

楽譜記述言語と正規表現を用いた形式的楽曲分析

Formal Musical Composition Analysis Using Music Notation and Regular Expression

森本 祥一
Shoichi Morimoto

1. まえがき

楽譜は、楽曲を記録するために演奏記号や音符、符号を使って記号化したものである。そこには音高や音価、演奏のための演奏記号といった様々な情報が含まれる。これらの情報をコンピュータで扱うために、様々な楽譜記述のための記法(楽譜記述言語)が提案されている[1]。これらの大半はそれぞれが独自の方法で画像表示したり、音声出力したりする機能を提供している。

一方で、楽譜上に音符を並べる際に守らなければならない規則がある。高さが異なる複数の楽音が同時に響いた音を和音というが、これらの和音同士を組み合わせで連結した時の相性の良い組み合わせと悪い組み合わせに関する規則をまとめた和声法がある[5]。また、複数の旋律が同時に鳴り響いても独立性を保ってバランスよく調和する状態を作り出す規則をまとめた対位法という理論もある[4],[7]。その他にも音および音楽に関する諸規則が多数存在し、これらを総称して音楽理論という。音楽教育においても、楽曲を理解するために重要な知識となっている。

このような音楽理論について、工学的なアプローチによる様々な研究がなされている[9]。国内でも、与えられたバスパートについて和声法に基づき上三声を導出するシステム[10]やソプラノパートに対して下三声を導出するシステム[1]、確率モデルに基づいて和声法や対位法を実現する研究[11]、音楽理論を論理学で捉え推論により音楽を創造する研究[8]、楽譜上で音楽理論の自動分析を行う研究[12]等、多数の研究が功績をあげている。

これらの各研究では、それぞれ入力となる“音”を表現する形式は異なる。複数の形式を扱っている研究もあるが、やはり限定的である。もちろん各記法間である程度相互変換がサポートされているが、完全に相互変換である記法は僅かである。

インターネット上などにこれだけ様々な記法で記述された楽曲が氾濫している現在、これらを統一して扱えることが望ましい。それにより、様々な記法の楽曲(楽譜)のデータを有効な資源として万人が利用できるようになる。つまり、利用者は楽曲がどのような記法で記述されているかを意識せずに、音楽理論の分析が可能になる。

よって本稿では、いかなる楽譜記述言語で記述された楽曲の上でも音楽理論に基づいた分析を行える技法について検討する。本研究の最終的な目標は、音楽理論を適用する対象となる楽譜がどのような形式でも統一して処理できるツールの開発である。更に『いつでも・どこでも・誰でも』といったユビキタス性を重視し、インターネット経由でブラウザにより気軽に利用できるツールを目指す。



図1 楽譜記述言語により記述された楽譜例

2. 楽譜記述言語と正規表現

楽譜記述言語は、大きく分けて (1) XML, (2) Ascii, (3) バイナリ形式の三種類に分類できる[1]。(1)の代表的なものとしては MusicXML や SMDL がある。XML 形式の楽譜記述言語は、データの拡張性や汎用性が強みである。(2)で著名なものとしては GUIDO[2]や abc, MusiXTeX, (3)では MIDI などがある。これらの記述言語間では、互換性がある、もしくは変換ツールが提供されている場合もある。

本稿では、これらの記述言語のうちまずは最もシンプルかつ直感的に理解しやすい記述である(2)について扱うこととした。

Ascii 形式の楽譜記述言語の大半は、音高を一般的な音名である a~g のアルファベットで、音価は数字で表す。以下はそれぞれ GUIDO と MusiXTeX のプリプロセッサである M-Tx による記述であり、どちらも図1に示す楽譜を表している。

- (GUIDO の場合)

```
[\meter<"4/4"> {c,e,g} {d,f,a} {d/2,g/2,b/2}]
```

- (M-Tx の場合)

```
Meter: 4/4
```

```
c d d2 []
```

```
C: eg fa gb []
```

このように、アルファベットと数字を使うことは共通しており、記法にもある程度の共通性がある。また、これらの記述は全て文字列として捉えることが可能である。つまり、楽譜上の全ての情報を文字列として表現することができる。

文字列として記述された楽曲が、各種音楽理論における規則を満たしているかを判断する、または規則を満たすように記述する、ということは、つまりは文字列に対する操作である。このような文字列に対する処理には、正規表現が有効である。正規表現を用いることにより、任意の文字列からある規則を満たした部分を探し出したり、特定の部分文字列に対して処理をしたり、といった文字列操作を容易に行うことができる。

†産業技術大学院大学 産業技術研究科,
School of Industrial Technology, Advanced Institute of
Industrial Technology

以上より、本稿では以降 Ascii 形式の楽譜記述言語で記述された楽譜に対し、正規表現を用いて音楽理論に基づいた楽曲分析を行う技法を考察する。

3. 正規表現による楽曲の分析

本章では、実際に正規表現を用いて各種音楽理論の規則を分析する技法について述べる。

3.1 対旋律の分析

対位法は音楽理論のひとつであり、複数の旋律が独立性を保ちつつ調和するよう重ね合わせる技法である。楽譜記述言語により記述された楽曲における旋律が、この対位法で定められた規則に基づいているかどうかを正規表現によりどのように分析するかについて述べる。

以下に自由対位法で定められた各旋律が守らなければならない規則のうち、幾つかを挙げる。但し、説明の簡単化のため使用する和音は I・IV・V7 とし、I IV I V7 I IV V7 I の和音進行を使用する。

原則 1: V7 における内音である B と F は、それぞれ B は C へ、F は E へ限定進行する (図 2)。



図 2 限定進行

原則 2: 増 1 度以外の増音程は禁止する (図 3)。

原則 3: 垂直音程は 3 度と 6 度のみ使用する (図 3)。曲頭では、1 度、5 度、8 度も使用することができる。



図 3 増音程の禁止と垂直音程

楽曲における各旋律が、原則 1 を満たしているかどうかを判断するには、B 音の直後に C 音がきているか、F 音の直後に E 音がきているかを確かめればよい。楽譜記述言語で記述された楽曲においては、文字列の中に b の次に c があるか、f の次に e があるかを判定すればよいが、音価やオクターブを示す数字が後ろに付く可能性もあるため、以下のような正規表現でマッチングを行う必要がある。

```
b \W* \d? \W? \d? \s* (?!c) | f \W* \d? \W? \d? \s* (?!e)
```

*と?は量指定子であり、*は直前の要素が 0 回以上、?は 0 か 1 回連続することを示す。また\d は 0 から 9 の任意の数字を、\s はスペースを示す。音の次にシャープやフラツ

トがくることもあり、また言語によっては音価を示す数字とオクターブを示す数字の間にスラッシュやプラスといった記号が挿入される場合があるため、\W によりアルファベットと数字以外の文字が間に入ることも想定している。?!は否定の先読み言明 (前方参照位置指定) であり、直後にその文字列がない場合にマッチする。つまり、上記の記述は、音価やオクターブを示す数字が後ろに付く・付かないに関係なく、B 音の次に C 音が、F 音の次に E 音がこない部分を表現している。楽譜記述言語で記述された楽曲文字列において、上記のパターンにマッチする部分があったならば、限定進行に違反していることが分かる。また、限定進行に違反している箇所が見つかった場合は文字列置換で b の次の文字を c に、f の次の文字を e に修正することが可能である。

次に原則 2 と原則 3 について考える。これらの規則は二音間の垂直音程が何度であるかを求める必要がある。このように音程を用いた判断は、対位法に限らず音楽理論において頻繁に用いられるため、以下のような擬似コードで表される処理を行う二音間の音程を求める関数を用意した。

```
function distance ($str1, $str2)
```

```
if ($str1 == 'c' && $str2 == 'd' | ...) return 長 2 度
else if ($str1 == 'c' && $str2 == 'e' | ...) return 長 3 度
else if ...
```

各音程になる二つの音の組み合わせを全て列挙しておき、比較したい二つの音を文字として引数で受け取り、それらがそれぞれ何度になるかを条件分岐で判断する。比較する二音の文字は、それぞれ正規表現で抽出し、関数に渡す。例えば、c と d が引数として渡された場合は長 2 度を返す。この戻り値を使って、増 1 度以外の垂直音程がないか、曲中に 3 度と 6 度以外の垂直音程がないかといった判断をすることができる。また、曲頭がどうかは、アンカー^を利用して判断することができるので、原則 2 と原則 3 についても分析可能である。

3.2 和声の分析

和声法は対位法と同様、音楽理論のひとつであり、和音の進行や各声部の付け方や配置を行う技法である。楽譜記述言語により記述された楽曲における和音や声部が、和声法で定められた規則に基づいているかどうかを分析する技法について述べる。

以下に和声法におけるカデンツの原則から導出した和音進行の可能性の規則を挙げる [5]。

規則 1: I の和音の次は II, IV, V, または VI がくる。

規則 2: II の和音の次は V がくる。

規則 3: IV の和音の次は I, II, または V がくる。

規則 4: V の和音の次は I または VI がくる。

規則 5: VI の和音の次は II, IV, または V がくる。

これらの規則についても、対位法の場合と同様に正規表現を用いて判定することができる。但し、前述の GUIDO と M-Tx の記述を比較しても分かる通り、和音の記述の仕方は楽譜記述言語によって多少異なるため、個別の対応が必要になる。例えば GUIDO の場合、調が C major で三和音

(図4)とすると、規則2について次のような正規表現でマッチングを行う。

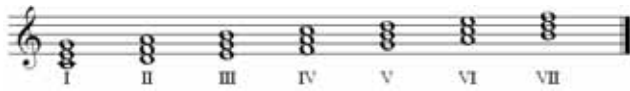


図4 C major における基本位置の三和音

```

\{ d \W* \d? \W? \d?, f \W* \d? \W? \d?, a \W* \d? \W? \d?
\} \{ s* (?! \{ g \W* \d? \W? \d?, b \W* \d? \W? \d?, d \W* \d?
\W? \d? \} )
    
```

つまり上記の表現は、IIの和音(D, F, A)の直後がV(G, B, D)の和音でない部分を表現しており、和音進行に違反している部分を見つけることができる。基本位置以外の和音についても直後にくるべき転回位置の和音を論理和で繋げて列挙することで、同様に違反箇所をマッチングすることが可能となる。II以外の和音進行についても同様である。

前述のように、言語によって和音の記法が多少異なるため、上記の表現のみでは全ての言語で和音進行違反を見つけることはできないが、規則自体が変わるわけではないので言語に合わせて多少修正した正規表現を追加することで簡単に解決できる。

3.3 楽式の分析

続いて楽式の分析法について述べる。楽式とは、楽曲の形式のことであり、曲全体の構成により幾つかの形式に分類することができる。例えば、提示部・展開部・再現部の三部で構成されるソナタ形式や、同じ旋律が異なる旋律を挟んで繰り返されるロンド形式といったものがある。楽式を判断するには、あるまとまった楽節が複数回現れるかどうかを判定すればよい。つまり、楽節を部分文字列とし、楽曲を記述した文字列全体に決まった展開で楽節の文字列が現れるかを判定する。このようなパターンマッチングは正規表現の得意とするところであり、容易に実現できる。

楽式分析の例として、カノン形式であるかどうかを判定する例を示す。カノン形式とは、対位法による楽曲のひとつで、ある声部が流れ、続いて別の声部が遅れてそれを模倣する。輪唱が良い例である。先行する声部をDux、追う声部をComesという。カノンにも様々な種類が存在するが、ここでは二声順行のカノンの厳格模倣について分析する。

楽曲の任意の一部(トークン)を切り出した時、その部分から全ての隣り合う音同士の音程を前述の関数を使って求める。音程が同じ度で推移している部分が見つかったら、

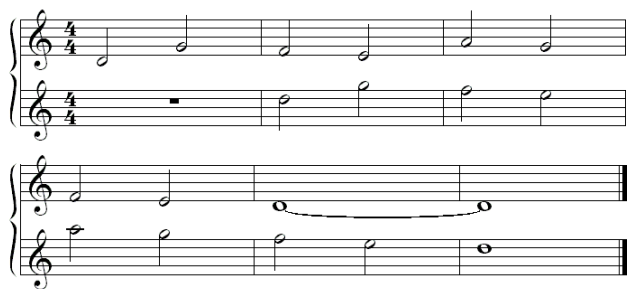


図5 カノンの例

更にそれらの音価を比較し等しければカノンと判定できる。図5に示すようなカノンを考える。これはGUIDOによって次のように記述できる。

```

\{ \meter<"4/4"> d/2 g/2 f/2 e/2 a/2 g/2 f/2 e/2 d/1 d/1,
\meter<"4/4"> _/1 d2/2 g2/2 f2/2 e2/2 a2/2 g2/2 f2/2 e2/2
d2/1}
    
```

上段と下段の休符を除く各音間の音程を前述の関数により度数を求め比較してみると、全て等しいことが分かる。次に、それぞれの音価が等しいかどうかを確かめる。音価を確かめるには、各音の音価のみを次の正規表現で抽出する。例えばPerlでは

```
m/[a-g] \W* \d? \ / (d?)/g
```

とすると、音価の数字の部分のみを得ることができる。上段、下段とも222222221となり等しい。つまりカノンであることが判断できる。

しかしながら、厳格模倣であってもComesが垂直音程を正確に模倣できるのは8度と同度のカノンのみであり、その他の場合は調によってはDuxとComesの音程の推移が等しくならない可能性がある[6]。反転カノンや逆行カノンも同様であり、更に拡大カノンや縮小カノンはDuxとComesの音価が等しくならない。よって、調号を取り除いて比較する、音程の推移を逆から比較する、各音価が倍数になっているかを判断する等、特別な処理が必要となる。例えば、図6のような2度のカノン进行分析する場合を考える。但しこの譜面は、楽曲の一部である。これをGUIDOにより記述すると以下ようになる。

```

\{ \meter<"2/4"> \key<"A"> c#4 | a/2 a/8 g#8 f#8 e/8 c#2/2,
\meter<"2/4"> \key<"A"> _/4 | _/4 d/4 b/2 b/8 a/8 g#8 f#8}
    
```

ここでは調号を除いて切り出した上段の文字列caagfeと下段の文字列dbbagfとの各音間の音程を比較すると等しく、また前述の正規表現で音価を比較すると428888...と等しくなり、カノンと判定できる。



図6 2度のカノン

4. 考察

3章で示したように、楽譜記述言語で記述した楽曲を文字列として正規表現で分析することにより、音楽理論における一部の規則を判定できることが分かった。また、分析だけでなく、文字列の置換を用いることで違反している箇所の修正や和声付け、対旋律付けも可能である。更に、カノンの分析において音価を抽出できることを示したが、これを利用してリズムの分析も可能である。

但し、正規表現では「どちらでも可」や「なるべく避ける」等の恣意的な判断を含む規則を分析することは難しい。

何らかの人手の介入が必要となる。同様に、和声付けや対旋律付けにおいて候補が多数ある場合などは、人手による判断が必要になる可能性がある。

また本稿で示した規則は比較的単純なものであり、また各音楽理論のうちほんの一部に過ぎない。更に各規則は個々に適用するのではなく、同時に満たしていなければならない規則が大半である。これら全てについて、正規表現を用いて確かめたわけではないが、楽譜記述言語により記述された楽曲は文字列となるので、多少判断条件が複雑になる可能性があるが、理論的には分析が可能である。

本稿では主に GUIDO や Perl を使ったが、正規表現を利用できる他のスクリプト言語でも可能であり、例で示した正規表現の基本的なアルゴリズムは他の楽譜記述言語でも共通で使用できる。また本稿では Ascii 形式の楽譜記述言語のみを対象としたが、XML による記述も同様に文字列であるので正規表現で一括して扱うこともできる。この場合は、XML としての特性を活かすことはできない。

5. おわりに

本稿では、楽譜記述言語により記述した Ascii 形式の楽曲を、正規表現を用いて音楽理論に従っているかどうか分析できることを示した。楽譜記述言語で記述することにより楽曲全体を文字列として扱うことが可能となり、正規表現を用いて楽曲の理論的な側面を文字列処理として実装し分析することができる。また、正規表現は PHP, Perl, Ruby といったスクリプト言語で利用できるため、CGI を用いてインターネット経由でいつでも誰でも利用できる Web サービスを構築することが可能である。我々は本稿で示した方法を用いて比較的単純な理論の分析を Web ブラウザから実行できるツールのプロトタイプ版(図7)を Perl で実装した。

今後は、各音楽理論の全ての規則を正規表現での文字列操作として順次実装していき、実現できた理論とできなかった理論、実装はできたが非効率な処理、といった項目を分析し評価する。

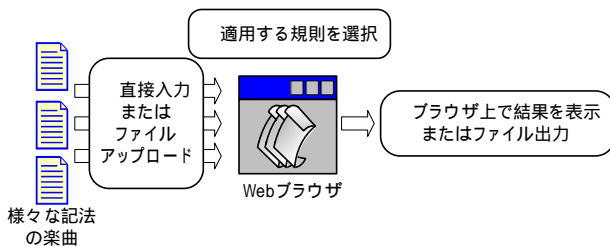


図7 プロトタイプ機能

参考文献

- [1] Castan, G., "Music Notation – formats –," <http://www.music-notation.info/>
- [2] Hoos, H., Hamel, K., Renz, K., and Kilian, J., "The GUIDO Notation Format -- A Novel Approach for Adequately Representing Score-Level Music," Proceedings of International Computer Music Conference 1998, pp. 451-454, 1998.
- [3] Miura, M., Obana, M., and Yanagida, M., "Evaluation of soprano donnee system for the theory of harmony," Acoustical Science and Technology, Vol. 23, No. 6, pp.328-

331, 2002.

- [4] 池内 友次郎, "二声対位法," 音楽之友社, 1965.
- [5] 池内 友次郎 他, "和声 理論と実習", 音楽之友社, 1964.
- [6] 岸辺 成雄 編, "音楽大事典 1 ~ 6," 平凡社, 1983.
- [7] 長谷川 良夫, "対位法," 音楽之友社, 1955.
- [8] 平田 圭二, "音楽知プログラミング試論," NTT R&D (社) 電気通信協会, Vol.50, No.9, pp.682-689, 2001.
- [9] 平田 圭二, "昨今の音楽情報処理における研究プロジェクトについて," 人間主体の知的情報技術に関する調査研究 VI, (財) 日本情報処理開発協会 先端情報技術研究所, pp. 179-192, 2003.
- [10] 三浦 雅展, 下石坂 徹, 斉木 由美, 柳田 益造, "和声学におけるパス課題についての回答確認システムの構築とその評価," 電子情報通信学会論文誌, Vol. J84, D-II, No. 6, pp. 936-945, 2001.
- [11] 米田 隆一, 西本 卓也, 嵯峨山 茂樹, "マルコフ確率場を用いた調認識, 自動和声付け, および自動対位法," 情報処理学会 研究報告 音楽情報科学, Vol.2005, No.129, pp.31-36, 2005.
- [12] 劉 剣利, 平賀 瑠美, 五十嵐 滋, 関口 由浩, "楽曲分析システム DAPHNE - 実際の楽譜上での自動分析," 情報処理学会 研究報告 音楽情報科学, Vol.1999, No.68, pp. 1-6, 1999.