

GTTMに基づく音楽表現手法再考

平田 圭二

平賀 譲

NTT コミュニケーション科学基礎研究所

図書館情報大学

hirata@brl.ntt.co.jp

hiraga@ulis.ac.jp

あらまし

本稿では、先に平田、青柳が提案した GTTM と DOOD に基づく音楽知識表現法をさらに改良、発展させるために、現在検討を加えている事項について述べた。提案された音楽知識表現法は有望な方向性を示すことはできたが、理論的にはまだ不備な点が幾つか残っている。そこで本稿では、まず本音楽知識表現法の設計原理に触れたあと、本法の抱える課題、問題点を整理、議論する。具体的な論点は、多声音楽の定義、タイムスパン簡約の解釈、音価と包摂関係、時間構造の冗長性などである。

Revisiting Music Representation Method based on GTTM

Keiji Hirata

NTT Communication Science Laboratories

Yuzuru Hiraga

University of Library and Information Science

Abstract

In a previous paper, we proposed a music representation method based on GTTM and DOOD. This paper discusses some key issues of the method, which would lead to its further improvement and enhancement. We think our method could prove to be on a right direction, whereas there are several issues remaining. In this paper, we mention the design principles of our method, clarify the issues and discuss how they can be resolved. Specifically, the issues include the definition of polyphony, interpretation of time-span reduction, duration and the subsumption relation, and redundancy of temporal structure.

1 はじめに

音楽の情報学的な意味を正しく解釈したり合理的な音楽システムを構築しようとするとき、音楽知識を計算機上でいかに表現するかという研究課題に辿りつく。この音楽知識表現という研究課題は、音楽理論、計算機科学、音楽認知に渡る幅広い知見と詳細な検討を必要とし、取り組むには非常に敷居が高い課題の一つと言ってよいだろう。また、この研究課題は唯一の正しい解を見出すような性質の課題でもないため、成果の評価も難しいと思われる。音楽知識表現という研究分野の重要性は言うまでもないが、現在提案されている音楽知識表現法の多くは、楽譜上の表層的な関係を記述するにとどまっていたり、音楽的な根拠が希薄なまま他領域の技法を無批判に適用しているように見受けられる。残念ながら研究分野としてまだ未成熟と言わざるを得ない。

平田と青柳は文献2で GTTM と DOOD に基づく音楽知識表現法(本稿では HA 法と呼ぶ)を提案した。当該研究分野において有望な方向の一つを示す試みではあったが、その理論的な考察にはまだ不十分な点があるので平賀がコメントを加えた。我々は、それらを下地として本稿を執筆する。

本稿での検討課題には以下の3点がある。各々関係する節を記す。

- (i) 旋律、和音、拍節などの対象をどう理解、解釈するか。対象が持つ本来的な意味を良く吟味し、

直観に合致するモデルを構築することが必要である。基本的に我々は GTTM に依拠している(3.1節)。

- (ii) その理解や解釈を計算機上の記号を用いた表現(項)にどう翻訳するか。多くの場合をできるだけ正確に効率良く記述することが必要である(3.2節, 3.3節, 3.4節, 4章)。
- (iii) 項をどのように操作するか。必要最小限で十分強力な表現力を持つ操作群を設定し、それら操作の持つ意味をきちんと理解する必要がある(3.3節, 4章)。

上の検討課題に関して、各節において HA 法の問題点を指摘し、論点を整理し、解決への糸口を与えようと思う。

続く第2章では、HA法が依拠する GTTM と DOOD に簡単に触れて、HA法自体を説明する。そして第3章、第4章で上記の研究課題について議論する。

2 HA法の概観

Generative Theory of Tonal Music (GTTM)[6]はLerdahlとJackendoffによって提案された音楽理論であり、計算機への実装に関して最も有望視されているものの一つである。Deductive Object-Oriented Database

(DOOD)[7] は知識表現法の一つであり、理論的基礎が確立されており形式的取り扱いが容易である。

2.1 GTTM

GTTM は聴取者の直観を形式化する音楽理論であり、いわゆる五線譜に書かれた西洋調性音楽を分析し解釈するための概念や手続きを提供している。五線譜上に書かれた楽曲は表層的な情報であり、GTTM はそこから暗黙的で深層的な情報を導き出す。Chomsky の変形生成文法の枠組にならったと言われている。GTTM が扱うのはホモフォニー¹に限定されている。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 [8; 9]。

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。結果として、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ことが多いだろう。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。タイムスパン簡約の結果はタイムスパン木として表現され (図 1)、タイムスパン木は GTTM の意味で楽曲の深層構造を表現している。本稿ではプロロンゲーション簡約は扱わない。

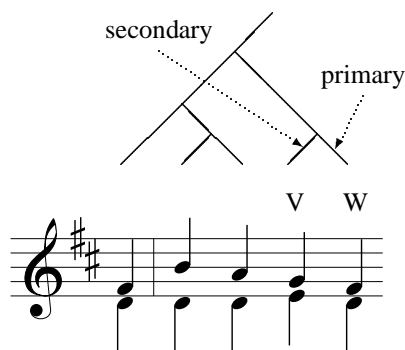


Figure 1: ホモフォニーのタイムスパン木の例

¹Homophony. 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。

タイムスパン木は、隣接する音や和音が階層的に作るグループを表現している。タイムスパン木をボトムアップに辿ると、重要でない音/和音から簡約 (削除) されていく様子が分かり、逆にトップダウンに辿ると重要な音/和音から精緻化 (elaborate) されていく様子が分かる。あるレベルのグループを代表する音/和音を head と言う (head はグループに含まれる音/和音から構成される)。タイムスパン木は基本的に二分木であり、本稿では、音楽構造的に重要な音/和音を含む枝を primary、そうでない方の枝を secondary と呼ぶ。

図 1 では primary 枝の下に 4 拍めの和音 W が出現し、secondary 枝の下に 3 拍めの和音 V が出現する。V と W は隣接しているのでグループを作り、グループの head は W 自身である。

2.2 DOOD

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 [7]。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ある項と抽象的な項を形式的に結び付けることができる。また DOOD は素性構造 [5] とも関連が深い。

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。本稿では、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。

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。オブジェクト項間の包摂関係は次のように定義される。基本項 p, q を比較し、 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ならば、 o_1 の方が具体的 (あるいは o_2 の方が抽象的) となる。属性の少ないオブジェクト項は包摂関係に関してより抽象的である。

オブジェクト項 x, y が与えられた時、 x と y の最小上界 (least upper bound, lub) は $\min(\{z \mid x \sqsubseteq z \wedge y \sqsubseteq z\})$ と定義され $lub(x, y)$ と書く。lub の直観的な意味は、ある 2 つのオブジェクト項の最大の共通部分であり、積集合を計算するイメージである。最大下界 (greatest lower bound, glb) も同様に定義される。

2.3 多声音楽の表現

HA 法は多声音楽²をタイムスパン簡約を応用して表現する手法であり、本節ではその HA 法によるオブジェクト項を紹介する。

多声音楽には様々な音楽的な構造が内包されているが、その内、HA 法のオブジェクト項はグルーピング構造と時間構造を顕在化する。グルーピング構造はタイムスパン簡約に基づく。時間構造は、音長を分析・合成するために、タイムスパン簡約に付加する

²Polyphony. 複数の独立した声部を持つ音楽のこと。

形で提案されたものである³。このオブジェクト項表現には、オブジェクト項の抽象化/具体化が多声音楽の簡約化/精緻化にうまく対応することが要請されている。HA法はこの対応が最も自然であるという原理に基づいている。

グルーピング構造 HA法による多声音楽の記述例を図2に示す。ここで $e_1 \sim e_6$ は音/和音の識別子である。図中、○の付いた枝を実際に ts オブジェクト

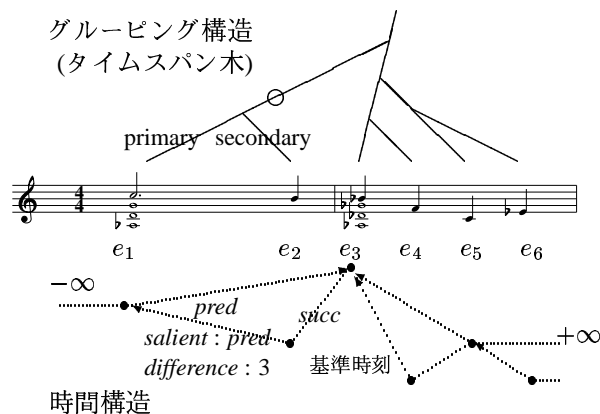


Figure 2: HA法によるグルーピング構造と時間構造の記述例

項と chord オブジェクト項で表現すると下のようになる。

```
ts(head : e1,
   at : t1,
   primary : ts(head : e1,
                at : t1),
   secondary : ts(head : e2,
                  at : t2))
```

```
e1 = chord(notes : {56,62,67,72},
           duration : 360,
           velocity : 60)
```

ここで、あるグループにおける salient な音は head 属性によって表され、notes 属性値は MIDI ノートナンバーの集合であり、duration 属性値は 1 小節 = 480 ticks とした時の値である (360 ticks は 3 拍)。

時間構造 図2下半分は、同じ多声音楽の持つ時間構造を表している。 t_n ($n = 1 \sim 6$) は e_n の発音時刻を表現するオブジェクト項とする。HA法の時間構造は、楽曲に含まれる音の生起時刻の情報を表現する際、以下の条件を満たすよう設計されている。

- (a) ある音の生起時刻はタイムスパン木において、その枝が secondary になるノードの primary の音との差分で与えられる。

³文献3で提案された音楽知識表現法 FRM も、GTTM に対する形式的表現の枠組として使用できるだろう。

- (b) 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。
- (c) 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。

図2において、まず和音 e_3 の発音時刻 t_3 を基準とする。次に和音 e_1 に着目すると、 e_1 は e_3 と $-\infty$ (負の無限大時刻) の間に発音し (順序に関する情報)、 e_3 から 4 拍めの時刻に発音する (時刻に関する定量的な情報、図中 $\leftarrow \dots$ で示す)。 e_1 の発音時刻 t_1 は下左の temp オブジェクト項で表現される。

```
t1 = temp(pred : -∞,          t2 = temp(pred : t1,
      succ : 0, (基準時刻)      succ : t3,
      salient : succ,          salient : pred,
      difference : 4)          difference : 3)
```

同様に、 e_2 は e_1 (pred) と e_3 (succ) の間に発音し、 e_2 にとってより重要な音は e_1 なので (salient : pred), e_1 から 3 拍めの時刻で発音する (difference : 3)。ここで、グルーピング構造における head 属性値と時間構造における salient 属性値は矛盾なく設定されている必要がある。実際、この時間構造は上記 (a)~(c) の要請を満たす。基準時刻を secondary 枝の音の発音時刻とすることも可能である。詳細は文献2を参照のこと。

3 HA法の課題

HA法の抱える課題について議論する。

以下、本稿では旋律を MML 的に記述する。単音はオクターブを省略し音名のみ C, D#, Gb のように書く。音価は C4, D#8, r16 のように連続して書く (4, 8, 16 はそれぞれ四分音符、八分音符、16分休符の音価を意味する)。和音は {C,E,G} あるいは {C,E,G}4 (四分音符の C, E, G から成る和音) と書く。音列は [G8 A8 {B, C}4 D16] のように書く。タイムスパン木の簡略記法を導入する。タイムスパン木のあるノードは (h l r) で表され、h が head, l が左枝, r が右枝を表す。また、本稿では簡約化/精緻化は抽象化/具体化と各々同義とする。

3.1 多声音楽の定義

問題点: HA法では、新に多声音楽のためのタイムスパン簡約相当のものを設計する必要があった (GTTMのタイムスパン簡約は元々ホモフォニーに対して定義されている)。しかし、多声音楽の形式的な定義を与えずに HA法を設計したため、多声音楽の簡約(抽象化)法として合理的な根拠を与えることができなかった。

多声音楽の定義として、音楽辞典には「時間的重畳を許す独立したホモフォニーあるいは多声音楽の階層的な集合」や「互いに独立であるが和音としても響くような複数の旋律が織り重なるテクスチャ」といった定義が記載されている。しかし計算機の立場からするとまだまだ形式化は不十分である。

タイムスパン簡約的視点に立つ多声音楽の形式的な定義において意見が分かれるのは、時間的に重なる音/和音/旋律の解釈である。

- HA 法では、時間的に重なる音/和音/旋律間にもタイムスパン的な順序があると考えた。
- Lerdahl の方法 [4] では、多声音楽は元々ホモフォニーであって、その声部のある一部分が時間方向に少しずれていると考えた。
- もう一つの可能性として、時間的に重なる音/和音/旋律を、そのまま順序が付かない独立した旋律であると考えられる方法もある。

以下、各々について説明する。

HA 法では、多声音楽をまずホモフォニーの集合に分解し、近接しているホモフォニー同士に重要度に関して順序を付けタイムスパン木として解釈した(図 3)。図中、四角で囲まれた部分がホモフォニーであり、楽曲全体は時間的に重畳するホモフォニーの集合として表現される。図右、1 拍めの C と Eb は時間的に重畳するホモフォニーの head であり、C と Eb の間にタイムスパン簡約としての順序が付いている。

次に、Lerdahl は文献 4 で多声音楽に対するタイムスパン簡約の拡張法を提案しており、そこから間接的に多声音楽の定義を知ることができる。まず HA 法と同様にホモフォニーの集合に分解した後、重要なホモフォニーに近接している他のホモフォニーを時間方向に適当に少しずらし整理させる。こうして多声音楽を部分的にホモフォニー化して GTTM のタイムスパン簡約を適用する。これを本稿では「整理によるホモフォニー化」と呼ぼう。しかし、整理のためにわずら幅が大きくなると、もともとがホモフォニーであったという仮定が不自然になる。実際、「カエルの歌」の輪唱の場合、分解して得られたホモフォニーを整理させるためには最小でも 4 拍ずらす必要がある。

上の 2 つの方法は、いずれも、同時に発音された音に対し何らかの順序を付けてある 1 つの要素で代表させる手法であった。これに対し、3 番目の多声音楽の解釈は、より直観的な面がある。例えば同時に 2 音だけ鳴る多声音楽において、この 2 音を順序が付かない独立した旋律であると考えれば、それは 2 つのタイムスパン木が輻輳しているような状況と見なせる。輻輳は和音と対極にある概念である。本質的に順序の無いところに無理に順序を付けようとしなければ、曖昧性も生じない。

解決へ向けて： 直観的には、いずれの手法にもある程度の合理性はあるように思える。むしろ多声音楽にはこれらの解釈が並立するような曖昧性や多義性が内包されているとさえ思える。よって、唯一標準的な形式的な定義を与えることを目指すよりは、ここでは触れなかった定義(例えば Narmour の暗意一実現モデル)なども並列させる枠組み作りを目指す方が賢明ではないかと考えている。

3.2 タイムスパン簡約の解釈

問題点： GTTM のタイムスパン簡約は重要でない音を削除し重要な音を残す。その簡約された音列は、実際の音列(旋律)そのものではなく、実際の音列と何らかの対応関係にあると解釈すべきものである。しかし HA 法では、タイムスパン簡約した音列を実際の音列と同一視し、HA 法自体に根拠を与えようとしたため、対応関係を明らかにすることに注意が向けられていなかった。

例えば、きらきら星変奏曲の主題と変奏の関係を考える。主題は [C4 C4 G4 G4 A4 A4 G4...] であり、第一変奏は [D16 C16 B16 C16 B16 C16 B16 C16 A16 G16 F#16 G16 F#16 G16 F#16 G16...] である。タイムスパン簡約は、この変奏を簡約すると(装飾音を取り除くと)主題の音列に一致するという関係を明らかにする。つまり、タイムスパン簡約の主張は、変奏を簡約して得た楽曲構造的に重要な音(骨格的な音)は、それが抽出されたタイムスパンの全長を支配する、ということである。ここで、あえて骨格的な音を(実際の音として)問題にするなら、骨格的な音は主題の各音と一致するので、発音時刻 = タイムスパンの開始時点、音価 = タイムスパンの全長、とするのが適当であると考えられる。しかし、GTTM のタイムスパン簡約では、飽くまで、骨格的な音の列は実際の音列とは別物と考えている。とは言うものの、Dibben の聴取実験 [1] では、骨格的な音を実際の音として被験者に聴取させて、肯定的な結果を得ている。

一方、HA 法は、簡約途中の音列も実際の音列として取り扱うことを目標としている。そのため、各音の発音時刻と音価の情報を簡約後も保存するような簡約手順を設計した。すると、簡約は、装飾音を単純に取り除き骨格的な音だけ残すという操作になる。例えば、第一変奏を HA 法で簡約すると [C16 r8. C16 r8. G16 r8. G16 r8. ...] となる(ここで 8. は符点八分の意味)。これを主題と比較すると、音価が短く、出現する位置が小節中のあちこちに散らばり聴こえが一致しないのではないかという懸念が生じる(図 3 参照でも同様の状況になっている)。これは今後とも検討を要すると思われる。

また、形式化する観点から考えると、HA 法では、各音の発音時刻と音価の情報を簡約後も保存するので、抽象化/具体化の際、項の関連する部分だけを削除/付加すれば良く、関連の無い部分は変更しない。この性質は、形式的な定義を与える上では好ましい。

これまでの結論： GTTM のタイムスパン簡約途中の音と実際の音の対応関係を整理する。GTTM のタイムスパン簡約に素直な方法(以下 LJ 法と呼ぶ)では、

(実音の)発音時刻 = タイムスパンの開始時刻、
(実音の)音価 = タイムスパンの全長、

となる。HA 法では、

発音時刻 = 骨格音の開始時点、
音価 = 骨格音全長、

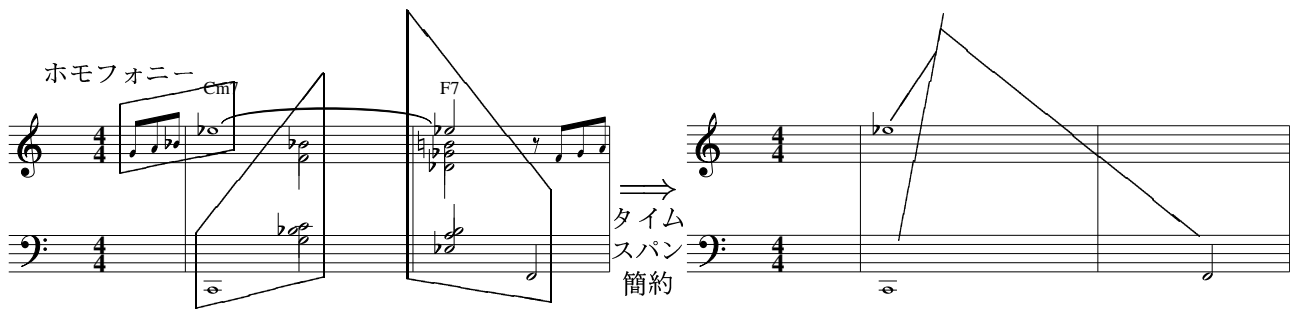


Figure 3: HA法における多声音楽のタイムスパン簡約

となる。

3.3 音価と包摂関係

簡約途中の音と実際の音の対応関係をさらに検討するために、音価と包摂関係との関連に注目する。

問題点: LJ法では音価が大きくなると抽象化されると考えるのに対し、HA法では逆に音価が小さくなると抽象化されると考える。

GTTMではタイムスパンが大きくなるほど簡約が進み抽象化されると考え、LJ法では、実音の音価 = タイムスパンの全長であるから、音価が大きいほど抽象化されると考える (例えば $\langle C8 \rangle \sqsubseteq \langle C4 \rangle$)。これは、タイムスパンが小さくなるほど、発音時間が特定されるので具体的になるとも見なせる。あるいは一定の時間区間をどう分割するかという、分割のための情報量という観点からでも説明できる。つまり、休符も一定時間を占有するという意味で、休符にも音符と同様の存在性を認めるのであれば、より不等分に分割する方が情報量は増える (より具体化する) ことになる。

一方HA法では、音価が小さいほど抽象化されるという逆の前提に立つ。この前提を導入した理由は、音符数が減ると抽象化されることと、音価を限りなく短くしていった極限を音符数ゼロと同等に見なしたことによる。これは、音符そのものが持つ情報量だけ考えるという観点に立っている。HA法では各音符に対する発音時刻と音価を別々に指定するので、たまたま音の鳴っていない区間が生じた時、結果としてその区間を休符のように見なすこともできる (SMFに似ている)。

音価と包摂関係の対応の差が最も顕著に反映されるのが多声音楽を表現するオブジェクト項に対する基本演算 lub と glb である。 lub は共通構造を抽出するものであり類似性照合などに用いられ、 glb は重ね合わせ (合体) を行う。標準的に、 lub と glb は次の条件 $lub(\top, x) = \top$, $lub(\perp, x) = x$, $glb(\perp, x) = \perp$, $glb(\top, x) = x$ を満たすように定義される (ここで \perp は最小要素である)。

オブジェクト項どうしの lub や glb を計算する際、特に音価 (duration 属性値) に関して 2 通りの定義を与えることができる。以下 a , b を音価とする。

LJ法	$lub(a, b) = \max(a, b)$
	$glb(a, b) = \min(a, b)$
HA法	$lub(a, b) = \min(a, b)$
	$glb(a, b) = \max(a, b)$

解決へ向けて: この問題についてはまだ結論が出る段階には達していないので、継続的に検討を続けたい。

3.4 時間構造の冗長性

問題点: HA法で導入した時間構造は、常にタイムスパン木と整合していなければならない。そのため記述が冗長になり、整合を取るためのコストが必要になってしまった。

HA法では、楽曲における音/和音の時刻を表現するため、時間情報の表現に要請される 3 つの条件を満たすよう時間構造を設計した (2.3 節)。基本的な考え方は、今着目している音は先行音 (predecessor) と後続音 (successor) の間に生起し、そのどちらか重要な音 (タイムスパン木情報から得る) を基準に時刻が定まるというものである (このような楽曲における音の時刻を表現する $temp$ オブジェクトの詳細については文献 2 を参照のこと)。HA法では、多声音楽を表現するために、タイムスパン木の各ノードごとに at 属性値としてこの $temp$ オブジェクトを設定して時刻を指定する。しかし、タイムスパン木情報から得る重要な音の情報が、常にタイムスパン木と整合している必要がある。

例として、 $[A8 B8 C8 D8 \dots F8 G8]$ という音列とそのタイムスパン木を考える (図 4 上半分)。このタイムスパン木では、A8 が secondary になるノードの primary は G8 であり (A8 は G8 に従属する)、B8 ~ F8 はすべて A8 に従属する。G8 の発音時刻が基準時刻である。図 4 下半分は、時間構造を表現している (もちろんこれ以外の解釈も有り得て、その場合は異なるタイムスパン木、時間構造になる)。B8 ~ F8 の $pred$ 属性はすべて A8 になっているが、これはタイムスパン木と重複した情報である。一方、B8 の $succ$ 属性が C8、C8 の $succ$ 属性が D8、 \dots , F8 の $succ$ 属性が G8 となるのは、 $temp$ オブジェクトでのみ表現できる情報である。ここで例えば、C8 の先行音が B8 であるという情報は、B8 の後続音が C8 であるという関係で表現されている。

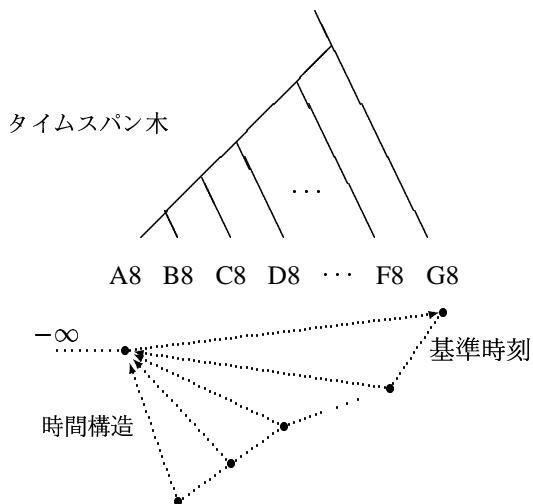


Figure 4: タイムスパン木と時間構造の対応関係

解決へ向けて: タイムスパン木と重複する *salient* 属性の情報を、項の構造を工夫したり何らかの継承機能等を活用し効率良く記述する方法を検討中である。

4 その他の課題

本章では、残りの未検討な課題を挙げる。

高次の照合: 先に、類似した旋律を検索するには、*lub* を用いて問合せ旋律との照合を行えば良いと述べた。しかし、実用的な類似旋律検索を行うためには、時間的な拡大や縮小、音高的な拡大や縮小にも対応できるような表現法と、時間や音高に関する相対化が必要となる。

そこで、拡大や縮小に対して不変であるような特徴として音価比 (隣接する音価の比) などを属性として導入することが考えられる。音価比と音長の関係は、ちょうど音程と音高の関係になぞらえることができよう。また、音高を相対化する時には、局所的なキー (調) を事前に知る必要がある。

さらに音楽に特徴的な問題として、和音の展開形を扱うような場合、音高のオクターブ要素に関して N と $N \pm 1$ が照合すると都合が良い。また、和音どうしを照合する時、その構成音の音高に関しても同様の場合が起こり得る。

しかし、これら拡張/縮小/相対化を含む高次の照合機能を DOOD の抽象化/具体化の関係 (\square) にうまく対応付けられるかどうかは難しい問題である。

緩い *glb*: *lub* の実用的な意味は良く理解できるものの、*glb* の方は *lub* の相補物以上の意義が希薄という指摘がある。さらに、オブジェクト項中で \perp になる枝が多く、*glb* が有効に機能する場面はそれほど多くないとも言われている。*glb(x, y)* は、 x と y の構造が矛盾しない限り x と y を重ね合わせる操作であるが、 x と y の構造が多少矛盾しても x と y をそれな

りに緩く重ね合わせる操作が定義できると表現力が高まるだろう。ここではその操作を *merge* と呼ぼう。

現在、まだ *merge* に形式的な定義を与えていないため、ここでは *merge* の動作例を挙げるにとどめる。単純な例として $merge(\{C4\}, \{E4\}) = \{C4, E4\}$ の場合は *glb* と同じ動作をする。より複雑な例として2つの旋律 [D4 C4] (タイムスパン木 T_3 は $\langle C4 D4 C4 \rangle$) と [C4 E4] (タイムスパン木 T_4 は $\langle C4 C4 E4 \rangle$) の *merge* を考える。ここで2つの旋律の C4 が同時刻とする (*temp* オブジェクトの導入によってこのような表現が可能となったことに注意)。その期待される解は [D4 C4 E4] となる。つまり $merge(T_3, T_4) = \langle C4 D4 \langle C4 C4 E4 \rangle$ あるいは $\langle C4 \langle C4 D4 C4 \rangle E4 \rangle$ となる。ここで $glb(T_3, T_4) = \perp$ となってしまふ。さらに、*merge* の逆演算 (上の例では [D4 C4 E4] から [D4 C4] あるいは [C4 E4] を算出する) も有用であると思われる。これは単なる部分列の抽出ではなく、意味のある部分タイムスパン木を (緩く) 抽出する操作である。

演算子群について: 本稿で取り上げた演算子は、包摂関係 (\sqsubseteq), *lub*, *glb* であった。さらに *merge* や逆 *merge* (未定義) などにも触れた。しかし、部分的な情報しか与えられていないような関数の値を推定したり、類似度に対応するような距離を導入したりといった実用的な機能を実現しようとする、上記の演算子群だけでは表現力が弱い。

実際に様々な応用システムを作成していく中で、適切な演算子群を探ることになると思われる。

Well-Formed なタイムスパン木: 音楽システムを作成する際、計算の結果として出て来た答えに意味があるかどうかを判定する上で Well-Formedness (整形形式性) は重要な役割を果たす。*ts* オブジェクトの場合、全ての属性の値が無矛盾に過不足なく記述されていれば well-formed であると言えよう。では *lub* や *glb* の答えは常に well-formed なのだろうか。

タイムスパン木のノードに関して、その head の作り方には ordinary, fusion, transformation, cadential retain の4通りがある [6]。ここまでの例における head は全て ordinary で作られていた。HA 法の *ts* オブジェクトは head 属性を独立に持っているの、そこに任意の head の値を設定することができる。では、head を ordinary 以外で作った場合も、*lub* や *glb* の演算結果は well-formed であろうか。

ここでは head を fusion で作る例を考える (図5)。これら2つのタイムスパン木 $T_5 = \langle \{C, G\}2 C4 G4 \rangle$, $T_6 = \langle \{C, E\}2 C4 E4 \rangle$ では head が fusion で作られている。*lub* の計算では、head 属性と primary 及び secondary 枝は独立に計算されるので、図のように head が $\{C\}2$ であるにもかかわらず、C4 が primary 枝の下に出現しない。このタイムスパン木は、属性が完全に揃っていないという意味では well-formed ではないが、少なくとも head の値は確定しているの、全く意味が無いとも言いきれない。今後、タイムスパン木の well-formedness について検討を加える必要がある。あるいは、この例のような振舞いを回避

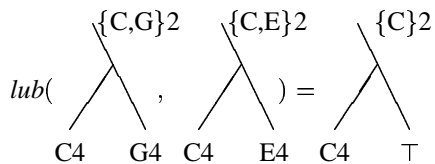


Figure 5: Well-formed でないタイムスパン木

する方法として, head を primary 枝と secondary 枝に関する制約として記述することも考えられる.

5 おわりに

本稿では HA 法に関して現在考察を進めている研究課題について述べた. 我々の目的は GTTM に基づいて計算機上の音楽知識表現法を設計することであり, 基本的に GTTM は所与のものとする立場を取っている. しかし, GTTM で与えられた概念が曖昧であるため, 音楽知識表現法を設計する際にそのギャップを埋めたり (例えばタイムスパン簡約の head 音価), 不足している概念を補充する必要があった (例えば時間構造).

最後に GTTM 自身の問題として一つ指摘しておきたいのはタイムスパン簡約の有効範囲の問題である. 16 小節程度の旋律までならタイムスパン簡約は有効に機能していると考えられるが, 32 小節や 64 小節以上の楽曲に適用した時も同様に意味のある分析結果が得られるのだろうか. むしろそのような大局的な構成については, プロロンゲーション簡約の方が有用な分析結果を提供してくれるかも知れない.

今後, 音楽知識表現の体系が幾つも提案されるようになると思われる. HA 法から改良, 発展したものもあれば, そうでないものもあるだろう. いずれにしても, 本稿での議論が参考となれば幸いである.

謝辞: 竹内好宏氏 (京都府立亀岡高等学校) より, GTTM に関して貴重な指摘や示唆を頂きました. 音楽知勉強会メンバの佐藤健教授 (国立情報学研究所), 東条敏教授 (北陸先端科学技術大学院大学) と, 草稿段階にて知識表現技術に関して議論して頂きました.

参考文献

- [1] Dibben, N., The Cognitive Reality of Hierarchic Structure in Tonal and Atonal Music, *Music Perception*, Vol.12, No.1, pp.1-25 (1994).
- [2] 平田, 青柳, 音楽理論 GTTM に基づく多声音楽の表現手法と基本演算, *情報処理学会論文誌*, Vol.43, No.2, pp.277-286 (2002).
- [3] 平賀, 音楽認知のための知識表現, 波多野諄余夫編 音楽と認知, 東京大学出版会, pp.97-130 (1987).

[4] Lerdahl, F., *Tonal Pitch Space*, Oxford University Press (2001).

[5] 小暮, 素性構造 (1) 及び (2), *人工知能学会誌*, Vol.8, Nos.2&3, pp.184-191&305-311 (1993).

[6] Lerdahl, F. and Jackendoff, R., *Generative Theory of Tonal Music*, The MIT Press (1983).

[7] 横田, 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).

[8] Widmer, G. Learning Expressive Performance: The Structure-Level Approach, *Journal of New Music Research*, Vol.25, pp.179-205 (1996).

[9] Uwabu, Y., Katayose, H. and Inokuchi, S., A Structural Analysis Tool for Expressive Performance, In *Proc. International Computer Music Conference 1997*, pp.121-124 (1997).