Vol.2017-MUS-116 No.22
2017/8/26

楽譜記述言語MusicXMLの拡張による楽典を考慮した和声解析フレームワークの提案

清水　祐輔[†1,a)]　沼尾　雅之[†1]

概要：本研究では楽譜記述言語である MusicXML を拡張することにより，和声解析における解析フレームワークの提案を行なう．MusicXML では楽譜の情報を全て記述することが可能だが，音符や拍子などの音楽記号を解釈するための処理が必要となる．また，楽曲は音楽記号だけでなく進行や拍などを考慮した上で解析を行うため，記号以外の楽典の知識も必要となる．さらに，和声解析など楽曲の構造解析は自動作曲や作曲支援といった応用を目的としている．構造解析によって得られた構造を別のフォーマットで記述すると楽譜から離れた領域で音楽を扱うことになる．本研究では，楽譜から拍子ごとの音符収集や，楽典の解釈を MusicXML へ拡張することにより，楽典を考慮した解析を容易にする．また，解析結果も記述できるように MusicXML を拡張することにより，モチーフ分析など高度な楽曲解析への応用も可能になり，和声解析における有用性を示した．

1. はじめに

和声解析とは，与えられた楽曲に対し計算機が和声 [1] を付与することである．和声とは楽曲の響きを表現するため作曲や編曲の基礎となる音楽理論であるため，計算機が和声を認識することができれば作曲支援，特徴抽出といった技術に応用することができる．

クラシック音楽の和声解析は広く行われており，既存研究では Hidden Markov Model(HMM) を用いた確率モデルでの和声解析や [2][3][4]，Lerdahl により発表された計算論的音楽理論である Tonal Pitch Space（TPS）[5] を応用したルールベースでの和声解析がある [6][7][8]．

しかしながら，既存研究では楽典をあまり考慮していない問題点がある．楽典とは，音楽の基礎的な理論であり，楽譜の読み書きをするための知識である [9]．人間は和声分析をする際楽典を適切に扱った上で和声を付与するため，楽典を考慮した和声解析は必須である．また，和声解析は応用研究のための技術であるため，ただ和声の結果を出力するだけでは応用研究に活かすことは困難であり，構成音や拍節，調の情報など得られた和声以外の解析の過程も考慮する必要がある．さらに，和声は必ずしも正解が一つとは限らないため，和声解析にはアルゴリズムや考慮する楽典には幅を持たせる必要がある．普段我々は楽曲の演奏や和声分析を行なう際，転調部分や演奏記号などに優先順位をつける．例えば拍子の関係を考慮したい人もいれば，転調を考慮したい人もいるなど，採用する楽典や音楽記号は異なる．和声を付与する際も順方向，逆方向，あるいは繰り返しながら行うなど，分析方法においても自分だけのオリジナルなアルゴリズムが存在する．和声解析を始めとする楽曲解析においても同様で，提案するアルゴリズムは人によって異なり，多様性が存在する．そのため様々なアルゴリズム，楽典の組み合わせによる和声解析を実現する必要がある．

多様性のある和声解析を実現するために，我々は楽曲記述言語 MusicXML[10] に着目した．MusicXML は，XML 形式で楽譜を記述するためのオープンフォーマットであり，MusicXML を用いた研究は数多く存在する [11][12]．MusicXML は元々解析用途に設計されたものではなく，楽譜記述用途のフォーマットであるため，解析に必要な拍子の概念や楽節などの楽典は全て記述されていないが，XML の拡張によって様々な情報を付与することができる．

以上により，本研究では，MusicXML の拡張を用いた，楽典を考慮した和声解析および和声解析を用いた応用研究のサポートを目的としたフレームワークを提案する．考慮する楽典やアルゴリズムを取捨選択することができ，様々な和声解析アルゴリズムを実現できるようにする．また，和声解析に必要な音符記号の構造化や和音，和声なども表現できるように MusicXML を拡張することで，和声解析のプロセスや結果を MusicXML 上に記載できるようにする．これにより，和声解析以外の楽曲解析や応用研究も可能にする枠組みを提案する．

2. MusicXML

MusicXML とは，楽譜を表現するための XML 形式のオープンフォーマットである．その他の記号楽譜ファイルとして Standard MIDI があるが，Standard MIDI はバイナリで楽譜を記述するため直感的に理解することは困難で

†1 現在，電気通信大学
Presently with The University of Electro-Communications
a) s1631064@edu.cc.uec.ac.jp

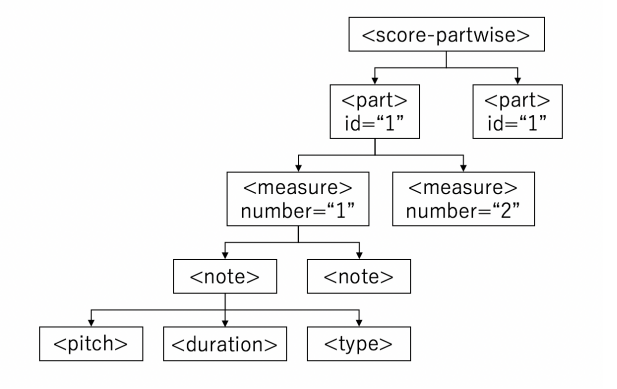

図 1　MusicXML の構造

```
<measure number='2>
  <note>
    <pitch>
      <step>C</step>
      <octave>4</octave>
    </pitch>
    <duration>
      4
    </duration>
  </note>
</measure>
```

図 2　XML の記述例

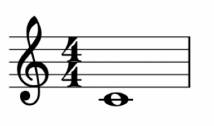

図 3　図 2 の出力例

ある．MusicXML は XML 形式を利用したフォーマットであり，要素をテキストで記述するためデータの意味が分かりやすい．さらに，要素は入れ子構造であり，小節単位，パート単位で情報を取り出すことも可能である．

MusicXML は階層的な記述がなされており，まず楽譜の記述形式を表す score-partwise があり，その中にパートを表す part タグが用意されている．パート内には小節を表す measure タグがあり，その中に音符などの要素がある，といった順に階層化されている (図 1)．

measure 内には，拍子などを指定する attributes タグ，音符を記述する note タグがある．(図 2) に全音符のド (図 3) を出力するための XML を記述例を示す．

2.1　attirbutes タグ

attributes タグ内では，基準の長さである division，主調を表す key，拍子記号を表す time が並ぶ．

divisions: 楽曲の中で扱う音符の最短調を，4 分音符を基準にして指定する．1 を指定した場合は，譜面で扱うことができる最短調の音符は 4 分音符，2 を指定した場合は 8 分音符となる．

key: 楽曲の主調を指定する．ハ長調を基準としてシャープとフラットの数を整数値で指定する．1 であればシャープがひとつ，つまりト長調となり，-1 であればフラットがひとつのヘ長調となる．長調であれば key 内に用意されている mode タグに”major”と記述し，短調であれば”minor”と記述する．

time: 拍子記号を指定する．拍の音符を表す beat-type と，拍子を表す beats が用意されており，それぞれ整数値で指定する．

2.2　note タグ

note タグ内では，主に音符や休符を記述する．

pitch: 音符の音高を指定する．音名をコードネームで記述する step と，オクターブを整数値で記述する octave が用意されている．

duration: 譜面上での音符の長さを指定する．divisions を基準にし，divisions で指定した長さの何倍か，という形で指定する．divisions が 1 で duration が 1 ならば 4 分音符，2 ならば 2 分音符，4 ならば全音符といった記述方法である．

rest: 休符を表す．

3.　楽典に基づく和声分析

本章では，人間が和声を付与する和声分析について述べる．和声分析の目的は，より良い演奏のためである．特に，トニック，ドミナント，サブドミナントを見つけ出し，緊張，弛緩を意識した演奏を行なうためである．また，和声だけでなく，和声分析の過程も楽譜に直接書き込むことによって演奏のクオリティを向上させる．

3.1　和音区間決定

まず，どの和音であるか考えるために，まずは和音の範囲を求める．和音は常に長さが決まっているものではなく，ある程度の範囲をまとめて和音とする場合がほとんどである．楽曲には拍節構造があり，和音は拍節毎，または複数の拍節をまとめて解釈の単位とする．複数の拍節が解釈の単位となる場合は複数の拍節をまたぐ 2 分音符や全音符，4 連符などが出現した時であり，その際はその音符の発音区間を 1 つの解釈の単位とする．演奏の際はその解釈の単位を意識し，できるだけ和声音が響くような演奏を心がける．

3.2　構成音から和音候補列挙

次に拍節毎の構成音から考えられる和音の候補を列挙する．構成音を見るだけである程度は候補を絞ることは出来るものの，それだけで和音の推定することは困難である．和音の推定には調とカデンツを活用する．楽曲には調の概念があり，和音を推定する場合は調号や臨時記号から調を認識する．調号の変化や臨時記号で転調や借用和音の判断を行なう．そこから，調上で考えられる和音を構成音から考える．借用和音が出現した際は目印を付け，ドミナントを意識し演奏する．

また，非和声音もここで適切に扱う．非和声音は和声に関係の無い音であるため，考慮せずに和声を推定する必要がある．経過音や刺繍音など，非和声音にも種類がある．それぞれの特徴から非和声音を判断し，和声と分離させる．非和声音に対してはその音が非和声音であると分かるように目印を付け，経過音は次の和声音に繋がるように演奏する，倚音はやや強めのタッチで演奏するなど，それぞれの非和声音の演奏方法を意識する．

3.3 カデンツに基づいた進行決定

次に，列挙した和音候補の中から最適な和声を求める．和声学には進行の基盤としてカデンツが定められており，トニック，ドミナント，サブドミナントの判断を行なう．例外はあるが，基本形としてトニックはI, ドミナントはV, サブドミナントはIVやIIが用いられることが多いため，これらの和音の可能性を考える．つまり，和声の前後関係を考慮する．例えば，ある楽曲に対してI，Vと判断した場合は，カデンツの進行から次の和音はIが来るだろう，というように認識すべき拍節の前の和音も考慮しながら和声分析が行われる．これらのステップを順方向，逆方向，または繰り返し考えることで和声が付与される．

以上により，人間は和声分析を行なう際，拍節構造，カデンツ，非和声音などの楽典を考慮して和声の付与を行なう．さらに，分析の過程で判明した情報を楽譜に書き込むことにより，演奏のクオリティを向上させている．

4. 和声解析フレームワーク

本章では, 第3章で挙げた和声分析のアルゴリズムから和声解析フレームワークの提案と，提案するフレームワークのステップに沿った和声解析手法について述べる．ここで，提案する和声解析フレームワークは，以下の5ステップの情報を取得できるツール，またはライブラリ群であると定義する．人間が和声分析を行う際のアルゴリズムを近似しており，それぞれのステップにおいて用意された楽典の解釈やアルゴリズムを取捨選択することにより，和声解析手法に幅を持たせることが可能になる．また，和声の出力だけでなく解析のステップ毎に情報の取得が可能であるため，和声解析を用いた作曲支援システムや特徴解析に応用だけでなく，和声解析結果以外の構成音やコード候補などの収集が可能になるため，和声解析以外の楽曲解析にも応用できる．

(1) 構成音収集
(2) 構成音に対して楽典の解釈を付与
(3) 和音候補の付与
(4) カデンツを考慮した候補の絞り込み
(5) 解析結果の記譜

4.1 構成音収集

和声解析の最初のステップとして，まずMusicXMLから拍節毎の構成音を収集する．ここで，既存研究ではRocherらがホモリズム変換によって音符を最小単位を分割することにより，複数の解析単位にも対応した構成音収集方法を提案したが，実際には音符を増やしている操作であり，楽曲の構造を崩した上での解析となる問題がある (図4)．

提案手法では，MusicXMLの要素であるnoteを直接参照する形を取ることにより楽曲の構造を崩さずに構成音収集を行なう (図5)．

4.2 楽典の解釈付与

収集した構成音に，楽典の解釈から生成したルールを付与する．提案手法では拍子を考慮した和声的強度の付与，

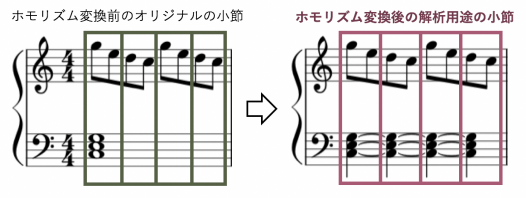

図 4 解析区間毎に構成音を分割するホモリズム変換

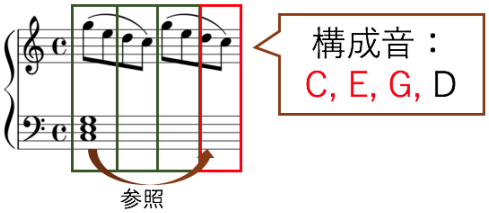

図 5 楽曲の構造を崩さない構成音収集

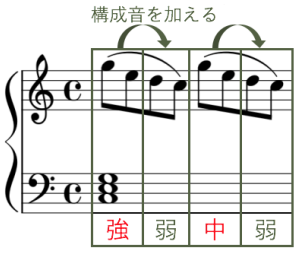

図 6 拍節外での拍子関係

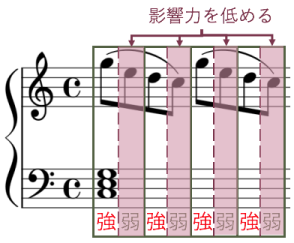

図 7 拍節内での拍子関係

楽節毎の楽曲分割がある．

4.2.1 拍子を考慮した和声的強度の付与

構成音収集の際，発音されているかどうかをバイナリで表現するのではなく，0から1への表現とすることで，構成音の和声的強度を求める．和声的強度を記述することにより，構成音がどの程度和声に影響しているかの情報を付与する．

強度を求める手法として，強拍，弱拍を考慮する．非和声音はそれぞれ固有の特徴があるが，弱拍に出現することが多いという共通の特徴がある．この強弱は音の大きさではなく，心理的な強さを表し，強拍部分に和声音が配置されることが多い．

提案手法では，解析区間自体での処理と，解析区間内での強拍，弱拍の処理がある．解析区間自体が弱拍の場合，弱拍の区間に対し，その直前の強拍の区間の構成音を足し合わせる（図6）．解析区間内の場合は解析区間を分割し，前半部分を後半部分に足し合わせる (図7)．それにより，弱拍では発音されていなくても暗黙的に響いている音を考慮できる．

4.2.2 楽節毎の楽曲分割

ひとつの楽曲は，楽節とよばれる小節のまとまりに分割することができる．2小節を動機，4小節を小楽節，8小節を大楽節と呼ぶが，特に，大楽節はそれ自体をひとつの楽曲とすることができる（図8）．転調は大楽節毎に行われることが多く，提案手法では転調を考慮するため，楽節毎の解析が必要となる．大楽節は通常8小節で構成されるた

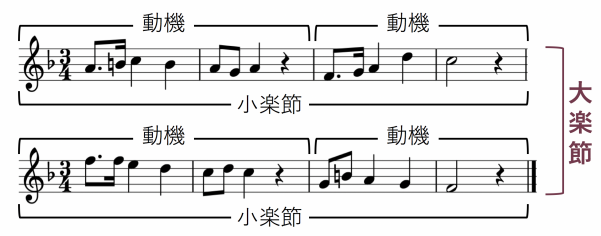

図 8　楽節の定義

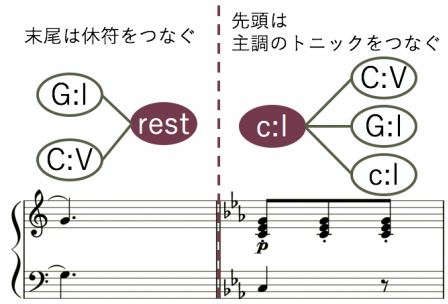

図 9　楽節毎の和音候補グラフ分割

め，提案手法では 8 小節毎に分割し，末尾には休符をつなぎ，先頭には主調のトニックをつなぐことで楽節を考慮した和声解析を行なう (図 9)．

4.3　ベーシックスペースによる和音候補付与

楽典の解釈を付与した構成音から候補となる和音を考える．和音を表現するための概念として，ベーシックスペースを用いる．ベーシックスペースとは和音の構成音のヒエラルキーを表現する概念である．ある音が，その和音の中でどれほどの影響力があるか示しており，以下のルールで level 毎に階層分けされている．ここで，ピッチクラスとは同じ音名を与えられている音の集合のことである．

- level a にあるピッチクラスはその和音の根音である．
- level b にあるピッチクラスはその和音の根音と 5 度音の 2 つである．
- level c にあるピッチクラスはその和音の構成音である．
- level d にあるピッチクラスはその和音が置かれている調の構成音である．

図 10 はハ長調の I の和音のベーシックスペースである．横軸をピッチクラス，縦軸をその和音における影響度としており，ピッチクラスの値が 0, すなわちドが 1 番影響力の高い音であることが分かり, その次に 7（ソ）, 4（ミ）が続いている．

ベーシックスペースを導入することの利点として, その和音の中でも実際に発音されていない音も暗黙的に響いている音であると考えることができる．例えば, レの音はハ長調の I の和音では発音されていないが, ベーシックスペースでは少なからずハ長調の I の和音に関わっていることが分かる．

ベーシックスペースと構成音を比較する事により, その和音と構成音との類似度を求める．ベーシックスペースを x, 構成音を y とした相関関数 $basicspace(x, y)$ を定義する．以下のステップで計算を行う．

level a	0											
level b	0							7				
level c	0				4			7				
level d	0		2		4	5		7		9		11
level e	0	1	2	3	4	5	6	7	8	9	10	11

図 10　ハ長調の I の和音のベーシックスペース

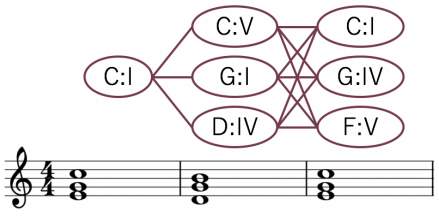

図 11　和音候補グラフ

(1) 構成音をピッチクラスへ変換する．
(2) ピッチクラスを 12 次元ベクトル上に配置. ここで, レベルはまだ d（=1）である．
(3) 比較を行うベーシックスペースを x とし，x と構成音のレベルを合わせる. これを y とおく．
(4) x と y を比較し, 異なっているレベルの差を合計する．

例として, 構成音がレ, ソ, シであれば，(1) ではまず 2,7,11 となる．次に（2）で [0 0 1 0 0 0 0 1 0 0 0 1] のようなベクトルになる．（2）で各ベーシックスペースに構成音のレベルを合わせる．例として, 以下のような配列になる．

- y（x = I/G） = [0 0 3 0 0 0 0 4 0 0 0 2]
- y（x = I/C） = [0 0 1 0 0 0 0 3 0 0 0 1]

(4) で, ピッチクラス毎に異なるレベルの差を合計する．計算式は以下となる．

$$basicspace(x, y) = \sum_{i=0}^{11} |x.basicspace[i] - y.basicspace[i]| \quad (1)$$

この合計値が構成音と和音の類似度を表し, 数値が小さい程似ているという結果になる．全ての調とその調上の和音のベーシックスペースと構成音を比較し, 最小値をとる和音を候補とする．

4.4　カデンツを考慮した進行の絞り込み

求めた和音候補から，和音の前後関係を考慮した和声を求める．提案手法では和音の候補をつなげたグラフを生成し，TPS を応用した計算コストを用いて Viterbi アルゴリズムによって和声を求める．

4.4.1　和音候補グラフ生成

求まった和音候補をノードとし, 解析区間を時系列とした和音候補グラフを生成する（図 11）．大楽節毎にグラフは区切られ，グラフの先頭には主調上のトニックが接続され，末尾には休符が接続される（図 9）．

4.4.2　和音間距離計算

上記により得られた和音候補グラフに対して, 坂本ら [7] が整備した TPS の和音間距離計算式を応用して遷移コストを付与する．求まった遷移コストから, Viterbi アルゴリズムを用いる事で楽曲全体として最適な経路を求める．以下のステップで和声を出力する．

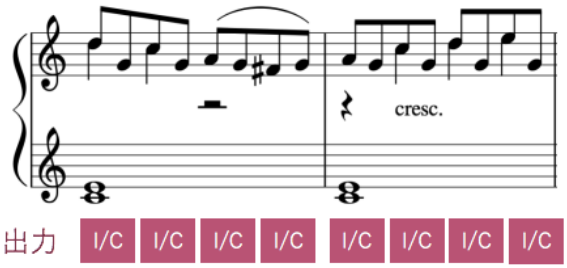

図 12　和声解析例

(1) 和音候補グラフのコストとして, TPS の和音間距離を応用した計算コストを付与する.
(2) Viterbi アルゴリズムを用いて最短経路計算を行う.
(3) 求まった経路を最適な和声進行とする.

4.5　解析結果の記譜

Viterbi アルゴリズムによって求めた進行を最適な和声とする (図 12).

5.　和声解析フレームワークに基づく MusicXML の拡張

本章では第 4 章で述べた和声解析フレームワークを実現するための MusicXML 拡張について述べる．和声解析フレームワークは，オリジナルの MusicXML の要素への拡張，更に楽典の指標である要素 metric の拡張，和声解析情報を記述する progression の拡張から成る．和声解析結果の MusicXML 表現を図 13 に示し，それぞれの拡張についての説明を行う.

5.1　従来の MusicXML 拡張

5.1.1　属性 metricals による小節の分割

楽曲解析を行う際，まず拍節毎の構成音を収集する．構成音収集では，解析単位として分割した小節毎で鳴っている音符を集めるため，各小節の分割単位を求める．小節の分割単位には拍節構造を考慮し，拍節毎で分割すればよい．適切な拍節は，4 分の 4 拍子の楽曲であれば 1 小節あたり 4 つ，4 分の 3 拍子の楽曲であれば 3 つというように，拍子記号を参照する事で求められる.

求めた拍節を要素 measure の属性として metricals を用意し、拍節数を整数値で記述することで，小節ごとの解析単位を取得できる.

5.1.2　属性 id の拡張による note 参照

要素 note の属性として，id を付与することで，metric 内で note を参照する．note を直接参照することで，情報を欠落させることなく情報を取得できる.

5.1.3　構成音の位置付けのための duration 拡張

構成音の小節内における位置付けとは，ある音符がどの解析区間に属しているかということである．ただ拍節内で発音されている構成音をまとめただけでは，その音符が拍節内のどこで発音されているか分からない．note の duration の属性に，発音開始時間と発音終了時間を表した begin と end を記述することで，音符が発音されている拍節を取得できる．単位は拍節とし，4 分の 4 拍子の小節であれば拍節数は 4 なので，0.0 から 4.0 の間で記述する．図 13 の例は 4 分の 4 拍子の楽曲であり，発音時刻が 0.0，終了時刻が 2.0 であることから，2 つ分の解析区間でこの音符が鳴っている 2 分音符ということが分かる.

```
<measure number='1' metricals='4'>
  <note id='2'>
    <duration begin='0.0'   end='2.0' />
  </note>
  ...
  <metric number='1' unit-beat='down'>
    <constitution>
      <component ref='2' beat='down'/>
      <component ref='3' beat='up'/>
      <component ref='8' beat='down'/>
      <component ref='9' beat='down'/>
      <component ref='10' beat='down'/>
    </constitution>
    <pitch>
      <C pron='1' strength='1.0'/>
      <Cis pron='0' strength='0.0'/>
      <D pron='1' strength='0.5'/>
      <Dis pron='0' strength='0.0'/>
      <E pron='1' strength='1.0'/>
      <F pron='0' strength='0.0'/>
      <Fis pron='0' strength='0.0'/>
      <G pron='1' strength='1.0'/>
      <Gis pron='0' strength='0.0'/>
      <A pron='0' strength='0.0'/>
      <Ais pron='0' strength='0.0'/>
      <B pron='0' strength='0.0'/>
    </pitch>
    <period number='1'/>
    <period number='2'/>
  </metric>
  <progression number='1'>
    <candidates repeat='1'>
       <candidate node-id='19' key='C'
         scale='4' basicspace='4.0'>
        <next node-id='24' cost='0'/>
        <next node-id='25' cost='3'/>
        <next node-id='26' cost='4'/>
      </candidate>
     </candidates>
    <harmony repeat='1' key='C' scale='4'/>
  </progression>
</measure>
```

図 13　和声解析結果の MusicXML 表現

表 1　楽曲解析のための XML 拡張における要素

要素名	概要
metric	拍節（解析単位）を示す要素
constitution	拍節内の構成音をまとめる要素
component	拍節内の構成音を表す要素
pitch	拍節内で発音されている音名を表す要素
period	楽節番号を示す要素

5.2　楽曲情報指標 metric

本節では楽典の解釈の指標について説明する．metric は主に楽曲情報について記述する．measure 内に生成され，拍節を単位として拍節毎の構成音についての情報が取得できる．属性として拍節番号を表す number と強拍，弱拍を表す unit-beat がある．生成した要素を表 1 に示す.

5.2.1　constitution による拍子を考慮した構成音表現

求めた拍節毎で発音されている音符をまとめる constitution を記述する．constitution 内には構成音のひとつひとつを表す component が記述される．属性として ref と beat があり，ref は note の id と関連付けされているため，note

表 2　和声解析情報要素 progression

要素名	概要
progression	拍節毎の和声を表す要素
candidates	和音候補をまとめる要素
candidate	和音候補を表す要素
next	次の拍節の和音候補を表す要素
harmony	和声を表す要素

の要素を直接取得できる．beat は強拍，弱拍を示しており，強拍であれば”down”，弱拍であれば”up”と記述する．

5.2.2　pitch による構成音表現

拍節内で発音されている構成音を表現する pitch を記述する．C から B までの音名を示した要素が pitch 内で記述され，それぞれに属性 pron と strength がある．pron は発音を示しており，発音されていれば 1，されていなければ 0 という表現となる．strength は和声的強度を考慮した構成音表現である．

5.2.3　period による大楽節毎の分割表現

period は楽節を表す要素で，属性 number により楽節番号を記述する．楽節番号を指定することで，楽節毎の構成音を取得する．例では period が 2 つ記述されているが，これは繰り返しにより 2 回小節を通っているという解釈となる．

5.3　和声解析情報要素 progression

progression では，主に和声の情報を記述する．measure 内に生成され，拍節を単位として拍節毎の和声についての情報が取得できる．和声解析の結果を，中間処理を含めて要素 progression に記述することで，還元譜を生成する．生成した要素を表 1 に示す．

5.3.1　和音候補 candidates

和音候補をまとめた要素 candidates を記述する．属性として repeat があり，何回目に通ったかの情報を整数値で表現する．candidates 内に和音候補 candidate が生成される．

candidate は 1 つの和音を表し，ひとつひとつに属性として node-id が付与される．key は調を英語表記で表し，scale は音度を整数値で表す．また，求めた類似度を属性 basicspace に記述する．

5.3.2　和音候補グラフを用いた next 生成

和声解析では，求まった和音候補をノードとし, 解析区間を時系列とした和音候補グラフを生成する．MusicXML には candidate 内に隣接ノードを表す next を記述する．candidate から遷移可能なノードを next として記述する．candidate の属性 node-id と対応付けされた node-id が付与されており，node-id から candidate を参照することができる．また，求めた遷移コストを next の属性 cost に記述する．

5.3.3　和声解析結果表現

Viterbi アルゴリズムによって求まった和声を progression 内の harmony に記述する．属性として key と scale があり，candidate における key，scale と同様の記述方法である．また属性 repeat も記述することで，繰り返し処理にも対応した和声を記述する．

6.　おわりに

本研究では，楽典を考慮した和声解析手法と，和声解析フレームワークを提案した．

和声解析手法では，MusicXML へ拍節の情報を付与することで，強拍，弱拍を考慮したピッチクラス変換が可能となり，非和声音を多く含む楽曲も正しく認識することができた．TPS を応用した和音間距離計算式により，和声を近似し，ベーシックスペースによる和音候補列挙法から構成音から和音候補を認識した．カデンツを考慮した進行の絞り込みを TPS と Viterbi に頼り切っている点や，臨時記号などまだ考慮できていない楽典も存在するため，今後は様々な楽典を考慮できる和声解析を目指す．

和声解析フレームワークでは，楽譜から拍子ごとの音符収集から和声解析解析までを自動化した．これにより和声解析を自動作曲や自動編曲といった研究への応用を可能にした．以上のことから，提案手法は楽曲解析，和声解析において有用なシステムであることを示した．今後は和声解析以外の楽曲構造解析における拡張も対応することにより，より高度な楽曲解析が自動化できるフレームワークを目指す．

参考文献

[1] 島岡 譲: 和声ー理論と実習（1），音楽之友社（1964）．
[2] 須見 康平, 糸山 克寿, 吉井 和佳, 駒谷 和範, 尾形 哲也, 奥乃 博: ベース音高と和音特徴の統合に基づく和音系列認識, Oxford University Press（2001）．
[3] 川上 隆: HMM を用いた旋律への和声付けに関する研究, 修士論文, 北陸先端科学技術大学院大学（2000）．
[4] 横井 史也, 平田 圭二, 竹川 佳成: ルール処理と HMM を統合した和音認識器の実現, JSAI（2015）．
[5] Fred Lerdahl. *Tonal Pitch Space*, Oxford University Press（2001）
[6] 小玉 貴史, 東条 敏:モード依存の Tonal Pitch Space, 第 29 回人工知能学会（2015）．
[7] Sakamoto, S., Matsubara, M., Arn, S. and Tojo, S.: ”Harmonic Analysis based on Tonal Pitch Space.” IEEE Conference on Knowledge and Systems Engineering, 2016.
[8] Thomas Rocher,Matthias Robine,Pierre Hanna,Robert Strandh.: *Dynamic chord analysis for symbolic music* , In Proceedings of the International Computer Music Conference（2009）．
[9] 石桁 真礼生，末吉 保雄，丸田 昭三，飯田 隆，金光 威和雄，飯沼信義: 楽典 理論と実習 (1965)
[10] Complex Types MusicXML: 入手先 ⟨http://www.musicxml.com/UserManuals/MusicXML/Content/CT-MusicXML.htm⟩（参照 2017-07-07）．
[11] 田村 直良, 後藤 大輔, 後藤 敏行: 楽譜記述言語 MusicXML からの楽譜自動点訳, 第 5 回情報科学技術フォーラム（2006）．
[12] 北原 鉄郎，橋田 光代，片寄 晴弘: 音楽情報科学研究のための共通データフォマットの確立を目指して, SIGMUS（2007）．