2B3-08

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

Formalization of Media Design Operations Using Relative Pseudo-Complement

平田 圭二[*1] 東条 敏[*2]
Keiji Hirata Satoshi Tojo

[*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
Japan Advanced Institute of Science and Technology

In this study, we propose a formalization of designing various art media. We consider that in design, contents are basically produced by arranging, transforming, and changing other contents and source materials, typically like collage. According to the knowledge representation technology, an art piece and contents can be represented by a feature structure, and we call the operations upon the feature structure media design operations, such as decoration, merging, abstraction, splicing, morphing, and so on. We put whole such variations of an art piece in *lattice* with regard to the subsumption relation between such feature structures, and we consider their meets and joins. Furthermore, we introduce the *relative pseudo-complement* for a structure to denote the relatively opposite arrangement, and discuss the operations upon art pieces in terms of strictly methematical notions.

1. はじめに

本稿では，種々のメディアデザイン操作を実現するために，相対擬補元の導入を提案しその具体的な処理方式について述べる．

我々はこれまで音楽 (旋律，和声，リズム) の形式化に取り組んできた [Hirata 03a]．人が音を知覚する時のゲシュタルトに基づく簡約関係 (reduction) を楽曲間に定義し [Lerdahl 83]，その簡約関係に基づく半順序を導入する．楽曲間に半順序関係が定義できれば，楽曲の領域を束として扱うことが可能となる．すると例えば，楽曲に対し最小上界，最大下界などの演算を適用することで，2 つの楽曲の共通部分を抽出したり，2 つの楽曲を足し合わせたりすることができる．

音楽以外の図形，造形などのメディアでもゲシュタルトに基づく簡約関係を見い出すことができれば，音楽と同様の形式化を適用することができる．すると，上の音楽の場合と同じく最小上界，最大下界などの演算適用が可能となる．メディアを加工，変形，修正する操作は一般にメディア依存だが，そのコンテンツをゲシュタルトのレベルまで抽象化して考えると，メディアに共通する抽象的な操作 (メディアデザイン操作) を考えることができる．メディアデザイン操作には例えば，足し合わせ (merger, superimposition/juxtaposition, splicing)，分割，置換，模倣，内挿/外挿，強調，引用，抽象化などがあろう．

しかし，上記のようなメディアデザイン操作の実現に際して，適用できる演算が最小上界，最大下界，半順序関係による抽象化/具体化しかない体系は，表現力が不足している．実際にこの不十分な体系で実現した演算は，類似度の計算 [Hirata 03b] と不完全な模倣 [Hirata 00] だけである．その表現力が不足している理由は，通常の実用的な演算体系では定義されている補元 (あるいはそれに相当する元) が定義されていないためである．補元が無いと，比例や重み付き和に相当する基本的かつ有用な演算が実現できない．

そこで我々は，種々のメディアデザイン操作を実現するため，相対擬補元の導入を提案しその具体的な処理方式を示す．相対擬補元を用いると，補元のように値が厳密に確定する演算は実現できないものの，処理方式を工夫することで実用性を高めることができる．相対擬補元を用いたメディアデザイン操作の実現は，デザイン支援システムの理論的基盤に貢献すると考えている．

連絡先: hirata@brl.ntt.co.jp

2. 属性構造，束，相対擬補元

2.1 属性構造

様々なメディアを統一的かつ形式的な方法で取り扱うためには，コンテンツを構成する部分や要素を属性構造で記述すると都合が良い．属性構造とは次のような feature–value の対の再帰的なリスト (属性構造):

$$\begin{bmatrix} f_1 & \begin{bmatrix} f_3 & v_3 \\ f_4 & v_4 \end{bmatrix} \\ f_2 & v_2 \end{bmatrix} \tag{1}$$

のことであり，属性構造は木構造と等価のデータ構造である．

二つの属性構造間が上位・下位関係にあるとは，下位の抽象的な構造に対して上位の構造はその下位の構造を含んでそれ以上の feature-value ペアが入っている状態を言う．このとき構造 F_1 が F_2 の下位にあるとき，$F_1 \sqsubseteq F_2$ と表記し，F_2 は F_1 を包摂 (subsume) すると言う．下記の二つの属性構造はいずれも属性構造 (1) を包摂するが，属性構造 (1) の変更を受けた個所が異なる．

$$\begin{bmatrix} f_1 & \begin{bmatrix} f_3 & v_3 \\ f_4 & [f_5 \;\; v_5] \end{bmatrix} \\ f_2 & v_2 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} f_1 & \begin{bmatrix} f_3 & v_3 \\ f_4 & v_4 \end{bmatrix} \\ f_2 & v_2 \\ f_6 & v_6 \end{bmatrix} \tag{3}$$

属性構造 (2) の構造は属性構造 (1) の木構造の葉の部分が詳細化されているのに対して (v_4 が $[f_5\ v_5]$ に)，属性構造 (3) の構造は木構造のトップレベルで枝分かれが増えている (属性 f_6 が増えている)．

属性値を取り出すために属性名を連鎖させるには '|' を用いる．簡単のために $f_1 \| f_n$ は属性の列を表し，'‖' の間の属性名が省略されていると考える．$f(F)$ は構造 F に対する属性 f の値である．属性構造 (1) の構造全体を F とすれば $f_2(F) = v_2$

であるが，$f_4(F)$ は未定義である．F の中に確かに f_4 は存在するがその値を取り出すためには $f_1|f_4(F) = v_4$ のように辿る必要がある．

2.2 束と meet, join

二つの属性構造 F_1, F_2 に対して，$F_1 \sqcup F_2$ は属性とその値について矛盾を起こさない限り両者の情報を統合した構造を指し，二つの構造の最小上界と考えらえる (join)．また $F_1 \sqcap F_2$ は両者の共通部分を指し，二つの構造の最大下界であると考えられる (meet)．$\top$ は最大の構造，$\bot$ は空の構造を表す．あるメディアの構造を数多く集めて，それらに $\sqcup, \sqcap$ の演算を導入した時，

べき等則 $a \sqcap a = a, a \sqcup a = a$
結合則 $a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c,\ a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$
交換則 $a \sqcap b = b \sqcap a,\ a \sqcup b = b \sqcup a$
吸収則 $a \sqcap (a \sqcup b) = a,\ a \sqcup (a \sqcap b) = a$

が成り立つならば，この構造の集合は束 (lattice) をなすという．束は $\sqcup, \sqcap$ に基づき自然に包摂の関係を作り，それらは全体で大きなグラフを構成する．

束はその中から上記の加工，修正に関する部分束を定義することができる．例えば「装飾に関する部分束」とは，属性構造の巨大な束の内，上位・下位関係が修飾という操作に限定された部分束を指す．

2.3 相対擬補元

束 L の元 a, b に対して $a \sqcap x \sqsubseteq b$ となる x のうち最大のものが一意に存在するとき，これを $a \supset b$ と書き a の b に対する相対擬補元 (relative pseudo-complement) と呼ぶ．任意の二元に対して相対擬補元が存在するとき，L は相対擬補束であると言う．相対擬補束には必ず最大元が存在し分配束 (次の分配則が成立する) である:

分配則 $a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c),\ a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$

逆に，L は最大元を持ち全順序集合であるならば相対擬補束となる [Landman 91]．

一般に，a は b の補元であるとは $a \sqcap b = \bot$, $a \sqcup b = \top$ と定義され，$b \equiv a^c$ と表記される．ブール束は相対擬補束であり，その補元は $a^c = a \supset \bot$ とも書ける．$\langle A, \sqcap, \sqcup, {}^c, \top, \bot \rangle$ がブール束であるとは，集合 A が有界 ($\bot$, $\top$ が存在する) な分配束でどの元にも補元が一意に存在するような場合である．図 1 にブール束，分配束，非分配束の例を示す．

3. メディアデザイン操作の形式的定義

3.1 メディアデザイン操作

様々なメディアによって表現されたコンテンツの加工，修正は，大きく，元のコンテンツを縮小する操作 (抽象化，分割，切断，meet)，拡張する操作 (装飾，接ぎ木，上置，併置，join)，それ以外 (内挿，外挿，モーフィング) に分けられる．

抽象化 (abstraction) 全体構造のうち深さを限定してある深さ以上の枝をすべて刈り取る操作を指す．

分割 (division) とは構造内の下位構造を分割することによって二つの下位構造を生成する操作を指す．分割において，根から分割点までの構造をコピーする．

$$\begin{bmatrix} f_1 & \begin{bmatrix} f_3 & v_3 \\ f_4 & [f_5\ \ v_5] \end{bmatrix} \\ f_2 & v_2 \end{bmatrix}$$
$$\Longrightarrow \begin{bmatrix} f_1 & [f_3\ \ v_3] \\ f_2 & v_2 \end{bmatrix},\ \begin{bmatrix} f_1 & [f_4\ [f_5\ \ v_5]] \\ f_2 & v_2 \end{bmatrix}$$

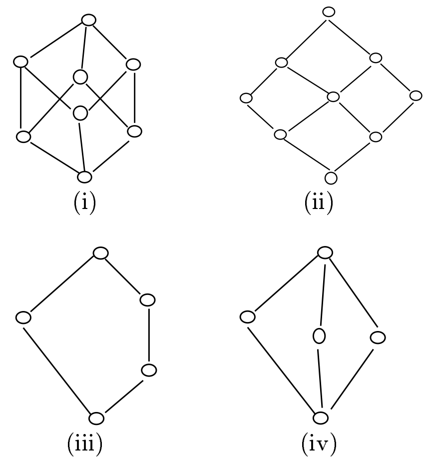

Figure 1: ブール束 (i)，分配束 (i)(ii)，非分配束 (iii)(iv)

切断 (cut) とは，特に属性構造の根における分割を指す．

装飾 (decoration) とは，ある構造の葉の部分のみを再帰的に別の属性構造に置換する操作である．属性構造 (2) は属性構造 (1) の $f_2|f_4$ における装飾である．

接ぎ木 (splicing) とは，ある構造を他の構造の一つの葉として埋め込む操作であり，空の葉の装飾 (置換) と見なせる．属性構造 (3) は，属性構造 (1) の根において，$[f_6\ v_6]$ を接ぎ木したものである．

上置 (superimposition) は，ある属性構造のある深さの属性の位置に，別の属性構造を並べて置く操作を指す．属性構造 (3) は，属性構造 (1) の根において $[f_6\ v_6]$ を上置したとも見なせる．

併置 (juxtaposition) とは，二つの属性構造の根において上置を行った場合を指す．特に矛盾なく上置が実行できる場合は join に等しい．切断の逆の操作と見なせる．

外挿 (extrapolation) 包摂関係にある二つの属性構造 $F_1 \sqsubseteq F_2$ に対して，$F_1 \sqsubseteq F_2 \sqsubseteq F_3$ であるような第三の構造 F_3 を外挿という．

内挿 (interpolaton) 包摂関係にある二つの属性構造 $F_1 \sqsubseteq F_2$ に対して，$F_1 \sqsubseteq F_3 \sqsubseteq F_2$ であるような第三の構造 F_3 を内挿という．

モーフィング (morphing) 二つの属性構造 F_1, F_2 に対して，$F_1 \sqcap F_2 \sqsubseteq F_3 \sqsubseteq F_1 \sqcup F_2$ であるような第三の構造 F_3 をモーフィングという．

3.2 準備

相対擬補元の定義より $a \sqcap (a \supset b) \sqsubseteq b$ が成立つ．ここでラムダ式 $f_{ab} = \lambda x. x \sqcap (a \supset b)$ を導入すると，上式は $f_{ab}(a) \sqsubseteq b$ と書き換えられる．

最小元を持つ相対擬補束をハイティング代数 (Heyting Algebra) または擬ブール代数という．ハイティング代数 H の元 a, b に対して $a \sqcup x \sqsupseteq b$ となる x のうち最小のものが一意に存在するとき，$\overline{a \supset b}$ と書き双対相対擬補元と呼ぶ．上と同様に，ラムダ式 $\overline{f_{ab}} = \lambda x.x \sqcup (\overline{a \supset b})$ を導入すると，$\overline{f_{ab}(a)} \sqsupseteq b$ を得る．

ハイティング代数の元 v_0 を繰り返し抽象化して得られる有限個の元を $v_0, v_1, \cdots, v_{n-1}, v_n$ とし，それらは $\bot \sqsubseteq v_n \sqsubseteq v_{n-1} \sqsubseteq \cdots \sqsubseteq v_1 \sqsubseteq v_0 \sqsubseteq \top$ を満たすとする．この時，元 v_i の列を v_0 の volume 列と呼ぶ．

以下特に断わらない限り，ハイティング代数 H の元 a, b は，具体的には，音楽，図形，造形などの個々のコンテンツや作品を表現している．

3.3 反対例から遠ざかる修飾

機能: 元 a, x が与えられた時，a に対して最も異なるように x に装飾を加える．つまり x を反対例として，a にできる限りの装飾を加える．

処理手順:

(1) $a \sqcap x$ を計算する．

(2) a の $a \sqcap x$ に関する相対擬補元 $a \supset (a \sqcap x)$ が求める修飾．

3.4 減衰器

機能: 減衰器関数を $y = att(x, v)$ と書く (ここで入力 x，減衰比 v，出力 y)．volume 列の要素 $v, v' \in V$ について，$v \sqsubseteq v' \Rightarrow att(x, v) \sqsubseteq att(x, v')$ が成立つ．さらに，$\bot = att(x, \bot)$ かつ $x = att(x, \top)$ が成立つ．

処理手順:

(1) 入力 x が与えられた時，volume 列 V の全要素 $v_0, v_n \in V$ に対して $v_0 \sqsubseteq x \sqsubseteq v_n$ でないような V を取ってくる．

(2) そのような $v, v' \in V$ について $att = \lambda xv.(v \supset x)$ とすると，$att(x, v)$ は減衰器として機能する．

3.5 モーフィング

機能: 減衰器を用いて 2 つのコンテンツの線形和を計算する．これは元 a と b 間のモーフィングに相当する．

処理手順:

(1) 条件 $a \not\sqsubseteq b \wedge b \not\sqsubseteq a$ が成り立つと仮定 (a と b は異なるコンテンツであると仮定)．

(2) a から volume 列 a_i を作り $att(b, a_i)$ を計算する．b の音量を調節する減衰器を a の volume 列から作る．

(3) 同様に b から volume 列 b_j を作り $att(a, b_j)$ を計算する．a の音量を調節する減衰器を b の volume 列から作る．

(4) a と b の内挿は $att(a, b_j) \sqcup att(b, a_i)$ (ただし $i + j = N$) として計算される．

ここで $i + j = N$ という条件は正規化のためであり，重み係数 a_i，b_j の和が常に「1」となるような制約を表現している．

3.6 事例 1 つの場合の模倣

機能: ある関数 f によって元 a を元 b に変換した ($b = f(a)$) という事例が与えられたとする．その関数 f に未知の元 x が入力された時の値 $y = f(x)$ を推定する (模倣する)．

処理手順:

(1) a と b より f_{ab} と $\overline{f_{ab}}$ を作る．

(2) $f_{ab}(x)$ と $\overline{f_{ab}(x)}$ を計算する．

(3) $f_{ab}(x) \sqsubseteq y \sqsubseteq \overline{f_{ab}(x)}$ なる y を探す．

3.7 問題点

上記メディアデザイン操作を実現する際の問題点を挙げる．

(i) ハイティング代数 H における相対擬補元 $a \supset b$ に対応する具体的な元 (項) の計算について，H の要素が有限個の具体項 (コンテンツ，作品) であれば，全要素を枚挙することで相対擬補元を求めることができる．しかし，H が無限あるいは膨大な要素を含む場合，任意の a, b に対して相対擬補元を一意に決定できるかどうか，そして決定できるならそれを具体的に求める効率的なアルゴリズムは知られていない．

(ii) 減衰器の出力は，入力だけでなく volume 列の値にも影響されるので，入力情報の損失が少ない高線型性の volume 列は入力に依存する．そのような適切な volume 列の求め方はまだ知られていない．

(iii) 模倣の処理ステップ (3) において，一般に $f_{ab}(x)$ と $\overline{f_{ab}(x)}$ 間には複数個 (あるいは無限個) の元が存在する．その中から適当な y を具体的に見い出すアルゴリズムを設計する必要がある．

4. おわりに

様々なメディアによるコンテンツや作品を属性構造で表現すると，従来の知識表現技術を応用することが可能になり，メディア非依存のレベルにおいてメディアの加工，修正といったデザイン操作を議論し実現できるようになった．実際，修飾，減衰器，モーフィングは相対擬補元の導入によって初めて実現可能となった操作であり，模倣は相対擬補元の導入によってより精度を上げることができた．

本形式化を適用するには，個々のコンテンツや作品間に半順序関係が定義されていなければならない．音楽の場合，認知的ゲシュタルトに基づく半順序関係 (楽曲間の簡約関係) を用いてその半順序関係を定義した．他の多くのメディア (例えば図形や造形など) でも認知的ゲシュタルトは自然に成立していると考えられる．

今後の課題を述べる．本稿で述べた形式化は，連続的でなく離散的 (基数的でなく序数的) である．しかし，演奏の表情付けやジェスチャ生成などのコンテンツ制作では，時間やある連続量を含む表現とその形式化が望まれる．そして，相対擬補元だけで表現力は十分なのかの考察も含めて，他のさまざまなメディアデザイン操作の機能と処理手順を考えていきたい．

References

[Hirata 00] 平田，青柳: バービーブン: 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 *2000-MUS-37*, pp.17–23, (2000).

[Hirata 03a] Hirata, K. and Aoyagi, T.: Computational Music Representation based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database, *Computer Music Journal* Vol.27 (3), pp.73–89, The MIT Press (2003).

[Hirata 03b] Hirata, K. and Matsuda, S.: Interactive Music Summarization based on Generative Theory of Tonal Music, *Journal of New Music Research* Vol.32 (2), pp.165–177, Swets and Zeitlinger (2003).

[Lerdahl 83] Lerdahl, F. and Jackendoff, R.: *A Generative Theory of Tonal Music*, The MIT Press (1983).

[Landman 91] Landman, F.: Structures for Semantics, Kluwer Academic Press (1991).