Vol.2010-MUS-85 No.4
2010/5/27

情報処理学会研究報告
IPSJ SIG Technical Report

旋律モーフィングアルゴリズムの形式的検証

平田圭二[†1]　東条敏[†2]　浜中雅俊[†3]

我々が提案した旋律モーフィングアルゴリズムの妥当性を形式的に証明する．まず証明の準備としてタイムスパン木の素性構造表現，楽曲構造束中の可聴な旋律，簡約パス，旋律の複雑さを計算する関数を導入し，それらに基づいて類似度と内挿を定義する．そして，アルゴリズムが内挿の定義を満たしているという定理を証明する．最後に証明の途中で定義した類似度と *join* 演算に関する問題点を議論する．アルゴリズムの妥当性を主張するために，これまでは例えば，多数の被験者を用いて評価実験を実施するような手段しかなかったが，計算論的音楽理論は別の手段を提供することを示した．

The formal verification of the melody morphing algorithm

Keiji Hirata,[†1] Satoshi Tojo[†2]
and Masatoshi Hamanaka[†3]

We formally prove the validity of the melody morphing algorithm that we proposed. As the preliminaries to the proof, we introduce a feature structure representation for a timespan tree, an audible melody in the lattice for musical structures, a reduction path and a function for calculating the melodic complexity, and define the melodic similarity and melodic interpolation. Then, the theorem that our algorithm satisfies the definition of the melodic interpolation is proved. Finally, we discuss the problems concerned with the melodic similarity and the *join* operation introduced for the proof. Up to now, conducting experiments with many subjects employed is almost the only way for demonstrating the validity of a developed algorithm. In contrast, computational musicology presents the potential for another way.

†1 NTT コミュニケーション科学基礎研究所 NTT Communication Science Laboratories
†2 北陸先端科学技術大学院大学 情報科学研究科 Japan Advanced Institute of Science and Technology
†3 筑波大学大学院 システム情報工学研究科 University of Tsukuba

1. はじめに

我々は計算論的音楽理論の構築を目指して，音楽理論に基づく楽曲の形式的表現法や代数的な楽曲操作体系の構築を行っている[1)]．曖昧で主観的と考えられている音楽的な知識や振る舞いを音楽理論の成果に基づいてモデル化，形式化することは，音楽認識だけでなく，音楽システムの設計や構築にも広く有用な知見をもたらすであろう．一般に，楽曲を操作するシステムでは記述力と簡潔さの間にトレードオフがある．このトレードオフに対処する 1 つの方法として，操作自体とそれらを組み合わせる意図の分離がある．つまり，ユーザにその意味を理解している操作を意図通りに組み合わせられるような自由度を与えるのである．

音楽理論や楽曲をモデル化，形式化することの利点には，操作自体と操作の意図を分離できることと，操作自体の意味 (semantics) を数学的に厳密に記述できることがある．前者についてのアナロジとして，我々が日常生活の中で様々な目的を達成するために四則演算を使いこなしている様子を思い浮かべて欲しい．意味を十分に理解した加減乗除と目的に沿ってそれらを組み合わせる方法論を提供することで，記述力と簡潔さをある程度両立させることができる．後者について，数学的に厳密な記述が計算機上への実装に大きく貢献するというのは明らかであろう．

既存の旋律から新しい旋律を作成する方法の中でも，旋律モーフィングはユーザの意図をシステムに伝達する directability[2)] という観点から実用的な手法の 1 つと考えられる．我々は旋律モーフィングアルゴリズムを提案し[3),4)]，そのアルゴリズムが実際に適切に動作しているかどうかを検証するため，実際に幾つかのサンプル旋律を被験者に聴取して貰い，モーフィング結果が入力サンプル旋律の内挿になっていることを確認し，アルゴリズムは妥当に動作していると結論付けた．

我々が提案したアルゴリズムは，束上の基本演算とそれらの組み合わせとして構成されている．もしその基本演算の意味と，それらを組み合わせたアルゴリズムの意味を形式的に記述することができれば，そのアルゴリズムが旋律モーフィングとしての形式的な仕様を満たしていることを数学的に証明することができるだろう．これは，アルゴリズムに多数のサンプル旋律を与えて被験者を使って 1 曲ずつ確認するような実験に頼らずにアルゴリズムの妥当性が厳密に主張できることを意味する．

続く第 2 章で，まず基本となるタイムスパン木の素性構造表現を説明し，アルゴリズム証明への準備として，楽曲構造束中の可聴な旋律，簡約パス，旋律の複雑さを計算する関数を導入し，それらに基づいて類似度と内挿を定義する．第 3 章で以前我々が提案した旋律モー

フィングアルゴリズムを紹介し，第 4 章ではそのアルゴリズムが内挿の定義を満たしているという定理を証明する．第 5 章では，証明の途中で定義した類似度と *join* 演算に関する問題点を指摘する．

2. タイムスパン木に基づく旋律の形式的表現

旋律とは音楽的イベントが時間方向に並んだものであり，音楽的イベントには単音や和音などがある．タイムスパン木とは，旋律に含まれる音楽的イベントどうしの音高と時間の近さによるグルーピングと拍節の強さによるグルーピングに基づいて，隣接する音の構造的重要度を決めながらボトムアップに構成された木のことである[5]．直感的には，動機や楽節に相当する数小節から数十小節の長さの旋律の静的な構造 (楽曲構造の分析) を表現したものである．

旋律 A，B に対するタイムスパン木を T_A，T_B のように表記する．一般に，1 つの旋律には複数のタイムスパン木が対応するが，本稿では 1 つの旋律に 1 つのタイムスパン木のみが対応すると仮定し，旋律 A とタイムスパン木 T_A を同一視する．曖昧にならない限りタイムスパン木に対しても A，B のように表記する場合がある．

2.1 楽曲構造束

本稿では知識表現法として素性構造を採用する[6]．素性構造では，対象 (タイムスパン木) が属性と値のペアの集合として表現され，値としてスカラー値だけでなく素性構造も許される (再帰的)．素性構造は二分木の構造をしており，ラベル付き非循環グラフ (Directed Asyclic Graph) に等しい．下に素性構造の例を示す:

$$F_1 = \begin{bmatrix} prim & v_3 \\ scnd & \begin{bmatrix} head & F_8 \end{bmatrix} \\ head & F_7 \end{bmatrix}$$

ここで $prim$，$scnd$，$head$ が素性ラベル (属性名) を，v_i がその値を表す．$prim$，$scnd$，$head$ はそれぞれ，タイムスパン木の各ノードにおける主枝，副枝，ヘッドを表している．属性 $scnd$ の値は素性構造である[⋆1]．

2 つの素性構造が包摂関係にあるとは，構造を根から辿る時に，一方の属性と値のペアの集合が他方のそれを包含していることを指す．例えば，上の F_1 は以下の F_2 や F_3 に包摂される．それを，$F_1 \sqsubseteq F_2$ 及び $F_1 \sqsubseteq F_3$ と表記する．

$$F_2 = \begin{bmatrix} prim & v_3 \\ scnd & \begin{bmatrix} prim & v_4 \\ scnd & v_5 \\ head & F_8 \end{bmatrix} \\ head & F_7 \end{bmatrix} \qquad F_3 = \begin{bmatrix} prim & \begin{bmatrix} prim & v_6 \\ scnd & v_7 \\ head & F_9 \end{bmatrix} \\ scnd & \begin{bmatrix} head & F_8 \end{bmatrix} \\ head & F_7 \end{bmatrix}$$

この時，タイムスパン木 F_2 を簡約して F_1 を得る，及び F_3 を簡約して F_1 を得ると言う．タイムスパン木を素性構造で表現することで，包摂関係が成立しているか否かの機械的判定が可能になる[7]．包摂関係を “`is_a`” 関係と見なすこともでき，例えば，$F_1 \sqsubseteq F_2$ の場合，F_2 `is_a` F_1 と解釈できる[⋆2]．

定義 A と B の交わり x とは $x \sqsubseteq T_A$ かつ $x \sqsubseteq T_B$ を満たす最大の元であり，$meet(A, B)$ あるいは $T_A \sqcap T_B$ と書く (図 1)．A と B の結び y とは $T_A \sqsubseteq y$ かつ $T_B \sqsubseteq y$ を満たす最小の元であり，$join(A, B)$ あるいは $T_A \sqcup T_B$ と書く．

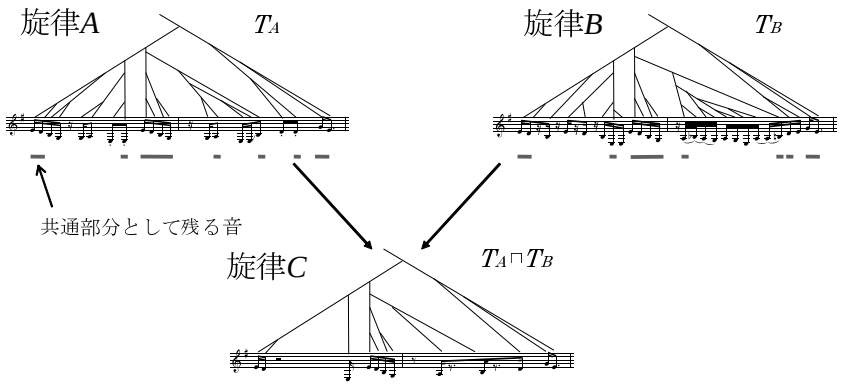

図 1 旋律 A と B の共通部分

⋆1 応用システム中で実現された素性構造には，本文で述べたタイムスパン木の情報に加えて，時間や順序に関する情報も含まれている．

⋆2 GTTM のタイムスパン木の簡約と本稿で述べた素性構造の包摂関係の対応付けには，理論的な不備が指摘されている[8]．

一般に $meet$ は，T_A と T_B に共通する部分木あるいは T_A と T_B の積集合に相当する．例えば，図 1 中の旋律 A，B のタイムスパン木を同時にトップダウンに辿ることで最大の共通部分 (下線を付した音) が抽出される．これは $T_A \sqcap T_B$ を計算することに等しい．同様に，$join$ は T_A と T_B を矛盾なく重ね合わせた木に相当し，T_A と T_B の和集合に相当する．任意の A, B に対して $meet$ と $join$ が存在することを仮定する (仮定の妥当性については 5.2 節で議論する)．

旋律とそれらの間の包摂関係から構成される束を楽曲構造束と呼ぶ．楽曲構造束は，全ての旋律とそれらを簡約して得られる全ての旋律を含む (ある 1 つの旋律とそれを簡約して得られる旋律だけを含むものではない)．つまりどんな旋律でも必ず楽曲構造束に埋め込むことができ，任意の 2 つの旋律に対して $join$ と $meet$ に対応する元が存在している．

2.2 可聴な旋律と簡約パス

定義 楽曲構造束の元であるタイムスパン木が過不足なく楽譜に変換できる時，可聴 (audible) であると言う．可聴な旋律だけから構成される楽曲構造束を可聴楽曲構造束と呼び可聴な項間の包摂関係を $\sqsubseteq$ と書く．

楽曲構造束には，可聴 (audible) な元と可聴でない元が含まれる．可聴な A，B に関して，$A \sqsubseteq B \Leftrightarrow A \sqsubseteq B$ である．本稿では以降，旋律は可聴なものに限定し，$meet$ や $join$ も同様に $\sqsubseteq$ を用いて定義されているものとする．

定義 $A \setminus B$ は，旋律 A に含まれているが $A \sqcap B$ に含まれていない音の集合である (タイムスパン木ではなく集合であることに注意)．

例えば図 1 の場合，A は 21 音から成る旋律であり，C は 12 音から成る旋律である．C の 12 音は全て A に含まれている (図中，共通部分に下線を付した)．本稿では集合 σ の要素数を $\#(\sigma)$ と表記する．$\#(A \setminus C) = 21 - 12 = 9$ である．

定義 $T_B \sqsubseteq T_A$ を満たす旋律 A，B に対し，A から B への簡約パスとは，図 2 の手順に従って，$A \setminus B$ に含まれる音を A から 1 音ずつ簡約して得られる旋律の列である．

一般に，図 2 の Step 2 おいて拍節的重要度が最小の音は複数個存在するので，簡約パスも複数通り存在する．A から $meet(A, B)$ への簡約パス上の旋律 C は，$T_A \sqcap T_B \sqsubseteq T_C \sqsubseteq T_A$ を満たす．

2.3 旋律の複雑さ

旋律とその旋律の複雑さあるいは旋律の持つ情報量を対応付ける関数 $|\cdot|$ は，$A \sqsubseteq B \Rightarrow |A| \leq |B|$ という条件を満たすならばどのような関数でも良い．例えば A に含まれる音の個数，タイムスパン木のノード数，木の深さで重み付けをしたノード数などが考えられる．

$N = \#(T_A \setminus T_B)$

Step 1: $T_n = T_A$ 簡約レベル$n = 0$ （$1 \leqq n \leqq N$）

Step 2 : $T_n \setminus T_B$に含まれる拍節的重要度が最小の音を1つ選び簡約してT_{n+1}を得る

Step 3: Step 2 をN回繰り返す

右例では$N = 3$ なので簡約パスには
$A \to x \to z \to B$ と $A \to y \to z \to B$がある

$B \sqsubseteq z \sqsubseteq x \sqsubseteq A$ と $B \sqsubseteq z \sqsubseteq y \sqsubseteq A$ が成り立つ

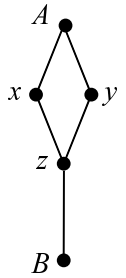

図 2 簡約パス

本稿では，$|A|$ の値として A に含まれる音の個数を採用する．さらに，任意の A, B に対して $join$ が存在し (2.1 節)，$|A \sqcup B| = |A| + |B| - |A \sqcap B|$ が成立することを仮定する (仮定の妥当性については 5.2 節で議論する)．

ここで関数 $|\cdot|$ に関する補題を与える．

補題 2.1 A から $A \sqcap B$ への簡約パス $A \sqcap B \sqsubseteq a_1 \sqsubseteq a_2 \sqsubseteq \cdots \sqsubseteq a_{n-1} \sqsubseteq A$ に関して，$|a_i| = i + |A \sqcap B|$ 及び $|a_i \sqcup B| = |B| + i$ である．

証明: まず，補題の前提より，$|a_i| = \#(a_i \setminus (A \sqcap B)) + |A \sqcap B| = i + |A \sqcap B|$ を得る．次に，本稿の仮定より，$|a_i \sqcup B| = |a_i| + |B| - |a_i \sqcap B|$ である．また $a_i \sqcap B = A \sqcap B$ なので，$|a_i \sqcap B| = |A \sqcap B|$．よって，$|a_i \sqcup B| = i + |A \sqcap B| + |B| - |A \sqcap B| = i + |B|$．

2.4 類似度と内挿

定義 A, B 間の類似度 S を次のように定義する[9]．

$$S(A, B) = \frac{|A \sqcap B|}{max(|A|, |B|)} \tag{1}$$

定義より明らかに $S(A, B) = S(B, A)$ である．もし $A = B$ の時は $S(A, B) = 1$ となり，$A \sqcap B = \bot$ の時は $S(A, B) = 0$ となる．

定義 A, B が与えられた時，$S(A, B) \leq S(A, \mu) \wedge S(A, B) \leq S(B, \mu)$ を満たすような旋律 μ を A, B の内挿であると言う (図 3)．

2 次元平面に図示したので，あたかも平面上の任意の 2 点間に距離が定義できるかのような印象を与えてしまうが，A, B は束上の元であり，A, B 間に連続的に旋律が存在して

情報処理学会研究報告
IPSJ SIG Technical Report

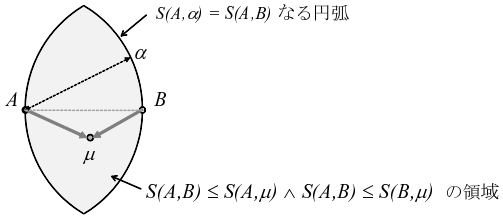

図 3 内挿の定義

いるわけではない．

3. 旋律モーフィングのアルゴリズム

旋律モーフィングの直感的な説明は，2 つの旋律が与えられた時の中間的な旋律である．文献 3) において，そのような中間的な旋律を計算するアルゴリズムを示した．以下の手順で A, B の内挿 μ を計算する (図 4)．

1. $meet(A, B)$ をとる ($T_A \sqcap T_B$ を計算する)．
2. A から $meet(A, B)$ への簡約パス上の旋律 C と，B から $meet(A, B)$ への簡約パス上の旋律 D を選ぶ．
3. $join(C, D)$ をとり ($T_C \sqcup T_D$ を計算し)，これを μ とする．

C を T_A 寄りにとると，出力に旋律 A の特徴を強く反映させることができる．C を $T_A \sqcap T_B$ 寄りにとると，出力にあまり旋律 A の特徴は反映されない．D のとり方に関しても同様である．

4. 旋律モーフィングアルゴリズムの証明

前章で示した旋律モーフィングアルゴリズムは，全モーフィングの計算 (図 6) に等しい．本章では，まず半モーフィング (図 5) に関する補題を示してから全モーフィングの妥当性を証明する．

補題 4.1 (簡約パス側の類似度) 旋律 A, B がある時，$A \sqcap B \sqsubseteq x \sqsubseteq A$ なる x に対して，$S(A, B) \leq S(A, x \sqcup B)$ である．

証明:

$$S(A, x \sqcup B) = \frac{|A \sqcap (x \sqcup B)|}{max(|A|, |x \sqcup B|)} = \frac{|x|}{max(|A|, |x \sqcup B|)} \quad (2)$$

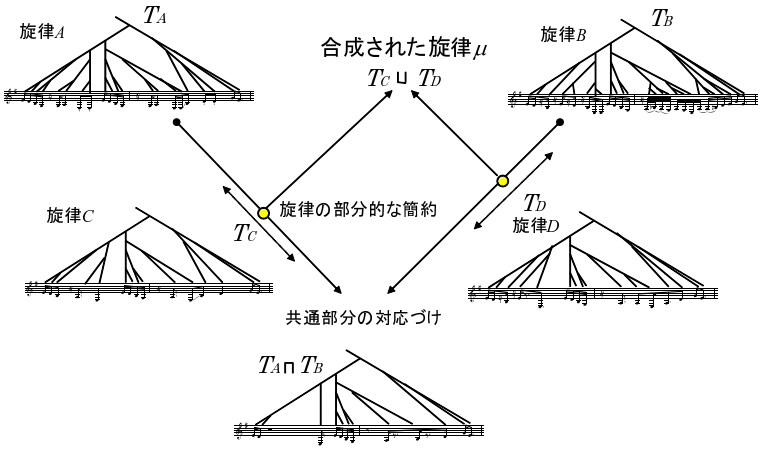

図 4 旋律モーフィングのアルゴリズム

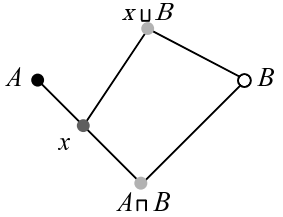

図 5 半モーフィングの計算

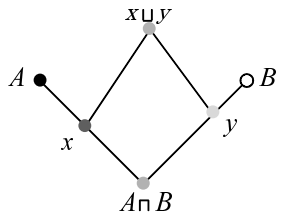

図 6 全モーフィングの計算

式 1 の値と式 2 の値の大小を比較するために場合分けを行う．

(C1) $|A| \leq |B|$．式 $1 = |A \sqcap B|/|B|$，式 $2 = |x|/|x \sqcup B|$ となる．式 2 の x に関して $x = a_i$ なる i をとることができる．補題 2.1 を用いて，式 $2 = |a_i|/|a_i \sqcup B| = (|A \sqcap B| + i)/(|B| + i)$ ただし $0 \leq i \leq n-1$ と変形できる．よって，式 1 $\leq$ 式 2．

(C2) $|B| \leq |A|$．x が $A \sqcap B$ から A に増加すると，$x \sqcup B$ は B から $A \sqcup B$ に増加するが，$|A| \leq |A \sqcup B|$ なので，$|x \sqcup B|$ の値はどこかで $|A|$ と等しくなる筈．つまり，$A \sqcap B \sqsubseteq a_K \sqsubseteq A$ かつ $|A| = |a_K \sqcup B|$ を満たすような a_K が存在する．式 2 の分母に関して場合分けを行う．

(C2-1) $A \sqcap B \sqsubseteq x \sqsubseteq a_K$．式 $1 = |A \sqcap B|/|A|$，式 $2 = |x|/|A|$．$A \sqcap B \sqsubseteq x \sqsubseteq A$ なので

式 $1 \leq$ 式 2.

(C2-2) $a_K \sqsubseteq x \sqsubseteq A$. $i \geq K$ なる i に関して，式 $2 = |a_i|/|a_i \sqcup B|$ である．補題 2.1 より，式 $2 = (|A \sqcap B| + i)/(|B| + i)$．次に式 2 と式 1 の差を計算する．式 2 − 式 1 の分子 $= |A \sqcap B| \cdot (|A| - |B|) + i \cdot (|A| - |A \sqcap B|)$，分母 $= |A| \cdot (|B| + i)$．分子，分母ともに正なので，式 2 − 式 $1 \geq 0$ を得る．よって，(C1)∼(C2-2) すべての場合において式 $1 \leq$ 式 2.

補題 4.2 (簡約パスの反対側の類似度) 旋律 A, B がある時，$A \sqcap B \sqsubseteq y \sqsubseteq B$ なる y に対して，$S(A, B) \leq S(A, y)$ である．

証明:

$$S(A, y) = \frac{|A \sqcap y|}{max(|A|, |y|)} \tag{3}$$

式 1 の値と式 3 の値の大小を比較するために場合分けを行う．

(C1) $|y| \leq |B| \leq |A|$. 式 $1 = |A \sqcap B|/|A|$. 式 $3 = |y \sqcap A|/|A|$. $A \sqcap B \sqsubseteq y \sqsubseteq B$ なので $y \sqcap A = A \sqcap B$. ゆえに式 1 = 式 3.

(C2-1) $|A| \leq |y| \leq |B|$. 式 $1 = |A \sqcap B|/|B|$. 式 $3 = |y \sqcap A|/|y|$. $y \sqcap A = A \sqcap B$ かつ $|y| \leq |B|$ なので，式 $1 \leq$ 式 3.

(C2-2) $|y| \leq |A| \leq |B|$. 式 $1 = |A \sqcap B|/|B|$. 式 $3 = |y \sqcap A|/|A|$. $y \sqcap A = A \sqcap B$ かつ $|A| \leq |B|$ なので，式 $1 \leq$ 式 3.

定理 4.3 全モーフィングは内挿の条件を満たす

証明: 補題 4.1 の結果に対し B を y に置き換えると $S(A, y) \leq S(A, x \sqcup y)$ となる．これと補題 4.2 より $S(A, B) \leq S(B, x \sqcup y)$ を得る．この式は A, B を入れ換えても成立するので内挿の条件を満たす．

5. 議　　論

5.1 類 似 度

例えば次の条件を満たす A，B_1，B_2 があるとする．B_1 と A，B_2 と A の間に直接の包摂関係 ($\sqsubseteq$) がなく，$A \sqcap B_2 \sqsubseteq B_2 \sqsubseteq B_1$ かつ $|A| \geq |B_1| \geq |B_2|$ である．この時，現在の類似度の定義 (式 1) では $S(A, B_1) = S(A, B_2)$ となる．この類似度の考え方では，A と B_j との *meet* を計算することで A と B_j の情報がどの程度失われるかを計量している．直感的には B_1 の方が失われる情報が多いが，正規化のための $1/max(|A|, |B_j|)$ の演算で常に $|A|$ の方が選択され B_j が考慮されないため，$S(A, B_1) = S(A, B_2)$ となってしまう．

上で述べた B_1 の方が失われる情報が多いという直感は，B_j を $A \sqcap B_j$ に順簡約するステップ数 r_j $(j = 1, 2)$ を比較して，$r_1 \geq r_2$ となることによって表現できる．そこで，簡約パス上の順簡約ステップ数 (とさらに逆簡約ステップ数) に基づくより精密な類似度の定義が考えることができよう．

5.2 *join* 演 算

2.1 節では，任意の A, B に対して *join* と *meet* が存在することと，$|A \sqcup B| = |A| + |B| - |A \sqcap B|$ が成立することを仮定した．つまり，*join* は 2 つの旋律 A，B に対して両者を包含するような複雑な旋律 $A \sqcup B$ の存在が計算論的に仮定できる時にのみ有効な演算である．また，*meet* に関しても同様に，A，B に対して両者の共通部分にあたる旋律 $A \sqcap B$ の存在が計算論的に仮定できる時にのみ有効な演算である．

本稿では詳しく述べなかったが，この仮定を成立させるため，現在の *join*, *meet* の定義では重ね合わされる音どうし，共有される音どうしが同じ音価，音高，時刻であるように制限している．ところがこの制限は実用的ではなく，相容れない構造や値を持ったタイムスパン木どうしの *join* や *meet* の計算が必要となる場面は多い．ここで相容れないとは，例えば，同時刻に生起した異なる音楽的イベント (例えば長さの異なる音符どうしや，音符と休符等)，secondary 枝の時間方向が異なるタイムスパン木のノード等である．

実用的な場面でも有効な演算ができるよう制限を緩め，*join* や *meet* を体系全体で一貫性をもって矛盾なく定義することは難しい (例えば，*join* の結果が常に可聴であることや，$|A \sqcup B| = |A| + \cdots$ が常に成立すること等)．文献 10) ではこの問題に対し，*join* の結果として例えば「3 であると同時に 4」，*meet* の結果として「3 あるいは 4」のような特殊な値を導入する手法に関し若干の議論を行った．さらに，*join* や *meet* をタイムスパン木を表現する項としてではなく，そのような項から成る集合と考える手法も考えられる．今後，これら手法により深い検討を加える必要がある．

6. おわりに

本稿で述べた旋律モーフィングアルゴリズムの形式的検証は，アルゴリズムが計算した全ての結果は内挿の条件を満たしているという意味において，アルゴリズムの健全性を証明したことに相当する．すると次はアルゴリズムの完全性に興味が湧くが，おそらく完全性は成立しないであろう．なぜなら，2.2 節で述べたように，B から $meet(A, B)$ への簡約パスは複数通りあるので，別の簡約パス上には現れているが今選んだ簡約パス上には現れていないような旋律が反例として存在するからである (その逆の場合もあり得る)．健全かつ完全な旋律モーフィングアルゴリズムの設計は今度の課題である．

旋律モーフィングアルゴリズムには，本稿で取り上げた内挿の考え方に基づくものだけでなく，外挿の考え方に基づくものも提案されている[11]．後者のアルゴリズムがどのような性質を持っているかを形式的に明らかにすること，またそれに先立ち適切な外挿の定義を与えることは，計算論的音楽理論において重要な課題であると考える．

参 考 文 献

1) 平田圭二, 東条敏, 相対擬補元を用いたメディアデザイン操作の形式化について, 第 19 回 人工知能学会 全国大会, 2B3-08 (2005).
2) 片寄晴弘, 橋田光代, 生成系音楽支援システムの Directability 視点からの考察, 情報処理学会研究報告 *2007-MUS-71*, pp.99–104 (2007).
3) Masatoshi Hamanaka, Keiji Hirata and Satoshi Tojo, Melody Morphing Method Based on GTTM, In *Proc. of ICMC 2008*, pp.155–158.
4) 浜中雅俊, 平田圭二, 東条敏, 計算論的音楽理論の応用, 情報処理学会誌, 道しるべ：計算の視点から音楽の構造を眺めてみると (5), Vol.49, No.11, pp.1334–1342 (2008).
5) Fred Lerdahl and Ray Jackendoff, *A Generative Theory of Tonal Music*, The MIT Press (1983).
6) Keiji Hirata and Satoshi Tojo, Lattice for Musical Structures and Its Arithmetics, *LNAI 4384* (Selected Papers from JSAI 2006, T. Washio et al. (Eds)), Springer-Verlag, pp.54–64 (2007).
7) Keiji Hirata and Tatsuya Aoyagi, Computational Music Representation Based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database, *Computer Music Journal*, 27(3), pp.73–89 (2003).
8) 平田圭二, 平賀譲, GTTM に基づく音楽表現手法再考, 情報処理学会研究報告 *2002-MUS-45*, pp.1–7 (2002).
9) Keiji Hirata and Shu Matsuda, Interactive Music Summarization based on Generative Theory of Tonal Music, *Journal of New Music Research*, 32:2, pp.165–177 (2003).
10) 平田圭二, 東条敏, 楽曲構造束とその上の演算系, 第 20 回 人工知能学会 全国大会, 1D2-4 (2006).
11) Masatoshi Hamanaka, Keiji Hirata and Satoshi Tojo, Melody Extrapolation in GTTM Approach, In *Proc. of ICMC 2009*, pp.89–92.