{k-1} \qquad \text{da cui}$$

$$\log_2 n \leq k - 1 < k \quad .$$

Tenendo conto della definizione di complessità minima di descrizione di sequenze di lunghezza $\ell(S)$ e del corollario (2), è immediato verificare il segunete teorema.

Teorema (4) - $\dfrac{\log_2 \ell(S)}{\ell(S)} < \dfrac{\psi(\ell(S))}{\ell(S)} \leq \Phi(S) \leq 1$

dove $\psi(\ell(S))$ è asintotico a $\log_2 \ell(S)$.

Dim. - Per la definizione di $\psi(\ell(S))$ si ha

$\psi(\ell(S)) \leq \Phi(S)\ \ell(S)$; per il corollario (2) si ha

$\log_2 \ell(S) < (\ell(S))$ e quindi $\dfrac{\log_2 \ell(S)}{\ell(S)} < \dfrac{\psi(\ell(S))}{\ell(S)}$.

Ricordando il corollario (1), segue l'enunciato.

Abbiamo dato una valutazione dell'intervallo entro cui possono variare i valori della complessità descrittiva di una sequenza S rispetto al lin̲guaggio dei processori P.

Ora vogliamo cercare quale è il numero $N(k)$ di processori per cui $\omega(P) = k$, cioé il numero di processori che produce sequenze di lunghezza k.

Ricordiamo che abbiamo definito un insieme di operatori la cui cardinalità è 48; (vedi nota ++).

Analizzando la definizione ricorsiva di processori, si ottiene facilmente il seguente sistema:

$$(1) \qquad \begin{aligned} N(1) &= 12 \\ N(k) &= 48 \sum_{j}^{[k/2]} (k-2j)\, N(k-j) \quad . \end{aligned}$$

Si può valutare un semplice estremo inferiore per i valori di $N(k)$.

Dal sistema (1) si ottine facilmente:

$$N(k) \geq 48(k-2)N(k-1)$$

$$N(k) \geq 48^{j}(k-2)(k-3)(k-4)\ldots(k-j-2)N(k-j-1).$$

Ed in particolare si ha:

$$N(k) \geq 48^{k-3}(k-2)!\; N(3) \ .$$

Sviluppando il fattoriale di $(k-2)$ secondo Stirling si ottiene che

$$N(k) \geq \frac{48^k}{3^k}\, k^k$$

Tenendo presente che le sequenze di lunghezza k sono 12^k, si può concludere che, quindi, esiste <u>almeno</u> una sequenza di lunghezza k processata da almeno

---

++ Nota: ammesso di avere scelto $\Sigma = \{0,1,\ldots,11\}$

$$\frac{N(k)}{12^k} \geq \frac{4^k}{3^k} k^k \qquad \text{processori.}$$

Essendo il risultato trovato estremamente alto, è necessario cotruire un algoritmo che, data una sequenza $S \in \Sigma^+$, trovi un processore che descrive la sequenza S in $\psi(S)$ passi, cioè tale che sia $\psi(S) = \ell(P)$ con $P \in P$, con un numero di passi di ordine inferiore a k^k.

Ecco un algoritmo che presenta questa proprietà.

<u>Algoritmo (1)</u>

Siano $x = S$ e $P = \text{-}$ le condizioni iniziali.

(1) E' $x = \sigma$? Se <u>SI</u> poni $P = \sigma P$ - <u>ALT</u>

SE <u>NO</u> va a (2)

(2) Esiste il massimo x_1 tale
che $x = ax_1bx_2$ con $\pi(x_k) = x_2$
per qualche $\pi \in \Pi$? Se <u>SI</u> va a (3)

Se <u>NO</u> va a (4)

(3) Poni $x = a_1b$

$s = \ell(a)+1$

$m = \ell(ax_1)$

$P = \pi(s,m)P$

Va a (1).

(4) Sia $x = y\sigma$ $(\sigma \in \Sigma)$

Poni $x = y$

$P = \sigma P$

Va a (1) .

Se, ad esempio, fosse $S = 121231234$ si avrebbero i seguenti passi di esecuzione:

$x = 121231$

$P = t_{1+}(3,5)$

$x = 12123$

$P = t_{0+}(3,3)\ t_{1+}(3,5)$

$x = 121$

$P = rt_{1+}(2,3)\ t_{0+}(3,3)\ t_{1+}(3,5)$

$x = 12$

$P = t_{0+}(1,1)\ rt_{1+}(2,3)\ \ldots$

$x = 1$

$P = t_{1+}(1,1)\ t_{0+}(1,1)\ \ldots$

$P = 1\ t_{1+}(1,1)\ t_{0+}(1,1)\ rt_{1+}(2,3)\ t_{0+}(3,3)\ t_{1+}(3,5)$

ALT

L'algoritmo descritto opera con un numero di passi proporzionale a

$$\sum_{1}^{n} k = \frac{n(n-1)}{2} \qquad \text{che è asintetico a} \quad \frac{n^2}{2}$$

su una sequenza di lunghezza n ed è quindi molto conveniente se si confronta l'ordine di grandezza di n^2 con l'ordine di grandezza di n^n.

Vediamo ora due diverse valutazioni del tempo di calcolo necessario per la determinanzione di un processore di una sequenza S di lunghezza n con l'algoritmo (1), e quindi con le proprietà richieste.

La seconda strada che è stata seguita rende conto della proporzionalità del tempo di calcolo con il quadrato della lunghezza della sequenza.

1) Dato S tale che $\ell(S) = n$, determiniamo un algoritmo intuitivo per calcolare il massimo x_1 tale che $S = ax_1\ bx_2$ con $\pi(x_1) = x_2$ per un operatore $\pi \in \Pi$ fissato.

a) Si fissi σ_n (l'ultimo elemento di S) e si determini se σ_n è ottenibile per mezzo dell'operatore π fissato dalla stringa $S-\sigma_n$.
L'esecuzione di a) richiede al più n-1 passi.

b) Si fissi $\sigma_{n-1}\ \sigma_n$ (gli ultimi due elementi di S) e si determini se $\sigma_{n-1}\ \sigma_n$ è ottenibile per mezzo dello operatore π fissato dalla stringa $S - \sigma_{n-1}\sigma_n$.

L'esecuzione di b) richiede al più n-3 passi.

c) ....
....
....

Il processo di ferma quando <u>non esiste</u> alcun $\sigma_{n-s_i-1} \ldots \sigma_n$ ottenibile per mezzo dell'operatore π fissato dalla stringa $S - \sigma_{n-s_i-1} \ldots \sigma_n$.

Il tempo di calcolo necessario per trovare il massimo x_1 con le proprietà che abbiamo richiesto) dato da:

$$t_o(n,s_i) = (n - 1) + (n - 3) + \ldots + (n-2s_i+1)$$

che è circa uguale a

$$s_i^2 \left(\frac{n}{2} - \frac{2s_i}{3}\right)$$

quindi

$$t_c(n,s_i) \leq n \frac{s_i^2}{2} \quad .$$

Il procedimento va avanti fino alla fine della costruzione del processore, trovando di volta in volta i diversi s_j.

Supponiamo allora che sia

$$n = s_1 + s_2 + \ldots . + s_j + \ldots + s_i \ .$$

Il tempo di calcolo necessario per la determinazione completa del processore sarà allora tale che

$$T_{c\pi}(n) \leq \frac{n}{2} \sum_{j-1}^{i} s_j^2 \leq \frac{n^3}{2} \quad .$$

Poiché inoltre tali operazioni devono essere compiute tante volte quanti sono gli operatori, si avrà in definitiva:

$$T_c(n) \leq 24n^3 \; .$$

2) Applicando una semplice estensione dell'algoritmo introdotto in (Paterson, 1973), si osserva che il tempo di calcolo necessario per calcolare il massimo x_1 tale che $S = ax_1 \; bx_2$ con $\pi(x_1) = x_2$ per qualche $\pi \in \Pi$ può essere valutato in un tempo proporzionale ad n, essendo n la lunghezza di S.

Cioé

$$t_c'(n,s_i) = \alpha \, n \quad .$$

Allora, supponiamo che sia

$$n = s_1 + s_2 + \ldots\ldots + s_j + \ldots + s_i \; .$$

Il tempo di calcolo necessario per la determinazione completa del processore sarà allora tale che

$$T(n) \leq i \; \alpha \; n$$

e poiché $i \leq n$, segue

$$T(n) \leq \alpha \; n^2$$

E' quindi dimostrato che l'algoritmo (1) opera con un numero di passi di calcolo proporzionale al quadrato della lunghezza della sequenza S di cui si vuole costruire il processore con le note proprietà che abbiamo richiesto.

Si può infine dimostrare che $\ell(P) \leq \text{Cost.} \; . \; \frac{n}{\log n}$.

Infatti, prendiamo per semplicità $\Sigma = \{0,1\}$;

le stringhe di lunghezza $2^k k$ sono le stringhe di lunghezza massima che non contengono ripetizioni di stringhe di lunghezza k.

Se $\ell(S) = 2^k k$ si avrà però che saranno ammesse in S ripetizioni di stringhe di lunghezza $\frac{k}{2}$.

Sostituendo la stringa di lunghezza $\frac{k}{2}$ con l'operatore di ripetizione nel processore ed erodendo la stringa considerata da S si ha

$(S) = 2^k k - \frac{k}{2}$.

Iterando questo procedimento si ottiene che la lunghezza di S che è stata erosa è:

$$2^k k - 2^{k/2}\ k/2$$

erodendo ogni volta $\frac{k}{2}$ simboli.

Il numero di erosioni che si ottiene con questo procedimento è il rapporto

$$\frac{2^k k - 2^{k/2}\ k/2}{k/2} = 2\ 2^k - 2^{k/2} .$$

Proseguendo ad applicare il procedimento alla stringa di lunghezza $2^{k/2}\ k/2$, alla stringa di lunghezza $2^{k/4}\ k/4$, e così via fino alla stringa di lunghezza $2^{k/2^s}\ k/2^s$ con $k = 2^s$, si avranno i seguenti rapporti:

$$\frac{2^{k/2}\ k/2 - 2^{k/4}\ k/4}{k/4} = 2\ 2^{k/2} - 2^{k/4}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots = \ldots\ldots\ldots\ldots\ldots$$

$$\frac{2^{k/2^{j-1}}\ k/2^{j-1} - 2^{k/2^j}\ k/2^j}{k/2^j} = 2\ 2^{k/2^{j-1}} - 2^{k/2^j}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots = \ldots\ldots\ldots\ldots\ldots$$

fino a $2^j = k$ cioè $j = \log k$.

Sommando tutti i rapporti si ottiene quindi il numero totale di erosioni e maggiorando questa sommatoria con una serie geometrica si ottiene:

$$2^k + 2^k + 2^{k/2} + 2^{k/4} + \ldots\ldots + 2^{k/2^s} \leq 2^k + \sum_1^k 2^j \leq 2^k + 2^{k+1} - 1 \leq 3.2^k$$

Quindi abbiamo trovato un estremo superiore per la lunghezza del processore P che dipende dalla lunghezza della seguenza:

$$\ell(P) \leq 3.2^k$$
$$\ell(S) = k.2^k = n$$

Passando ai logaritmi si ha:

$$\log n = k + \log k$$

e quindi

$$\frac{1}{2} \log n \leq k$$

$$\frac{\ell(P)}{\ell(S)} \leq \frac{3}{k} \leq \frac{3.2}{\log n}$$

Abbiamo quindi trovato il risultato desiderato

$$\ell(P) \leq \text{Cost.} \cdot \frac{n}{\log n}$$

E' stato realizzato un programma in APL che costruisce un processore di lunghezza minima su una stringa di lunghezza n e che calcola la complessità descrittiva $\Phi(S)$ sulla stringa data.

Ecco alcuni risultati sperimentali ottenuti con questo programma su sequenze di 72 simboli dell'alfabeto $\Sigma = \{0,1,\ldots,11\}$ di alcuni brani per piano.

Sono stati scelti brani formati da note di uguale durata.

Chopin - Studio 25, n. 10 (prime 6 battute)

S = 6 5 6 7 8 9 8 7 8 9 10 11 10 9 10 11 0 1 2 3 2 3
4 3 4 5 4 6 5 6 7 6 7 8 7 8 9 7 8 9 8 9 10 11 10
11 10 9 8 9 8 9 8 7 6 7 6 7 6 5 4 3 2 1 11.

$\Phi(S).72 = 14$

Chopin - Studio 25, n. 8 (prime 6 battute)

S = 3 5 3 5 3 5 1 3 4 6 5 1 5 3 5 3 5 3 1 3 4 6 5 1 3
5 6 7 8 10 11 0 0 3 1 8 0 10 0 10 0 10 8 10 6 8 5
6 3 5 3 5 3 5 1 3 4 6 5 1 5 3 5 3 5 3 1 3 4 6 5 1 .

$\Phi(S).72 = 17$

Czerny - Studio 849, n. 1 (prime 6 battute)

S = 7 0 2 4 2 0 7 0 2 4 2 0 7 0 2 4 2 0 7 0 2 4 2 0
7 2 4 5 4 2 7 2 4 5 4 2 7 2 4 5 4 2 7 2 4 5 4 2
7 4 5 7 5 4 7 4 5 7 5 4 5 2 4 5 4 2 5 2 4 5 4 2 .

$\Phi(S).72 = 18$

Czerny - Studio 849, n. 6 (prime 6 battute)

S = 7 2 0 7 0 11 7 2 0 7 0 11 7 2 0 7 0 11 7 2 0 11 0 4
7 4 2 7 2 1 7 4 2 7 2 1 7 4 2 7 2 1 7 4 2 1 2 5 7 5
4 7 2 0 7 0 11 7 2 5 7 5 4 7 2 0 7 0 11 7 2 5 .

$\Phi(S).72 = 18$

Clementi - Gradus ad Parnassum, Studio n
(prime 4 battute e mezzo)

S = 10 9 0 10 2 0 3 2 5 4 7 5 3 2 0 10 10 9 0 11 2 0 3 2
5 3 2 0 10 9 7 5 5 4 7 5 3 2 0 10 11 0 2 0 9 10 0 10
5 4 7 5 9 7 10 9 0 11 2 0 10 9 7 5 5 4 7 6 9 7 10 9 .

$\Phi(S).72 = 16$

Clementi - Gradus ad Parnassum, Studio n. 7

(4 battute e mezzo centrali)

S = 7 6 7 6 7 5 4 2 1 2 1 2 1 2 4 5 7 6 7 6 7 5 4 2 1 2
4 2 1 9 11 1 5 4 5 4 5 2 9 5 7 6 7 6 7 4 10 7 7 6 7
6 7 4 1 7 9 8 9 8 9 5 2 9 3 2 3 2 3 0 5 3 .

$\Phi(S).72 = 22$

Stringa pseudocasuale di 72 elementi.

$\Phi(S).72 = 31$

I risultati ottenuti sono da considerare esemplificativi; per studio valido si dovrà fare una classificazione di più autori, di stringhe più lunghe e più corte.

Il tempo di calcolo necessario per trovare il processore e per valutare $\Phi(S)$ per una stringa di 72 caratteri è stato circa di dieci secondi per stringa.

Il tempo di calcolo necessario per la determinazione del processore può essere limitato superiormente da un termine inferiore, per n grande, ad $\alpha \cdot n^2$.

Infatti, in base ai risultati precedentemente ottenuti si ha:

$$T(n) \leq i\,\alpha\, n$$

$$\ell(P) = j \leq \text{Cost.} \cdot \frac{n}{\log n}$$

e quindi si ottiene:

$$T(n) \leq \text{Cost.} \cdot \frac{n^2}{\log n}$$

PROSPETTIVE

La descrizione formale di processi musicali è lo strumento necessario per affrontare e risolvere diversi tra i più significativi problemi concernenti il mondo musicale odierno.

Le ambiguità dei vecchi sistemi notazionali vengono eliminate necessariamente nell'atto del compositore o dell'interprete informatico di descrivere i fenomeni musicali i tutti i parametri che li determinano.

La complessità dei fenomeni e la possibilità di descriverli in forma parametrica inducono all'utilizzo di strumenti di descrizione formali che garantiscano la completa determinazione e la economia descrittiva degli eventi sonori che si desiderano.

Il compositore ha la possibilità di essere interprete delle sue stesse composizioni senza dover ricorrere ad intermediari.

L'interprete, cioè chi si accinge alla descrizione di testi musicali preesistenti, ha la possibilità di rendere oggetto inalterabile la sua interpretazione o le sue diverse interpretazioni dello stesso testo; non è più il supporto sonoro il testimone di un'interpretazione.

L'approccio formale alla descrizione dei fenomeni musicali permette lo sviluppo di uno strutturalismo della concezione musicale.

Questo strutturalismo dovrebbe portare a due risultati:

a) alla definizione di alcuni concetti di base sulle forme musicali ottenuta per mezzo di uno studio approfondito delle forme stesse;

b) alla possibilità di classificazioni più significative e strutturate di materiale sonoro ai fini di indagini musicologiche e di creazioni di archivi musicali economici ed ordinati.

Un sistema informatico rivolto allo studio di processi musicali potrebbe essere un valido strumento per la didattica delle conoscenze e delle esperienze musicali ed offrirebbe senz'altro enormi vantaggi di tempo per gli interessati ad un'attività di questo tipo (basta confrontare, ad esempio, il tempo necessario per imparare a suonare uno strumento qualsiasi con il tempo necessario ad imparare a programmare con un linguaggio qualunque).

Per arrivare ad avere degli strumenti informatici validi nel campo musicale è necessario che si rivolgano gli sforzi individuali verso una partecipazione ben maggiore delle diverse conoscenze che vengono ottenute.

Si deve rendere accessibile a tutti un centro che offra la possibilità di conoscere:

a) bibliografia;
b) discografia;
c) conoscenze attorno ai diversi centri che operano nel settore specifico;
d) conoscenze attorno ai sistemi di analisi e di sintesi ed ai programmi elaborati in questi centri;
e) introduzione nelle scuole di musica di nozioni attorno alle discipline di informatica musicale;
f) sovvenzioni alla ricerca teoretica ed applicata.

Lo scambio di queste informazioni è indispensabile per uno sviluppo più rapido e per evitare lo spreco di sforzi ripetuti in direzioni già percorse.

Riferimenti bibliografici

Paterson, M.S. - "*String-Matching and Other Products*" - Atti del Convegno Annuale AICA - Pisa - 1973

Rufer,J. - *Composition with twelve Notes* - Barrie & Rockliff Ed. - 1954

Schoenberg, A. - "*Funzioni strutturali dell'armonia* - Il Saggiatore - 1967.

** Desidero ringraziare il Prof. Alberto Bertoni, il Prof. Giovanni Degli Antoni, il Prof. Giancarlo Mauri ed il Dott. Mauro Torelli senza l'aiuto ed i suggerimenti dei quali non sarebbe stato possibile realizzare questo lavoro.