

音楽理論 GTTM に基づく多声音楽の表現手法と基本演算

平田 圭二[†] 青柳 龍也^{††}

本論文では Generative Theory of Tonal Music (GTTM) という音楽理論と演繹オブジェクト指向データベース (DOOD) という知識表現法に基づく多声音楽の表現手法, 及びそうして表現される多声音楽に対する基本演算について述べる. 今後の技術や文化の発展を考えると, 音楽知識を計算機上で表現し操作する必要性はますます高まるものと思われる. 従来の音楽記述手法では, 五線譜上で上下左右に隣接する表層的な音どうしの関係しか記述していなかった. 我々の手法は音楽理論に基づき, 音楽的な深層構造を考慮して多声音楽を表現する. その結果, DOOD で表現されたオブジェクト間の包摂関係と GTTM の定義する抽象化, 具体化の関係を適切に対応付けることができた. さらに, 本手法で表現された多声音楽に対して包摂関係, 最小上界, 最大下界という基本演算に着目し, それらが持つ音楽的な意味に検討を加えた.

Representation Method and Primitive Operations for a Polyphony based on Music Theory GTTM

KEIJI HIRATA[†] and TATSUYA AOYAGI^{††}

This paper presents a new representation method for a polyphony based on a music theory, Generative Theory of Tonal Music (GTTM), and a knowledge representation scheme, Deductive Object-Oriented Database (DOOD), and the primitives for a polyphony represented by our method. Considering the future development of technology and culture, we think that the requirements for representing and manipulating musical knowledge on a computer will increase in future. Conventional music description languages allow us to describe just the surface relationships between a note and its surrounding notes on a score. Our method can represent a polyphony, taking into account its deep musical structures. Consequently, we can make the appropriate correspondence between the subsumption relation of objects in the DOOD and the abstraction and instantiation relations stipulated by GTTM. This paper also discusses the operations of the subsumption relation, least upper bound and greatest lower bound, and investigates their musical meaning.

1. はじめに

今後ますますインターネットが普及しマルチメディア技術が身近になるにつれ, 音楽をコミュニケーションのメディアとして使う機会が増えていくだろう. 例えば, 各個人が新しい音楽を創造しそれをやりとりするようなコミュニケーションが一般化すると, 音楽による自己表現や創造性の支援が重要になるだろう. その際には, 音楽が持つ意味やそこに暗黙的に込められた意図を考慮しながら, 計算機上で適切に処理を行う音楽システムが必要になる.

音楽は曖昧で主観的という特徴を持つが, 音楽理論と

はそのような音楽が持つ意味やそこに暗黙的に込められた意図を分析, 解釈するための理論である. 上記のような音楽システムを開発するには, 音楽理論を積極的に援用すべきであるが, 従来の音楽システムの多くはそれを行ってこなかった. その理由は, 音楽理論が計算機上での実装やアルゴリズム化を前提としてないため, および音楽理論は音楽そのものの理論であって, 演奏したり聴取するための理論ではないからである⁹⁾. 従来の音楽システムの多くはシステム固有の限定的な楽曲分析の方法を採用しているが, そのような方法では音楽的な構造や意味を十分考慮することができず, 質の高い楽曲分析や生成はあまり期待できないと考える.

我々は音楽理論に基づく音楽システムを開発している. 音楽理論を積極的に援用した音楽システムを設計するに際して決定すべき事項が 2 つある. 1 つはどのような音楽理論に基づくのか, もう 1 つはどのような表現手法を採用するかということである.

[†] NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

^{††} 津田塾大学 情報数理科
Department of Mathematics and Computer Science,
Tsuda College

音楽理論に関して、定義や規則が厳密に記述してあり、理論全体が緻密に矛盾なく構築されているものが望ましい。これは、アルゴリズム化する際の開発者の主観や無意識の具体化をできるだけ排除するためである。

知識表現に関して、表現すべき知識は、推論を行うための規則と推論の対象となるデータの2つに大きく分けられる。知識表現手法は、これまで様々な手法が提案されているが、表現すべきことを簡潔に述べられる記述力と正確さが求められる¹⁸⁾。

我々は、上の議論をもとに、音楽理論として Generative Theory of Tonal Music (GTTM)¹⁷⁾ を採用し、表現手法として Deductive Object-Oriented Database (DOOD)^{14),22)} ☆を採用し¹⁰⁾、一連の音楽システムを作成した^{6)~8),11),12)}。これより、次の2つの成果を得た。(1) 多声音楽^{☆☆}としての楽曲をその音楽的な意味とともに内部表現する手法を開発した。つまり、DOODにより表現されたオブジェクトの間の包摂関係を GTTM の定義する抽象化、具体化の関数に適切に対応づけることができた。(2) その手法に従って表現された多声音楽に対する音楽的な基本演算を提案した。DOODにより表現された項に対する最小上界と最大下界の演算は、楽曲間の共通部分を計算する演算と重ね合わせ部分を計算する演算に対応する。この2つの基本演算を用いると、多声音楽の類似度を判定するアルゴリズムや編曲のアルゴリズム等を構成することができる。

本論文では、GTTM と DOOD に基づく多声音楽の表現手法、及びそうして表現された多声音楽に対する基本演算について述べる。本論文の構成は次の通りである。まず第2章で GTTM について概説し、第3章では DOOD について概説する。第4章で GTTM と DOOD を用いた多声音楽の表現手法とその特徴について述べ、第5章で基本演算について述べる。第6章で議論を行い、最後の第7章でまとめと今後の課題について述べる。

2. Generative Theory of Tonal Music

GTTM は、音楽に関して専門知識のある聴取者の直観を形式的に記述するための理論として Fred Lerdahl と Ray Jackendoff により提唱された^{17),19)}。五線譜に記述された楽曲(ホモフォニー^{☆☆☆})を表層構造として、

☆ DOOD という用語はもともと国際会議や研究分野の名称として用いられていたが、本論文では知識表現手法の名称として用いる。

☆☆ Polyphony. 複数の独立した声部を持つ音楽のこと。

☆☆☆ Homophony. 主声部の旋律に対し簡単な和声的な伴奏を付されたもので、単旋律的に解釈できるような様式。

それに解釈や分析を加え、そこに内在するさまざまな階層構造を深層構造として顕在化させる。Chomsky の変形生成文法の枠組にならったと言われている。楽曲を分析する理論はこれまで幾つか提案されているが、その中でも GTTM が最も緻密に構築されており計算機への実装に適しているであろうと考えられている^{1),20),21)}。

2.1 理論の概要

この理論はグルーピング構造分析、拍節構造分析、タイムスパン簡約 (time-span reduction)、プロロンゲーション簡約 (prolongational reduction) という4つのサブ理論から構成されている。

グルーピング構造分析は、連続したメロディをより短い部分(フレーズ)に分割することである。長いメロディを歌う時にどこで息継ぎすべきかを見つけるような分析である。

拍節構造分析は、楽曲の拍(指揮者がタクトを振るタイミング、あるいは聴取者が曲に合わせて手拍子を打つタイミング)を見出し、さらに四分音符、二分音符、1小節等のレベルにおける強拍と弱拍を同定する分析である。

タイムスパン簡約は、あるメロディはそれより長いメロディを簡約化(reduction)、抽象化することによって得られるという直観を表現したものである。例えば、ある楽曲の全体は最終的にその調の主音に簡約化されると考える。タイムスパン簡約は、グルーピング構造分析と拍節構造分析の結果に基づいて、部分を合わせて全体にまとめあげるという意味でボトムアップに行われる。

プロロンゲーション簡約は、次の2つの直観を表現したものである。1つは、メロディ中のある音はそれまでの音の繰り返しとして或は次の展開をもたらしために出現しているというものである。もう1つは、メロディの全体や部分は、一般に、緊張と弛緩というパターンを持っているというものである。プロロンゲーション簡約の実行は、タイムスパン簡約の結果に基づいて、トップダウンに行われる。つまり、楽曲中から繰り返しのあるいは展開的に最も重要な音を選び出し、その音を手がかりにして楽曲を分割していく。

GTTM の規則は、分析を進めて木構造を生成するための構成規則 (well-formedness rule) と構成規則適用に関する知識である選好規則 (preferencne rule) の2種類から成る。楽曲を分析することは、タイムスパン簡約及びプロロンゲーション簡約を行うことである。これらの簡約結果は、各々タイムスパン木及びプロロンゲーション木として表現される(タイムスパン木の例は図1を参照)。これらの木が、GTTM の意味で楽曲の深層構造を表現している。人それぞれで楽曲の解釈が異なる

ように、1つの楽曲には、一般に、複数通りの分析が可能である。これは、GTTMの分析では選好規則による解釈の絞り込みが働いても、タイムスパン木及びプロロンゲーション木が複数通り導かれたことに対応する。

2.2 タイムスパン簡約

上述したように、プロロンゲーション簡約はタイムスパン簡約に基づいているので、GTTMによる楽曲分析の第一段階はタイムスパン簡約である。我々はまずタイムスパン簡約の形式化を目指す。

タイムスパン木は以下のようにボトムアップに構成される。まず、楽曲に含まれる隣接する音や和音どうしがグループを作る。そのグループに含まれる音や和音の中でタイムスパン木において重要な音あるいは和音はそのグループを代表する (head と呼ばれる)。タイムスパン木における重要度は、グルーピング構造分析と拍節構造分析の結果から決定される。さらに、隣接する head どうしで同様にグループが作られる。このグルーピング作業を楽曲全体が1つのグループになるまで続ける。

図1は、あるホモフォニーとタイムスパン木の例である。タイムスパン木は基本的に二進木であり、重要な

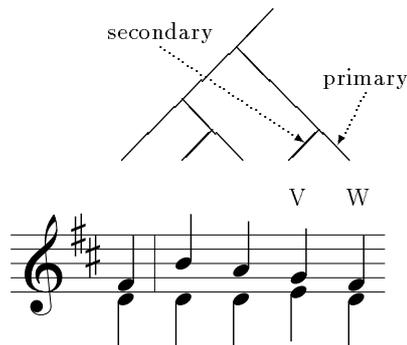


図1 ホモフォニーとタイムスパン木の例

Fig. 1 Example of Homophony and Time-Span Tree

枝を primary, そうでない枝を secondary と呼ぶ。タイムスパン木の最下端の枝には五線譜上の各音 (和音や単音) が現れる。図中, primary 枝の下には1小節め4拍めの和音 W がぶら下がり, secondary 枝の下には1小節め3拍めの和音 V がぶら下がる。和音 V と W は隣接するのでグループを作る。このグループの head は和音 W である*。ある楽曲をタイムスパン簡約するとは、楽曲中の重要でない音 (例えば和音 V) を削除することである。

* head の作り方には, ordinary, fusion, transformational, cadential retain の4通りがある。この場合は ordinary である。

3. DOOD

前述したように、音楽は曖昧で主観的という特徴を持つ。ここで曖昧とは、解釈の多義性を解消するために必要な情報が欠落しているか、部分的にしか与えられていないことを意味する。これを知識表現の立場から見ると、ある属性が欠落しているか、属性の値ではなくその型だけが与えられている (宣言されている) ということに対応する。従って、分析によって顕在化された楽曲の深層構造が複数通り存在するという多義性を表現するためには、属性の欠落や属性の型宣言が記述できなければならない。また、主観性とは、複数通り存在する深層構造のいずれを採用するかに対応している。この深層構造の採用はユーザの価値判断を反映しており、お互い矛盾するようなものを採用する場合もある。この主観性には事例やコーパスに基づく推論¹⁶⁾が有効であろうと考えている。その事例やコーパスはデータでもあり規則でもあるので、データと規則が同一の形式で表現されるような表現手法を採用することで、高い可読性と効率的な推論が可能となるだろう。その代表的なものは一階述語論理である。

DOOD は、属性の欠落や属性の型宣言を記述するために一階述語論理を拡張した手法である²²⁾。さらに DOOD では、演繹規則によって項間の包摂関係を定義することができ、この包摂関係によって曖昧な項とそうでない項を形式的に結び付けることが可能となる。また DOOD は素性構造¹⁵⁾と関連が深い。

3.1 オブジェクト項

DOOD では、対象をオブジェクトとその属性の集合から成るオブジェクト項として表現する。本論文では、オブジェクト項を $o(\dots, l : v, \dots)$ と記述する。ここで o は基本オブジェクト項, $l : v$ は属性, l は属性ラベル, v は属性値を表わす。ここで便宜的な記法を導入する。 $p(\dots, l : v, \dots)$ というオブジェクト項がある時, $p.l$ と表記するとその値は v と等しいとする (ドット記法と呼ばれる)。包摂関係 (\sqsubseteq) とは「情報量が多いオブジェクト \sqsubseteq 情報量が少ないオブジェクト」という意味である。あるいは「具体的なオブジェクト \sqsubseteq 抽象的なオブジェクト」, 「特殊 \sqsubseteq 一般」を意味する。以降、本論文ではオブジェクト項とオブジェクトを同一視する。

3.2 包摂関係

オブジェクト間の包摂関係は以下の演繹規則によって定義される。今 $o_1 = p(\dots, l_m : v, \dots)$, $o_2 = p(\dots, l_n : w, \dots)$ とすると, o_1, o_2 間の包摂関係は,

$o_1 \sqsubseteq o_2 \leftarrow \forall l_n \exists l_m (l_m = l_n \wedge o_1.l_m \sqsubseteq o_2.l_n)$ と定義される. o_2 の全ての属性 l_n について o_1 のそれが全て具体的ならば, o_1 の方が具体的 (あるいは o_2 の方が抽象的) という意味である. 属性の少ないオブジェクトは包摂関係に関してより抽象的である. 基本オブジェクトが異なるようなオブジェクトどうしでは包摂関係が成り立たない.

簡単な例として note , $\text{note}(\text{pitch} : C)$, $\text{note}(\text{pitch} : C, \text{octave} : 5)$ の3つのオブジェクトを考える. ここで note は抽象的な1つの音を表す概念であり, 外延的にはあらゆる音から成る集合と等価である. $\text{note}(\text{pitch} : C)$ は note より具体的な音が音階 (ピッチクラス) の情報しか与えられていない音のことであり, 例えば「Cの和音は C E G から成る」と言う時の C の音のことである. さらに具体的な音が $\text{note}(\text{pitch} : C, \text{octave} : 5)$ であり, 例えばピアノ鍵盤上の C5 の音等が該当する. 実際に $\text{note}(\text{pitch} : C) \sqsubseteq \text{note}$ 及び $\text{note}(\text{pitch} : C, \text{octave} : 5) \sqsubseteq \text{note}(\text{pitch} : C)$ が成立する.

次にオブジェクトを要素に持つ集合の包摂関係を定義する. オブジェクト o_{ij} ($j = 1 \sim N_i$) を要素に持つ集合を O_i とする. O_1, O_2 間の包摂関係として様々な定義が考えられるが, 以下に代表的な2つを演繹規則として記述する.

$$\begin{aligned} O_1 \sqsubseteq_H O_2 &\leftarrow \forall s \in O_1 \exists t \in O_2 s \sqsubseteq t \\ O_1 \sqsubseteq_S O_2 &\leftarrow \forall t \in O_2 \exists s \in O_1 s \sqsubseteq t \end{aligned}$$

\sqsubseteq_H はいわゆる集合の Hoare 順序に, \sqsubseteq_S は Smyth 順序に等しい. 例えば, $b_1 \sqsubseteq b_2$ かつ $d_1 \sqsubseteq d_2$ とすると, $\{b_1, d_1\} \sqsubseteq_H \{a, b_2, c, d_2\}$, $\{a, b_1, c, d_1\} \sqsubseteq_S \{b_2, d_2\}$ が成り立つ.

集合要素間に選言^{*}の意味がある場合には Hoare 順序を, 連言^{**}の意味がある場合には Smyth 順序を用いる.

4. 多声音楽の表現

ピアノやギターで演奏される音楽は多声音楽であり, それらの譜面も, 通常, 多声音楽として記述されている. クラシックでは殆んどすべての楽曲が多声音楽として記述されており, ポピュラーでも多くの楽曲が多声音楽で記述されている. しかし従来の研究において開発された音楽システムの多くは, 多声音楽を直接的に分析や生成の対象にしていなかった. そこで本研究では GTTM に基づいて多声音楽を扱うことを目指す.

本章では, GTTM と DOOD に基づいて多声音楽を計算機上で直接表現するためのオブジェクトを設計する. この設計が満たすべき条件は, タイムスパン簡約が規定する多声音楽間の抽象化の関係と, DOOD によるオブジェクト間の包摂関係が適切に対応付けられることである.

4.1 タイムスパン木の表現

まず, C5 と E5 の2音から成る2つの旋律を考える (図2). タイムスパン木のノードを表現するオブ

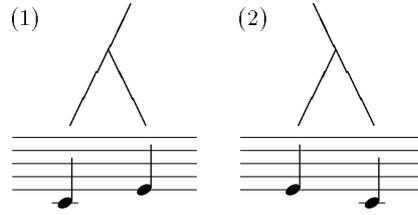


図2 簡単なタイムスパン木の例
Fig. 2 Examples of Simple Time-Span Trees

ジェクト ts に $head$, $primary$, $secondary$ の属性を持たせる. 最下端枝は $head$ 属性のみを持つ ts オブジェクトで表現する. これは要素が1つしかない特殊なグループに相当する. 図中 (1) のタイムスパン木は以下のようなオブジェクト K_α で表現する.

$ts(\text{head} : C5,$
 $primary : ts(\text{head} : C5),$
 $secondary : ts(\text{head} : E5))$

ここで, タイムスパン木において重要度の低い E5 の枝を削除して得られるタイムスパン木は以下のようなオブジェクト K_β で表現される.

$ts(\text{head} : C5)$

この時 $K_\alpha \sqsubseteq K_\beta$ が成り立つ.

ところが, この手法では, 図2 (2) のタイムスパン木のオブジェクトも K_α に等しくなってしまう. これは, $primary$ 属性と $secondary$ 属性間の時間関係が記述されていないからである. 従って, 時間関係の情報を記述する属性を付加しなければならない.

4.2 時間構造の満たすべき条件

我々は, 楽曲に含まれる音の生起時刻の情報を表現するために以下のような条件を課す.

- ある音の生起時刻はタイムスパン木においてより重要な音に対する差分で与えられる.
- 生起時刻の差分の情報を抽象化すると音間の順序の情報になる.
- 生起時刻や順序の情報を抽象化すると縮退した順序になる.

* Disjunction. 論理和とも言う.

** Conjunction. 論理積とも言う.

上の (a) は元のタイムスパン簡約において暗黙的に仮定されていた条件である。(b) は旋律に関して我々が持つ直観に由来する。例えば、音高の並びは等しいが各音の音長は異なるような 2 つの旋律がある場合、我々はこの 2 つの旋律はお互い似ていると感じるのであろう。これは、各音の並びという抽象的なレベルでも旋律を捉えているからである。(c) も我々が持つ直観に由来する。順序が縮退するとは、「音 a は音 b より先」という順序が「音 a は音 b より先あるいは同時」という緩い順序になることである。例えば、分散和音*を抽象化すると同じ構成音からなる和音が得られるだろう。この抽象化の前半、つまり分散和音の各音間の差分の情報が捨象されて順序になる段階は (b) が担っている。後半、つまり順序が 1 つの和音に縮退する段階は (c) が担っている。

4.3 時間構造の表現

前節の条件を満たすために、ある音の生起時刻を以下 4 つの属性を使って表現する。今着目している音の周囲にある音の中で、ある 1 つが先行音となり、別の 1 つが後続音となる。先行音か後続音のいずれかはタイムスパン木においてより重要な音であり、生起時刻はその音からの差分として指定される。この方針に従った時間構造の例を図 3 (1) に示す。図中 p, q, r は四分音符の音

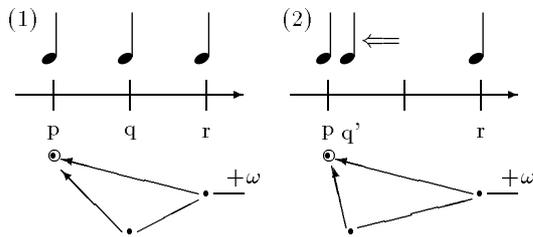


図 3 時間構造の例

Fig. 3 Examples of Temporal Structures

長を持つ音とする。まず p の生起時刻を旋律全体の基準と仮定する (図中 ⊙ で示す)。ここで便宜的に正無限大時刻 (+∞ と記す) と負無限大時刻 (-∞) を導入する。すると、r の先行音は p で、後続音 (の時刻) は +∞ である。r は p から 2 拍めの時刻に生起する。同様に音 q は先行音が p、後続音が r で、p から 1 拍めの時刻に生起する。図中、↖ の矢印が、その音にとってタイムスパン木においてより重要な音の方を指示している。

本論文では、生起時刻を表現するために上述の 4 つの

* その和音を構成する音が同時ではなく分散して奏されるようなもの。

属性を持つオブジェクト temp を導入し、先行音、後続音、より重要な音、差分を指定する属性として *pred*, *succ*, *salient*, *difference* を持たせる**。これより例えば、図 3 (1) 音 q の生起時刻を表現するオブジェクト T_q は

$\text{temp}(\text{pred}: T_p, \text{succ}: T_r,$

$\text{salient}: \text{pred}, \text{difference}: 1/4)$

と表せる (4.2 節条件 (a) を満たす)。ここで T_p , T_r は各々先行音 p、後続音 r の生起時刻を表現する temp オブジェクトである。 $T_p.\text{salient} = \text{pred}$ なので、音 q は p よりも後に、r よりも先に生起することが分かる。*difference* の 1/4 は四分音符長を表す。これに対し、差分の情報を捨象し順序の情報だけ残すと以下のようなオブジェクト T_s が得られる。

$\text{temp}(\text{pred}: T_p, \text{succ}: T_r,$

$\text{salient}: \text{pred}),$

すると $T_q \sqsubseteq T_s$ が成り立つ (4.2 節条件 (b) を満たす)。

次に、4.2 節条件 (c) の例として、図 3 (2) の音 q' を考える。q' にとって r より p の方がタイムスパン木においてより重要な音なので、q' が縮退できる音は p しかない。これより、我々は順序の縮退による抽象化の関係を表現する新しい包摂関係を導入する。

$T \sqsubseteq T.(T.\text{salient})$ ただし $T \sqsubseteq \text{temp}$

図 3 (2) の例では、(1) の q が p の時刻に縮退して q' となっている。今 $T_q.\text{salient} = \text{pred}$ なので、 $T_q \sqsubseteq T_q.\text{pred} = T_p$ となる (4.2 節条件 (c) を満たす)。

4.4 オブジェクト項の構文

タイムスパン木と時間構造を表現するオブジェクト項の定義を図 4 に示す。図中、 $\text{objclass}(\text{name}_1: \text{objclass}_1, \dots)$ の *objclass* はオブジェクトクラス名、*name*₁ は属性名、*objclass*₁ は属性値のオブジェクトクラス名あるいはデータ型という意味である。Rational は有理数を、Rational+ は正の有理数を、0 は基準時刻を、Pitch は C5 や E6 等の音高 (音名とオクターブ位置の組) を表す。x | y は x または y という意味を、{x} は x から成る集合を表す。

各音ごとの生起時刻を指示するために、ts オブジェクトに *at* 属性を導入し、その属性値を temp オブジェクトとする。chord オブジェクトの *notes* 属性は、和音を構成する個々の音の音高 (Pitch) を表す。notes

** 時間構造を表現するオブジェクトの構成法は幾つか考えられる。

(i) 旋律全体の基準時刻 (図 3 音 p の生起時刻) から時間構造のネットワークをトップダウンに辿るもの。(ii) 旋律に含まれる各音を根とするような木を考えボトムアップに辿るもの。本論文では (ii) を採用した。

```

ts(head: chord, at: temp | 0,
   primary: ts, secondary: ts)
chord(notes: {Pitch}, duration: Rational+)
temp(pred: temp | 0 | - $\omega$ , succ: temp | 0 | + $\omega$ ,
      salient: pred | succ,
      difference: Rational)

```

図4 オブジェクト項の定義

Fig. 4 Definitions of Object Terms

属性値が要素 1 つだけの集合の時それは単音と同等であり、和音の要素間の関係は連言的なのでその集合間の順序は Smyth 順序に従う。duration 属性は和音が鳴っている時間的長さ（音長）を表し、 $1/N$ は N 分音符長を意味する。temp オブジェクトの各属性の意味は前節を参照のこと。

例として、図5に我々の手法で表現した多声音楽のタイムスパン木と時間構造を示し、図6に P_1 （図5）のタイムスパン木と時間構造を表現する ts オブジェクトを示す。

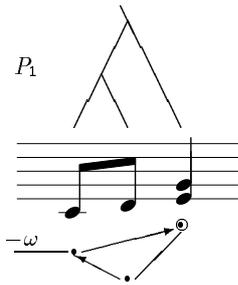


図5 タイムスパン木と時間構造の例

Fig. 5 Example of Time-Span Tree and Temporal Structure

ここで ts オブジェクトに関してコメントを加える。まず、包摂関係は半順序であり、我々の手法では is_a 関係を用いないので、ts オブジェクトの集合は包摂関係に関して束を構成できる。また、temp オブジェクトを抽象化するとそれを含む ts オブジェクトも抽象化される。よって、同じ音から構成されている多声音楽でも、より抽象的な時間構造を持つ多声音楽の方が抽象的になる。

4.5 強簡約仮説と多声音楽

ホモフォニーを対象とする GTTM は「強簡約仮説」を採用している¹⁷⁾。強簡約仮説は、(1) 楽曲は厳密に階層的に聴取されること、(2) 構造的に重要でない音は周囲のより重要な音と関連づけて聴取されることを主張する。ここでさらに (1) は、(1-1) 1 つの音は常に 1 つのグループに、そして 1 つのグループは常に 1 つの上位格

```

ts(head: K3,
   at: 0,
   primary: ts(head: K3,
               at: 0),
   secondary: ts(head: K1,
                 at: TC5,
                 primary: ts(head: K1,
                             at: TC5),
                 secondary: ts(head: K2,
                               at: TD5)))

```

$$K_1 = \text{chord}(\text{notes} : \{C5\}, \text{duration} : 1/8)$$

$$K_2 = \text{chord}(\text{notes} : \{D5\}, \text{duration} : 1/8)$$

$$K_3 = \text{chord}(\text{notes} : \{E5, G5\}, \text{duration} : 1/4)$$

$$TC_5 = \text{temp}(\text{pred} : -\omega, \text{succ} : 0, \\ \text{salient} : \text{succ}, \text{difference} : 1/4)$$

$$TD_5 = \text{temp}(\text{pred} : TC_5, \text{succ} : 0, \\ \text{salient} : \text{pred}, \text{difference} : 1/8)$$
図6 図5 P_1 の多声音楽を表現する ts オブジェクトFig. 6 Object Terms Representing Polyphony of P_1 in Fig.5

ループに属する、(1-2) 同じ階層の各グループの時間幅（タイムスパン）は重ならないという意味である。本手法では、グループを表す個々の ts オブジェクトに時間情報を記述する temp を at 属性として明示的に付加した。これより、時間幅が重畳したグループ（即ち多声音楽）も記述できるようになった。本手法は (1-1) と (2) に基づいて多声音楽を形式化したものと考えられる。

タイムスパン簡約が主張しているのは、簡約後に残る head はそのグループ全体に影響を及ぼすということだけである。その head の発音時刻をどのようにすると簡約後の楽曲に対する聴取者の認識が簡約前のそれに類似するかということには触れていない。head の発音時刻に関して、簡約後もそのままの時刻にしておく、あるいは、グループの先頭に移動する⁵⁾ の 2 通りが考えられる。上述したように、本手法が対象とする多声音楽では時間幅が重なるグループが存在する可能性があるため、head の発音時刻をグループの先頭に移動すると、元の楽曲中には含まれていなかった新たな関係が生じてしまう可能性がある。従って本手法では、簡約後に残る head の発音時刻は移動しないものとする。

5. 多声音楽に対する基本演算

ts オブジェクトの領域を基に完備束を構成できるので、ts オブジェクトに対する基本的な演算として包摂関係 (\sqsubseteq)、最小上界 (least upper bound, lub)、最大下界 (greatest lower bound, glb) が考えられる。各々の数学的な意味は良く知られている。

まず、包摂関係をタイムスパン簡約の立場から考える。包摂関係は、ある多声音楽とそれをよりタイムスバ

ン簡約した多声音楽を、或はタイムスパン木においてより精緻化した多声音楽を関係付ける。

5.1 最小上界

包摂関係を用いると lub は次のように定義される。オブジェクト x, y が与えられた時、 x と y の最小上界 $lub(x, y)$ とは $min(\{z | x \sqsubseteq z \wedge y \sqsubseteq z\})$ である。手続き的に定義すると次のようになる。

$$\begin{aligned}
 lub(a, b) &= \begin{cases} \top & a \neq b \\ a & a = b \end{cases} \\
 lub(o, o(l : v)) &= o \\
 lub(o(l : v_1), o(l : v_2)) &= o(l : lub(v_1, v_2))
 \end{aligned}$$

ただし、 a, b は基本オブジェクトであり、 \top はオブジェクトが構成する完備束の中の最大要素であり、 $o(l : \top) = o$ である。

lub は、2つの多声音楽のタイムスパン木をトップダウンに見て、最も大きく共通する部分を取り出す。それは、2つの多声音楽からタイムスパン木において重要な音を順に削除し、はじめて同じ形になった時点での多声音楽である。

図7に、多声音楽 P_1, P_2 に対する lub の計算結果を示す。この lub の計算において、属性情報が十分に

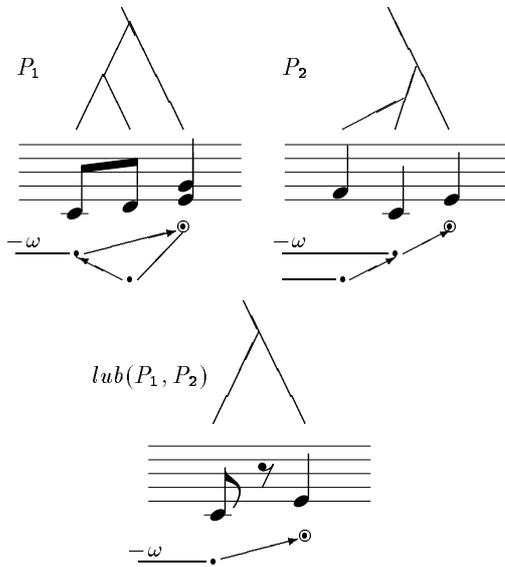


図7 lub の計算例
Fig. 7 Example of Calculating lub

揃っていない不完全なオブジェクトは削除した。 P_1 の T_{C5} と P_2 の T_{C5} を表現する temp オブジェクトは等価である。 chord オブジェクトの $duration$ 属性に関して、 $lub(d_1, d_2) = d_1$ if $d_1 < d_2$ と仮定したので、

$lub(P_1, P_2)$ の $C5$ の $duration$ は $lub(1/4, 1/8) = 1/8$ となる。

5.2 最大下界

lub と同様に、 glb も包摂関係と \perp (最小要素) を用いて定義できる。ただし $o(l : \perp) = \perp$ である。

glb は2つの多声音楽のタイムスパン木をトップダウンに見て、2つの多声音楽の各音の時間構造が矛盾しない限りタイムスパン木を重ね合わせていく。もし、どちらかの多声音楽の末端ノードまで到達したら、もう一方の残りの多声音楽をその場所に展開する。例えば生起時刻を表現する temp オブジェクトどうしの glb が \perp になると、上の定義より、 glb 計算結果全体が \perp になる。

図8に、多声音楽 P_1 と P_3 に対する glb の計算結果を示す。この glb の計算において、 chord オブジェクト

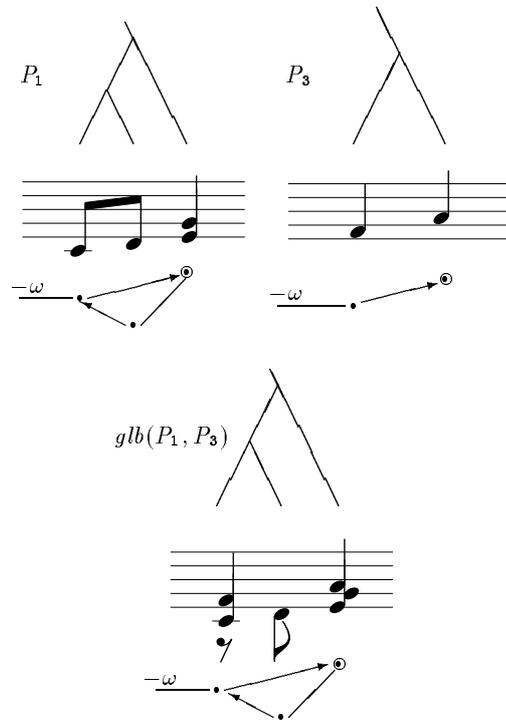


図8 glb の計算例
Fig. 8 Example of Calculating glb

の $notes$ 属性に関して $glb(n_1, n_2) = n_2 \cup n_1$ とし、 $duration$ 属性に関して $glb(d_1, d_2) = d_2$ if $d_1 < d_2$ とした (従って $glb(1/4, 1/8) = 1/4$)。尚、 $glb(P_1, P_2) = \perp$ であることに注意。

6. 議 論

6.1 関連研究

GTTM を利用しない音楽記述手法には, Balaban の音楽オントロジー²⁾, Cope の signature/SPEAC³⁾, Dannenberg の Nyquist⁴⁾ 等の研究事例がある. これらは基本的に五線譜上のある音がどの音と隣接して出現しているかという表層の関係を記述している. ユーザはこれらの関係を用い, 独自の音楽理論に基づいて旋律や和声のパターン照合機能やパターン合成機能を実現する. しかし音楽的に妥当な分析や生成を実現するには, 音どうしの表層的な関係をなぞるだけでは不十分であり, 豊富な背景知識を必要とする²¹⁾. 従って, 上記の音楽記述手法上に構築された音楽システムは, 音楽家, ジャンル, 年代, 地域等の文脈や状況に大きく依存し, 適正に動作する範囲が極めて限定的になってしまう.

我々の手法は音楽理論 (GTTM のタイムスパン簡約) に基づいているので, これが豊富な背景知識の役割を果たし, 文脈依存度が減少し適切に動作する範囲が広がる. GTTM のタイムスパン簡約に基づいているので, 音楽タスクを実現するために提供されるインタフェースは包摂関係, *lub*, *glb* となる.

一方, GTTM を利用した研究事例には, Widmer の演奏表情付け学習システム²¹⁾, Arcos らの CBR を用いた演奏表情付けシステム¹⁾, Uwabu らのコーパスからのプロロンゲーション簡約の学習²⁰⁾ などがある. Widmer と Arcos は知識表現手法そのものに主眼を置いておらず, GTTM の分析結果だけがアプリオリに与えられている. Uwabu ではタイムスパン木やプロロンゲーション木に関する演算については触れられていない. 従ってここで比較検討は行わない.

6.2 利 点

GTTM に基づく形式的な音楽知識表現手法の利点は, 音楽アプリケーションや音楽アルゴリズムを体系的に設計できるということである. 例として, 類似度を判定するアルゴリズムと編曲のアルゴリズムを挙げる.

まず, 類似度を判定するアルゴリズムとは, 多声音楽 x, y, z が与えられている時, x は y と z のどちらにより類似しているかを判定するものである⁸⁾. $lub(x, y)$ と $lub(x, z)$ を包摂関係によって比較し, $glb(x, y)$ と $glb(x, z)$ も同様に比較する. lub だけでは判定できない場合, glb も用いることで, より精密な判定が可能になった.

次に, 編曲のアルゴリズムとは, 写像を模倣する問題として編曲タスクを形式化したものである⁷⁾. 我々の問題設定では, 未知の入力課題曲 X 及びある課題曲 E

とその編曲結果 A から成る編曲事例を入力として与える (ここで使用する事例は 1 つである). すると, X を編曲するとは, $f(E) = A$ を満たすような写像 f に対して, $f(X)$ の値を推定することである. そのような f は多数存在するが, 我々は音楽的に合理的であるために幾つかの条件を課し, $f(X) = glb(lub(A, X), E)$ という近似ができることを示した. よって解は一意に求まる. このような写像の設計が可能になったのは, 多声音楽をオブジェクトとして表現し, 数学的な計算対象として形式化できたからである.

同様の音楽アプリケーションを, 例えば Nyquist で実現しようとする, 音楽アプリケーション毎に多くのヒューリスティクスやアドホックな特徴空間を導入する必要が生じ, 実現された関数は適用範囲が限定されたものになるであろう. 従って, 体系的な音楽アプリケーション設計は困難であると思われる.

6.3 問題点

我々の手法には以下 2 つの問題点がある: (i) タイムスパン木と時間構造を付加するためには音楽の専門知識及び GTTM の理解が必要であり, (ii) それら深層構造に関する情報を付加する作業がユーザにとって負担となる.

まず (ii) に対しては専用エディタによる対処を考えている (図 9)⁷⁾. このエディタにより *head*, *primary*, *secondary* の関係, 時間構造における *salient*, *difference* の情報を指定することができる. 図中, ユーザは GTTM のタイムスパン規則により α の枝が β の枝に従属していると考え, α の枝を β に貼付けている. この時, β が *primary* に, α が *secondary* に, γ が *head* に対応する. こうして, ユーザは自分の多声音楽の解釈をタイムスパン木と時間構造として効率的に指示することが可能となる.

次に, 音楽の専門知識を持たない多くの人々にとって (i) が障害となる. この解決のため, 我々は GTTM に基づく分析の自動化に取り組んでいる¹³⁾. この自動化により, ユーザは必要最小限だけシステム動作に介入すれば良い. ただし, ユーザ意図に沿わない自動化や過度の自動化は, ユーザにとって制御性の低下を招きやすい. そこで, 対話性を高めたり, システムの内部状態を効果的に説明する機能を付加することも検討している.

7. ま と め

本論文では, 音楽理論を援用し, 音楽が持つ意味を考慮に入れた音楽知識表現手法を提案した. 我々の手法は, 音楽的な直観を素直に表現でき, 不完全ではあるが形式的な意味も定義することができた. 本手法が適切に

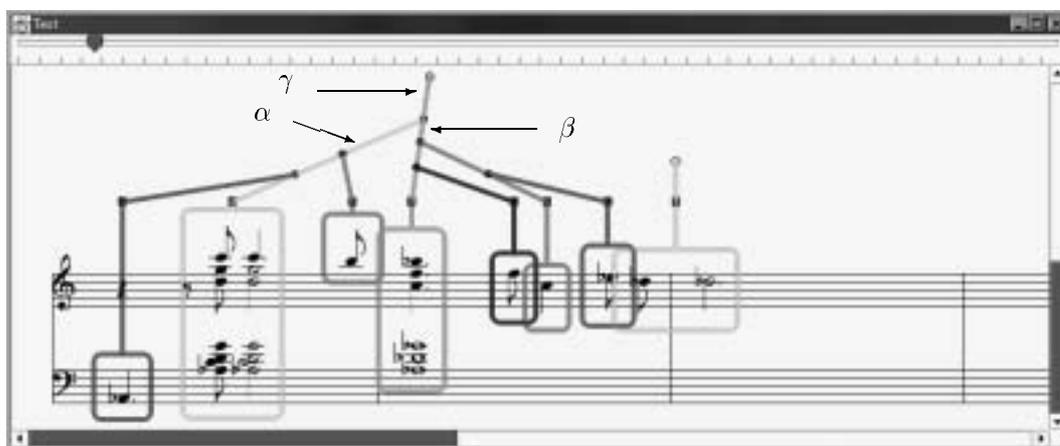


図9 現在開発中のエディタのウィンドウ
Fig. 9 Window of The Editor Under Development

機能している理由は、確立された音楽理論 (GTTM) に依拠したためであると考えられる。

しかし、現在の ts オブジェクトと temp オブジェクトには、一部重複して表現されている情報があるので、継承のメカニズム等をうまく導入して表現法をできるだけ効率化する必要がある。また、より正確な表現を得るためには、音楽を多角的に分析するのが有効であろう。これは、さらに別の音楽理論を援用することを意味し、例えば、プロロンゲーション簡約や Narmour の暗意-実現 (Implication-Realization) モデルの援用が考えられる。

音楽は人間の知識活動の所産である。音楽学は、計算機という前提を置かずに、長年音楽における人間の知的活動を記述し分析してきた。しかし、今後の技術や文化の発展を考えると、音楽知識を計算機上で表現し操作する必要性はますます高まるものと思われる。本研究が今後の音楽学と計算機科学の橋渡しとなれば幸いである。

参考文献

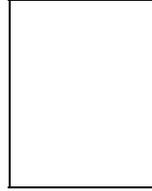
- 1) Arcos, J. L. and de Mántaras, R. L.: Combining AI Techniques to Perform Expressive Music by Imitation, *AAAI Workshop: Artificial Intelligence and Music*, pp. 41-47 (2000).
- 2) Balaban, M.: The Music Structures Approach to Knowledge Representation for Music Processing, *Computer Music Journal*, Vol. 20, No. 2, pp. 96-111 (1996).
- 3) Cope, D.: *Experiments in Musical Intelligence*, A-R Editions, Inc. (1996).
- 4) Dannenberg, R. B.: Machine Tongues XIX: Nyquist, a Language for Composition and Sound Synthesis, *Computer Music Journal*, Vol. 21, No. 3, pp. 50-60 (1997).
- 5) Dibben, N.: The Cognitive Reality of Hierarchic Structure in Tonal and Atonal Music, *Music Perception*, Vol. 12, No. 1, pp. 1-25 (1994).
- 6) 後藤真孝, 平田圭二: ハービー君: 演繹オブジェクト指向に基づいてジャズらしいコードにリハーモナイズするシステム, 情報処理学会研究報告, Vol. 96, No. 75, pp. 33-38 (1996). 96-MUS-16.
- 7) 平田圭二, 青柳龍也: パービーブン: 音符レベルでユーザ意図を把握して編曲を行う事例ベースシステム, 情報処理学会研究報告, Vol. 2000, No. 94, pp. 17-23 (2000). 2000-MUS-37.
- 8) 平田圭二, 青柳龍也: パービーブン: ジャズ和音を生成する創作支援ツール, 情報処理学会論文誌, Vol. 42, No. 3, pp. 633-641 (2001).
- 9) 平田圭二, 後藤真孝: 音楽システムを考える, 情報処理学会研究報告, Vol. 2001, No. 45, pp. 47-54 (2001). 2001-MUS-40.
- 10) 平田圭二, 平賀瑠美: 演繹オブジェクト指向データベースを用いた演奏生成アプリケーションオントロジー, 人工知能学会誌, Vol. 14, No. 6, pp. 1033-1040 (1999).
- 11) Hirata, K. and Hiraga, R.: Next Generation Performance Rendering - Exploiting Controllability, *Proc. International Computer Music Conference, ICMA*, pp. 360-363 (2000).
- 12) 平田圭二, 平賀瑠美: ハービーブン: 2 段階演奏表情付け法によるインクリメンタルな演奏生成システム, 情報処理学会研究報告, Vol. 2001, No. 16, pp. 19-26 (2001). 2001-MUS-39.
- 13) 井田健太郎, 平田圭二, 東条敏: GTTM に基づくグルーピング構造及び拍節構造の自動分析の試み, 情報処理学会研究報告, Vol. 2001, No. 103, pp. 49-54 (2001). 2001-MUS-42.
- 14) Kifer, M., Lausen, G. and Wu, J.: Logical

Foundations of Object-Oriented and Frame-Based Languages, *Journal of ACM*, Vol. 42, No. 4, pp. 741–843 (1995).

- 15) 小暮潔: 素性構造 (1) 及び (2), 人工知能学会誌, Vol. 8, No. 2&3, pp. 184–191&305–311 (1993).
- 16) Kolodner, J.: *Case-Based Reasoning*, Morgan Kaufmann (1993).
- 17) Lerdahl, F. and Jackendoff, R.: *A Generative Theory of Tonal Music*, The MIT Press (1983).
- 18) Russell, S. and Norvig, P.: エージェントアプローチ 人工知能, 共立出版 (1997). 古川康一監訳.
- 19) 竹内好宏: 音楽の構造解析とその応用, bit 別冊 コンピュータと音楽の世界, 共立出版, pp. 224–240 (1998).
- 20) Uwabu, Y., Katayose, H. and Inokuchi, S.: A Structural Analysis Tool for Expressive Performance, *Proc. International Computer Music Conference*, ICMA, pp. 121–124 (1997).
- 21) Widmer, G.: Learning Expressive Performance: The Structure-Level Approach, *Journal of New Music Research*, Vol. 25, pp. 179–205 (1996).
- 22) 横田一正: 演繹オブジェクト指向データベースについて, コンピュータソフトウェア, Vol. 9, No. 4, pp. 3–18 (1992).

(平成13年6月15日受付)

(平成13年12月18日採録)



平田 圭二 (正会員)

昭和33年生. 1987年 東京大学情報工学専門課程博士課程終了. 工学博士. 同年 NTT 基礎研究所. 1990年から1993年 (財) 新世代コンピュータ技術開発機構 (ICOT). 1999年より現職. 音楽知プログラミング及び並行論理型プログラミングの研究に従事. 日本ソフトウェア科学会会員, KLIC 協会理事. 共訳書「コンピュータ音楽 - 歴史・テクノロジー・アート」(東京電機大学出版局).



青柳 龍也 (正会員)

昭和35年生. 1988年 東京大学情報工学専門課程博士課程終了. 同年電気通信大学助手. 音楽情報処理及びCAIの研究に従事. 1998年より津田塾大学助教授.