

## 言語クラス EMG を用いた不完全なシーケンスからの構造推定手法

池田 剛, 乾 伸雄, 小谷 善行  
(東京農工大学・工学部・情報コミュニケーション工学科)  
{ikedata, nobu, kotani}@fairy.ei.tuat.ac.jp

### 概要:

我々はこれまでに、メロディーなどの楽曲シーケンスのなす形態的な構造を、文法ベースで表現するために、言語クラス EMG を提案し、その処理系として、与えたシーケンスに対してその形態的構造を EMG を用いて解析し出力する処理システム "MAGIEnvironment" を実装した。このシステムは、多様なシーケンスを扱うために、文法記述として膨大な規則記述を用意しなくても、適した規則を自動的に生成することが可能である。これを適用規則処理と呼んでいる。また、構造が、どのくらい拘束要素を持っているのかという点で評価することも行った。これにより、譜面情報のような正規化されたシーケンスである完全なシーケンスにおいてはその構造を取得することが可能となった。

シーケンスを扱う際、音値は繰返しの判定などに厳密な値を要求される。しかし、記譜上以外での入力方法ではこの音価要素は完全には表現されない。これは繰返しの判定などに多大な悪影響を及ぼす。この音価要素に対して柔軟な処理を行うことができれば、手段や技量を選ばない、幅広いソースからシーケンスの構造を得ることが可能になる。

本稿では、まずはこれまで提案および実装した EMG と、適用規則処理と構造評価付けを用いた完全なシーケンスに対する構造解析の概要について述べ、さらに不完全なシーケンスからの構造推定手法について述べる。

## The method of presuming sequence structure from incomplete sequence using language class EMG.

Takeshi Ikeda, Nobuo Inui, Yoshiyuki Kotani  
Dept. of Information and Communication Science, Tokyo University of Agriculture Technology  
{ikedata, nobu, kotani}@fairy.ei.tuat.ac.jp

### Summary:

We proposed the language class EMG, in order to express the morph structure which musical sequences using grammar, such as a melody. And we implemented processing system "MAGIEnvironment" which analyzes and outputs the morph structure using EMG from given sequence. In this system, in order to treat various sequences, even if it does not prepare a many rule description, it is able to generate automatically the rule for which were suitable. We also tried to evaluate generated structure in whether have restricted elements. It is able to obtain the structure in a complete sequence like score description.

In case sequence processing, time value is required strict value of the judgment of a repetition. However, by the input method except on score, this element is not expressed completely. This has a great bad influence on the judgment of a repetition. If flexible processing can be performed to this element, it will become possible to acquire the structure of a sequence from various sources. This paper describes about outline of EMG, adaptive production process and evaluating structure. And also describes method to presume structure from incomplete sequence.

### 1.はじめに

我々は、楽曲を構成するメロディーなどの音符列(以下、シーケンスと呼ぶ)の形態的構造をコンピュータにて容易に表現および処理するために、文法ベースである言語クラス EMG(Extended Music Grammar)を提案した[1]。

文法ベースでのシーケンスの表現は L-Sysytem[2]や GTTM[3]などがあり、また、文法ベースとでなくても知識表現の一つとしてのフレームワークとしては MS[4]などがあるが、EMG は、楽曲の構造のうち、繰返しなどの拘束を伴った形態的構造を扱うのに最適化された言語クラスである。文脈自由文法 CFG(Context Free Grammar)を元に、一つの記号が書き換わることにより

導出される構造の部分的な構造を指定できる機構を持ち、この指定構造間の同一性をもって繰返しを表現(繰返し拘束)し、また、この指定部位間のある決まった関係を表現(関係拘束)することが可能である。EMG の導出は指定部位が常に下層部を指しているため、コンピュータ上での受理導出動作はボトムアップで行うことができる。

我々は、この EMG を処理するための処理系として "MAGIEnvironment" (以下 MAGI) を実装した。このシステムは、あらかじめ用意した文法記述に従って、入力されたシーケンスにおける構造を、文法の受理導出処理を行うことにより生成するものであり、生成された構造を見ることにより「どの部分が拘束関係にあるか」ということを判断することができる。

MAGI を動作させるためには、構造生成(解析)の元となる文法記述を用意しなくてはならないが、入力されるシーケンスの対象を汎用的なものにすることを考えると、この文法を明確に記述することは非常に難しい。様々な形態的構造のシーケンスを取り扱うのであれば、それに相当した文法記述、特に規則記述を用意すれば一応は解決する(これに関するものは我々は[5]にて検証した)。ただし、これに関しても、記述数が膨大に増える。これは理論的な面ではなく現実的な面で処理コストの大幅な増大を招いてしまう。そこで我々は、この文法記述について、基本的な記述要素だけを与え、それ以外の、特に拘束に関したものについては、システムが自動的に判断し、半ば規則を自動的に生成しながら構造を生成する手法を考案した。システムが生成する規則は着目している生成中の部分構造を結びつける際に、それに最も適した(決められた範囲でもっとも拘束を持っている)規則を自動的に生成しているという点が最も大きな特色であり、これを「適応規則処理」(Adaptive Production Process)と呼んでいる。

適応規則処理は、様々な規則を生成させるため、これにより生成される構造もたくさん出現する。今、我々が求めたい構造は「形態的な特色をもっとも有した構造」であるので、生成された構造をこの観点から評価付けする必要がある。我々は、拘束が係ることによりシーケンスの各音符の自由選択度が減少することに着目し、これがもっとも低いものを最尤の構造とし、これの評価値を計算する機構を導入した。これは、受理導出構造を生成する際にも、選択候補の除去などに有効に使用できる。

これら一連の処理によって、譜面情報をそのままシーケンス情報化したような「完全なシーケンス」であるならば、ほとんどの場合で解析不可能を起こすことなく形態的構造を取得することができる。

ところで、メロディーなどのシーケンスを入力することにおいて、最近のマルチメディアや音楽そのものの大衆化が相まって、音楽的な訓練を受けていない初心者がシーケンスを入力する機会が増えてきた。音符が音高と音価で構成されているとして、これを入力する時は、実

際にリアルタイムで入力した方がスムーズに入力できるのは言うまでもない。しかし、初心者にとってリアルタイムでほぼ正確にシーケンス、特に音価要素を入力することは容易ではない。無理に入力したとしてもそれは明らかに「不完全な」シーケンスとなってしまう。

我々はこのような不完全なシーケンスを完全なシーケンスに補正することを目標として掲げる。これを実現するための段階として、入力するシーケンスを完全なシーケンスではなく、音価において不完全なシーケンスでも形態的構造を推定できるようにすることを目指す。形態的構造が推定できれば、その構造内の拘束状態から適切な音価に補正することは比較的容易である。この構造推定手法として、音価の拘束条件である繰返しの判断基準を柔軟にすることを基本に、さらには事例を併用する手法を提案する。これにより、音価要素が大幅に崩れていたとしてもそれなりの構造を推定することが可能であると考えられる。

## 2. 言語クラス EMG の概要

言語クラス EMG は、前述の通り、シーケンスの形態的特色を表現することに特化した文法ベースの型のフレームワークである。ここでいう「形態的」とは、繰返しや音高シフトなどのシーケンス上に現れる特色を直的に示したものである。シーケンスにおいてはこれらは「拘束」という形で現れる。拘束関係にある部位は、一方のシーケンス状態が他方のそれに影響を与える、または影響を与えると見なせる状態であり、たとえば繰返しの場合は一方が決定するともう一方は一意に決定する、音高シフトの場合は、一方が決定するともう一方は先頭の音符だけが自由となる、などの状態を指す。

現在のところ EMG は、形態的情報より高位の知識情報(特に意味構造)は扱わない。たとえばメイン=サビの関係において、EMG はこれらを形態的特色上の理由から分割することは扱うが、これをもって特定の部位をメイン(サビ)と認識することは行わない(形態的情報での状態に限定するのであればこの限りではない)。日本語において、構文と意味構造の代表的なものである係り受け構造が異なると同様に、音楽においても意味に関しては、それ専用のフレームワークを用意すべきだと考える。

EMG の文法定義は fig.2-1 のようになっている。CFG にある 4 要素に、EMG 独自の 2 要素が付加されている。なお、文法定義の詳細および導出については[1]を参照されたい。

非終端記号は、CFG と同様に、それ自体が別の非終端記号列、または終端記号列に書き換わるものである。この書き換え過程がそのままシーケンスの構造として表現される。また、書き換えの開始として提示する記号が始記号であり、これも非終端記号に含まれる。すなわち、

構造はこの始記号をトップとした木構造として表現されることになる。

終端記号は、EMG では「音要素」と定義する。音要素とは音符やコードなどを示すものである。本稿では以後終端記号はすべて「一つの単音音符」とする。すなわち、始記号から導出される構造の末端部分を抜き出すとこの終端記号列となり、これがシーケンスとなる。

$G=(N, T, S, R, D, P)$

N: 非終端記号の集合(#~)

T: 終端記号の集合(\$~)

S: 始記号

R: 関係記号の集合(%~)

D: 導出識別記号の集合(\*x)

P: 規則記述の集合

#### ○規則記述の例

**規則種別1**(非終端記号 非終端記号列)

#SC #SC(\*1\*2)#SC(\*1\*3);

#SC #SC(\*1)#SC(\*2)/%OLV(\*1,\*2);

**規則種別2**(非終端記号 終端記号列)

#SC \$NOTE;

#### ○導出

- (1)同一の導出構造識別記号で示された部位は同一
- (2)関係記号で示された部位は、記号で示された関係が成立する

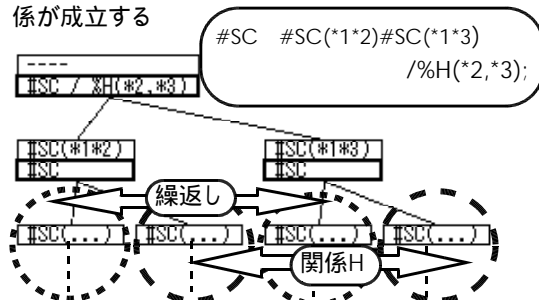


fig.2-1: EMG の定義・規則記述・導出

導出構造識別記号は、ある非終端記号が書き換わるとき、その書き換わることにより導出された構造を部分的に指示するものである。この記号は規則記述を見ても分かるとおり、規則右辺の非終端記号に付随して記述される。たとえば #SC(\*1\*2) とは、非終端記号 #SC は、右辺の非終端記号ないし終端記号が 2 つ並んでいる規則を用いて書き換わり、それ以下の構造において、部分的な構造をそれぞれ\*1,\*2 と識別する。EMG では、一つ上の規則で、その下の規則の書き換え記号数を限定する。

関係記号は、先の導出構造識別記号で指示された部位の形態的な関係を意味するものである。たとえば %R(\*1,\*2) という記述は、\*1 と\*2 には R という関係があることを意味する。これにより、R という関係がないものは除外される。なお、各関係記号が意味する関係は今のところ、終端記号列であるシーケンスの状態を見ることによって判断できる関係であるが、構造を着目する関

係を認めないわけではない。

繰返し拘束は、規則右辺の非終端記号の記述に使用する導出構造識別記号の使用状況で設定される。EMG の原則として、同一の導出構造識別記号は同一の構造であることを規定する。同一の構造であれば当然ながら同一のシーケンスであり、これをもって繰返しが表現される。

このように EMG は、シーケンスの形態を構造木の形で表現するように特化されたものである。

### 3.MAGIEnvironment

我々は、前章のように定義した EMG を実際にコンピュータ上で処理するためのシステムとして "MAGIEnvironment" (MAGI = Music, Acquisition, Generation, Interaction) を構築した。MAGI は、あらかじめ用意した文法記述を元に、入力されたシーケンスの形態的構造を、文法が受理できるかという形で生成するというシステムである。現在のところ入力できるシーケンスは単旋律のものに限られる。MAGI では、ただ単に入力シーケンスを文法記述に従って処理するわけではなく、現実的な処理を実現するために様々な手法が取り入れられている。

#### 3-1.受理処理(構造解析)

受理処理はボトムアップで行われる。つまり、初期状態としてシーケンスそのものを用意し、それが結びつくことのできる終端記号を検索し、実際に結びつく。そして、ある連続した領域を構成する記号列が規則で示されたとおりになっている場合に限り、その規則の左辺非終端記号にまとめられる。これはちょうど導出過程の逆を行っていることになる。最終的に、対象のシーケンス全体が一つの非終端記号にまとめられ、かつ、それが始記号として示された記号であるならば、その構造をもって受理導出構造とし、文法はシーケンスを受理できるとする。なお、EMG は構造的な同一性も要求するため、実際には生成された部分構造を常に保持するようになっている。詳しくは[1]を参照されたい。

#### 3-2.適応規則処理

当然ながら、文法ベースの処理システムにおいて、すべての動作を司るのは文法記述であり、これなしには受理処理や生成処理を行うことができない。しかし、我々が対象としているシーケンスの事象において、たいいていのシーケンスを受理できるような文法を用意するのは非常に難しい。もちろん論理的には可能であるが、汎用的

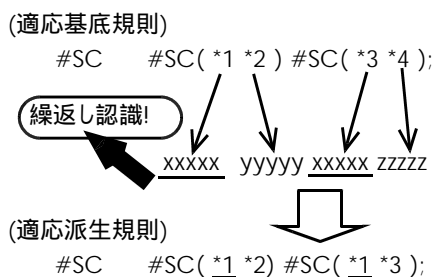
な面や現実的な面で不都合が多く発生することは容易に考えられる。

そこで、我々は、規則において、純粹に記述によって与える文法をごく基本的なものとし、実際の規則記述は対象のシーケンスに合わせて自動的に生成する手法を考案した。これにより、シーケンスがあらかじめ指定した基本的な形をなしているのであれば解析可能であり、同時に、そのシーケンスに適応した規則が自動的に生成することができる。この処理を、規則をシーケンスに合わせて適応させることから「適応規則処理」(Adaptive Production Process)と呼んでいる。

適応規則処理においてあらかじめ用意する文法記述は、非終端記号、終端記号、始記号、関係記号の種類と、導出構造識別記号を伴った規則の「基本的な」記述だけである。そして、処理時に基底な規則記述を実際のシーケンスの状態に合わせた「派生的な」記述を生成する。よって、適応規則処理は処理時に規則を肉付けすることが主の動作である。このとき、あらかじめ与える基底な規則を「適応基底規則」、処理によって派生された規則を「適応派生規則」と呼ぶ。なお、便宜上、文法 G には生成された適応派生規則も含まれることを前提とする。

ある適応基底規則 Pf と、それによって得られた基底派生規則 Pi においては次のような関係がある。また、適用基底規則と適用派生規則の例を fig.3-1: に、実際にシステムに与える規則記述を fig.3-2: に示す。

- (1) Pi は Pf に繰返し拘束、関係拘束を付加したも
- (2) 導出木において、Pi が適用された部分に Pf を適用しても構造は文法を充足
- (3) 適応規則処理を行う関数を AP(Pf,St) (St: 着目している下層部構造) とすると、Pf と St が一意であるならば常に Pi=AP(Pf,St) となり一意の適応派生規則が生成



関係拘束に関しても同様に設定する

fig.3-1: 適用基底規則と適用派生規則の例

適応派生規則は元となった適応基底規則を具体化したものといえる。この具体化の決定要素として実際のシーケンスの状態が使用される。適応基底規則は繰返し拘束も関係拘束も設定されていないので、規則が示した基本

的な構造(右辺書き換え記号の書き換える構造の分割数)を満たしていれば、どのような構造にも適用できる。そして、実際に結びついた構造において、各導出構造識別記号間の構造に繰返し(同一)や関係が認められる場合は、そのように規則を変形して、その規則にすりかえることになる。適応規則処理は、ボトムアップ型の受理処理の規則適用部分を変更するだけで実現できる。

(下層部の直線的構造を形成する規則群)

```
#LNT $NOTE;
($NOTE は扱える音符の集合を示す。この規則
は#LNTが$NOTE内の音符のどれかに書き換
わる(あるいはそのような規則自体の集合))
#LN #LN(*1*2) #LNT(*3)
/ %LNI_1_2_N(*1);
#LN #LN(*1) #LNT(*2);
#LN #LNT(*1);
#LNS #LN(*1);
#LNS #LN(*1*2);
#SC #LNS(*1) / %LNI_1_4_2(*1);
```

(適応規則処理の対象規則[適応基底規則])

```
#SC #SC(*1*2) #SC(*3*4);
#SC #SC(*1*2) #SC(*3);
#SC #SC(*1) #SC(*2*3);
#SC #SC(*1) #SC(*2);
#SC #SC(*1*2);
```

関係記号 %LNT\_~ は、単独の部位における音符数の制限を意味する  
始記号は #SC である

fig.3-2: システムに与える規則記述

前述の関係(3)より、同じ適応派生規則および対象の下層部構造ならば、適応規則処理によって出力される適応派生規則は常に一意とする。このとき、St が間接的に示すシーケンスにおいて、拘束が最も強く係っている適応派生規則が生成される。我々の構造生成の目的は、対象のシーケンスの構造のうち、もっとも拘束が係っているものであるため、求めたい適応派生規則も自ずとそのようになる。また、そのような観点からすれば、適応規則処理は、無駄な構造の生成を抑制するという利点もあり、効率的な構造生成に貢献すると考えている。

なお、適応規則処理はすべての規則において行うわけではなく、指定した規則だけに行うようになっている。実際 fig.3-2: の規則記述を用いると、終端記号に近い部分は直線的な構造となり、それを単位として適応規則処理を用いて構造を組み立てることになる。

### 3-3. 構造の評価付け

システムは文法から考えられるすべての構造を生成す

る。我々が求めたい構造は、シーケンスの形態的特色を最も表したものである。これを探し出す機構が必要となる。そこで我々は、得られた構造において音符の自由選択性に基づいた構造の評価付けを行う。

前章の fig.3-2:において、終端記号から非終端記号 LNT へ還元する規則では、図中の記述にもあるように、終端記号の集合 NOTE のどれかが選択される。この規則だけでは音符が NOTE で示したものであるならばどれもこの規則を適用できる。しかし、上部の構造において、たとえば繰返し拘束が係っている場合は、この選択は一意となる。これを用いて、シーケンス内音符において、生成されたシーケンスより上の構造がどの程度自由に選択できるかを数値的に算出することが可能である。これは同時に表層要素であるシーケンスの状態と、それを構成する構成要素の分離して扱うことも示している。

いま、3 音符のシーケンス  $Note3=\{n1, n2, n3\}$  があるとして、これの各音符のすべてが自由に選択できる場合の評価値  $E(Note3:St3)=3ee$  ( $St3$ :  $Note3$  を構成している構造、 $ee$ : 音符一つの選択性に対する定数)とする。当然、この値は  $Note3$  内の音符数が増えるほど、構成音符のすべてが自由に選択できるならば、この値は音符数に比例して増加する。しかし、今度は 6 音符のシーケンス  $Note6=\{n1, n2, \dots, n5, n6\}$  があり、かつ、これを構成する構造にて、前半と後半が同一であることが示されている場合(たとえば  $cdecde$  など)は、音符の自由選択性は前半(または後半)3 音符分しか存在しない。このときの評価値  $E(Note6:St6)=3ee$  とする。

つまり、評価値は、完全に自由に選択できる音符の長さを基準としたものとする。この値が小さいほど自由選択性のある音符が少なくなり、それが最も小さい構造がもっともシーケンスの形態的特色を示しているものであるといえる。

評価値の個々の計算は各要素(音高・音価)ごとに計算し、最後にこれらをひとまとめにする。ある構造の評価値において、音高に着目した評価値が  $E_p()$  で、音価のそれが  $E_t()$  であるとすると、これらを統合した評価値  $E()$  は次のように計算する。

$$E = E_p + E_t \quad (E3-1)$$

なお、音符一つの評価値はあらかじめ与えた定数(要素ごとに与える)とする。評価値はこれを単位としたものとなる。

ある二つのシーケンス  $Sa, Sb$ (構造は  $Ta, Tb$ 、それぞれの評価値は  $E(Ta), E(Tb)$ )が連結しているとき、それらに何ら拘束がない(繰返しも関係もない)場合、連結されたシーケンス  $S_{a+b}$  およびその構造  $T_{a+b}$  の評価値は、次のように各構造ごとの評価値を和にしたものとする。

$$E(T_{a+b}) = E(T_a) + E(T_b) \quad (E3-2)$$

次に、二つの領域に繰返し拘束があるときは、自由に

シーケンスの音符を選択できる範囲が繰返しの分だけ減少することになる。このような場合、つまり、 $Sa$  と  $Sa$  が連結して  $S_{a+a}(T_{a+a})$ となる状態では、評価値は  $Sa$  単体のものだけとする。

$$E(T_{a+a}:rep) = E(T_a) \quad (E3-3)$$

さらに、二つのシーケンスおよび構造にある関係  $R$  が成立して構造が連結されている状態を  $T_{a+b}:R$  とすると、 $E(T_{a+b}:R)$  は、何も拘束がない状態  $E(T_{a+b})$  よりも評価値は減少することになる。これは、 $Ta$  および  $Sa$  が決定すると  $Tb(Sb$  の音符)は完全に自由に選択できなくなるからである。関係  $R$  によって減少する評価値の割合を  $D(R)$  とすると、この状態の評価値は次のように計算する。

$$E(T_{a+b}:R) = D(R)E(T_a) + D(R)E(T_b) \quad (E3-4)$$

この  $D(R)$  は、関係記号  $R$  の種類によって異なる。単に関係が係っていることを評価するものと、厳密に評価値の減少を計算できるものが存在する。

評価値を計算する上では、部分構造の下層部構造( $Ta, Tb$  など)のそれが必要となるため、評価値計算も受理処理と同様にボトムアップで行う必要がある。MAGI では、着目領域の構造生成が終了した直後にその構造の評価値を計算するようになっている。また、構造評価値は、効率的な処理を行うために、評価の低い(評価値の大きい)構造を除去するためにも使用することができる。この構造除去において受理処理は効率が大幅に上昇する。

なお、構造除去によって、取得される最尤構造は理論的には近似的なものになるが、関係記号の機能が構造自体を参照しない(シーケンスだけを参照にする)場合はそれほど影響はない。

## 4.不完全なシーケンスへの対応

現実的な問題として、入力するシーケンスの質の問題がある。たとえば、譜面に記述された情報を入力として用いる場合、譜面からシーケンスを起こす必要があるが、この際、シーケンサや譜面ソフトなどを用いることが多い。しかしこの方法は、初心者などには必ずしも気軽な方法とは言えず、また、入力に対する人的コストも決して小さくない。

もっとも気軽な方法といえば、鍵盤などからのリアルタイム入力が考えられるが、ここから得られたシーケンスから構造を把握することは容易ではない。特にこれは形態的特色であるほど顕著であり、音価の拘束に関しては致命的である。

音価の繰返しとは「音価値が完全に一致している」であるが、あるフレーズを入力者が繰返して演奏していると思っけていても、実際それを数値化した際にはそうとは言えない違いが頻繁に発生する。

我々は、より現実的な入力手段から得られたシーケン

スからも構造を把握できるように、解析をよりロバストに行えるようにすることを目指す。これが実現すれば、入力者や手段を選ばないより身近なシーケンス入力システムが実現できると考えている。

本稿ではこのロバストな構造生成を実現するための基本的な手法と、予備的な実験を示す。

#### 4-1.完全・不完全なシーケンス

問題を明確にするために、ここで二つのシーケンスを定義する。

まず、シーケンサや譜面ソフトなどを用いて、記譜情報に相当するものを厳密に再現したシーケンスを「完全なシーケンス」と定義する。特に繰返しに関しては、繰返しが数値的に見ても繰返しになっているものとする。また、これを入力して得られた構造をそのシーケンスの持つ本来の構造であるとする。

一方、鍵盤などからのリアルタイム入力を含む厳密でないシーケンスを「不完全なシーケンス」と定義する。これは、特に繰返しを演奏するという行為と、入力されたその結果が必ずしも一致しないシーケンスであるとする。MAGI でも単純にこれを入力として受け入れてもそれなりの構造を生成することができるが、それはそのシーケンスを完全に信用したものであり、求めるべき構造ではない。

今、我々は、不完全なシーケンスを音価において不完全であるというものと位置づけ(つまり音高値は完全なものである)、このシーケンスを前処理なしで完全なシーケンスから得られた構造と近いもの、最終的には同一なものを推定することを目標とする。

#### 4-2.音価要素の繰返し判定の柔軟化

前述の通り、音価要素における不完全なシーケンスにおいて、繰返されている部分では、大まかには似ているにしても、厳密には一致しない。今回は厳密に一致しないものでも繰返しであると認識することを、認知的観点ではなく機械的処理にて行うことを考える。

シーケンス内音価要素の一部分である T1 と T2 において、それが音価の繰返しを判断するためには次のような判断を行っている。

- (1) T1 の音符数と T2 の音符数が同一
- (2) T1, T2 の対応する音符の音価が完全に一致

不完全なシーケンスでは、(2)の部分で問題が生じることは明白である。コンピュータ上ではこの音価値はテンポや分解度を基準とした絶対的な音価値(4 分音符

=480 など)で表現するが、この値が 1 でも異なれば繰返しだとは判断できなくなる。これを回避するためには、(2)の判断を閾値などのマージンを用いて柔軟にすることが考えるが、よりの確に判断するために、閾値を導入する代わりにいくつかの判断基準を追加する。

- (1) T1 の音符数と T2 の音符数が同一
- (2) T1 の音価和と T2 の音価和の差が範囲内
- (3) T1, T2 の対応する音価の差が範囲内
- (4) T1, T2 を先頭の音価値を基準として相対化(部分相対音価)し、各音部分相対音価の差が範囲内

音価の繰返しであるときは、そのシーケンス間では音符数が一致しており、各部位のシーケンスの音価和が一致し、対応する音符同士でも一致している。さらに、各部位の先頭の音価を基準として部分相対化してもそれぞれが一致している。繰返しはこれらの要素がすべて成立している。このうち(1)以外の各要素にそれぞれ個別の閾値を与えて、すべてが成立する時に繰返しであると判断する。厳密に数式化すると次のようになる。なお、シーケンスの音価要素を  $T_1 = \{t_{11}, t_{12}, \dots, t_{1n}\}$   $T_2 = \{t_{21}, t_{22}, \dots, t_{2n}\}$  とする。

- (1) は音符数が一致しているの自明

$$(2) \sum_{i=1}^n t_{1i} / \sum_{i=1}^n t_{2i} \geq TH_{sum} \quad (E4-1)$$

(ただし  $\sum_{i=1}^n t_{1i} \leq \sum_{i=1}^n t_{2i}$  そうでない場合は分母を逆にする)

- (3) すべての i において  $t_{1i} / t_{2i} \geq TH_{pair} \quad (E4-2)$

(ただし  $t_{1i} \leq t_{2i}$  そうでない場合は分母を逆にする)

- (4) T1(T2)の全音価について  $t_{11}(t_{21})$  を 1.0 とした部分相対音価を求める。

$$tr_{1i} = t_{1i} / t_{11} \quad tr_{2i} = t_{2i} / t_{21} \quad (\text{双方とも } i \text{ は } 2 \text{ 以上})$$

このとき  $tr_{1i} / tr_{2i} \geq TH_{rate} \quad (E4-3)$

(ただし  $tr_{1i} \leq tr_{2i}$  そうでない場合は分母を逆にする)

どの判定式においても、閾値が 1.0 の時が完全な一致を要求し、値が小さいほど幅が広がり、様々なシーケンスを受け付けるようになる。

なお、評価値については、先の評価値計算では規則の

記述形式だけを参照して求めているため、実際のシーケンスの細かい内容については参照していない。よって緩和した判断式を用いた場合でも評価値の計算方法に変化はない。

### 4-3. 閾値の影響と事例の併用

前述のように、閾値を用いて判断基準を緩和することにより、ある程度不完全なシーケンスでも繰返しを推定することが可能である。しかし、閾値を構造生成時に一律に設定すると、値を大きくするほど、本来繰返しではない、あるいは、繰返しとは到底認められないものまでをも認識してしまう可能性がある。

これを回避するために我々は、あらかじめ用意した既存の完全なシーケンスから得られた規則と部分構造を用いて、動的に閾値を変化させることを考えている。

事例を使用する利点としては、閾値を動的に可変させながら適用できるという点がある。静的に小さい閾値を設定すると、基準の緩和がほかの部分にも波及してしまうが、動的に変化させることに避けることができる。

構造を生成する際、一つの構造を生成するには、一つ以上のシーケンスで見えて並んでいる下層部構造群を頂上適用規則の記述状態にあわせてつなぎ合わせて形成される。適応規則処理ではこの下層部構造群を状態を見て頂上適用規則を生成していたが、本来は、下層部構造群から頂上適用規則が選択されることになる。ボトムアップでの構造生成の場合、この下層部構造群は既知であるため、これと実際に適用された頂上適用規則を事例として持っておき、構造推定時にはその事例から規則を選び、その時の閾値は選択された事例の確率値から次式を用いて決定する。なお、式中の各パラメータは閾値の種類ごとに独自に定める。

$$TH = TH_{base} - (TH_{base} - TH_{limit}) \times \frac{P_{this} - P_{min}}{P_{max} - P_{min}} \quad (E4-4)$$

THbase: 基準閾値

THlimit: 限界閾値

pthis: 選択された規則についての確率

Pmin: 同一条件下の事象の最低確率

Pmax: 同一条件下の事象の最大確率

統計は、下層部構造群の条件付き確率で行い、確率値の最大値と最低値をあらかじめ設定した閾値の範囲に線形で割り当てる。これにより、統計的によくある部分構造であるほど柔軟に受け付けるようになる。部分構造の範囲は、理論的には終端記号までに至る部分を取得することは可能であるが、事例を集める現実的な点から、頂上部を含めて2～3階層とする。

なお、事例とともに適応基底処理も同時に行なう、その際にはあらかじめ設定した閾値を静的に使用する。

### 5. 解析結果および推定の予備実験

まず、完全なシーケンスとして用意したシーケンスを適応規則処理にて構造を生成した結果を fig.5-1: に示す。これは、コラール"Aus meines Herzens Grunde"におけるソプラノパートを抜き出して解析したものである。構造は、直線的な構造で区切られたいくつかの細かい部分に分かれ、それらが上層部の規則によってつながっていることが分かる。また前半の繰返しに対応する規則が生成されている。

次に、不完全なシーケンスの推定のために、予備的な解析結果を示す。前章でも述べたとおり、閾値は適応規則処理での静的な設定と、事例として得られた規則における動的な設定が存在するが、今回は事例においてはクローズデータ(事例に解析元のシーケンスが含まれる)にて行なった結果を示す。先のシーケンスの前半を抜き出したものにおいて、不完全なシーケンスを再現するために、各音符の音価を30%以内に乱数値を乗して音価列にばらつきを与えたものを対象とする。静的な閾値だけを設定し事例を用いずに解析した結果を fig.5-2: に、そして、先の事例を用いて解析した結果を fig.5-3: に示す。なお、静的な閾値はどの判断式においても0.8とした。この結果、事例なしのものでは繰返しが認識されていないが、事例を用いると今までは認識できなかった繰返しが認識し、fig.5-1:と同様な構造が得られていることが分かる。

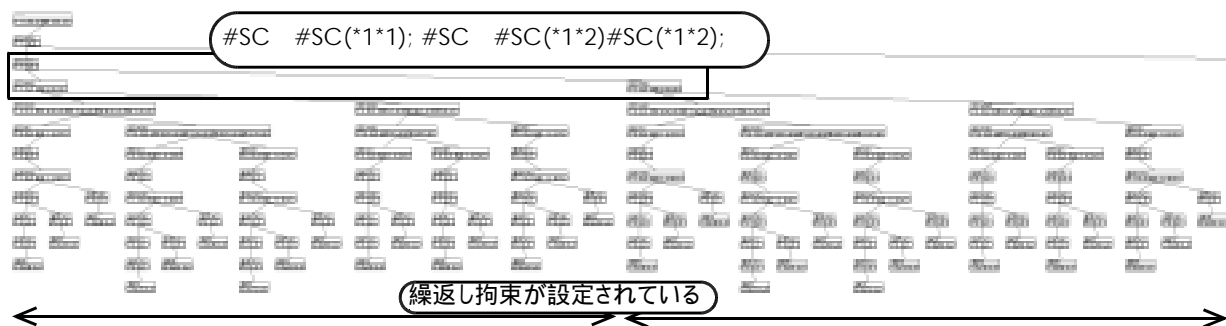


fig.5-1: Aus meines Herzens Grunde の解析結果(前半だけ掲載)(完全なシーケンス)



fig.5-2: 不完全なシーケンスの構造推定結果(事例未使用)

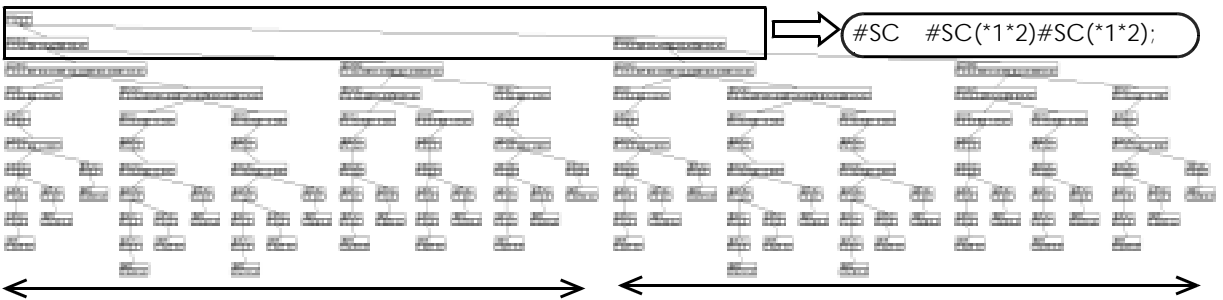


fig.5-3: 不完全なシーケンスからの構造推定結果(事例併用)

## 6.考察・展望

取得された構造を見ても分かるように、完全なシーケンスに関しては、シーケンスにそれなりの拘束があるならば、それが規則として現れている。この構造は EMG の存在理由である「形態的な構造」を示している。逆に言えば、形態的ではないもの、たとえば認識での区切りなどについては必ずしも一致しない。ただしこれに関しては関係記号にそのような機能を追加する(たとえば GTTM のグルーピングなど)ことにより、ある程度は対応できると考えられるが、目的の構造によって導入すべき関係記号を選んでいく必要がある。

一方、構造の推定についてだが、静的に閾値を低下させて柔軟にするだけでは、完全に構造を推定できないことは今回の予備実験からも明白であり、事例を用いた局所的な閾値の変化は必要不可欠であることが分かる。

今回の推定は、クローズドデータで行なったが、最終的にはオープンデータでもそれなりの構造を得られるようにすべきである。ただしそのためには事例の調整が必要となる。今回は閾値を静的に設定した適応規則処理と、事例を用いた動的な閾値設定を併用したが、適用規則処理で生成した規則と事例内規則の関係が規則選択に問題が生じることも考えられる(なお現在は、適応規則処理によって生成した規則が事例規則内に同一のものがあるかどうかをチェックし、ある場合は検索時に置き換えている)。また、事例として蓄える規則がたくさんあるという仮定ならば適用規則処理は必要ないが、それが現実として可能かどうかは現在調査中である。どちらにしても、今後、オープンデータに対応するために事例の効果的な使用についてより考究する必要があると考える。

## 7.まとめ

本稿では、文法クラス EMG を用いた EMG 処理系 MAGIEnvironment を実装した。このシステムを実現するに当たって、様々なシーケンスを取り扱うために文法内規則を半自動的に生成し処理する適応規則処理を考案し、最尤の形態的構造を取得するために構造の評価付けを行った。また、不完全なシーケンスにも対応するために、音価における繰返し拘束の判定を閾値をもって緩和する手法を提案した。今後は用いる事例の調整・緩和基準の検討など、総合的によりロバスタなシーケンスからも構造が得られるようにし、構造からシーケンスを補正する処理を実現したい。

## 参考文献

- [1]池田剛, 乾伸雄, 小谷善行: 音楽記述文法 EMG により楽曲受理処理系の試作, 情報処理学会音楽情報科学研究会報告 SIGMUS47-20, 2002
- [2] J. McCormack: Grammar Based Music Composition, Complexity International, 1996
- [3]F. Lerdahl, R. Jackendoff: A Generative Theory of Tonal Music, MIT-Press, 1983(1996)
- [4]M. Balaban; The Music Structures Approach to Knowledge Representation for Music Processing, Computer Music Journal 20-2, 1996
- [5]池田剛, 乾伸雄, 小谷善行: 音楽記述文法 EMG による楽曲の規則的構造の抽出, 第 44 回プログラミングシンポジウム, 2003