

# **La segmentazione**

**Goffredo Haus**  
**Massimiliano Pancini**

# SOMMARIO

<b>INTRODUZIONE</b> .....	<b>3</b>
DIVERSI APPROCCI .....	3
PROBLEMATICHE .....	3
GLI OPERATORI MUSICALI.....	4
<b>OPERATORI RITMICI</b> .....	<b>6</b>
<i>L'operatore ritmico/strutturale</i> .....	9
<b>OPERATORI MELODICI</b> .....	<b>11</b>
TRASPOSIZIONE .....	12
INVERSIONE SPECULARE.....	13
RETROGRADAZIONE.....	14
IL RICONOSCIMENTO DELLA PRESENZA DI OPERATORI MELODICI NELLE PARTITURE .....	15
COMPOSIZIONE DI OPERATORI MELODICI.....	18
<i>Composizione della trasposizione e dell'inversione</i> .....	18
<i>Composizione della trasposizione e della retrogradazione</i> .....	18
<i>Composizione della traslazione, della retrogradazione e dell'inversione</i> .....	19
COMPLESSITÀ .....	19
<b>OPERATORI ARMONICI</b> .....	<b>21</b>
LA VERTICALIZZAZIONE .....	21
L'OPERATORE DI RICONOSCIMENTO FUNZIONE TONALE (RFT) .....	23
L'OPERATORE CADENZA ARMONICA (CA).....	25
L'OPERATORE RIDONDANZA ARMONICA (RA).....	26
LE COMBINAZIONI DI OPERATORI UTILIZZATE .....	28
<b>I PARAMETRI DEL SISTEMA</b> .....	<b>30</b>
LA SUPERFICIE RITMICO-MELODICA.....	30
<i>La valutazione</i> .....	31
LA SUPERFICIE ARMONICA .....	31
VALORI STANDARD .....	32
<b>UN ESEMPIO: IL TEMA DELLA SINFONIA K40 DI W.A.MOZART</b> .....	<b>33</b>
<b>GLI ALGORITMI</b> .....	<b>35</b>
ALGORITMI DI DELIMITAZIONE DEI TEMI .....	42
<i>Commenti</i> .....	48
LEGENDA.....	50
<b>RIFERIMENTI BIBLIOGRAFICI</b> .....	<b>51</b>

# Introduzione

In questa trattazione presenteremo i risultati ottenuti presso il LIM nell'area di ricerca dell'informatica musicale che si occupa dell'analisi automatica delle opere musicali, (di qualsiasi tipo e forma), chiamata segmentazione.

Segmentare un brano significa evidenziarne i contenuti principali. Affrontando l'analisi musicale nel suo aspetto linguistico-espressivo, il compito è quello, dapprima, di aggregare gruppi di elementi (note o figurazioni) ciascuno dei quali, isolatamente, è scarsamente significativo.

Complementarmente ci si deve mettere in grado di discernere, nell'economia della composizione, i più "importanti" tra tutti gli aggregati di informazione ricostruendo la struttura compositiva ad alto livello. Inoltre parleremo dello studio delle relazioni che intercorrono tra le unità tematiche di un brano, al fine di ricondurre la struttura architettonica di un brano ad un insieme organico di trasformazioni (resintesi).

Queste valutazioni possono essere eseguite su di una partitura come su di una registrazione audio, parlando nei due casi rispettivamente di segmentazione in campo simbolico e fonico. Al momento attuale, gli studi sulla conversione tra le due rappresentazioni procedono, accorciando il divario tra gli approcci. Una volta ottenuta una scomposizione del brano in oggetti musicali, la traccia che si ottiene per backtracking partendo dall'insieme degli oggetti musicali fino al tessuto originale è un modello dell'idea compositiva.

## Diversi approcci

La ricerca sulla segmentazione prevede diverse scelte di criteri per esaminare la struttura di una composizione. Qualora si fosse in possesso della registrazione audio di un'esecuzione, i parametri osservabili direttamente e significativi sono quelli che la mente umana elabora inconsapevolmente, quali ad esempio la dinamica, la variazione nel tempo del brano, la durata dei silenzi e lo scostamento del valore preciso riconducibile alla possibile durata della nota. Nell'ambito delle registrazioni audio si tratta di porre l'attenzione sulle caratteristiche sonore che l'esecutore o il compositore stesso usano ai fini espressivi, e dopo aver creato gli strumenti per misurare quest'informazione, elaborarla in un modello creato ad hoc in grado di valutare la plausibilità di questo risultato. I modelli impiegati potrebbero per esempio essere modelli statistici e adattivi, reti neurali in grado di apprendere il riconoscimento di un attributo per ogni regola scelta. La nostra ricerca è focalizzata sull'estrazione di materiale "generativo" da partiture simboliche, risalendo dalla notazione al fraseggio e infine alla semantica.

Per realizzare quest'obiettivo, sono stati introdotti gli operatori musicali, ovvero un insieme di strumenti scientifici in grado di valutare e rilevare matematicamente la presenza di vari tipi di oggetti musicali.

Definiamo Oggetto Musicale qualsiasi tipo di informazione strutturata del linguaggio musicale.

Si sceglie di trattare con "oggetti musicali" per formalizzare un concetto linguistico generale: parlando di materiale tematico si introdurrebbero restrizioni sensibili a criteri estetici, storici e formali. Noi stiamo cercando strumenti applicabili nel modo più astratto e flessibile. Inoltre nemmeno i testi di estetica musicale chiariscono esaurientemente il concetto di Tema Musicale, ed il riconoscimento umano stesso è influenzato da fattori culturali. Più precisamente, il punto di contatto tra l'estetica e la nostra formalizzazione è la variazione di entropia introdotta dall'oggetto musicale. Se la combinazione di simboli dell'alfabeto musicale genera "sorpresa", attenzione nell'ascoltatore o soddisfazione di un'aspettativa, è perché questi percepisce la variazione di entropia nella partitura (e di riflesso nell'esecuzione).

L'oggetto musicale è così definito e riconosciuto in presenza di una variazione del flusso di informazione veicolato dal brano.

## Problematiche

Il problema generale di ogni formalizzazione di linguaggi umani (e quindi vale per il caso particolare in cui si parli di arte) è la non perfetta categoricità delle norme che definiscono il valore semantico delle espressioni.

Nel caso della letteratura, della musica e perfino delle arti figurative, esse variano nel tempo, nelle aree geografiche e nel contesto culturale.

Ogni teoria in grado di catturare il maggior numero di proposizioni del linguaggio in esame, difficilmente sarà categorica, quasi sicuramente avrà casi singolari di non aderenza (si pensi alle “forme idiomatiche” per il linguaggio letterario).

Inoltre il linguaggio musicale ha una sintassi pluridimensionale in cui vi è coesistenza di ritmo, melodia e armonia. Il più semplice alfabeto musicale è già la combinazione di almeno due informazioni simultaneamente per ogni simbolo, o se preferiamo, informazione di base. La definizione di nota associa all'altezza di un suono (la frequenza) una durata nel tempo.

Inoltre, pur tralasciando gli accenti e il battere ed il levare, per avere una caratterizzazione dell'informazione elementare dovremmo aggiungere a queste due dimensioni almeno il volume con cui quella nota è suonata, ed il timbro dello strumento che la deve suonare.

La semplificazione del modello nelle sole prime due dimensioni è comunque già sufficiente, in quanto processi mentali di interpretazione a cui accenneremo in seguito non ci obbligano a distinguere strumenti diversi, e l'intensità sonora nei generi di musica studiati non è mai relativa ad una particella del linguaggio come la nota, ma viene attribuita proprio al riconoscimento di strutture espressive più complesse.

Ogni linea melodica (parte reale) composta da note, è scomponibile in altre 2 dimensioni immediatamente percepite dall'ascoltatore. Queste dimensioni sono l'andamento ritmico - melodico (dato appunto dalla combinazione delle note, seguendo le regole della sintassi musicale) e l'andamento armonico, che forma appunto la dimensione quasi indipendente delle funzioni tonali. Possiamo aggiungere che il linguaggio musicale, per questa sua complessità dimensionale, è molto meno formalizzabile di altri linguaggi naturali.

Esistono alcune regole sintattiche, ma dai tempi dei greci gli analisti cercano una spiegazione pragmatica della dimensione comunicativa, senza riuscirci. Solo alcuni movimenti elementari armonici (TONICA- SOTTODOMINANTE - DOMINANTE-TONICA) o melodici (sensibile - tonica) o alcuni elementi di punteggiatura ritmica (pause, note lunghe) restano definiti in tutti i periodi storici.

Ancora, una partitura che esprime le stesse idee tematiche di un'altra può sembrare diversissima a causa della scrittura diversa: lo strumento con cui si sviluppa un brano (timbro) influisce sulla scelta stilistica e quindi si ripercuote sulla notazione.

Un altro fattore critico è la robustezza del modello del linguaggio: un segmentatore di partiture ha l'esigenza di massimizzare in fase di progetto la compatibilità con le forme possibili del passato.

A differenza di un linguaggio formale, che nelle specifiche deve prevedere l'espandibilità per forme future, il modello dovrà essere configurabile. Nei secoli, infatti (più di cinque, dell'uso della tonalità ad oggi) è cambiata la lunghezza dei temi, le forme musicali e le regole armoniche, la lunghezza delle cadenze ed il concetto stesso di tema, spostandosi dalla melodia verso l'influsso dell'armonizzazione.

L'approccio per strutture formali analitiche permette la generazione di un dominio di dati utilizzabili sia come risultato della segmentazione, sia come stadio intermedio di scambio tra l'approccio verticale ed orizzontale dell'analisi musicale. Operatori melodici ed armonici, benché molto differenti tra loro, producono risultati complementari e compatibili.

Si è così scelto di scomporre in tipi diversi di analisi, con un approccio “quasi diagonale”: computazioni diverse in momenti diversi, ma che condividano le informazioni prodotte.

## **Gli operatori musicali**

Inizialmente si sono considerate spie significative ed invarianti la localizzazione degli accenti e le ripetizioni di pattern ritmici e melodici. Questo requisito è la più elementare caratteristica nell'esposizione tematica.

La sintassi musicale ed il lato percettivo della melodia riflettono nella ricorrenza di strutture del linguaggio la necessità umana di richiedere, per l'apprendimento, un certo numero di ripetizioni. Inoltre, sempre per lo stesso motivo, è richiesta una punteggiatura ritmica per organizzare mentalmente il materiale acquisito.

Stabilita l'importanza della ridondanza, si è potenziato lo strumento per la segmentazione rendendolo in grado di riconoscere non solo la ripetizione dei pattern, ma delle rielaborazioni che essi possono subire.

La nostra mente ha meccanismi che elaborano gli stimoli auditivi ben diversi dal riconoscimento testuale di un qualsiasi parser.

Si pensi ad esempio alla tendenza innata di creare mentalmente macrostrutture riconosciute come incisi o frasi: una mente musicalmente allenata può trattare insieme di note e suoni diversi come un'unica maneggevole informazione.

Un altro esempio sono i filtri mentali che trovano immediatamente similitudini esclusivamente in base al movimento dei suoni, bypassando timbri di strumenti diversi e altezze assolute diverse; questi filtri inoltre ignorano anche il mismatch di alcuni intervalli, se questi non sono preponderanti nell'economia dello svolgimento della melodia,

Un esempio di come si sia modellato questo sistema reale, e' la descrizione della melodia come successione differenziale di intervalli anziché di altezze assolute, dalla semplice osservazione che la mente riconosce e ricorda le melodie in modo indipendente dall'altezza a cui vengono ascoltate.

Si e' introdotto così il riconoscimento della trasposizione, cioè di una ripetizione non fedele ma traslata nota per nota di n gradi sulla scala musicale, e dell'inversione, strumento compositivo meno consueto ma ugualmente utilizzato dai compositori. Fino alla retrogradazione, quasi impercettibile per l'ascoltatore ma utilizzata al pari delle altre manipolazioni nella fuga, ed alla combinazione di ognuno di questi artifici.

Questi modelli di "rielaborazione" permettono di riconoscere ognuno un certo tipo di ridondanza in modo molto affidabile. Di contro, sono fragili a causa del necessario rigore nel confronto nota per nota di due melodie.

Una relazione orientata alle note di una melodia implica che ogni nota soddisfi l'equivalenza per l'altezza e per la durata.

Nel caso di variazioni o abbellimenti, una piccola differenza vanifica il confronto.

Non essendo abbastanza efficace l'introduzione di un concetto di "tolleranza", inutile già nel semplice esempio della riduzione, che trasforma la singola nota in un gruppetto, si e' sentita sempre più l'esigenza di aggirare l'approccio contrappuntistico, e procedere con un'analisi Shenkeriana, tale da individuare precise zone di importanza tematica e da discernere le note importanti da quelle ornamentali o secondarie.

Per individuare così le zone generate dalle macrostrutture più importanti si e' creato un interprete dell'armonia, armonia intesa come tessuto armonico.

Se intendiamo con "sonorità" l'insieme delle note suonate in un qualsiasi istante X del brano, una volta individuata la tonalità e' possibile ricondurre ogni sonorità dapprima ad un'armonia, e successivamente ad una funzione tonale, cioè ad un significato espressivo. Questo indipendentemente dalla complessità della disposizione dei suoni simultanei e dal numero di strumenti che li producono. Eseguendo la partizione dell'insieme di tutti i possibili suoni contemporanei, si individua una "spine" che segnala all'analista aree di stasi espressiva.

Rifacendoci ancora una volta al modello percettivo, noi osserviamo il movimento nel tempo da ognuna di queste funzioni ad ogni altra. Questo perché questa trasformazione implica un'evoluzione del discorso musicale, e di conseguenza in alcuni casi queste "mutazioni" saranno riconducibili ad una "punteggiatura musicale" (cadenze).

L'ultima frontiera toccata ed ancora espandibile e' la ricerca dell'interazione tra questi differenti filtri (ritmico, melodico, armonico) e la capacità di adattare il riconoscimento armonico ad ogni genere di musica, modellando insieme di regole anche per le musiche etniche, il jazz e tutte le forme non direttamente riconducibili all'estetica della musica tonale europea.

# Operatori ritmici

Il primo dei tre piani studiati e' quello ritmico. Con questa analisi si ricava una descrizione di come la melodia si articola nel tempo. Ricordando che la melodia e' composta di intervalli melodici e ritmo, e' immediato dedurre l'importanza della congruenza degli accenti nel riconoscimento melodico.

Ricordiamo che un accento rappresenta il riferimento "concettuale" di un aspetto percettivo. E' cio' che giustifica la "carica ritmica" che una nota si trova ad assumere in base alla sua posizione all'interno della battuta.

In virtu' di quanto detto, segue che partendo dalle sole informazioni di partitura, data una melodia si puo' rapidamente

```
{first case}
  IF note.duration = prev.duration THEN
    {the previous rhythmic motion is kept }
    note.accent := FundAcc;
    {the accent is the fundamental metrical one }

{second case}
  ELSE IF note.duration < prev.duration THEN
    BEGIN
      utLiv := LastLevel(scan, Unit^TempoCor);
      {compute what is the last level filled with previous notes }
      IF note.duration < utLiv THEN
        {start a level of lower subdivision }
        note.accent := strong
        {because the note duration is strictly < UintTempoCor,
         doesn't have to consider the metric on movements, so it is always strong }
      ELSE IF note.duration = utLiv THEN
        {on could conclude with current note the level over the last evaluated one,
         except the case when it be the 2nd part of a ternary level }

        IF (utLiv = Unit^TempoCor) OR
           (it's a ternary tempo AND (utLiv = subdivision)) THEN
          {check that there insn't a strong beat due to metric, i.e. quaternary}
          nota.accent := FundAcc
        ELSE
          nota.accent := weak
          {ending a binary progression}}

      ELSE {note.duration > utLiv}
        note.accent := FundAcc;
    END

{Third case}
  ELSE {note.duration > prev.duration}
    BEGIN
      utLiv := LastLevel(scan - prev.duration, Uint^TempoCor);
      {it is the level reached before the previous note, it is required to know from which
       level we are getting back }
      IF utLiv = Unit^TempoCor THEN
        note.accent := FundAcc
        {it could be a weak accent if provided for by fundamental metric}
      ELSE
        Note.accent := Strong;
```

Figura 0

arrivare ad assegnare nota per nota gli accenti ritmici tramite un automa. (fig. 0)

Nell'analisi ritmica si assegna ad ogni figura l'accento ritmico che essa assume nell'esecuzione. Il concetto di "battuta" e' riferito a dove il tempo, appunto, si appoggia. Per evolvere nel tempo, inteso sia come concetto musicale, sia come dimensione cronologica, modelliamo l'azione nel tempo come la successione di eventi, siano essi note o pause, che occupano una certa quantita' di tempo. Ogni quantita' e' relativa alla notazione ed alle indicazioni metronomiche anziche' essere predeterminata in secondi.

Avendo ogni nota associata una durata, implicitamente essa possiede un punto di partenza, ancora una volta sincronizzato (nel modello) all'istante dopo la fine dell'evento precedente, e non ad un istante "assoluto". In questo modo e' esplicita la relazione tra le varie note e su come ognuna di esse tragga la sua carica dinamica dalla sua disposizione nella battuta in funzione di tutti gli eventi. La battuta e' divisa in movimenti, che a loro volta sono suddivisi in unita' ritmiche minori.

Come in ogni struttura musicale, all'interno della battuta vi e' una gerarchia: il movimento con maggior appoggio e' sempre il primo della battuta, detto appunto "in battere". Esso determina l'istante in cui si appoggia l'unita' ritmica, e tutto cio' che segue quell'istante acquisisce sempre piu' carica dinamica, perdendo importanza statica e appoggio ritmico.

Il movimento finale della battuta e' chiamato "levare" proprio per indicare lo slancio che prepara ad un nuovo appoggio, come succede quando un musicista batte il tempo con la mano o il direttore d'orchestra muove la bacchetta.

All'interno della battuta si alternano movimenti "forti" e "deboli", che rispettano comunque sempre il rapporto di subordine con il primo movimento.

Nei trattati si schematizza questa caratteristica come un F (forte) sul primo tempo, e nessuna indicazione dinamica relativa ai movimenti successivi deve sovrastare questo movimento, per il resto della battuta.(fig. 2)

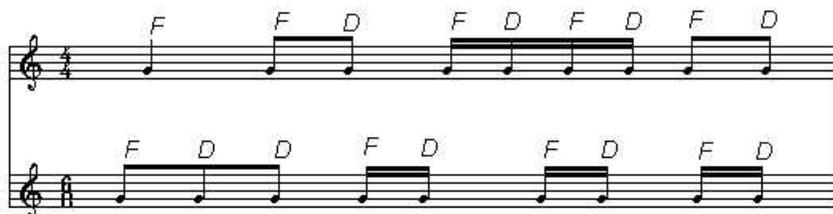


Figura 1

Analogamente, tutti i movimenti che precedono l'ultimo devono avere meno slancio ritmico di esso. Questo implica anche che rispetto al levare abbiano maggiore appoggio, quindi maggior volume. Ogni movimento a sua volta ha una suddivisione, che può essere binaria o ternaria, in cui le note assumono accento più forte se iniziano sul movimento, debole se sono sulle altre suddivisioni. Estendendo ricorsivamente il ragionamento, possiamo pensare ad ogni ulteriore divisione come contenente al suo interno due o più tempi (di solito sotto la suddivisione del movimento le divisioni sono binarie), di cui il primo possiede accento più forte dei successivi.

Un automa, ricorsivo per conseguenza di quanto detto, sarà in grado di percorrere i vari livelli di suddivisione, muovendosi tra di essi quando la figurazione si sposta verso note di valore minore o maggiore. Esso avrà lo scopo di assegnare ad ogni nota l'importanza ritmica, intesa nel significato di accento. Questo automa classificherà gruppi irregolari e sincopi come eccezioni, e in questi casi particolari attribuirà loro l'accento relativo alla loro particolare figurazione, e proporzionato al livello di suddivisione a cui appartiene (Fig. 3).



Figura 2

Utilizzando l'algoritmo presentato, siamo in grado di ricavare, da due sequenze date di note  $S1$  e  $S2$  di lunghezza  $n$ , le corrispondenti sequenze di accenti:

$$D1 = \langle X1, X2, X3, X4, X5, \dots, Xn \rangle$$

$$D2 = \langle Y1, Y2, Y3, Y4, Y5, \dots, Yn \rangle$$

e da esse un vettore d'errore  $E$  che riporta un valore nullo dove gli elementi di  $D1$  e  $D2$  coincidono ed un valore unitario dove essi differiscono.

Introduciamo così una metrica binaria, cioè basata sul binomio "match/ non match".

Il vettore di errore  $E$  rappresenta la distanza di due oggetti musicali "sequenza" (definiti intuitivamente come successione di simboli musicali), siano essi esaminati sul piano ritmico come mostrato adesso, sia sul piano melodico come vedremo successivamente. La metrica del confronto ritmico per accenti è semplice, in quanto determina la coincidenza degli accenti "correlati", dove la correlazione è data dalla stessa posizione assunta all'interno delle diverse sequenze.

il vettore  $E$  sarà ottenuto sottraendo ad uno ad uno gli elementi correlati dei due vettori  $D1$  e  $D2$ .

Sulla base di  $E$  si valuterà poi la somiglianza dei due frammenti (per quanto riguarda gli accenti) indicandola ad esempio come  $1 - |E|/n$ .

Ad esempio, indicando con  $F$  l'accento forte e con  $D$  quello debole, se fossero date le sequenze di accenti

$$D1 = \langle F, F, D, F, D \rangle$$

$$D2 = \langle F, D, F, D, D \rangle$$

si otterrebbe il vettore d'errore

$$E = \langle 0, 1, 1, 1, 0 \rangle$$

da cui un indice di somiglianza pari a  $1 - 3/5 = 0,4$ .

Adesso, si estende il confronto tra sequenze di note ad un altro attributo, la durata, per la ricerca di figurazioni "simili". Questo strumento, anch'esso limitato dal punto di vista informativo, andrà ad integrarsi con il riconoscimento degli accenti per completare l'analisi ritmica strutturale.

Lo scopo di questa parte di analisi è infatti la sintesi della capacità di riconoscere un gruppo di durate simili.

Questo deve poter avvenire, inoltre, scandendo le sequenze nel senso di percorrenza proprio della notazione musicale, da sinistra verso destra, o muovendosi al contrario, da destra verso sinistra. Infatti come già detto il riconoscimento ritmico potrà contribuire ad isolare elaborazioni della stessa melodia, riproposte in punti diversi della partitura. Per poter essere compatibile con tutti gli oggetti musicali che vedremo, la ricerca di occorrenze di ripetizioni ritmiche deve essere più flessibile e generale possibile. Si noti come tra due patterns ritmici vi possa essere coincidenza di accenti anche qualora essi siano molto diversi, ed analogamente come lo stesso pattern suonato in battere ed in levare cambi completamente l'idea compositiva. Da qui, la complementarità dei due metodi.

Per il riconoscimento delle occorrenze si procede in modo analogo a quanto fatto nel caso degli accenti, si parte cioè dalle due sequenze di note da confrontare  $S1$  ed  $S2$  e si ricavano le corrispondenti sequenze di durate

$$L1 = \langle X1, X2, X3, X4, X5, \dots, X6 \rangle$$

$$L2 = \langle Y1, Y2, Y3, Y4, Y5, \dots, Y6 \rangle.$$

Confrontando fra loro queste ultime si costruisce un vettore d'errore  $E$  che riporta come nel caso precedente le differenze riscontrate. Da questo si potrà infine ricavare un valore indicativo della somiglianza strutturale fra i due frammenti. Come è facilmente intuibile la differenza fra la versione non retrograda e quella retrograda di questo operatore sta nella costruzione del vettore  $E$ . Nel primo caso esso sarà costruito confrontando le due sequenze  $L1$  ed  $L2$  da sinistra verso destra mentre nel secondo caso la direzione di scansione di queste sequenze sarà opposta, ovvero  $L1$  da sinistra a destra e  $L2$  da destra a sinistra.



Figura 3

Come si può immediatamente vedere, i tre frammenti di figura 4 sono quasi uguali. La differenza fra i primi due sta nelle prime due note che sono leggermente diverse in durata mentre il terzo è dato dalla retrogradazione (cioè un operatore melodico di cui vedremo più avanti una trattazione dettagliata) del primo con la stessa modifica sulle ultime due note. Da essi si ricavano rispettivamente i seguenti tre vettori delle durate

$$L1 = \langle 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/4 \rangle$$

$$L2 = \langle 3/16, 1/16, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/4 \rangle$$

$$L3 = \langle 1/4, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 3/16, 1/16 \rangle$$

Da questi, confrontando il primo con il secondo vettore in forma non retrograda, si ottiene il seguente vettore d'errore

$$E12 = \langle 1, 1, 0, 0, 0, 0, 0, 0, 0 \rangle$$

che indica una somiglianza pari a  $1 - 2/9 = 0,78$ .

Confrontando il primo con il terzo in forma retrograda si ottiene

$$E13 = \langle 1, 1, 0, 0, 0, 0, 0, 0, 0 \rangle$$

che indica di nuovo una somiglianza pari a  $0,78$ .

Dal confronto fra il secondo e il terzo, in forma non retrograda, si ottiene invece

$$E23 = \langle 1, 1, 0, 0, 0, 0, 0, 1, 1 \rangle$$

cui corrisponde una somiglianza pari a  $1 - 4/9 = 0,56$ .

Ora che abbiamo formalizzato le singole "componenti ritmiche", si può definire da esse un unico operatore ritmico, inteso come il successo del riconoscimento per entrambi gli operatori. Essi possono quindi essere visti come un unico operatore.

## L'operatore ritmico/strutturale

Poiché, come detto, due frammenti potranno essere considerati equivalenti del punto di vista ritmico solo se entrambe gli operatori presentati nei due paragrafi precedenti li riconosceranno come tali, unificheremo il confronto in modo tale che l'equivalenza ritmica possa essere descritta tramite un solo operatore.

Escludendo i casi in cui gli operatori siano retrogradati (V. “operatori melodici”) entrambi gli operatori presentati coincidono con la trasformazione identica, quindi l’operatore che li andrà a sostituire sarà anch’esso la trasformazione identica.

La complementarità sarà formalizzata dalla scelta di combinare le informazioni ritmiche (accento e durata) in coppie. Di conseguenza l’operatore non agir  su una sequenza o di accenti o di durate ma su di una sequenza di coppie. Il confronto avverr  nel seguente modo.

Date le sequenze di note da confrontare  $S1$  ed  $S2$ , si ricavano le corrispondenti sequenze di coppie (accento, durata)

$$DL1 = \langle (D11,L11), (D12,L12), \dots, (D1n,L1n) \rangle$$

$$DL2 = \langle (D21,L21), (D22,L22), \dots, (D2n,L2n) \rangle$$

e da esse il vettore d’errore  $E$  che sar  per  dato della somma degli errori riscontrati per ogni coppia, ossia un punto se l’accento non coincide e un altro punto se la durata non coincide.

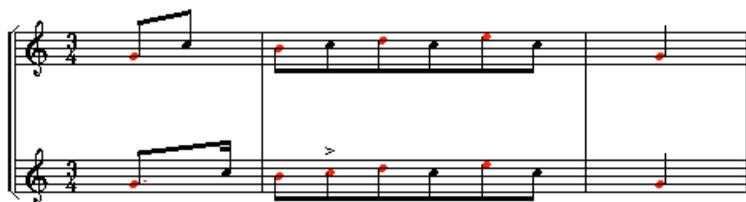


Figura 4

Considerando l’esempio in figura 5 (in cui la quarta nota del secondo rigo   stata forzata ad avere accento forte) si otterrebbe

$$DL1 = \langle (F,1/8), (D,1/8), (F,1/8), (D,1/8), (F,1/8), (D,1/8), (F,1/8), (D,1/8), (F,1/4) \rangle$$

$$DL2 = \langle (F,3/16), (D,1/16), (F,1/8), (F,1/8), (F,1/8), (D,1/8), (F,1/8), (D,1/8), (F,1/4) \rangle$$

e da esse il seguente vettore d’errore

$$E = \langle 0+1, 0+1, 0, 1+0, 0, 0, 0, 0, 0 \rangle$$

cui corrisponde un *indice di somiglianza*  $1 - 3/9 = 0,67$ .

Nel caso retrogrado (che da quanto anticipato finora risulta comunque comprensibile) non sar  necessario valutare gli accenti, in quanto questa esigenza non costituir  pi  un discriminante percettivo. L’operatore coincider  allora con l’operatore delle occorrenze ritmiche descritto precedentemente e il confronto ignorer  gli accenti.

Questo strumento, sebbene preciso e veloce, non fornisce da solo alcuna garanzia, in quanto spesso un pattern ritmico e’ l’artificio compositivo usato comunque in una composizione anche sottendendo materiale tematico diverso. Il pattern ritmico e’ infatti un anch’esso oggetto musicale, sebbene poco caratterizzante.

Lo studio del piano ritmico lavora quindi in concomitanza con l’analisi melodica (fornendo comunque informazioni che possono essere riprese anche in analisi pi  complesse), ponendosi come filtro per determinare le ricorrenze ritmiche, ma anche per evidenziare i punti pi  significativi per il confronto analitico. L’automa per la determinazione degli accenti si occupa di partizionare le note in classi di equivalenza in base alla loro posizione all’interno della battuta, riconoscendo il “battere” (tempo forte, stasi) ed il “levare”(tempo debole, slancio) della battuta, in divisioni sempre pi  fini. Il parsing della dimensione ritmica ha il vantaggio di avvenire in tempi lineari. Essendo poi il modello matematico isomorfo alla teoria musicale relativa, basata esclusivamente su regole matematiche, esso e’ inoltre consistente.

# Operatori melodici

Mentre la sezione ritmica fornisce principalmente una serie di risorse di impiego generale per analizzare la partitura, e' fondamentale esaminare la melodia con una strategia aderente al tipo di analisi in progetto.

Ogni oggetto musicale introduce informazione in una partitura. L'ascoltatore "umano" ha la necessita' di frammentare il flusso di informazione. Il primo approccio e' quindi la ricerca di rielaborazioni del materiale tematico. Il collegamento tra la segmentazione e l'analisi musicologica e' l'attitudine mentale di dover accorpate unite' informative semplici in strutture piu' complesse.

Il modo con cui le sequenze di suoni vengono rielaborate va dalla semplice riproposizione della frase musicale in oggetto alla sua ripetizione variata, alla trasposizione (tonale e non) ed a metodi piu' artificiosi e meno basati su canoni percettivi quali l'inversione e la retrogradazione.

In questi ultimi casi, anche l'analisi musicale sposta il suo campo d'azione verso l'informazione simbolica, astraendosi da meccanismi percettivi e poggiandosi ancora piu' rigorosamente sul modello formale. Nonostante questo, hanno la stessa importanza storica delle trasformazioni piu' facilmente riconoscibili dall'ascoltatore.

Una volta introdotti questi operatori, possiamo studiare sul modello formale le relazioni tra essi. Ad esempio, la semplice ripetizione di una melodia e' considerata come la trasformazione identica indotta dell'operatore "trasposizione" di quella melodia, portando l'attenzione su un impiego piu' generale dello strumento "operatore".

Iniziamo ora a definire questo modello per quanto riguarda la scala musicale.

Esistono due distinti modi per misurare l'altezza (o alfabeti) e quindi per misurare la distanza tra le note: in base alla distanza cromatica, cioe' per semitoni (e quindi in modo indifferente al contesto tonale), e diatonica, cioe' con riferimento alla tonalita' in uso e di conseguenza alle regole compositive.

Essendo la melodia una successione di note necessariamente connotate da altezza, oltre che da altri attributi, il confronto avviene sulla successione delle altezze. Inoltre, la perfetta equivalenza tra l'informazione insita nell'elenco delle altezze assolute e nella sequenza di intervalli melodici (in cui ogni nota e' definita dalla distanza dalla nota precedente), porta a scegliere il secondo approccio (differenziale).

La stragrande maggioranza degli ascoltatori percepisce il linguaggio musicale principalmente sul movimento melodico e non sulle altezze assolute. Il confronto con cui l'ascoltatore riconosce un pattern melodico e' quindi basato sulla differenza tra le altezze, oltreché naturalmente sul ritmo.

La teoria musicale ha modellato questa caratteristica dell'ascolto umano. Per un modello formale, necessiteremo di due relativi sistemi enumerativi adatti.

Nel caso assoluto, che indicheremo come distanza reale, intenderemo il numero di semitoni che le separa due note, e avremo la seguente enumerazione:

$$do = 0, re = 2, mi = 4, fa = 5, sol = 7, ecc.$$

in quanto, ad esempio fra il do ed il re, sulla scala temperata esiste un'altra nota che può essere indicata come do diesis o re bemolle. Si osservi che le distanze riportate sopra sono calcolate a partire dal do, per semplicità.

Un intervallo melodico sarà ovviamente la differenza delle due altezze (es.  $do-mi=hr(mi)-hr(do)=4-0=4$  dove  $hr$  e' la funzione che restituisce l'altezza reale).

Nel caso di misurazione dell'altezza relativa alla tonalità ci riferiremo invece al numero di gradi che separa le due note sulla scala della tonalità corrente (distanza tonale), avremo invece:

$$do = 0, re = 1, mi = 2, fa = 3, sol = 4, ecc.$$

si useranno entrambi questi sistemi a seconda della necessita'.

Riprendendo e riassumendo quanto detto, si e' in grado di ricondurre queste elaborazioni a tre categorie principali, riconosciute da altrettanti operatori:

## Trasposizione

## Inversione

### Retrogradazione

La presenza di una di queste elaborazioni o una loro combinazione, indica che la creatività del compositore ha agito sul modo di proporre l'oggetto musicale.

In tal caso è confermata l'importanza dell'oggetto elaborato, per la scelta stessa di non introdurre nuova informazione (intesa come nuovo materiale musicale) ma di utilizzare quella preesistente.

## Trasposizione

Nel caso di **trasposizione** il modello ricalca perfettamente il processo cognitivo che rende l'ascoltatore in grado di apprendere l'informazione tramite riascolto

Si indica con  $T_x$  l'operatore di trasposizione di grado  $x$  su  $\Sigma$  così definito:

$$T_x(h) = h + x$$

e lo si estende all'insieme di parole generate su  $\Sigma$  nel seguente modo:

$$T_x(hw) = T_x(h)T_x(w)$$

Siano ad esempio, 1, 2, 3, 4, 5 simboli dell'alfabeto  $\Sigma$ , applicando alla sequenza "1 2 3 4" una trasposizione di primo grado otterremo:

$$T_1(1234) = T_1(1)T_1(234) = \dots = T_1(1)T_1(2)T_1(3)T_1(4) = 2345$$

Come detto, un caso particolare di questo operatore è  $T_0$  (trasposizione di grado 0) che traspone ogni simbolo dell'alfabeto in se stesso, cioè:  $T_0(h) = h$ .

Il significato musicale ci riconduce ai due sistemi notazionali introdotti sopra, ognuno relativo ad un tipo di trasformazione: la trasposizione reale e quella tonale.

La trasposizione reale, detta anche modulante, di una sequenza di note è lo spostamento di tale sequenza verso l'alto o verso il basso nella scala in modo che, ad ogni intervallo melodico, la distanza reale misurata in semitoni tra le note della sequenza trasposta sia identica a quella della sequenza originale. Osservando la figura 6 si può vedere sul primo rigo una scala di do maggiore; sul secondo rigo è stata riportata la sua trasposizione reale di quattro semitoni (ottenendo una scala di mi maggiore). È facile verificare che la distanza fra una nota e la successiva è la stessa in entrambe i casi (calcolata in semitoni: 2-2-1-2-2-2-1).



Figura 5

Il caso della trasposizione tonale è invece più complesso: lo spostamento di ogni nota della sequenza è effettuato verso l'alto o verso il basso di uno stesso numero di gradi sulla scala della tonalità corrente. Il che la rende molto diversa della trasposizione reale in quanto le distanze relative in semitoni fra le note del frammento trasposto non saranno uguali a quelle

del frammento originale. (nell'esempio in figura 7, gli intervalli del primo rigo, espressi in semitoni, sono 2,2. Per il secondo ancora 2,2, e per il terzo 2,1)

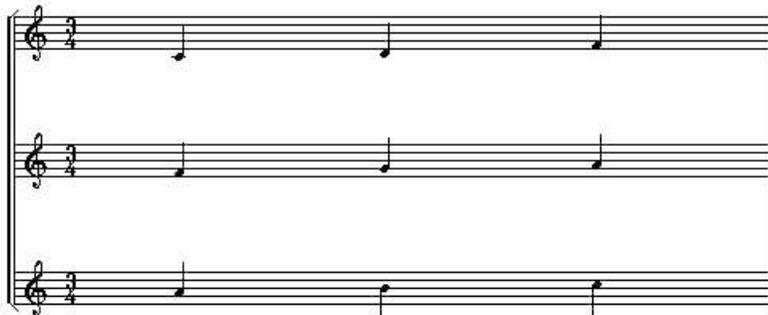


Figura 6

## Inversione Speculare

Si dice operatore di **inversione speculare** su  $\sum$  rispetto ad  $i$  e si indica con  $I_i$  un operatore così definito:

$$I_i(h) = i - (h - i) = 2i - h.$$

Anche in questo caso estenderemo tale operatore alle parole generate su  $\sum$  nel seguente modo:

$$I_i(hw) = I_i(h)I_i(w)$$

L'operatore di inversione speculare inverte quindi il valore del parametro passatogli come argomento in modo simmetrico rispetto ad un asse di simmetria (o nota di riferimento) indicato dall'indice  $i$ .

E' interessante osservare che la variabilità dell'asse di simmetria realizza un aspetto singolare di questo operatore, cioè che una inversione comporta necessariamente un'implicita trasposizione della sequenza. Vale a dire che, applicando l'inversione, resta implicitamente applicata anche una trasposizione

L'operatore di inversione speculare compie la funzione di invertire tutti gli intervalli armonici. Nell'ambito musicale questa trasformazione, che nella letteratura specifica è detta anche imitazione per moto contrario o meno propriamente rovesciata, riveste grande importanza. Essa è stata utilizzata molto spesso negli ultimi secoli da autori come Bach e Beethoven fino a Schoenberg.

In analogia con quanto visto per le trasposizioni, anche con l'operatore di inversione speculare, è possibile riconoscere due tipi diversi di trasformazioni musicali degli oggetti del brano: l'inversione reale e quella tonale.

Anche per questa elaborazione si possono avere il caso di inversione speculare reale o modulante, in cui sono invertiti tutti gli intervalli armonici (da ascendenti a discendenti e viceversa, in modo che gli intervalli della sequenza invertita siano identici a quelli della sequenza originale a meno dell'andamento) e inversione tonale, in cui la distanza fra le note rimane la stessa (misurandola in gradi della scala), ma i tipi di intervalli rispetto alla sequenza originale potranno cambiare. (S.Guagnini e M.Pancini)

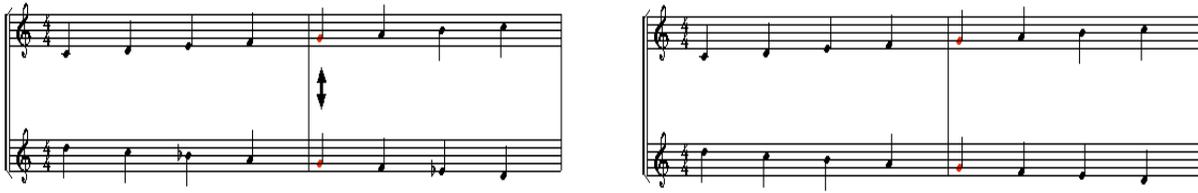


Figura 7 Inversione reale e tonale

L'esempio in figura 8 divide i due casi:

a sinistra riporta sul primo rigo una scala di do maggiore, mentre sul rigo sottostante è stata riportata una inversione speculare reale rispetto al sol.

A destra, si vede come la stessa scala sia stata elaborata con l'inversione tonale sempre rispetto al sol: mentre nella scala di do sul primo rigo la distanza fra la seconda nota (re naturale) e la terza (mi naturale) è di due semitoni (quindi un intervallo di seconda maggiore), dopo l'inversione tale distanza si è ridotta ad un solo semitono passando da un do naturale ad un si naturale (quindi un intervallo di seconda minore).

## Retrogradazione

Si definisce operatore di **retrogradazione** su  $\Sigma$  e lo si indica con  $R$  un operatore così definito:

$$R(h) = h$$

$$R(wh) = hR(w)$$

L'operatore di retrogradazione non agisce sull'altezza delle note ma sull'ordinamento degli elementi che costituiscono una sequenza.

Ad esempio, siano 1, 2, 3, 4 simboli dell'alfabeto  $\Sigma$ , avremo:

$$R(1234) = 4 R(123) = 43 R(12) = 432 R(1) = 4321$$



Figura 8 esempio di retrogradazione

## Il riconoscimento della presenza di operatori melodici nelle partiture

Partendo da un frammento melodico di  $n$  note, si costruisce un Vettore Caratteristico di  $n-1$  posizioni, in cui sono elencate le distanze tra due note vicine. Questo vettore rappresenta la melodia nella sua evoluzione che, come già detto, caratterizza la componente principale del riconoscimento umano.

Resta da definire il concetto di distanza (tra note consecutive) che sarà calcolato in semitoni, nel caso in cui l'operatore in esame sia valutato nella sua accezione reale, o contando i gradi della scala compresi nell'intervallo tra le due note, per il caso tonale.

Infine, a seconda dell'operatore, viene scelto un semplice e veloce algoritmo per verificare che due vettori di uguale lunghezza (relativi a frammenti melodici distinti) possano essere riconosciuti come uno la trasformazione dell'altro tramite l'operatore in questione. L'algoritmo produce stavolta un vettore di errore  $E$  che rappresenta la distanza delle due melodie secondo la metrica introdotta dall'operatore stesso e su cui quindi andare a stabilire la rispondenza dei frammenti ad ogni singolo criterio.

Nel caso della trasposizione, il vettore  $E$  sarà ottenuto sottraendo ad uno ad uno gli elementi corrispondenti dei due vettori precedenti. Se tutte le componenti di  $E$  saranno nulle i due frammenti saranno perfettamente relazionabili tramite l'operatore di trasposizione. Quanto più saranno numerose ed elevate in valore assoluto le componenti non nulle del vettore  $E$  tanto meno sarà possibile stabilire una relazione tra le sequenze tramite questo operatore.

Questo in virtù del principio che la trasposizione (tonale/reale) lascia inalterata la configurazione dei salti (in gradi/semitoni), ed è quindi riconoscibile utilizzando il semplice algoritmo di sottrazione tra elementi corrispondenti dei vettori



Figura 9

**Esempio:** la melodia nell'esempio di figura 10 genera il vettore caratteristico

$$V1 <2,2,1,2,2,2,1>$$

essendo questa la distanza in semitoni tra i gradi della scala di do maggiore riportata sul primo rigo.

Il secondo rigo riporta una scala maggiore anch'esso, anche se di mi, stavolta.

Gli scarti intervallari saranno quindi sempre quelli della scala maggiore, e cioè ancora il vettore caratteristico

$$V2 = <2,2,1,2,2,2,1>$$

Dovrebbe essere chiaro che il confronto è avvenuto per trasposizione reale, in quanto sono state calcolate le distanze per semitoni. Il vettore:

$$E = V1 - V2 = <0,0,0,0,0,0,0> = 0$$

Il confronto ha riconosciuto perfettamente la trasposizione

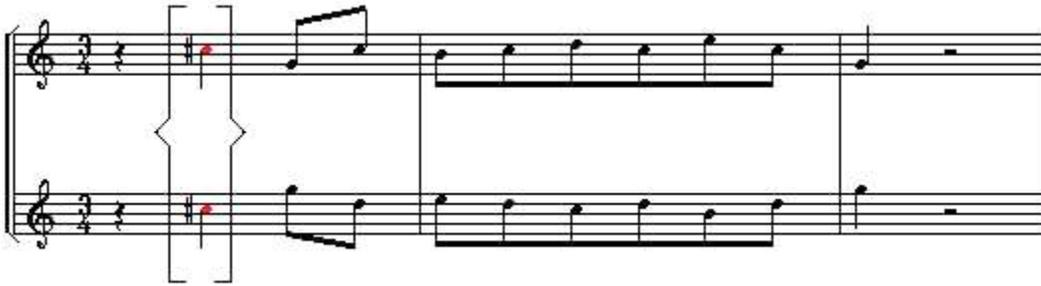


Figura 10

Per il riconoscimento della presenza di un'inversione speculare il procedimento è quasi analogo, con la differenza che il vettore d'errore  $E$  non verrà costruito tramite la sottrazione degli elementi corrispondenti dei vettori  $V_1$  e  $V_2$  ma tramite la loro somma. Questo perché dopo il processo di inversione, gli intervalli sono invertiti nella direzione, quindi restano complementari

Vediamo velocemente in pratica un riconoscimento di inversione, dall'esempio di figura 11: si può vedere un'inversione tonale (quindi con distanze calcolate in gradi tonali) rispetto al  $do\#$ . Si ottengono in questo caso i seguenti due vettori caratteristici

$$V_1 = \langle 3, -1, 1, 1, -1, 2, -2, -3 \rangle$$

$$V_2 = \langle -3, 1, -1, -1, 1, -2, 2, 3 \rangle$$

e dalla loro somma

$$V_1 + V_2 = E = \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle$$

Cioè la conferma che i due frammenti sono l'uno l'inversione tonale dell'altro.

Per il riconoscimento delle sequenze retrograde, nonostante si seguano i criteri visti per gli altri tipi di trasformazione, si opera un ribaltamento di una sequenza tramite inversione dell'ordine delle note, cosicché la prima nota divenga l'ultima, la seconda la penultima e via di seguito. Questo suggerisce che una prima relazione riscontrabile tra i vettori caratteristici delle due sequenze sarà il rovesciamento dei termini, o semplicemente tramite la percorrenza della melodia in direzione opposta (da destra a sinistra) al momento di creare uno dei due vettori caratteristici. Inoltre la distanza fra note consecutive verrà mantenuta a meno del segno, poiché la sequenza risultante sarà disposta al contrario dell'originale. Vale a dire che anche la retrogradazione, come l'inversione, inverte gli intervalli armonici (da crescenti in decrescenti e viceversa).

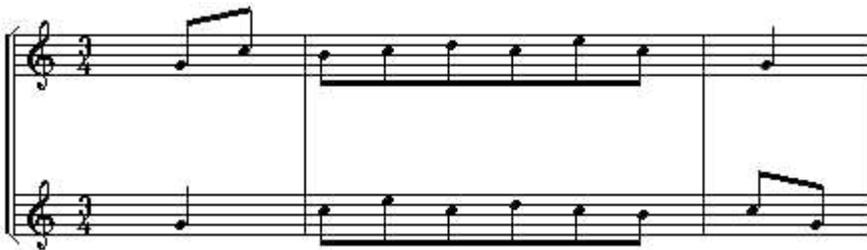


Figura 11

Sul primo rigo di figura 8 è stato riportato lo stesso frammento di figura 7 e sul secondo la sua retrogradazione; è facile verificare che i vettori caratteristici  $V_1$  e  $V_2$ , ad essi associati sono

$$V_1 = \langle 3, -1, 1, 1, -1, 2, -2, -3 \rangle$$

$$V_2 = \langle 3, 2, -2, 1, -1, -1, 1, -3 \rangle$$

e da essi, sottraendo l' $i$ -esimo elemento di  $V_1$  all' $(n-i-1)$ -esimo elemento di  $V_2$ , si ottiene

$$E = \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle$$

che ci conferma la relazione intercorrente fra i due frammenti.

Riassumendo, l'effetto di una retrogradazione sul vettore caratteristico di una sequenza è il suo ribaltamento con cambiamento di segno.

## Composizione di operatori melodici

I tre operatori finora descritti possono essere composti tra loro in più modi generando dei nuovi tipi di trasformazioni sulle sequenze di note.

Ad esempio è possibile trasporre una sequenza e poi invertirla, cioè applicare la trasformazione

$$s' = I[ T(s) ] = I \cdot T(s)$$

Anche se la composizione di operatori rimane riconoscibile dai singoli confronti, e' interessante ora elencare le varie composizioni di operatori per studiarne l'efficacia e le proprietà a livello operativo.

Si noti innanzitutto che la composizione degli operatori formalizzati gode della proprietà simmetrica ovvero che, se  $X$  e  $Y$  sono due operatori qualsiasi, vale  $X \cdot Y = Y \cdot X$ .

## Composizione della trasposizione e dell'inversione

La composizione della trasposizione e dell'inversione si denota come segue:

$$T \cdot I$$

Si dimostra che esso gode delle stesse proprietà della inversione:

$$\text{Sia } x \in \Sigma,$$

l'inversione speculare di  $x$  rispetto all'asse di simmetria  $i$  è data

$$I_i(x) = 2i - x$$

e la sua traslazione di grado  $t$

$$T_i[I_i(x)] = I_i(x) + t = 2i - x + t$$

Ossia

$$T_i[I_i(x)] = 2i - x + t = 2(i + \frac{1}{2}t) - x = I_{(i + \frac{1}{2}t)}(x)$$

Ovviamente, tutte le considerazioni fatte precedentemente sulla duplice caratterizzazione (tonale e reale) dell'operatore  $I$  continuano a valere anche per l'operatore  $T \cdot I$ .

## Composizione della trasposizione e della retrogradazione

La composizione della trasposizione e della retrogradazione si denota come segue:

$$T \cdot R$$

E' interessante osservare che, essendo l'operatore di trasposizione un'identità per il vettore caratteristico, la composizione di questi due operatori è, per quanto concerne il riconoscimento, del tutto equivalente all'applicazione della sola retrogradazione.

L'operatore  $T \cdot R$  sarà utilizzato in sostituzione dell'operatore  $R$ .

Va inoltre osservato che potendo l'operatore  $T$  essere sia tonale che reale, l'operatore appena introdotto  $T \cdot R$  erediterà questa caratteristica.

## Composizione della traslazione, della retrogradazione e dell'inversione

La composizione della trasposizione, della retrogradazione e dell'inversione si denota come segue:

$$T \cdot R \cdot I$$

Questa composizione di operatori introduce nuove considerazioni. Infatti sia la retrogradazione che l'inversione invertono l'andamento armonico degli intervalli di una sequenza. Questo comporta che il riconoscimento delle trasformazioni apportate da questo operatore richiederà un diverso modo di costruzione del vettore d'errore  $E$ .

Sia data la sequenza

$$s = \langle 5, 6, 7, 9 \rangle$$

con

$$V = \langle 1, 1, 2 \rangle$$

l'applicazione di una inversione rispetto al punto 8 genera

$$s' = \langle 11, 10, 9, 7 \rangle, \quad V' = \langle -1, -1, -2 \rangle$$

la retrogradazione di  $S'$  infine produce

$$s'' = \langle 7, 9, 10, 11 \rangle, \quad V'' = \langle 2, 1, 1 \rangle.$$

Nell'esempio si è trascurato l'operatore di trasposizione poiché, per quanto visto, non influisce sul vettore caratteristico.

La trasformazione che questo nuovo operatore induce su tale vettore è data dalla sovrapposizione delle trasformazioni portate da ciascun operatore singolarmente. Il ribaltamento dell'ordine con cambio di segno indotto da  $R$  viene sovrapposto al cambio di segno portato da  $I$  causando l'elisione mutua dei due cambi di segno. Il riconoscimento della trasformazione sintetizzata da  $T \cdot R \cdot I$  produce un vettore d'errore  $E$  ottenuto sottraendo gli elementi antagonisti dei due vettori caratteristici, ossia sottraendo il primo di un vettore all'ultimo dell'altro, il secondo al penultimo e così via.

Il vettore  $E$  decreta ovviamente la somiglianza dei due frammenti come negli altri casi.

Con questa prima famiglia di strumenti, si possono esaminare con pochi calcoli diverse ipotesi di elaborazioni contrappuntistiche tra due frammenti melodici.

Soprattutto si è fornito un modello formale espandibile anche verso il riconoscimento di ogni tipo di manipolazione da parte del compositore sugli oggetti musicale, passato e futuro.

## Complessità

L'approccio per operatori melodici reperisce i dati dalle informazioni di partitura, e conservandone la potenza espressiva le rende estremamente maneggevoli dal calcolatore.

Purtroppo questa velocità nel riconoscimento di rielaborazioni tematiche ha insita nell'eshaustività del metodo un suo limite: in assenza di metodi complementari per la selezione di oggetti musicali, l'unico modo per essere certi di non trascurare materiale rilevante è un confronto totale in cui ogni frammento melodico viene confrontato con tutti gli altri di ugual lunghezza, e questo per ogni possibile lunghezza accettabile.

Questo significa che ponendo un limite di 2 (o 3) note di lunghezza minima per la definizione di un oggetto musicale (che nel caso del tema del Tristano di Wagner è di 1 sola nota per parte reale), ci si può aspettare di dover eseguire un numero di confronti asintotico a  $O(N^2)M$ , con:

N numero medio di note per parte(staff)

(e quindi N-1 numero coppie di note consecutive della partitura, N-2 numero di incisi di 3 note consecutive, ecc.)

M numero di parti reali presenti nella partitura

Ricordiamo che il confronto di un frammento melodico in realta' comprende tanti confronti quante sono le note che lo compongono, moltiplicato per il numero C di operatori valutati.

Per confrontare le coppie di note consecutive, si eseguiranno N-2 confronti per la prima coppia, N-3 confronti per la seconda, e cosi via. Ognuno di essi implica 1 solo confronto (1 solo movimento melodico)

Per verificare l'elaborazione di incisi di due note, sono quindi necessari

$$[(N-2) + (N-3) + (N-4) \dots 2] * M$$

confronti . Nel caso di incisi di 3 note, il numero di confronti sara'

$$[(N-3) + (N-4) \dots 2] * M * 2$$

nel caso di incisi di 4 note, il numero di confronti sara'

$$[(N-3) + (N-4) \dots 2] * M * 3$$

ecc... quindi il costo complessivo e' dell'ordine quadratico.

Un altro problema di questo metodo e' la sua specificita': per le manipolazioni descritte il risultato e' ottimale, ma in presenza di altri artifici quali la riduzione o i cambi di armonizzazione, qualora venisse espanso, incorrerebbe in costi computazionali ben maggiori.

Per questo e' stata affiancata la famiglia di operatori armonici, a cui viene demandata quella che in analisi musicale puo' essere definita "capacita' selettiva di discernimento", oltre che a superare i limiti derivanti da una scansione orientata alla singola nota: l'impossibilita' di estrapolare informazioni sulla semantica dei risultati, sulla loro importanza (se non valutandola euristicamente in base alla lunghezza delle frasi trovate) o di stabilire tolleranze entro le quali alcuni mismatching restino accettabili per quantita e per importanza.

# Operatori armonici

L'approccio armonico ha completato il sistema di analisi a livello semantico - contestuale, al servizio della segmentazione. Il limite di ogni approccio statistico e analitico per operatori come quello visto finora, è la località delle soluzioni. Senza una cognizione strutturale di metalinguaggio musicale, alla base della generazione fraseologica, ogni riconoscimento resta fine a se stesso. Eventuali regole generali che possano governare e spiegare a priori l'espressività della composizione sarebbero di aiuto fondamentale nella comprensione della loro costruzione.

Sebbene non siano ancora state individuate se non parzialmente regole come quelle descritte sopra, l'analisi musicale fornisce tramite una sua branca, lo studio dell'armonia, un corpus di regole studiate ed estrapolate a posteriori dalle opere degli autori. Esso è uno strumento conoscitivo con cui analizzare il brano specifico. Sebbene estremamente utile, questa base di conoscenza è da usare con delicatezza e cautela, a causa della sua intrinseca non univocità.

Gli operatori melodici già in nostro possesso rappresentano l'analisi sintattica del linguaggio musicale. La valutazione dell'armonia è uno strumento in grado di creare ipotesi per la delimitazione di frasi esistenti nella trama notazionale. In realtà l'armonia fornisce diverse chiavi di interpretazione, ma ai nostri scopi servono semplici direttive il più possibile epurate dalla necessità dell'apporto umano.

Alla base di questo, si noti bene, non c'è l'esigenza di uno strumento discriminante, ma di uno strumento riconoscitivo, per quanto sottile sia la differenza. Questa differenza con i metodi precedenti rende complementare questo approccio per lo spostamento dell'attenzione verso le regole di analisi anziché quelle di "sintesi" in materia di composizione (sfruttate dagli operatori melodici e ritmici). L'approccio è anche coerente sul piano concettuale: mentre la melodia rappresenta una visione del dettaglio, ed il suo studio è rivolto in direzione Top-Down, lo studio delle strutture profonde del brano è basato sull'astrazione, a livelli sempre più globali.

Da questo segue immediatamente un nostro importante postulato: l'armonia è scarsamente sensibile alla variazione melodica. Riflettiamo un attimo sulla necessità dell'integrazione di questo metodo. Un frammento ripetuto numerose volte risulta un tema plausibile, ma ogni arrangiamento sfrutta per l'accompagnamento diverse figurazioni con la caratteristica di ricorrenza ritmico-melodica. Ascoltando un qualunque brano sinfonico si nota come tra le parti dell'orchestra compaiano frammenti ripetuti pedissequamente, spesso anche stereotipati in una prassi compositiva.

A rendere indispensabile un approccio contestuale è l'osservazione che questo materiale possa costituire l'idea compositiva fondamentale in altre composizioni. L'analisi non fornisce oggi caratterizzazioni sufficienti per poter discriminare gli oggetti principali da quelli secondari su nessuno dei piani ritmico melodico e armonico presi singolarmente.

Ciò che mette in relazione gli oggetti è il rapporto di importanza che, in base a regole prevalentemente estetiche, li distingue in materiale tematico e accompagnamento. Questo può essere tracciato con un'esame congiunto delle tre dimensioni, un metodo "diagonale" in cui ogni famiglia di operatori opera una selezione sui dati ricavati.

## La verticalizzazione

Questo operatore è applicato all'insieme dei suoni prodotti, e non agisce analizzandone né i rapporti di altezza né i ritmi, ma esplorandone la contemporaneità. Pertanto, il dato fondamentale che esso cerca è la localizzazione temporale dell'evento sonoro. Esso può essere identificato come una funzione di scelta che, avendo per dominio l'insieme di tutti i suoni (note) del brano, porta in un'insieme denominato "armonia" un suono per volta, o nessuno, qualora siano esauriti i suoni simultanei. Detto operatore ha la capacità di discernere i suoni detti ridondanti, per i quali l'analisi armonica non è sensibile, operando un modulo 7 sui gradi (cioè eliminando tutte le ottave sovrapposte in un accordo), in modo però sensibile al pitch (tenendo conto quindi di intervalli diversi per ampiezza in semitoni, ma che interessano gradi uguali).

Si definisce operatore di verticalizzazione e lo si indica con  $V: \Sigma \rightarrow P(\Sigma)$  un operatore così definito:

$$V_T(h): A_T = A_T U \{h\} \text{ se } T(h) = T$$
$$V_T(h): I(A_T) \text{ altrimenti}$$

Come si può vedere l'operatore insiemistico di verticalizzazione aggiunge un elemento all'insieme A quando questo evento è contemporaneo a tutti gli eventi nell'insieme A (e quindi, ponendo l'attenzione sui casi base in cui A abbia un solo elemento o sia vuoto, è ovvio come tutte le note di un'armonia risultino tutte sincronizzate). Per quanto detto sulle classi di equivalenza e per le proprietà delle operazioni insiemistiche, è ovvio come la non ridondanza sia garantita dall'operazione di unione e dal concetto di insieme.

Nel semplice esempio di figura 12, il frammento è preso da un corale di J.S. Bach, in cui rappresentiamo ogni intervallo di un quarto con il pedice unitario. Gli accordi non devono necessariamente essere costruiti di ugual durata e ritmicamente regolari, in questo caso ci siamo concessi una semplificazione scegliendo un frammento omoritmico. Così il primo quarto avrà pedice 1, il secondo pedice 2, ecc.



Figura 12

Alla lettura della terza voce, tenore, colonna di sinistra, e alla lettura della voce di basso, colonna di destra, avremo la situazione armonica:

$$V_1(): A_1 = \{mi, sol, si\} \quad ( = Mi bemolle maggiore )$$

$$V_2(): A_2 = \{mi, sol, si\} \quad ( = Mi bemolle maggiore )$$

$$V_3(): A_3 = \{re, fa, si\} \quad ( = Si bemolle maggiore )$$

$$V_4(): A_4 = \{reb, mi, sol, si\} \quad ( = Mi bemolle maggiore con settima di dominante )$$

$$V_5(): A_5 = \{do, mi, la\} \quad ( = La bemolle maggiore )$$

L'operatore di verticalizzazione ha costruito gli accordi ai vari intervalli di tempo. E' immediato riconoscere la natura dell'accordo, esplicitata tra parentesi, applicando le basi della teoria armonica. Mentre questo esempio è stato tutto scritto con la notazione tradizionale per immediatezza, il risultato numerico dell'elaborazione, sarebbe stato:

$$A_1 = \{2, 4, 6\} \quad A_2 = \{2, 4, 6\} \quad A_3 = \{1, 3, 6\} \quad A_4 = \{1, 2, 4, 6\} \quad A_5 = \{0, 2, 5\}$$

Una prima generalizzazione è catturata dall'esempio di figura 13:



Figura 13

Questa è la battuta successiva, nel corale preso ad esempio (quindi ha ancora i tre bemolli in chiave). Inizialmente, leggendo la riga del soprano, esso incontra tre armonie banali:

$$V_1(si): A_1 = \{si\} \quad V_2(sol): A_2 = \{sol\} \quad V_3(do): A_3 = \{do\}$$

Leggendo poi la riga del contralto, incontra una nota dalla partenza simultanea alla prima (incontrata all'istante 1), ma di durata diversa da quella che è divenuta poi  $A_1$ .

Ecco allora come agisce l'operatore, considerando che  $A_1 = A_1^{1/8} \cup A_1^{3/8}$ , dove  $A_1^{1/8}$  indica il primo ottavo della durata di  $A_1^1$  e  $A_1^{3/8}$  indica il complementare di questa durata rispetto ad  $A_1$ .

$$V_1(re): A_1 = \{si, re\}$$

Ora però il prossimo cambio armonico è un ottavo dopo l'inizio di  $A_1$  e una parte di armonia è ancora da rappresentare, esattamente i tre ottavi esclusi dalla verticalizzazione rispetto alla nuova nota. Quindi  $A_2$  sarà la nuova armonia di tre ottavi e le armonie già esistenti aumenteranno l'indice di 1.

$$V_2(mi): A_2 = \{mi, si\}$$

Ancora, quindi, avremo creato un'armonia della durata di un ottavo e rimarrà dall'armonia sezionata ancora un quarto. E ancora le armonie già esistenti aumenteranno l'indice di 1.

$$V_3(fa): A_3 = \{fa, si\} \quad V_4(re): A_3 = \{re, si\}$$

A questo punto, l'armonia iniziale di due quarti è stata letteralmente fatta a pezzi, lasciando al suo posto quattro armonie di un ottavo ciascuna. L'accordo che alla prima passata era stata numerata come secondo, ora è finito in quinta posizione, per lasciare posto alla scomposizione del precedente.

Vediamo ora come termina la verticalizzazione :

$$V_1(): A_1 = \{si, re, fa\} \quad V_2(): A_2 = \{do, mi, fa, si\}$$

$$V_3(): A_3 = \{re, fa, si\} \quad V_4(): A_3 = \{re, si\}$$

$$V_5(): A_5 = \{mi, sol, si\} \quad V_6(): A_6 = \{mi, la, do\}$$

Si noti come, essendo in questo esempio la parte del contralto quella che "muove" più velocemente (si indica così il cambio di nota in un contesto armonico), determina il ritmo con cui gli accordi si susseguono. Questo è un caso, come è una coincidenza dell'esempio il fatto che il basso abbia lo stesso ritmo del contralto.

La regola generale da evincere è come la nota più breve determini una nuova valutazione del cambio armonico, e la sua frammentazione. Questo non vuol dire che l'esito dell'intero programma sarà una frammentazione anziché una segmentazione, in quanto, una volta ottenuti tutti i singoli momenti sonori, sarà possibile decidere tra essi quali siano determinanti e quali possano o debbano invece essere eliminati. Infatti, al termine della scansione si scorgono ancora una volta nitidamente gli accordi e, tra essi non è difficile riconoscere le armonie principali e le armonie di passaggio, servendosi dell'ausilio di pochissime regole formali.

## L'operatore di riconoscimento funzione tonale (RFT)

Questo operatore è applicato all'insieme dei suoni presenti nell'accordo, quindi è anch'esso un operatore insiemistico. Tuttavia, esso ha dominio in un insieme delle parti del set iniziale contenente i 12 suoni della scala cromatica, e codominio in un insieme simile a quello delle sole classi di equivalenza relative ad ogni grado.

Più semplicemente questo operatore, dato un qualunque accordo, dovrebbe interpretare il grado della scala su cui esso è costruito, e trarre conclusioni in merito alla ragione per cui è stato costruito in quella posizione. Quest'ultima fase, sebbene espressa sinteticamente, necessita di un riscontro globale, quindi a questo operatore sarà richiesto il già arduo compito di riconoscere l'armonia rappresentata dall'argomento e sintetizzarla in un metalinguaggio musicale.

Il compito dell'operatore RFT sarà una riduzione di tutto un insieme di note in un simbolo scelto tra 3 classi, ammettendo al più un ulteriore simbolo di indefinitezza che però deve rimanere altamente improbabile. Questo perché si auspica che in un brano musicale, istante per istante, l'ambiente sonoro denoti una tra le tre regioni di Tonica, Dominante o Sottodominante.

Si definisce operatore di riconoscimento funzione tonale e lo si indica con **RFT:  $P(\Sigma) \rightarrow \{T, D, SD, 0\}$**  un operatore così definito:

---

<sup>1</sup> infatti  $A_1$  dura due quarti, e la nota da inserire dura un ottavo.

$RFT(A) = T$  se  $A$  è accordo di *Tonica* (cioè costruito sul I o VI Grado)

$$RFT(A) = D$$

se  $A$  è accordo di *Dominante* (cioè costruito sul V o VII Grado)

$$RFT(A) = SD$$

se  $A$  è accordo di *SottoDominante* (cioè costruito sul II o IV Grado)

$$RFT(A) = 0 \quad \text{altrimenti (Caso indefinito)}$$

Come si può intuire, una volta in possesso di un insieme di note, risultato della verticalizzazione della partitura, il lavoro dell'operatore RFT consiste nell'ordinare i gradi presenti nell'accordo in modo che ad un grado ben definito, che verrà riconosciuto come *Basso Armonico*, segua un intervallo dispari, cioè di terza, o almeno di quinta. Nel caso questo non sia possibile, l'accordo verrà riconosciuto come meno stabile, e si accetteranno intervalli via via più dissonanti secondo le regole dell'armonia tonale, quindi di sesta, di quarta ed infine di settima e di nona.

Scopo di questa ricerca non è l'acquisizione del basso armonico, ma della fondamentale dell'accordo, cioè del grado su cui, per terze, si è costruito l'accordo. Essa spesso coincide per motivi estetici e acustici con il basso armonico, e quando questo non accade, comunque il basso ci aiuta nella ricerca.

Per decidere quale nota considerare basso armonico, ci viene incontro una semplicissima regola armonica, che decreta come ogni armonia si appoggi sulla nota più grave acusticamente, quindi di altezza minore.

In questo caso torna utile la duplice notazione delle note tenuta in memoria, in quanto il confronto dei pitch ci dirà immediatamente quale sia il basso, e il grado fornirà ancor più velocemente la posizione tonale dell'accordo, come vedremo alla fine del procedimento. Immediatamente dopo, si riconoscono tutti gli intervalli generati dalle altre note rispetto al basso.

Una volta riconosciuto il basso e gli intervalli soprastanti, si esaminano da subito, come detto sopra, quelli dispari, in particolare ponendo attenzione alle seguenti regole di "ridondanza":

- Se l'accordo contiene note ad un intervallo di terza e di quinta dal basso armonico, abbiamo l'accordo in stato fondamentale, è costruito sulla nota del basso, che ne è la fondamentale. Questo è lo stato più stabile, un chiaro segnale che siamo in presenza di un accordo il cui significato (sia esso tonica, dominante o sottodominante) è messo in risalto.
- Se l'accordo contiene note ad un intervallo di terza e di sesta dal basso armonico, abbiamo la triade in primo rivolto, cioè la nota su cui l'accordo è costruito (fondamentale) è stata innalzata di una o più ottave, e quindi al basso è rimasta la terza dell'accordo. Per ottenere la fondamentale è sufficiente sottrarre due gradi al basso armonico o, a seconda della rappresentazione dei dati e quindi della comodità di accesso ai gradi dell'accordo, prendere la nota che dista una sesta dal basso armonico. Si noti l'equivalenza delle due scelte.
- Se l'accordo contiene note ad un intervallo di quarta e di sesta del basso armonico, abbiamo la triade in secondo rivolto. La fondamentale e la terza immediatamente sopra sono state innalzate di una o più ottave, lasciando al basso la nota che era a una quinta di distanza dalla fondamentale. Ora "la quinta dell'accordo" è passata a ricoprire il ruolo di basso armonico. Per riottenere la fondamentale è sufficiente sottrarre ad esempio quattro gradi al basso armonico (scendere cioè di una quinta).

Visualizziamo quanto detto ora nella figura, dove, nell'ordine, sono riportati i tre casi descritti:



Figura 14

Analogamente, nel caso di un accordo di 4 note distinte, possiamo trovarci nel caso in cui la nota in più sia una settima:

- eliminati i gradi dell'armonia distanti intervalli di terza e quinta dal basso, se rimane solo una nota ad una distanza di settima, essa denota che l'accordo è una dissonanza. Compositivamente la settima è una piccola complicazione che deve essere trattata con alcune cautele, ma questo esula oltre i nostri scopi. Infatti, nell'analisi, un accordo con settima, quando non indica modulazioni transitorie, può essere considerato della stessa classe della triade che avremmo comunque, tralasciando la settima.
- Se i gradi dell'armonia si dispongono in modo che ci siano intervalli di terza, quinta, e sesta, siamo in presenza di un accordo con settima, in primo rivolto. Possiamo scegliere di scendere di due gradi (terza) rispetto al basso, che è l'equivalente di riconoscere la fondamentale dell'accordo accordo nella nota ad una sesta di distanza dal basso armonico.
- Se i gradi dell'armonia si dispongono in modo di avere intervalli di terza, quarta e sesta, siamo in presenza di un accordo con settima, in secondo rivolto. Analogamente, la fondamentale è la nota che precede il basso nella scala per quattro gradi. Infine, se i gradi dell'armonia si dispongono formando intervalli di seconda, quarta e sesta, siamo in presenza di un accordo con settima, in terzo rivolto. Calcoliamo la nota su cui è costruito l'accordo salendo di un grado (seconda) rispetto al basso.

Può capitare che un insieme di note non abbia sufficiente informazione per permetterci di capire univocamente quale grado armonizzi: abbiamo detto che un accordo è un insieme di almeno tre suoni, quindi nel caso in cui i suoni simultanei siano soltanto due o al peggio uno, pur non contravvenendo a nessuna regola armonica, ci troviamo con una ambiguità.

L'operatore deve comunque essere definito: nel caso di due note verifica se una di esse possa essere quinta dell'altra, limitando solo al modo l'ambiguità dell'accordo (che resta privo della sola nota "caratteristica"):

- Nel caso di due note ad una terza di distanza, si assume che la nota inferiore sia la fondamentale.
- Se distano per una seconda, si può riconoscere una settima in terzo rivolto.

Nel caso degenerare in cui l'armonia sia composta da un suono solo, esso è chiaramente l'unica informazione a disposizione per determinarne la fondamentale, e quindi l'assunzione più plausibile è proprio che esso sia la fondamentale stessa. Armonie degeneri di questa specie vengono valutate in un modo dedicato, che tenga conto della loro diversa valenza "armonica". Un'armonia degenerare è trattata come i simboli risultanti dalla verticalizzazione: rappresenta l'ambiente sonoro per tutta la durata dell'armonia degenerare. Pertanto non può essere tralasciata. Inoltre, quando accade che si possa percepire una sequenza di suoni singoli senza ambiente armonico sottostante, è probabile che l'intenzione del compositore sia stata portare all'attenzione dell'ascoltatore esattamente quella frase. Pertanto, se una frase costituita da armonie "degeneri" raggiunge la lunghezza ammissibile per essere considerata tema, essa entra di diritto nell'elenco dei potenziali temi.

Nel caso in cui la tonalità non sia DO Maggiore, l'accordo deve essere correlato al suo significato percettivo relazionandolo alla tonalità d'impianto. L'accordo assume significato in base al grado della scala su cui è costruito. Grazie a questa informazione si estrae l'indicazione "linguistica" dell'accordo stesso. Una tonalità diversa da quella di riferimento trasla il baricentro sonoro, e la fondamentale di ogni accordo può essere corretta sottraendo al grado riconosciuto dall'operatore la tonica della tonalità. Una ulteriore scelta, collassa a due a due le classi dei gradi in tre classi relative alle funzioni principali Tonica, Dominante e Sottodominante.

## L'operatore Cadenza Armonica (CA)

L'operatore CA riduce a composizione di semplici formule logiche il problema complesso della individuazione delle cadenze, assegnando valore di verità ad una qualunque sequenza di armonie esaminata, e verificando tutte e solo quelle che possono essere cadenze armoniche.

La corrispondenza posta tra ogni insieme ed ogni funzione tonale, premette ora di trattare ogni accordo come un simbolo del linguaggio, un'informazione unitaria.

Data la sequenza di accordi  $A1, A2, A3, \dots, Am$  elementi di  $P(\Sigma)$

$$CA(A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots, A_n) = CA(A_i, A_n)$$

Abbiamo il seguente caso base:

$$CA(A_i, A_n) = CA_{Inizio}(A_i) \cdot CA_{Fine}(A_n)$$

Con:

$$CAInizio (Ai) = (RFT(Ai)=T) V (RFT(Ai)=D) V CAFine (Ai-1)$$

$$CAFine (An) = (RFT(An)=T) V (RFT(An)=D)$$

Si noti come tra le condizioni determinanti affinché l'armonia i-esima sia riconosciuta quale possibile inizio di tema, basti supporre che l'armonia precedente ad essa implichi una conclusione tematica.

La necessità di soddisfare contemporaneamente la condizione di inizio tema assieme a quella di fine tema sono vincoli sul singolo risultato, e non precludono all'operatore la possibilità di formare un insieme relativamente grande di oggetti musicali selezionati.

## L'operatore Ridondanza Armonica (RA)

Descriviamo qui l'operatore ideale di Ridondanza Armonica. Esso, applicato alla sequenza di M accordi

$$(Ai, Ai+1, Ai+2, Ai+3, \dots Ai+M)$$

ricerca all'interno del brano una sequenza analoga

$$(Aj, Aj+1, Aj+2, Aj+3, \dots Aj+M)$$

Potremmo quindi considerarlo l'equivalente dell'operatore T per gli operatori melodici nel suo caso banale, l'operatore T0, cioè l'identità. Questo operatore però agisce su insiemi, e non è limitato al caso in cui le sequenze combacino perfettamente. Si noti come la riduzione degli accordi in 3 tipi ne abbatta la numerosità rispetto alle note, ma non l'informazione veicolata, che anzi è maggiore. Inoltre esso opera sulla trasposizione identica tonale, poiché, a meno di modulazioni riconoscibili con altri mezzi, esso non risente di alterazioni.

La teoria musicale fornisce una collezione di regole basate sul concetto di tonalità, e ogni movimento armonico è previsto e "regolamentato" all'interno di essa. In questo modo possono essere valutate le più piccole varianti nel tessuto armonico e nei movimenti tonali, applicando tutte le regole e restrizioni presenti nella base di conoscenza dell'armonia. A noi serve solo la parte "macroscopica" di questo codice, in quanto stiamo cercando zone altamente informative, quali frasi e temi, e non zone istantanee quali sonorità di passaggio o ornamenti del compositore.

L'esperienza dei limiti trovati agli operatori melodici ha sottolineato la necessità di creare la nuova famiglia di operatori esente dalle difficoltà intrinseche di un'analisi orientata alla nota. Perciò, se ad una trasposizione tonale deve essere consentita una tolleranza, allo stesso modo si può prevedere che una sequenza di accordi non si ripeta pedissequamente, ma abbia leggere variazioni. Spesso una frase termina dapprima su un accordo di dominante per lo specifico senso di sospensione, per poi essere ribadita sulla armonia di tonica per dare il "sapore conclusivo".

In questo caso l'operatore RAX può ignorare x armonie differenti, nella sua valutazione. Si noti come questa "feature" si traduca sul piano pratico nell'insensibilità dell'algorithmo verso un numero anche elevato di note confrontate senza successo, purché tutte restino confinate nei limiti temporali ben delimitati dal movimento armonico variato. Un grande vantaggio di questo operatore è che la tolleranza di una armonia errata non deve considerare anche il caso in cui questa armonia sia assente.

Nel caso del confronto orientato alle note, una nota poteva essere, tramite il processo di riduzione, sdoppiata ad esempio in nota reale ed appoggiatura. Sebbene la nota in più vanificasse completamente la procedura, shiftingo nel confronto le successive, essa non differenzia l'oggetto musicale. Gli accordi significativi, invece, si susseguono ad intervalli relativamente ampi, cosicché l'assenza di un accordo in un pattern sia una differenza sufficientemente grave da compromettere il confronto.

Un'altra differenza con l'operatore T, basato sulle regole di sintesi, che forniva il riconoscimento di strutture e tecniche usate dai compositori, l'operatore RAX filtra e trattiene da subito le sequenze che trova ridondanti, classificandole in base alla loro ripetitività. Questa cernita però non è eseguita senza controlli, in quanto ogni cadenza deve rispettare almeno vincoli di lunghezza .

I vincoli di lunghezza eliminano molto bene gli incisi, altamente ripetitivi ma fuorvianti nella ricerca tematica a meno di avere la capacità di aggregare frammenti tematici a distanze non prevedibili tra loro, catturandoli solo nelle loro strutturazioni in frasi.

Sebbene l'operatore RA, comportando un'operazione di substring searching, sembri rappresentare in termini e modelli matematici l'operazione più costosa dal punto di vista computazionale, esso ha semplicemente un costo quadratico, riducibile ulteriormente con classi di algoritmi già ufficialmente riconosciuti.

Il vantaggio computazionale di questo metodo è la limitata variabilità della quantità di dati, e l'estrema maneggevolezza di un numero ridotto di accordi, rispetto all'insieme iniziale di note. A questo si aggiunge la restrizione a determinate lunghezze plausibili di sequenze accordali. Infatti si effettua il riconoscimento di gruppi di accordi con determinata lunghezza, anziché un esame esaustivo di tutte le possibili sottostringhe della spine iniziale.

Questa proprietà sulla lunghezza delle stringhe plausibili ha senso in analisi armonica, mentre è molto meno verificabile in una analisi orientata alla nota. Nel caso peggiore, quindi, in cui si decida di cercare tutte le cadenze in un brano, a partire dalla lunghezza del brano fino ad arrivare al fraseggio basato sulla singola armonia, non si supera comunque la complessità già connaturata a questo programma al momento in cui esso è stato ripreso. In compenso il lavoro sviluppato comporta due notevoli vantaggi:

- La ricerca viene eseguita su strutture a livello superiore, nell'ambito del linguaggio musicale.
- Le cadenze sono, in numero, non maggiori dell'ordine delle note di un brano, anche se un caso in cui si eguagliano è altamente improbabile e non significativo.

All'aumentare del metro armonico (cioè la frequenza con cui si cambia armonia) per multipli della durata media delle note, il rapporto tra numero di armonie e numero di note tende a decrescere con andamento logaritmico. Inoltre il numero di armonie tende a ridursi quando le note sono disposte su più pentagrammi, in quanto un accordo è per definizione generato da più note.

Questo succede quasi sempre nelle partiture. Se per esempio un brano fosse un corale, a 4 voci (pentagrammi), scrittura omoritmica e 32 note da un quarto per rigo, con battute in 4/4 e cambiamento armonico ogni mezza misura (cioè ogni 2/4), avremmo:

128 note (32 note per pentagramma, moltiplicato per i quattro pentagrammi)

contro

16 armonie (32 note perché ogni voce "muove" contemporaneamente, diviso 2, in quanto ogni armonia dura 2/4, quindi due note).

## Le combinazioni di operatori utilizzate

L'uso degli strumenti per l'analisi armonica è vincolato alla possibilità di operare sul significato sonoro e non sulle singole note. Questo significato è una sorta di "valore aggiunto" dalla simultaneità dei suoni che il compositore valuta attentamente in fase di composizione. E' praticamente la differenza tra l'entropia di ogni possibile aggregazione di lettere dell'alfabeto valutata nella totalità delle stringhe costruibili e l'entropia delle parole valutata nell'insieme "vocabolario".

Per questo, prima di ogni valutazione, nel momento stesso in cui decidiamo di usare l'ausilio che l'analisi ci mette a disposizione, e' necessario preparare i dati che verranno utilizzati, e che saranno il risultato di elaborazioni partendo dalla notazione della partitura.

Come detto all'inizio della presentazione degli operatori armonici, il sistema di riconoscimento analizza il brano sulle tre dimensioni ritmico, melodica e armonica. Questa analisi puo' essere congiunta, configurando il sistema tramite la combinazione degli operatori. La possibilità di eseguire una ricerca basata solo su ricorrenze ed elaborazioni melodiche era già collaudata nelle ricerche precedenti. Oltre a questa, i due operatori armonici potranno essere combinati, singolarmente o in tandem, a tutti gli strumenti già esistenti.

I due operatori armonici principali verranno applicati a tutte le possibili sequenze di M armonie, dove M rappresenta un range di lunghezze proposto dall'utente e successivamente leggermente ottimizzato dal sistema.

Avremo quindi, per la sola parte relativa all'analisi armonica, le seguenti combinazioni di operatori:

$$\begin{aligned} & CA (A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots A_{i+M},) \\ & RA (A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots A_{i+M},) \\ & RA (CA (A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots A_{i+M},)) \\ & MEL.OP.(CA (A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots A_{i+M},)) \\ & MEL.OP.(RA (A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots A_{i+M},)) \\ & MEL.OP.(RA (CA (A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots A_{i+M},))) \end{aligned}$$

Chiaramente intendendo con **MEL.OP.** l'applicazione di qualsiasi combinazione di operatori melodici. Essi, all'ultimo stadio, raffinanò l'output ottenuto dalla "indagine armonica". Si noti che in questo caso è stato modificato l'algoritmo esistente e sono stati velocizzati i calcoli eseguiti sui frammenti melodici. Questo perché il frammento candidato non viene ricercato tra tutti i gruppi di note ma viene direttamente proposto dalle nuove funzioni che a tal fine isolano un certo gruppo di note.

Finora sono stati presentati due gruppi di operatori, separandoli a livello concettuale: gli operatori ritmici e melodici sono dedicati all'analisi della melodia mentre quelli armonici producono una rielaborazione della partitura esplicitando e sintetizzando informazioni necessarie per una lettura "globale" della partitura. Schematizziamo la componibilità degli operatori raggruppandoli in tre stadi, uno dei quali sarà "composito". Ognuno di questi stadi è progettato per riconoscere una caratteristica musicale che possa essere spia della presenza di un tema, e con opportuni adattamenti, come abbiamo visto, combinato agli altri.

La prima categoria è quella di tutta la famiglia di operatori melodici, unita a quella degli operatori ritmici. Entrambi agiscono su sequenze di note, e il loro obiettivo è organico nel cercare elaborazioni tematiche dislocate in un qualsiasi ordine sparso in tutta la partitura. Questo stadio è riconosciuto "composito" non solo a causa dell'interazione di due dimensioni complementari dell'informazione musicale, ma anche per la cooperazione di tutti gli operatori che compongono entrambe le famiglie di metodi. La riunione operata ha come criterio il modus operandi simile di questi strumenti di analisi. Alla base della combinazione con altri metodi, inizialmente, vi era la complessità computazionale causata dalla ricerca esaustiva sulla partitura completa. Questo metodo è stato così designato per essere sempre lo stadio finale di ogni segmentazione, al fine di poter elaborare (quando possibile) un insieme di dati già ridotto.

Quando l'analisi non è basata soltanto sulle linee melodiche, viene innescato un processo di "selezione e verifica", in modo da utilizzare lo stadio ritmico-melodico come "controllo di qualità dei risultati", risultati selezionati dagli stadi precedenti. Infatti in questo modo si valorizza il compito di un controllo atto a valutare le caratteristiche intrinseche ad un singolo frammento, non costringendolo a valutare metodicamente ogni tipo di informazione notazionale in partitura. Questo perché i frammenti non significativi generati iterativamente sono di un ordine superiore rispetto al numero di oggetti musicali effettivamente presenti.

I metodi armonici hanno invece il vantaggio di selezionare oggetti candidati senza dover costruire tutti i frammenti, e per questo si prestano bene alla fase di selezione. Se impiegati come unico operatore, ognuno dei due stadi operanti sull'armonia fornirà una lista ordinata di potenziali oggetti. La lista di output sarà quindi la chiave di volta per poter mettere i diversi stadi in comunicazione tra loro, tramite i loro risultati.

Definito un ordine per i due stadi armonici, il prototipo inizierà l'elaborazione passando le informazioni a sua disposizione al metodo selezionato che occupa una posizione prioritaria rispetto agli altri. Il suo output diventerà l'insieme di dati su cui lo stadio successivo eseguirà le sue valutazioni, e così il terzo.

Notiamo innanzitutto come gli operatori armonici impongano una serie di adattamenti preliminari della partitura, a causa della particolare forma di dati di cui necessitano per eseguire la ricerca. La forma accordale della partitura non è una informazione di partitura, e come tale deve essere generata solo quando essa è utile all'elaborazione. Pertanto, ogni volta che l'utente sceglie uno stadio armonico di analisi, il programma preprocessa la partitura con l'operatore di verticalizzazione e riconoscimento tonale.

Tra i due operatori armonici si è scelto quello di "Cadenza Armonica" come iniziale, in quanto è quello che non opera, a sua volta, un confronto esaustivo tra stringhe di simboli. Nonostante i simboli in questo caso indichino macroeventi, cioè accordi che possono raggruppare anche decine o addirittura centinaia di note ognuno, l'altro stadio armonico offre ancora un approccio statistico, e quindi più oneroso.

Inoltre la qualità semantica dell'operatore di Cadenza Armonica ha avuto un ruolo importante in questa scelta: mentre ha immediatamente senso pensare di isolare in un brano le frasi in qualche modo "ben formate" armonicamente e successivamente esaminare quali sono i contesti sonori più ripetuti, l'ordine inverso causerebbe una perdita inevitabile di materiale utile potendo cercare tra tutte le cadenze, comprese quelle "sintatticamente inesatte". Ricordiamo che una selezione sintattica a livello armonico presuppone una delimitazione semantica, almeno a livello di ubicazione temporale, dei fraseggi melodici.

Quando l'operatore di Cadenza Armonica è attivo, esso ha il compito di passare allo stadio successivo la lista ordinata in base ai suoi valori interni di plausibilità. Questa lista viene creata in base al grado di articolazione della cadenza e alla sua struttura interna. Nel caso in cui sia l'unico stadio selezionato, esso fornisce in output soltanto i primi oggetti trovati, tanti quanti ne sceglie l'utente, tralasciando gli altri.

L'operatore di Ridondanza Armonica, può compiere i confronti di somiglianza ordinati per articolazione e per metrica delle imprecisioni riscontrate nel confronto, indifferentemente andando a selezionare tutte le possibili sottosequenze di simboli armonici della spine della partitura oppure, qualora l'operatore di cadenza Armonica sia stato attivato, su ogni entry della lista dei risultati fornita dallo stadio precedente.

Anche questa fase restituisce una lista ordinata per rispondenza a criteri statistici, diversi da quelli utilizzati per lo stadio precedente, e anche questa fase nel caso si trovi ad essere l'ultima del processo, taglia la lista trattenendo solo il numero di oggetti richiesti dall'utente.

Infine, lo stadio di operatori melodico-ritmici, valuta i risultati che arrivano dagli stadi precedenti, se ve ne sono, scegliendo i frammenti in cui riconoscono meglio le manipolazioni su cui sono calibrati.

Nel caso in cui essi lavorino in modo stand-alone, senza l'ausilio della selezione tramite criteri armonici, essi fanno scorrere l'intera partitura confrontando ogni frammento, escludendo solo quelli sotto un minimo di note in comune e quelli più lunghi di una soglia definita dall'utente.

Pertanto, rifacendoci all'elenco del paragrafo precedente, ecco come ogni combinazione di operatori incide sul flusso di informazione tra i vari stadi del processo:

**CA (.)** *sceglie e restituisce all'utente le N cadenze piu' articolate e rispondenti ai parametri di ingresso armonici e all'ordine di presentazione*

**RA (.)** *sceglie e restituisce all'utente le N cadenze piu' ripetute e rispondenti ai parametri di ingresso armonici e all'ordine di presentazione*

**MEL.OP.(.)** *valuta tutti i frammenti, creando indici di somiglianza per il piano ritmico e quello melodico, la combinazione di queste informazioni è un valore discriminante di scelta su tutto l'insieme. Restituisce gli N frammenti con punteggio maggiore*

**RA (CA ( . ))** *Seleziona tutte le frasi armoniche ben formate sintatticamente, e tra queste sceglie e restituisce le N cadenze piu' ripetute e rispondenti ai parametri di ingresso armonici e all'ordine di presentazione*

**MEL.OP.(CA (.))** Seleziona tutte le frasi armoniche ben formate sintatticamente, traccia sulla partitura per ogni cadenza un gruppo di frammenti melodici che la cadenza stessa sottende, in base a regole che rispettano una tolleranza nei valori ritmici, e offre questi frammenti agli operatori melodici affinché scelgano i più plausibili.

**MEL.OP.(RA ( $A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots, A_{i+M}$ ))** Seleziona tutte le frasi armoniche ripetute, inserendo priorità in base a quanto sono ripetute. Traccia sulla partitura per ogni cadenza un gruppo di frammenti melodici che la cadenza stessa sottende, in base a regole che rispettano una tolleranza nei valori ritmici, e offre questi frammenti agli operatori melodici affinché scelgano i più plausibili.

**MEL.OP.(RA (CA ( $A_i, A_{i+1}, A_{i+2}, A_{i+3}, \dots, A_{i+M}$ )))** Seleziona, ordinandole per grado di articolazione, tutte le frasi armoniche ben formate sintatticamente. Effettua una scelta sulla base di quanto esse vengano ripetute eliminando quelle ascoltate una sola volta e fornisce i dati agli operatori melodici partendo dalle melodie armonizzate dalla cadenza più ripetuta.

NB: in questo caso, per chiarezza espositiva, si è sintetizzato con “operatori melodici” tutti gli strumenti di analisi del piano melodico e ritmico, in quanto comunque essi appartengono alle computazioni effettuate nello stesso stadio.

## I parametri del sistema

Nell’interfaccia utente, il sistema prevede diversi parametri. Ognuno di essi permette all’utente di influire in diversi modi nel riconoscimento di una sequenza di note o di una cadenza accordale candidate ad essere elementi di un tema.

### La superficie ritmico-melodica

Innanzitutto vi è la possibilità di decidere quali parti della partitura possano contenere il tema, e quali contribuiscano a creare l’armonia, al fine di escludere quando necessario strumenti quali batteria o percussioni generiche. Si può ulteriormente forzare la mano, senza perdere troppo in generalità, suggerendo la singola parte tra quelle selezionate, candidata a contenere il tema, mentre le altre conterranno solo materiale di riscontro melodico o armonico.

Il modulo del vettore “Errore”, definito parlando degli operatori, verrà moltiplicato per il parametro “Altezza” o “Durata”, progettati nell’interfaccia utente per differenziare le due diverse superfici (quella ritmica e quella melodica). Entrambi sono un controllo che permette di specificare quanto il singolo errore debba “pesare” sul riconoscimento del frammento completo, in ciascuna analisi.

Un parametro relativo ad entrambi i controlli riguarda la “tolleranza”, intesa come il massimo errore percentuale al di sotto del quale due frammenti possono essere considerati equivalenti. Esso viene inserito come un valore bidimensionale, cioè doppio, di cui uno riguarda la superficie ritmica, l’altro quella melodica.

Essendo l’analisi melodica rivolta alle differenze di fraseggio, è sorto il problema di come valutare anche i segni espressivi, gli abbellimenti e altri segni complementari alla nota (intesa come simbolo altezza-durata). Si è risolta la questione parametrizzando l’influenza di questi “attributi” (così vengono identificati) tramite un elenco in cui spuntare quelli determinanti ai fini della differenziazione ed un valore “peso”. Nel caso in cui gli attributi selezionati siano considerati determinanti, una mancata corrispondenza tra i due frammenti esaminati, genererà un valore pari al peso inserito, che si sommerà a quello prodotto dal vettore d’errore.

Per la superficie ritmica viene invece offerta all’utente la possibilità di scegliere quanto la corrispondenza degli accenti nel ritmo sia rilevante ai fini delle differenze tra due pattern ritmici. Tramite selezioni, si può designare ciascuna delle due superfici (ritmica o melodica) come sufficiente per stabilire l’equivalenza di due frammenti, mentre se questo non è espressamente richiesto, è necessario il successo per entrambi i confronti.

Essendo in una melodia più percepibile la nota alta, quando vi siano più note contemporaneamente sulla stessa linea melodica (si noti come si stia ancora parlando di melodia, in quanto questo prescinde dal raggruppamento accordale dell’analisi armonica attraverso tutti i pentagrammi), si assume di default che sia la nota acuta quella che viene scelta e inserita nella linea melodica tra tutte quelle simultanee e di ugual durata. Vi è comunque un’opzione che permette di segnare, se attivata, come questo “privilegio” sia invece da riservarsi alla nota grave.

Dovendo lavorare con dati simbolici, importati spesso da formati diversi spesso più poveri sotto l’aspetto espressivo notazionale, è stata implementata una funzionalità chiamata “analisi a campionamento”. Questa analisi filtra il set di dati della partitura prelevando ad una determinata risoluzione le altezze delle note. Questo vuol dire che una volta scelta una figura musicale come intervallo di campionamento, si scorre la partitura prelevando ad intervalli regolari l’altezza della nota

suonata in quel momento. Se si sceglie il valore “Croma” come frequenza di campionamento, da una qualsiasi partitura si avrà sempre come risultato una sequenza di crome. Per ogni croma la corrispondenza con la partitura è data in base all’istante in cui essa inizia: definendo  $t_j$  l’istante a cui inizia la j-esima croma, essa assumerà l’altezza della nota che nella partitura originale inizia all’istante  $t_j$  o che è cominciata prima ma in quel momento è in esecuzione.

Un parametro “frammento” specifica la minima lunghezza accettabile per una proposizione. Poiché nel caso di analisi a campionamento, non si può parlare di proposizioni, il significato che questo parametro assume è quello di indicare il numero minimo di campioni che devono coincidere (entro i limiti di tolleranza) affinché si possa procedere con il confronto degli elementi successivi.

E’ stato introdotto un gruppo di parametri, chiamati break-makers, allo scopo di operare in modo fine sulla divisione di una frase musicale. Con questi parametri si definisce l’influenza di particolari figurazioni musicali sulla continuità del fraseggio. Infatti è possibile definire il massimo numero di pause consecutive o la durata massima di un silenzio da considerarsi accettabili per un oggetto musicale.

E’ possibile definire se una proposizione può frammentare anche figurazioni particolari che normalmente, per continuità, si suppone siano atomiche, cioè indivisibili nel fraseggio: tuplet (gruppi irregolari) e legature .

Infine è lasciata all’utente la libertà di effettuare la ricerca del materiale tematico su un singolo pentagramma piuttosto che su tutta la partitura, o di restringere l’insieme delle proposizioni utili a quelle sole frasi con caratteristiche psicoperceptive proprie di una cadenza (melodica) relative all’andamento ritmico o melodico.

In un’apposita sezione relativa agli operatori, si possono configurare ed selezionare le elaborazioni della melodia che verranno poi ricercate nella partitura.

## La valutazione

Una sezione di parametri denominati premio, penalità, delta, tema, punteggio piloteranno la misura in cui le differenze generate nel confronto debbano influire sul riconoscimento. Una coppia di parametri premio penalità assicura che una volta trovati due frammenti correlati, uno di essi risulti privilegiato (il primo ad essere presentato, in ordine cronologico) e l’altro invalidato, in ragione dei valori assegnati.

Un valore “tema” indica quante note, numericamente, un frammento debba contenere affinché non sia da valutarsi “troppo corto” per essere un tema.

Il parametro “punti” compie la stessa valutazione esaminando non più il numero di note ma il punteggio raccolto dalla melodia delimitata, alla fine dei confronti.

Il parametro “delta” rappresenta invece una tolleranza. Questo parametro indica un limite in numero di note al di sopra del quale due frammenti vicini nella partitura possano essere considerati “indipendenti”, e al di sotto del quale si possano inglobare in un’unica melodia tematica entrambi i frammenti selezionati, assieme a tutti gli eventi che li dividono.

Infine, i valori di verosimiglianza relativi alle superfici melodica e ritmica verranno combinati in un unico risultato. Un’ultima coppia di valori stabilisce il peso con il quale ognuna delle due superfici contribuisce a valutare il risultato finale.

## La superficie armonica

Armonicamente, i parametri rappresentano la calibrazione che l’analista osserva al primo sguardo della partitura: ogni quanti movimenti ci si aspetta il cambio di accordo, quanto è articolato il movimento armonico, quali sono i ritardi espressivi e gli accordi non significativi.

Per la frequenza con cui cambiano le armonie (o, equivalentemente, per la durata attesa di ogni singola regione armonica) è stabilito un parametro che può anche essere affidato ad una analisi automatica del calcolatore. Un musicista riesce quasi immediatamente a individuare il movimento accordale di un brano, e con i diversi valori predeterminati a disposizione può pilotare intuitivamente la precisione del procedimento tramite il Metro Armonico .

Per il fraseggio armonico, si può scegliere un valore minimo di armonie sotto il quale non scendere per riconoscere una possibile linea armonica. Questo perché ogni genere ed ogni periodo hanno fraseggi diversi, e ci si aspetta che, in funzione di questa diversità, la melodia del tema si snodi sopra diverse tessiture armoniche, più o meno elaborate.

Analogamente, vi è un limite massimo di lunghezza per i temi riconosciuti in questo modo, espresso però nel più significativo e consueto termine di “battute”. Questa ambivalenza ha il fine di indicare la minima informazione in termini “semantici”, cioè di quantità di contenuti, lasciando il limite superiore indicato in termini strutturali alla forma musicale.

Per rendere l'elaborazione più efficiente, è stato inserito nei parametri di questa sezione anche il numero massimo di temi da estrarre, cosicché la computazione possa terminare non appena possibile. Questo è un parametro semplice da utilizzare per un utente non esperto, ed è stato realizzabile grazie alla particolare forma di ricerca tematica Top-Down descritta in seguito nella sezione "algoritmi".

Infine, un parametro "tolleranza" permette di eliminare dall'analisi un numero di accordi probabilmente creati come effetti collaterali del movimento delle singole voci, inglobandoli in accordi più significativi. Un riquadro in questa sezione permette poi la selezione di quali metodi dovranno contribuire cooperativamente alla segmentazione, tra tutti quelli citati (armonici e melodici).

## **Valori Standard**

La segmentazione è automatica nel procedimento, e funziona bene con un set di parametri standard, che tuttavia possono essere adattati alle peculiarità del genere e della forma di musica esaminata, per aumentare l'efficacia del riconoscimento.

# Un esempio: il tema della sinfonia k40 di W.A.Mozart

Per seguire l'analisi automatica, si è preso un brano dal repertorio classico, scelto tra quelli maggiormente conosciuti e dalla trama compositiva elaborata. Si è scelto un brano per cui l'applicazione dei soli operatori melodici incontrasse dei problemi, alla ricerca di un test che costringesse il sistema ad una analisi globale.

Al primo passo vi sono una serie di dati, detti di "stadio intermedio", prodotti dagli operatori "Verticalizzazione" e "Riconoscimento Funzione Tonale". Questa è la "spine" che conterrà la sequenza di accordi, dapprima come spettro di suoni contemporanei, secondariamente come significato associato ad essi nel linguaggio musicale.

Esaminiamo innanzitutto la linea melodica contenente il tema del primo movimento della sinfonia K40 di Mozart, sottolineando il materiale tematico che, fin d'ora, possiamo anticipare verrà riconosciuto:



Figura 15 Il tema della sinfonia K40 di mozart

Si noti che esso è costituito da una trasformazione identica del primo inciso di 3 note, per 3 volte. La struttura tematica però è di ben più ampio respiro: dopo questo inciso ripetuto, la melodia prosegue con un ampio intervallo melodico, per poi scendere con una scala di note ribattute con lo stesso pattern ritmico dell'inciso iniziale. Questo spunto tematico viene poi riproposto elaborato per trasposizione tonale. La seconda parte del tema è data anch'essa da un frammento melodico ripetuto due volte, identicamente fino alla penultima nota. Alla seconda ripetizione, però, la nota finale viene sostituita da una scala discendente alla dominante, per avere una cadenza conclusiva.

A questo punto, tra i due frammenti vi sono 5 note di differenza. Inoltre l'ultima nota del frammento melodico è "distante" anche in una metrica direzionale, in quanto nel primo frammento la melodia scende, mentre nel secondo sale.

A questo punto, sebbene sia possibile, non è semplice calibrare i parametri del sistema in modo da ottenere il riconoscimento di parti di questo tema da cui risalire al tema intero. Inoltre, essendo il tema costituito da un fraseggio

articolato, difficilmente si ripresenterà più di due o tre volte invariato o comunque elaborato soltanto tramite gli operatori visti. Poche note differenti, magari soltanto appoggiature o abbellimenti, vanificano la ricerca per operatori melodici.

Infine, nella scrittura orchestrale, la parte del tema è spesso ripartita tra le voci, e quindi letteralmente inesistente, a livello di melodia. Aggiungendo al confronto la caratteristica di analisi armonica, verrà dapprima creato nella memoria del calcolatore un tabulato procedendo come nell'esempio di figura 17, in cui le note vengono opportunamente frammentate. Alla fine di questa fase si hanno le armonie effettive della partitura, cioè insiemi di suoni contemporanei, con le seguenti caratteristiche

- 1) ogni singola nota ha la stessa durata (e lo stesso punto di inizio nel tempo) di tutte le altre note nel medesimo accordo (proprietà di simultaneità)
- 2) accordi consecutivi differiscono per almeno una nota.

Figura 16

Alla fine di questa fase, sono stati creati accordi anche di durate non significative, e quindi inutili. Inoltre gli accordi sono insiemi di note, con tutte le informazioni utili ancora implicite nell'accordo.

METRO ARMONICO : 1/2 -----			
Armonia 0	Battuta 1	Durata= 1/1	Inizio a 0/1 ANALISI : GRADO 1 -- RIVOLTO 35
Armonia 1	Battuta 2	Durata= 1/2	Inizio a 0/1 ANALISI : GRADO 6 -- RIVOLTO 35
Armonia 2	Battuta 2	Durata= 1/2	Inizio a 1/2 ANALISI : GRADO 1 -- RIVOLTO 35
Armonia 3	Battuta 3	Durata= 1/1	Inizio a 0/1 ANALISI : GRADO 6 -- RIVOLTO 35
Armonia 4	Battuta 4	Durata= 1/1	Inizio a 0/1 ANALISI : GRADO 1 -- RIVOLTO 35
Armonia 5	Battuta 5	Durata= 5/8	Inizio a 0/1 ANALISI : GRADO 2 -- RIVOLTO 46
Armonia 6	Battuta 5	Durata= 3/8	Inizio a 5/8 ANALISI : GRADO 7 -- RIVOLTO 46
Armonia 7	Battuta 6	Durata= 3/4	Inizio a 0/1 ANALISI : GRADO 2 -- RIVOLTO 46
Armonia 8	Battuta 6	Durata= 1/4	Inizio a 3/4 ANALISI : GRADO 3 -- RIVOLTO 35
Armonia 9	Battuta 7	Durata= 7/8	Inizio a 0/1 ANALISI : GRADO 3 -- RIVOLTO 56
Armonia 10	Battuta 7	Durata= 1/8	Inizio a 7/8 ANALISI : GRADO 2 -- RIVOLTO 46
Armonia 11	Battuta 8	Durata= 7/8	Inizio a 0/1 ANALISI : GRADO 3 -- RIVOLTO 35
Armonia 12	Battuta 8	Durata= 1/8	Inizio a 7/8 ANALISI : GRADO 7 -- RIVOLTO 36
Armonia 13	Battuta 9	Durata= 1/1	Inizio a 0/1 ANALISI : GRADO 6 -- RIVOLTO 35
Armonia 14	Battuta 10	Durata= 1/1	Inizio a 0/1 ANALISI : GRADO 5 -- RIVOLTO 34
Armonia 15	Battuta 11	Durata= 1/2	Inizio a 0/1 ANALISI : GRADO 6 -- RIVOLTO 6
Armonia 16	Battuta 11	Durata= 1/2	Inizio a 1/2 ANALISI : GRADO 1 -- RIVOLTO 35
Armonia 17	Battuta 12	Durata= 1/1	Inizio a 0/1 ANALISI : GRADO 5 -- RIVOLTO 34
Armonia 18	Battuta 13	Durata= 1/2	Inizio a 0/1 ANALISI : GRADO 6 -- RIVOLTO 56
Armonia 19	Battuta 13	Durata= 1/2	Inizio a 1/2 ANALISI : GRADO 1 -- RIVOLTO 35
Armonia 20	Battuta 14	Durata= 1/1	Inizio a 0/1 ANALISI : GRADO 4 -- RIVOLTO 35
Armonia 21	Battuta 15	Durata= 1/1	Inizio a 0/1 ANALISI : GRADO 4 -- RIVOLTO 35
Armonia 22	Battuta 16	Durata= 3/4	Inizio a 0/1 ANALISI : GRADO 3 -- RIVOLTO 35
Armonia 23	Battuta 17	Durata= 1/2	Inizio a 0/1 ANALISI : GRADO 3 -- RIVOLTO 1234567
Armonia 24	Battuta 17	Durata= 1/2	Inizio a 1/2 ANALISI : GRADO 2 -- RIVOLTO 56
Armonia 25	Battuta 18	Durata= 1/2	Inizio a 0/1 ANALISI : GRADO 3 -- RIVOLTO 35
Armonia 26	Battuta 18	Durata= 1/2	Inizio a 1/2 ANALISI : GRADO 2 -- RIVOLTO 56
Armonia 27	Battuta 19	Durata= 5/4	Inizio a 0/1 ANALISI : GRADO 1 -- RIVOLTO 36
Armonia 28	Battuta 20	Durata= 1/2	Inizio a 0/1 ANALISI : GRADO 3 -- RIVOLTO 35

Figura 17

La seconda fase di riconoscimento funzione tonale, individuata la tonalità, riconosce dalle note dell'accordo la sua funzione tonale, cioè il grado su cui l'accordo è stato costruito. Nel corso di questa fase, inoltre, in base al parametro "metro

armonico” gli accordi troppo veloci vengono raggruppati, in base a criteri sintattici, in strutture più profonde, cioè meno elaborate. Lo stato della memoria della macchina, alla fine di questo operatore, è riassumibile nel tabulato prodotto dal calcolatore in Figura 17.

A questo punto si ha una sequenza di simboli relativi agli eventi armonici, strutturata in forma di lista. Su di essa verrà eseguita nell’esempio riportato una ricerca di cadenze . Partendo dal primo accordo : I - VI - I , ma anche I - VI - I - VI - I se si prosegue fino alla quarta battuta, mentre arrivando alla ottava, si può ottenere la frase I - VI - I - VI - I - II - VII - II - III - II - III -VII. Si noti come la cadenza può essere autentica e terminare con un primo grado (sarebbe corretto terminasse con una coppia di accordi delle funzioni di Dominante il primo e Tonica il secondo), o sospesa e terminare con un accordo di dominante.

Questa selezione di cadenze non indaga direttamente cercando il tema, ma isolando tra tutte le possibili sequenze di accordi quelle che normalmente sottendono linee melodiche complete. In assenza di meglio si può utilizzare soltanto questa funzione, che sfrutta la sintassi della composizione. Nel nostro caso, tutte le frasi candidate, vengono confrontate tra loro, isolando la trama armonica più ripetuta, che è poi quella che va dalla seconda battuta alla quindicesima, e che contiene effettivamente il tema sopra riportato.

A questo punto l’algoritmo va a prendere la melodia nella traccia suggerita dall’utente e, con un piccolo aggiustamento ancora sintattico, “ritaglia” il tema.

## Gli algoritmi

Prima di esaminare gli algoritmi, si mostra parte della struttura dati utilizzata per convertire la classica organizzazione dei dati in liste correlate, propria dei sequencer, nella spine dedicata alla rappresentazione verticale della partitura.

### Analisi Armonica

L’oggetto principale è l’entità “armonia” che contiene i dettagli necessari per l’analisi della struttura accordale del brano, nonché quelli prodotti nel corso di detta analisi. Viene qui riportata una descrizione leggermente semplificata per chiarezza.

```
class HARMONY: public CObject // list of Events
{
protected :
LIST Chords; // list of chords
private :
    int CurrentAddress;        // pointer to the current harmony
    int MaxBars;              // theme: max length allowed ,measured in bars
    int MinChordPerTheme;     // theme: max length allowed ,measured in chord changes
    int Root;                 // tonic note
    char KeySignature;        // key signature
    int Accuracy;             // index: proportion of chords we need to analyse before decide if the piece is in a major or minor tonality
    int Mode;                 // mark if the piece is in major or minor scale
    int ModulationRoot;      // “

public:
    LIST Themes;              // list of results of analysis
    Fraction DiffDurationAllowed; // maximum value (duration) of a note that can be ignored. Is a tolerance .
    bool AutomaticPreprocessing;

    Fraction MaxChord;        // maximum duration of a chord: if the chord is longer, there is a tonality change
    int Process ;             // flag: which operators use?
    int Accidents ;           //accidents
    int MaxThemes ;           // number of themes
    int Length ( void ) const { return Chords.Length(); }
    Fraction MetroArmonico;   // duration of a chord (average)
    void HARMONY::StringMatching(); // operator RA
```

```

void HARMONY::SortTheme(); // sort themes by length (in bars )
void HARMONY::ResortThemes(); // sort themes by length (in chords)
/*OMITTED: Set of function to manage the object list */
void HarmonicParser (); // operator CA
void HARMONY::harmony_tracking(SCORE *,CProgressCtrl *); //Create the harmonic spine
bool HARMONY::Collapse(SPINE *,SPINE *); //simplify the spine
void HARMONY::CollapseChords(SPINE *, SPINE *);//simplify the spine
void HARMONY:: ComputeHarmonicSpine(); //simplify the spine, deleting all ornamental chords and dissonances.
Fraction HARMONY::TonalitaEMetricaAutomaticaTM(); //calculate the key signature and the average length of a chord

HARMONY( void )
{
m_startime = m_endtime = DiffDurationAllowed =CurrentAddress = 0;
Accidents=NO_SIGNATURE;
MaxChord =Fraction(4,4);
MaxThemes=3;
Process=H_ANALYSIS;
MaxBars=16;
MinChordPerTheme=SIMPLE; /*at least 3 chords indicate a cadence, es.I-V-I*/
}

~HARMONY ( void ){ Clear(); }

void HARMONY::SetDiffDurationAllowed(int ); // set the duration value of non - significative note

friend class Fraction;
friend class LIST;
friend class NOTE;
friend class REGIONETONALE;
friend class SPINE;
friend class BarElement;
friend class TIMETRACK;
};

HARMONY

```

Harmony;

```

class SPINE: public COBJECT // evento armonico: ho una modifica nel
                        // tessuto armonico
{
private:
    int bar;           // bar number
    int Root;         // tonic note of the tonality or modulation on which chord play
friend class LIST;
public:
    LIST NotesInChord;

    int ChordMask;    // bitwise mask : store the configuration of chord
    int ModalChordMask; // bitwise mask : store the type of each interval in the chord

    int FundamentalOfChord; // fundamental note of the chord
    int Inversion;         // inversion
    BassPitch;           // the lowest note in chord
    bool CanEnd;         // flag to mark a chord that can be a cadence start point
    bool CanStart;       // flag to mark a chord that can be a cadence end point

    friend class Fraction;

    SPINE( void ){ scan_armptr = scan_armptr =Root= 0;FundamentalOfChord = INT_MAX;}
    ~SPINE( void ){ Clear(); }
    void CreateChordComposition(int tipo,int grado); // encode the chord in harmonic notation
    int ReadThirdGradeOfChord (); // evaluate the mode of chord
    int ReadFifthGradeOfChord (); // evaluate the type of chord
    int ReadSeventhGradeOfChord (); // evaluate the type of seventh in the chord

    int Length(); // return the number of note in the chord
    int Bar() // return the position (in bar) of chord in the piece

    bool:: IsCadence(); //evaluate if a cadence can end with this chord
    void InsertChordDuration(Fraction); //force the duration of a chord to a value
    void InsertChordStart(Fraction); // force the start time of a chord to a value
    void InsertNoteInChord(NOTE*); //insert a note in a chord
    void ReadChordDurationAndStart (Fraction *,Fraction *); // read duration and start time of a chord
    bool IsQuasiEqualInDuration(NOTE*,NOTE*); // compare values of 2 notes using a tolerance value
    NOTE* GetNoteFromChord( int ); // get the nth note from a chord
    void DisposeChord(int , int ); // calculate which grades are filled in a chord
    SameFunctionDifferentChord(); // Macro that compare two chord and return true if they are of the same family (tonal function)

    void SPINE::FindBass(void) ; // find the lowest pitch
    int SPINE:: CountHalvesTones (int); // calculate distance in semitones of between a note and the bass
    int SPINE::Trace(NOTE* ); // write in the mask the configuration of chord
    void SPINE::Normalize(NOTE* ); // transform the note in relative position in current scale
        void SPINE::Normalize(NOTE* , int ); // transform the note in relative position in a chord
        void SPINE::SortNoteInChord (void); // recognize from a set of notes (chord) the name and type of chord
        void SPINE:: Chord_Recognizer (void); // main function: find the meaning of chord
        int SPINE::ExpectedModeInTonality(int); // function of a chord (built onto the nth grade of tonality) : return the
        mode expected for the chord
    bool SPINE::StronglyAtonalChord(void); // verify if a modulation occur
    bool SPINE:: StronglyTonalChord onale(void); // verify if chord belong at tonality
    bool SPINE::FundamentalAltered(int ); // store if the grade has an accident
};

```

Vediamo innanzitutto come funziona l'algoritmo di analisi armonica

Possiamo pensare che per ogni accordo venga richiamata la procedura sottostante.

```
Chord_Recognizer(void)
{
    Find Bass();
    Sort Note In Chord();
}
```

Essa ,nell'ordine, riconosce il basso armonico, e avendo questo come riferimento, ristrutturata tutto l'accordo.

La prima chiamata alla funzione Find Bass produce la seguente elaborazione.

```
FindBass()
{
    if (RecognizedChordBass)
    {
        for(count=0; count < #notes in chord ; count++)
        {
            Get Next Note From Chord
            if ( (Note.pitch < BassPitch)&&(IsNotAnHarmonicPedal (Note.pitch)))
            {
                BassPitch= Note->pitch ;
                index     = count      ;
            }
        }
        Note = GetNoteFromChord(index);
        Normalize(Note) ;
        Bass = (Note.grade %7) ;
        RecognizedChordBass = True ;
    }
}
```

Vediamo come all'inizio della procedura si valuta un flag che porta l'informazione riguardo all'avvenuta analisi dell'accordo.

Nell'elaborazione iniziale dei dati alcuni accordi, quelli iniziali, vengono preanalizzati, per individuare la tonalità'.

Ogni accordo, una volta riconosciuto, viene "segnato" in modo da non richiedere ulteriori calcoli, avendo tutte le informazioni al suo interno.

Si nota la funzione Normalize, riportata sotto.

```
void SPINE::Normalize(NOTE* Note, int norm)
{
    if (Note ->grado < norm) Note ->grado += 7 ;
    Note ->grado -= (norm) ;
    Note ->grado %= 7 ;
    SetFlagNormalizationDone ;
}
```

```
void SPINE::Normalize(NOTE* Note)
{
    Normalize(note, Keynote()); // Tonic calibration
}
```

In questo caso la funzione overloaded normalizza la nota rispetto alla tonalità'.

Infatti si sta ricavando il grado specifico della tonalità', e l'interesse si e' già' spostato dal pitch della nota alla sua valenza tonale.

Nella prossima pagina vediamo invece l'algoritmo alla base della determinazione dei gradi dell'accordo.

```

void SPINE:: Sort Note In Chord()
{
  ResetMask;      // reset the mask of intervals of chord . We call it "ChordMask"
  ResetModeMask; // reset the mask of mode interval (minor or major) of chord . We call it "ModalChordMask"

  for(index =0; index < gradi.Length();index ++) //CYCLE: Cost P (where P = total number of staff)
  {
    Element      = GetNoteFromChord (index);
    Normalize (Element); // first normalize respect tonality, then by bass
    Normalize (Element, Grade);
    Cicle_FindAndDeleteNoteRepeated(note, From 0 to index) WORST CASE (P-1)*(P-2)
    HalvesTones  = CountHalvesTones (Element -> pitch);
    Element -> pitch = HalvesTones;
    interval     = Trace(Element);
    DisposeChord (Element ->pitch , interval ) ;
  }
  Inversion=0;
  switch(ChordMask) // ChordMask is a bitwise mask, now the process looks for which flag are marked.
  {
    case THIRD + FIFTH:          Inversion = 35;Corrector= 0;break; // TRIAD (NO INVERSION)
    case THIRD + FIFTH + SEVENTH: Inversion = 7; Corrector = 0; break; // SEVENTH (NO INVERSION)
    case THIRD + SIXTH:          Inversion = 6; Corrector = 5; break; // 1st INVERSION TRIAD
    case FOURTH+ SIXTH:          Inversion = 46; Corrector = 3; break; // 2nd INVERSION TRIAD
    case THIRD + FIFTH + SIXTH:  Inversion = 56; Corrector = 5; break; // 1st INVERSION SEVENTH
    case FIFTH + SIXTH:          Inversion = 56; Corrector = 5; break; // 1st INVERSION SEVENTH
    case THIRD + FOURTH+ SIXTH:  Inversion = 56; Corrector = 5; break; // 1st INVERSION SEVENTH
    case THIRD + FOURTH:         Inversion = 34; Corrector = 3; break; // 2nd INVERSION SEVENTH
    case SECOND+ FOURTH+ SIXTH:  Inversion = 24; Corrector = 1; break; // 3rd INVERSION SEVENTH
    case SECOND+ FOURTH:         Inversion = 24; Corrector = 1; break; // 3rd INVERSION SEVENTH

    default : if (ChordMask&(THIRD+ FIFTH)) { Inversion = 35; Corrector = 0; } //TRIAD
              else if(ChordMask &( THIRD + SIXTH)) { Inversion = 36; Corrector = 5; } //TRIAD 1st INVERSION
              else if (ChordMask &( FOURTH + SIXTH)){ Inversion = 46; Corrector = 3; }
              else { Inversion = 1234567; Corrector = 0; }

    break;
  }
  Grade += Corrector; // essendoci poi una sottrazione, meglio avere un margine
  if (Grade >6) Grade %=7 ;
}

```

Si può notare come un ciclo iniziale esamini le note dell'insieme "accordo", filtrandole dei doppi (anche se non identiche ma a distanza di ottava).

Sull'indicazione dei costi computazionali, si vedrà alla fine di questa sezione, uno studio sulla quantità dei dati in gioco.

Una volta marcati i gradi tonali presenti nell'accordo nella maschera orientata al bit (bitwise), si confronta la configurazione presente con alcune configurazioni note.

La struttura "Switch" cerca il riconoscimento di disposizioni canoniche, partendo dalle più significative e stabili, via via verso i casi particolari.

Gli "if" annidati alla fine della procedura permettono di classificare i casi non ancora identificati generalizzando le caratteristiche delle tipologie principali di triade. Qui sotto vediamo come la funzione Trace modifichi il vettore di bit facendo un'OR con la maschera relativa ad un intervallo

```

int SPINE:: Trace (NOTE Grade)
{
  switch( Grade )
  {
    case 2: Interval = THIRD ; break;
    case 4: Interval = FIFTH ; break;
    case 6: Interval = SEVENTH; break;
    case 1: Interval = SECOND ; break;
    case 3: Interval = FOURTH ; break;
    case 5: Interval = SIXTH ; break;

    default : Interval = 0 ;
  }
  ChordMask += Interval;
  return(Interval); //ChordMask is a bitwise mask.
// NOW flag of interval are marked.
}

```

Si può notare come nel precedente algoritmo sia stata utilizzata la tonica per ricondurre tutto nella tonalità di impianto del brano. Sebbene la tonalità possa essere specificata nello standard NIFF, non e' detto di essere sempre nelle condizioni di averla come dato in ingresso.

Una semplice procedura prova a determinare la tonalità quando questa non e' specificata. Contemporaneamente viene eseguito un controllo qualitativo sulla frequenza con la quale le armonie si susseguono.

Essa si basa su un test su n accordi consecutivi (il valore Accuracy definito in fase di sperimentazione nella lista delle armonie) ed in base agli accordi riconosciuti e al loro modo , associa la tonalita piu' plausibile per la sequenza dei primi n accordi

Fraction HARMONY::TonalitaEMetricaAutomaticaTM()

/\*Tonalita' e Metrica sono 2 funzioni separate. Vengono inglobate in 1 solo ciclo di scansione della lista per motivi di efficienza\*/

```

bool IsTheBegin =true , ConditionNoSkip;
int i,j,tt;
int modal, Root;
NOTE notetmp;
Fraction Duration,Compare, Tempo;
SPINE ChordStart;
char FirstAccident = ' ';
char TonAlt=' ';

Harmony.GoToBegin ();
do
{   ChordStart =spine.CurrentChord();
    for (i=0;i< DaAnalizzare->Length();i++)
    {
        notetmp = GetNoteFromChord(i);
        InsertNoteInTonalContext(notetmp, Root)
    }
    if (DiagnosticTonalEvent() ) //returns true when all required notes of tonality are found or a modulation is checked
    {
        tt= GiveTonality(&modal, Root,Accident);
        // investigate on the scale mask of tones and halftones; returns mode and tonic grade of scale, the last accident (sorted
        by comparison)
        if (IsTheBegin) // this is the first tonal centre found
        {
            if (Root!= tt%7)
            {
                IsTheBegin =false;
                Root =tt%7;
                Harmony.Mode= modal;
                TonAlt= Accident;
            }
        }
        } else SetModulation();
    }
    InsertDurationOfChordInASortedList(ChordStart);
    ThereIsAnotherChord = NextChord();
} while (ThereIsAnotherChord==true);

i= (LengthOfSortedListDuration/2)
if (LengthOfSortedListDuration<5 i -= 1;

Compare = MoreFrequentChordDuration();
i--;
Tempo = FindTempo //Tempo is "tempo" of piece
do {
    i++;
    Duration = GetFromPositionInSortedDurationList(j);

```

```

    ConditionNoSkip = i < LengthOfSortedListDuration-1;
} while(((IsIntegerRatio(Duration/Compare))||IsIntegerRatio(Tempo/Duration))&&( ConditionNoSkip));
Harmony.GoToBegin (); //chordal controls on scale found
do
{
    ThereIsAnotherChord = spine.NextChord();
    ChordStart =spine.CurrentChord();
} while ((ChordStart->Length()<3)&&( ThereIsAnotherChord)); // The program look for the first chord, jumping
incomplete chords

if (ThereIsAnotherChord!= false) //prevent from singular case on which the spline is a melodic line, without chords
{
    ChordStart.ChordRecognizer();

    if (FundamentalOfChord is not equal to root and is not the sixth grade (relative minor scale))
        {
            if (FundamentalOfChord is not dominant of the root ( or chord is not
            "major") and
            is not dominant of the relative minor (or chord is not "major")
            { // looking for near tonalities
            if ((Root is fourth grade respect FundamentalOfChord and is a major mode)
            Root += 4;
            else if (Root is fourth grade respect a minor third up the FundamentalOfChord and is a minor mode)
            Root += 6
            else Root = ChordStart. FundamentalOfChord; //no information: keep the Fundamental as Root
            Root%=7;
            }
            }
            else
            {
            if (Root==(( ChordStart. FundamentalOfChord +5)%7)&&( ChordStart. ReadThirdGradeOfChord ()!=MAJOR))
            {
            Root=((Root+5)%7);
            Harmony.Mode= MINORE;
            }
            }
            if (NoKeySignatureInScoreNotation)
            {
            int quoz=0; int ton=0;
            if (Modo==MINORE) ton=5;
            if (Tonality is F+ or D- or B-, or TonAlt is "flat")
            while (ton!= Root) {quoz--; ton=(ton+3)%7;} // back in the 5th circle
            else
            while (ton!=Root) {quoz++; ton=(ton+4)%7;} // forward in the 5th circle
            KeySignature = quoz ;
            }
            SetTonality(Root,TonAlt);

            return(Duration);
        }
}

```

## Algoritmi di delimitazione dei temi

Mostriamo qui come si opera sulla lista di accordi (SPINE) per delimitare aree sulla partitura in cui la probabilita' di trovare un tema o un oggetto musicale sia maggiore. Nel corso di questa fase vengono posti dei marker di "possibile inizio oggetto" e "possibile fine oggetto", con i quali verranno poi valutati gli oggetti che si vengono a delimitare tra essi. Gli oggetti piu' complessi e articolati saranno inseriti in un'altra lista chiamata Themes, che fa riferimento una volta ancora alla lista SPINE

```
HARMONY::HarmonicParser()
```

```
{
    int buffer, Counter;
    bool ThereIsAnotherChord = true, StartThemePoint=false;
    SPINETRACK *ChordStart, *ChordEnd, *temp;
    HARMONY TempList;

    Harmony.GoToBegin ();
    ChordStart =spine.CurrentChord();
    ChordStart ->CanStart=true;
    ThereIsAnotherChord =Harmony.NextChord();
    While (!EndOfTrack) //CYCLE PERFORMING SEGMENTATION : LINEAR COST
        { ChordStart=Harmony.CurrentChord();
          ChordStart ->CanStart =false;
          if ((ChordStart->Grade==0) || (PrecedentChordCanBeEndOfCadence))
              {
                  ChordStart -> CanStart = true;
                  PrecedentChordCanBeEndOfCadence =false;
              }
          ChordStart -> CanEnd = PrecedentChordCanBeEndOfCadence = ChordStart ->IsCadence();
          GetNextChord();
        }

    Harmony.GoToBegin();
    while (!EndOfTrack) //cycle: bounding the themes
        { // LINEAR COST combined with K user defined costant in both FOR internal Cycle
          if (ChordStart -> CanStart)
              {
                  SaveActualPositionInList (&buffer);
                  for (Harmony.GetNextChord; ((!EndOfTrack) && (GoOn (buffer))));GetNextChord()
                      {
                          pointer=ActualPositionInList;
                          if ((ChordEnd ->CanEnd)&&((pointer-buffer)> MinChordPerTheme))
                              TempList.AddTheme(buffer,CurrentAddress);
                      }
              }
          if (TempList.Themes.Length()>0) //We keep only the longest cadence
              {
                  Counter=(only_using_cadence_detection ? 1 : TempList.Themes.Length());
                  for (;Counter>0;Counter--)
                      {
                          ScrollTheme=(Results*)TempList.Themes[TempList.Themes.Length()-1];
                          Harmony.Themes.Ins(Themes.Length(),ScrollTheme);
                          TempList.Themes.Del(TempList.Themes.Length()-1);
                      }
              }
          while (TempList.Themes.Length()!=0) TempList.Themes.Del(0);
          CurrentAddress=buffer;
        }
    ThereIsAnotherChord =Harmony.NextChord();
    ChordStart=Harmony.CurrentChord(); //legge l'armonia corrente
```

```
}  
Deleteoverlapping();  
}
```

**COST = O(H) where H= Number of Chords in score**

void HARMONYTRACK::StringMatching()

l'algoritmo e' sdoppiato per essere adattato a due utilizzi simili ma in contesti diversi:

se questo metodo e' usato come sistema unico per determinare il tema, la ricerca e' eseguita direttamente usando solo la traccia SPINE , con un algoritmo di ricerca di sottostringhe.

Se invece un metodo impiegato in precedenza fornisce una lista di temi candidati, è necessario prendere ogni possibile tema e confrontarlo con tutti gli altri

Quindi se un lista dei "temi candidati" è già presente

```
Cycle 1:
{  GetThemeFromList (jT)
  Cycle 2:
  {  GetThemeFromList (iT)
    RecurrenceFound =false;
    if Conditions
    {  Cycle 3:
      {  Cycle 4:
        {  Cycle 5:
          {  score=0;
            Cycle 6:
            {
              if (symbols are different)
                if (SameFunctionDifferentChord ()) score += MIN_PENALTY;
                else score += MAX_PENALTY;
            }
            if (cycle has not been aborted) //score of errors is below the threshold
            {  ThemesList.Del (iT);
              ThemesList->AddItWeight (jT);
              RecurrenceFound =true;
            }
          }
        }
      }
    }
  }
}
```

**Ciclo 1 :** la variabile 'jT' e' il contatore che scorre la lista dei temi, dall'inizio alla fine

**Ciclo 2 :** la variabile 'iT' e' come jT, e indica il tema della lista da comaprare con il frammento puntato da jT. Essa scorre tutti i frammenti fino alla fine della lista a partire da quello successivo alla posizione di jT

**Ciclo 3 :** vengono assegnate alla variabile 'j' I valori delle varie lunghezze per le stringhe di simboli da confrontare, per rendere possibile produrre risultati di lunghezze differenti, cercando dapprima quelli piu' lunghi ('MaxChord' simboli) fino al minimo consentito (MinChordPerTheme).

Quando un tema viene verificato con ragionevole certezza, il ciclo termina forzatamente

**Condizioni :** da qui, I cicli sono subordinati ad alcune condizioni: anche se il numero delle differenze riscontrate in un confronto e' accettabile, esse non devono superare meta' del numero di accordi che compongono la frase

Inoltre, le frasi non devono sovrapporsi, nemmeno parzialmente .

Se una sequenza non verifica questo requisito, essa è immediatamente cancellata.

**Ciclo 4 :** una variabile 'm' rappresenta lo scostamento 'offset', per verificare che il secondo tema del confronto non sia contenuto nel primo, a partire ad esempio dalla terza o quarta armonia

Esso varia da 0 a j simboli prima dell'ultimo accordo.

Questo ciclo scorre tutte le sottosequenze di j simboli nel secondo tema partendo da m simboli dopo l'inizio del primo tema. E' valida anche in questo ciclo la stessa condizione d'uscita del ciclo esterno .

**Ciclo 5 :** la ricerca e' eseguita sulla SPINE : le strutture nella lista dei temi segnano il punto di inizio e di fine dell'oggetto selezionato sulla SPINE

Il contatore 'i' scorre il piu' lungo tra I due temi confrontati (la lista dei temi e' ordinata), scandendo patterns di j simboli ad ogni iterazione. Viene valutata ancora una volta la condizione di abbandono dei cicli.

**Ciclo 6 :** Confronto di accordi di due sequenze partendo dal punto indicato nella struttura "tema" combinato con gli "offset" i e m per la prima e la seconda sequenza, vengono confrontati j simboli.

Se (score > Threshold) abbandona il ciclo ed elimina il pattern considerato.

altrimenti, semplifichiamo l'algoritmo in un metodo brute-force

```
ConditionExit= False;
Cycle 1
{ Cycle 2
  {
    Cycle 3
    { score=0;
      Cycle 4
      {
        if (symbols are different)
          if (SameFunctionDifferentChord ()) score += MIN_PENALTY;
          else score += MAX_PENALTY;
        }

        if (cycle has not been aborted)
          if (ThemeAlreadyFound) AddItWeight());
          else { KeepTheme(i,j)      ThemeAlreadyFound =true; }
        }
        ThemeAlreadyFound =false;
        ConditionExit=(Themes.Length() > MaxThemes);
      }
    }
  }
SortThemeByHarmonicalLength();
PurgeRedundancy ();
```

in entrambi i casi , l'ultima istruzione e':

```
SortThemeByRecurrenceAndHarmonicalLength ();
```

**Ciclo 1** : assegna alla variabile 'j' la lunghezza della sequenza da confrontare, per permettere di trovare risultati di lunghezze differenti, da MaxChord fino a MinChordPerTheme.

Quando un tema viene verificato con ragionevole certezza, il ciclo termina forzatamente

*Questo ciclo ha un costo con un fattore  $K$ , dove  $K$  e' un intervallo di valori tra la sequenza piu' lunga accettabile e quella piu' corta trovate dalla computazione precedente.*

**Ciclo 2** : la variabile 'i' punta ai possibili temi, nella lista delle armonie(SPINE).

Vengono fatti passare i valori dall'inizio della lista, fino alla j-esima posizione a partire dal fondo.

Anche questo ciclo termina forzatamente alla condizione di "possibile tema trovato"

*Questo ciclo contribuisce con un fattore di costo  $M$  dove  $M$  è il numero di temi trovati dalla computazione precedente*

**Ciclo 3** : la variabile 'm' riferisce ai frammenti che devono essere confrontati con quello puntato da "i". Essa indica la loro posizione nella lista delle armonie(SPINE).  $M$  parte dal fondo della lista, avanzando verso il minimo valore di "i".

*Questo ciclo conferisce al costo computazionale un fattore  $(M-1)$  dove  $M$  è il numero di temi trovati nella computazione precedente*

**Ciclo 4** : Confronta per j accordi due sequenze di armonie che partono da i e m.

Se (score > Threshold) abbandona il ciclo e scarica la sequenza esaminata

*Questo ciclo ha il costo di un algoritmo di confronto tra stringhe. Considerando j come lunghezza dei pattern confrontati,  $j < \text{MaxChord}$ . MaxChord e' un limite superiore per la lunghezza, nel calcolo del costo di questo ciclo*

In entrambi i metodi vengono impiegati i parametri riportati qui sotto:

len	=	lunghezza della traccia SPINE (numero totale di accordi presenti)
WeightOfErrorAllowed	=	soglia oltre quale l'errore determina il rifiuto del tema;
MaxBarsTheme	=	Massima lunghezza consentita per i temi (in battute);
MinChordPerTheme	=	numero minimo di armonie di cui deve essere costituito un tema

Si calcola poi il massimo numero di accordi in un tema leggendo la SPINE e scegliendo il massimo numero di accordi contenuti nel valore MaxBarsTheme.

Andiamo adesso ad esaminare il risultato di una statistica sulla mole di informazioni contenuta nei files NIFF esaminati in una sessione di alpha test. La prima osservazione riguarda come, nel corso dell'elaborazione stessa, si producano dati più significativi.

Questi sono il risultato dell'aggregazione dell'informazione a diversi livelli, ma anche della selezione a parità di livello delle informazioni.

Contemporaneamente all'affidabilità dei risultati elaborati da questi valori, si ha una riduzione notevole della quantità delle informazioni da valutare.

Questa diminuzione si riflette immediatamente sulla dimensione di memoria impiegata e sui tempi di computazione.

Inoltre, essendo i cicli spesso basati sul controllo degli accordi, è stata eseguita una valutazione statistica sulla profondità media dell'accordo, con dati reali.

Si intende con profondità il numero di note che formano l'accordo.

Spesso i controlli su ogni accordo hanno un fattore di costo per l'intero algoritmo di  $C \times (C-1)$ , che in una partitura a 10 pentagrammi (10 note suonate contemporaneamente) è 90, ma alla luce dei dati valutati, sappiamo andare da 2 a 4.

Gli accordi possono essere composti al caso limite da tutti i 12 semitoni della scala cromatica, anche se un accordo di sei note diverse è già un caso rarissimo.

Eseguita la semplificazione delle note ridondanti a meno di ottave (classe 12), l'accordo tende a conservare comunque le poche note consonanti.

Inoltre dalle tavole si vede come le semplici regole sintattiche che isolano gli ornamenti siano in grado di ridurre ancora le effettive armonie, sempre più sensibilmente al crescere della mole di dati (le didascalie della tavola 1 approfondisce tutti questi aspetti).

**Tavola 1 Occupazione di memoria nel sistema armonico**

Nome brano	numero note	numero armonie	profondita' media	Prob. accordi con n note					
				1	2	3	4	5	6 o piu'
<b>Bolero</b>	20465	5391		0.0178	0.0692	0.2040	0.1606	0.1185	0.4300
A semplificazione avvenuta		1133	3.611650	0.0000	0.1458	0.3445	0.3145	0.1502	0.0459
<b>Qualcosa di Grande</b>	3826	1297		0.0099	0.0936	0.3054	0.4434	0.1330	0.0197
A semplificazione avvenuta		203	3.651961	0.0093	0.0501	0.0786	0.2907	0.4063	0.1658
<b>Sinfonia k.440 di Mozart</b>	12063	2735		0.0230	0.0282	0.1192	0.0969	0.1784	0.5547
A semplificazione avvenuta		671	3.608631	0.0313	0.1133	0.3100	0.3607	0.1371	0.0492
<b>Fuga in Re minore, Bach</b>	2207	1619		0.3663	0.6343	0.0000	0.0000	0.0000	0.0000
A semplificazione avvenuta		354	3.016901	0.0226	0.1836	0.5509	0.2458	0.0000	0.0000
<b>Corale di Bach</b>	259	82		0.0000	0.0000	0.1220	0.8902	0.0000	0.0000
A semplificazione avvenuta		50	3.235294	0.0000	0.0000	0.7800	0.2400	0.0000	0.0000
Lirica: canto e pianoforte. - - - - -									
<b>Valzer: Libiamo dai lieti calici</b>	1120	839		0.2658	0.5626	0.1728	0.0000	0.0000	0.0000
A semplificazione avvenuta		256	2.315175	0.0703	0.6055	0.2695	0.0586	0.0000	0.0000
<b>Aria: Bella Figlia dell'Amor</b>	1696	438		0.0023	0.0137	0.2945	0.1324	0.2466	0.3128
A semplificazione avvenuta		155	3.666667	0.0129	0.1419	0.3419	0.2581	0.1742	0.0774
<b>Madama Butterfly</b>	1622	1390		0.4727	0.5281	0.0000	0.0000	0.0000	0.0000
A semplificazione avvenuta		223	2.133929	0.1480	0.5785	0.2735	0.0045	0.0000	0.0000
<b>Aria: La ci darem la mano</b>	1776	1273		0.3158	0.4493	0.2357	0.0000	0.0000	0.0000
A semplificazione avvenuta		285	2.346154	0.0983	0.5193	0.3298	0.0526	0.0035	0.0000
<b>Nessun Dorma</b>	1366	1222		0.4632	0.5377	0.0000	0.0000	0.0000	0.0000
A semplificazione avvenuta		205	2.169903	0.1171	0.6098	0.2683	0.0098	0.0000	0.0000
<b>Ritorna Vincitor</b>	4190	3421		0.2727	0.7273	0.0000	0.0000	0.0000	0.0000
A semplificazione avvenuta		475	2.426471	0.0821	0.4358	0.4590	0.0253	0.0000	0.0000

## Commenti

Si noti la riduzione nel passare dal numero di note, al numero di accordi della spine, al numero di armonie reali (almeno quelle estrapolate dal calcolo automatico)

Nel corale di bach si nota immediatamente il riconoscimento di una struttura accordale, dalla distribuzione del numero di note per accordo

Soprattutto si nota come la fase di semplificazione interpreti le armonie di triade e ridimensioni notevolmente la complessita' armonica, riconducendola a quella reale

Anche la profondita media tendente a 3 indica un andamento accordale. In effetti piu' si va verso la polifonia piu' la media si alza, nella melodia accompagnata resta prossima a 3.5-3.6, valore perfettamente plausibile (triade dell'accordo+ nota della melodia, che una volta su due e' anche nota consonante nella triade)

Inoltre anche la riduzione da numero di note a numero di armonie significative si impenna quando la struttura armonica e' libera, scendendo fino ad un rapporto di 1/4 nel caso del corale di Bach e a valori bassi nella musica barocca, dove la struttura accordale e' rigidamente governata da regole, e caratterizza l'estetica e le regole compositive del brano. Ancora, il rigore armonico corrisponde ad una percentuale nulla di armonie "complesse", perfettamente in linea con l'analisi classica

Nei brani orchestrali come la sinfonia k40 di Mozart o il Bolero di Ravel e' significativa la caduta del numero percentuale di armonie a 6 o piu' note.

Questo indica l'epurazione quasi totale della ridondanza nel linguaggio musicale

In tutti i brani lirici e' estremamente interessante notare come, eliminando o collassando tra loro armonie non significative, quali possono essere le sonorita' istantanee create dalle note di passaggio dell'accompagnamento e/o dagli abbellimenti della voce, si porti piu' peso verso l'informazione completa,cioe' le armonie con un numero di note tra le 3 e le 4. Viene cioe' interpretata in automatico l'informazione ad un livello piu' articolato. Si segnala che la partitura scelta era una melodia di canto accompagnata al pianoforte. Questo e' anche il motivo del piccolo valore di profondita' media delle armonie

Caratteristica comune a tutte le partiture di canto solo con accompagnamento, e' che la vera riduzione in quantita' dei dati d'ingresso viene fatta dopo la creazione della spine. Questo fatto trova spiegazione nel fatto che il pianoforte condensa tutta l'informazione armonica dell'orchestra, quindi il raccoglimento per armonie non riduce quanto la loro interpretazione semantica

**Tavola 2: Tempi di elaborazione nel sistema armonico**

Nome brano	numero Note	staves	Tempo 1	Tempo 2	Tempo 1&2	Tempo 1&3	Tempo 2&3	Tempo 1&2&3
<b>Bolero</b>	20465	15	0' 19" 80	0' 21" 88	2' 18" 75	1' 20" 90	0' 17" 20	2' 6" 54
<b>Qualcosa di Grande</b>	3826	8	0' 1" 90	0' 1" 80	0' 2" 71	0' 3" 05	0' 1" 65	0' 4" 51
<b>Sinfonia k.440 di Mozart</b>	12063	11	0' 6" 78	0' 7" 83	0' 9" 17	0' 5" 40	0' 6" 30	0' 10" 09
<b>Fuga in Re minore, Bach</b>	2207	2	0' 1" 27	0' 1" 11	0' 4" 20	0' 1" 30	0' 1" 04	0' 4" 23
<b>Corale di Bach</b>	259	4	0' 0" 12	0' 0" 08	0' 0" 10	0' 0" 10	0' 0" 07	0' 0" 10
Lirica: canto e pianoforte. - - - - -								
<b>Valzer: Libiamo dai lieti calici</b> 1" 18	1120	4	0' 0" 84	0' 0" 72	0' 1" 16	0' 0" 77	0' 0" 74	0'
<b>Aria: Bella Figlia dell'Amor</b> 0" 75	1696	6	0' 0" 59	0' 0" 43	0' 0" 76	0' 0" 50	0' 0" 40	0'
<b>Madama Butterfly</b> 0" 90	1622	2	0' 0" 90	0' 0" 81	0' 0" 90	0' 1" 45	0' 0" 79	0'
<b>Aria: La ci darem la mano</b> 1" 59	1776	3	0' 0" 98	0' 0" 89	0' 1" 58	0' 1" 04	0' 0" 97	0'
<b>Nessun Dorma</b> 1" 10	1366	2	0' 0" 79	0' 0" 70	0' 1" 09	0' 0" 84	0' 0" 67	0'
<b>Ritorna Vincitor</b> 7" 88	4190	2	0' 4" 15	0' 3" 33	0' 7" 35	0' 3" 56	0' 3" 29	0'

**LEGENDA**

- 1) Uso degli Operatori Armonici di ricerca cadenza
- 2) Uso dello string matching (senza approssimazione)
- 3) Uso dell'opzione di verifica dei risultati tramite gli operatori melodici A&B combinazione di A e B (riesame dell'output di A con il metodo B)

# Riferimenti bibliografici

E.Cambouropoulos Musical Parallelism and Melodic Segmentation,  
Proceedings of the III Journées d'Informatique Musicale, Caen, 1998

G.Haus, Elementi di informatica musicale, Jackson, 1984

S.Guagnini, Metodi e strumenti per la segmentazione automatica di partiture musicali rappresentate in NIFF,  
tesi di laurea in Scienze dell'informazione, Università degli studi, Milano, 1998

Tim Hoffman, William P. Birmingham: *A constraint satisfaction approach to tonal harmonic analysis*,  
*Electrical engineering and computer science department*, the University of Michigan CSE-TR-397—99

F.Lerdhal, R.Jackendoff, A Generative Theory of Tonal Music,  
The MIT Press, 1983

F.Lonati, Metodi, algoritmi e loro implementazione per la segmentazione automatica di partiture musicali,  
tesi di laurea in Scienze dell'informazione, Università degli studi, Milano, 1990

M.Pancini, Segmentazione di informazione musicale testuale e fonica per la generazione automatica di indici in data base  
multimediali a oggetti,

tesi di laurea in Scienze dell'informazione, Università degli studi, Milano, 2000

## **Aiuti per la comprensione musicologica**

Renato Dionisi :*Lezioni di Armonia Complementare*, Zanichelli 1992

C.Kuhn, Il linguaggio delle forme nella musica occidentale,  
ISP/IEC DIS 10743, unicopli, 1992

[27] Walter Piston,: *Armonia*, Edt 1992