

音高変化のパターン抽出による  
変奏曲の構造解析

筑波大学  
図書館情報メディア研究科

2013年3月

二本松 由明子

# 目次

|       |                               |    |
|-------|-------------------------------|----|
| 第 1 章 | はじめに                          | 6  |
| 1.1   | 本論文の流れ                        | 6  |
| 第 2 章 | 基本事項                          | 7  |
| 2.1   | 楽曲の構造解析                       | 7  |
| 2.1.1 | 抽象化による構造                      | 7  |
| 2.1.2 | パターンによる構造と音高変化                | 9  |
| 2.1.3 | 音高変化のパターンと簡約化                 | 10 |
| 2.2   | 楽曲のデータ形式                      | 11 |
| 2.3   | 変奏曲                           | 13 |
| 2.3.1 | 変奏曲とは                         | 13 |
| 2.3.2 | 変奏の技法                         | 16 |
| 第 3 章 | 研究概要・目的                       | 19 |
| 3.1   | 本研究で対象とする楽曲                   | 19 |
| 3.2   | 変奏曲の構造解析                      | 19 |
| 3.3   | 音高変化のパターンの分析と簡約という視点からの主題変奏関係 | 20 |
| 3.4   | 第 2 章以降の流れ                    | 21 |
| 第 4 章 | 先行研究                          | 23 |
| 4.1   | 音列パターン分類に基づく楽曲の主題・変奏関係の解析     | 23 |
| 第 5 章 | 前処理                           | 27 |
| 第 6 章 | 処理 1:特定の音高変化パターンの抽出と簡約化       | 30 |
| 6.1   | 概要                            | 30 |
| 6.2   | 手法                            | 30 |
| 6.2.1 | 辞書パターンと重要だと考えられる音             | 33 |
| 6.2.2 | 抽象化を用いた一致判定                   | 34 |
| 6.2.3 | 辞書パターン同士の関係                   | 35 |
| 6.3   | 出力例と考察                        | 35 |
| 第 7 章 | 処理 2:正規表現を用いたパターン辞書への対応       | 39 |
| 7.1   | 概要                            | 39 |
| 7.2   | 手法                            | 40 |

|        |                      |    |
|--------|----------------------|----|
| 7.3    | 出力例と考察               | 44 |
| 第 8 章  | 処理 3:楽曲に頻出するパターンの辞書化 | 47 |
| 8.1    | 概要                   | 47 |
| 8.2    | 手法                   | 47 |
| 8.2.1  | 音高変化列中のパターン出現数の計算    | 48 |
| 8.2.2  | 類似するパターンの判断          | 50 |
| 8.2.3  | パターン出現数の計算結果         | 50 |
| 8.2.4  | 辞書化するパターンの条件         | 51 |
| 8.2.5  | パターンの辞書化             | 53 |
| 8.2.6  | パターン辞書の統合            | 55 |
| 8.3    | 出力例と考察               | 56 |
| 8.3.1  | 生成した辞書パターンの分析        | 56 |
| 8.3.2  | 生成した辞書を用いたパターン抽出と考察  | 61 |
| 第 9 章  | まとめ                  | 70 |
| 第 10 章 | 謝辞                   | 73 |
| 第 11 章 | 参考                   | 74 |

## 表目次

|   |  |    |
|---|--|----|
| 1 | トラック数が2のMIDI形式のファイルの概要 . . . . .       | 12 |
| 2 | 辞書パターンとして用いたパターン一覧 . . . . .           | 31 |
| 3 | 辞書パターン up ([表2]の一部を再掲) . . . . .       | 34 |
| 4 | 辞書パターン up ([表3]の再掲) . . . . .          | 39 |
| 5 | 正規表現パターンと該当する文字列の例 . . . . .           | 40 |
| 6 | 辞書パターンとして用いたパターン一覧 ([表2]の再掲) . . . . . | 41 |
| 7 | [表6]に対応する正規表現辞書パターン . . . . .          | 42 |
| 8 | 正規表現辞書パターンの例 ([表7]の一部) . . . . .       | 53 |

## 図目次

|    |   |    |
|----|---|----|
| 1  | 抽象化による構造の例：階名とリズム                         | 8  |
| 2  | 抽象化による構造の例：変化の方向                          | 8  |
| 3  | 顕著なパターンの例                                 | 9  |
| 4  | MIDI ノート番号と鍵盤の対応                          | 13 |
| 5  | 「きらきら星変奏曲」第1変奏（右手）冒頭の MIDI ノート番号          | 13 |
| 6  | 区分的変奏曲の例                                  | 15 |
| 7  | 連続的変奏曲の例                                  | 16 |
| 8  | a) 和声の変化の例（ベートーヴェン：ピアノソナタ Op.2 の1）        | 17 |
| 9  | b) 動機の延長の例（ベートーヴェン：弦楽四重奏曲 第一番 Op.18 の1）   | 17 |
| 10 | f) 旋律の装飾の例（ベートーヴェン：ピアノソナタ Op.53）          | 18 |
| 11 | きらきら星変奏曲 主題と変奏の冒頭（右手）の比較と分析               | 20 |
| 12 | 音列パターンの定義                                 | 24 |
| 13 | 「きらきら星変奏曲」第1変奏（右手）冒頭の MIDI ノート番号（[図5]の再掲） | 25 |
| 14 | 「きらきら星変奏曲」処理結果                            | 26 |
| 15 | 音高変化の数値列計算例                               | 28 |
| 16 | 「きらきら星変奏曲」第1変奏（右手）冒頭の MIDI ノート番号と音高変化     | 29 |
| 17 | パターン抽出と重要な音判定の例                           | 32 |
| 18 | パターンによる簡約プログラムの流れ                         | 33 |
| 19 | turnA-を表す装飾記号                             | 33 |
| 20 | ハ長調とイ長調の音階と音高変化                           | 35 |
| 21 | 「きらきら星変奏曲」第1変奏右手について抽出を行った結果              | 36 |
| 22 | 「きらきら星変奏曲」第1変奏右手について抽出を行った結果の一部（1回目）      | 37 |
| 23 | 「きらきら星変奏曲」第1変奏右手について抽出を行った結果の一部（2回目）      | 37 |
| 24 | 「きらきら星変奏曲」第3変奏右手について抽出を行った結果の一部           | 38 |
| 25 | 正規表現パターンによる簡約プログラムの流れ                     | 43 |
| 26 | 「きらきら星変奏曲」第1変奏右手について抽出を行った結果              | 45 |
| 27 | 頻出パターン辞書化プログラムの動作概要                       | 48 |
| 28 | パターン出現数計算の流れ                              | 49 |
| 29 | パターン出現数計算結果の一部                            | 51 |
| 30 | [図29]のうち辞書化する条件に適合するパターン                  | 52 |
| 31 | 生成した辞書ファイルの例（最低音数以上、変化の幅なし）               | 54 |
| 32 | 生成した辞書ファイルの例（最低音数、変化の幅なし）                 | 54 |
| 33 | 生成した辞書ファイルの例（最低音数以上、変化の幅あり）               | 55 |

|    |  |    |
|----|--|----|
| 34 | 生成した辞書ファイルの例（最低音数、変化の幅あり） . . . . .                        | 55 |
| 35 | 生成辞書内のパターン分類 a) の出力例 . . . . .                             | 58 |
| 36 | [図 35] をパターン数順に並べ替えたもの . . . . .                           | 59 |
| 37 | 生成辞書内のパターン分類 b) の出力例（パターン数の多い順） . . . . .                  | 60 |
| 38 | 「きらきら星変奏曲」第 1 変奏右手から、生成した辞書パターンを抽出した結果 . . . . .           | 62 |
| 39 | 「きらきら星変奏曲」第 1 変奏右手から、特定パターンの抽出を行った結果（[図 26] の再掲） . . . . . | 63 |
| 40 | 「きらきら星変奏曲」第 1 変奏右手から、生成した辞書パターンを抽出した結果（一部） . . . . .       | 64 |
| 41 | 変化の幅の設定により矛盾が生じる例 . . . . .                                | 65 |
| 42 | ピアノソナタ K.331 第 1 楽章 第 1 変奏右手に生成した辞書を適用した結果 . . . . .       | 67 |
| 43 | 複数の旋律でパターンが見られる例 . . . . .                                 | 69 |

# 第1章 はじめに

人間が音楽を聴く、あるいは演奏、作曲をするなどの音楽活動を行うにあたって、「楽曲の構造を解析する」ことが重要であると考えられている。しかし、「楽曲の構造」という言葉が具体的に何を指すのかということについては人によって様々な解釈や意見があり、共通に認められた定義は存在しない。ここでは「楽譜に明記されないが、演奏や聴取にあたって認識される特徴」のことを幅広く指すこととする。

楽曲の構造解析は、音楽に関する様々な場面で行われていると考えられている。例えば音楽を聴いていて、楽曲中の旋律に区切りやまとまり（グルーピング構造）があることや、2つの旋律が類似していることが直感的に理解できるのは、旋律の区切りや共通性という楽曲の構造を聴き手が無意識に解析しているからだと言える。他にも、音楽学において楽曲の意図を理解するための「アナリゼ」という楽曲分析の観点としても、楽曲の構造に着目することがある。研究分野においても、Lerdahlらが計算機的に楽曲構造解析を行うための理論 GTTM<sup>1)</sup>を提案し、浜中らがその理論を基に楽曲のグルーピング構造を自動抽出する研究<sup>2)</sup>などを進めているが、人間がどのような手順で楽曲構造の分析を行なっているのか、その詳細については分かっていない。本研究では楽曲中の音高の変化に着目し、楽曲の構造解析の自動化を試みることで、音楽活動の支援や、人間の音楽認知の解明の手がかりとなることを目的とする。

## 1.1 本論文の流れ

本論文では、変奏曲中の音高変化のパターンを抽出し、それを基に簡約を行うシステムについて述べる。論文全体の流れとして、まず第2章では本研究に関する基本事項として、主に楽曲構造や楽曲を扱うデータ形式、そして変奏曲の説明を行う。次に第3章にて本研究の概要と目的を述べ、第4章では先行研究を紹介する。第5章では本研究で扱うシステムに共通する前処理について説明する。そして、第6章、第7章および第8章にわたり本研究で実際に取り組んだ内容について述べ、第9章にてまとめを行う。

## 第2章 基本事項

### 2.1 楽曲の構造解析

楽曲構造としてどのようなものが考えうるかということに関して、『音楽とコンピュータの世界』4章1節「音楽分析・認知研究への招待」<sup>5)</sup>では、次の4つが挙げられている。

- 分割による構造
- 制約による構造
- 抽象化による構造
- パターンによる構造

まず、分割による構造とは、全体は部分構造に分割できる、逆に部品を組み合わせた複合物として構造を捉えるという観点である。例として、楽曲は複数の楽章から成り、楽章は小節の集まりで構成され、小節は拍に分割されるという構造がこれにあたる。他にも、楽曲全体がいくつの部に分かれているかという観点で分類を行う音楽形式の理論（楽式、楽曲形式）が挙げられる。例えば楽曲形式の1つであるソナタ形式は、楽曲が序奏（含めないこともある）、提示部、展開部、再現部、終始部から構成される楽曲のことを表している。

これに対して制約による構造とは、曲が音楽としてまとまっていたり、特定のスタイルやジャンルに属するために満たすべき条件から捉える構造のことを指す。例として西洋の調性音楽を挙げる。調性音楽とは楽曲が何らかの調性（ハ長調、ニ短調など）を持っているということを指し、この調性によってそれぞれの高さの音の役割や重要性が決まる。このようにして生まれる構造が制約による構造である。

抽象化による構造とは、分割や制約による構造のような個々の単位や属性に共通して存在する性質や特徴である。また、パターンによる構造とは、抽象化や分割の結果が特徴的な事例に集約され、曲や旋律の照合や識別といった操作の対象となりうるものである。これらについては本研究との関連が深いため、以下で詳しく述べる。

#### 2.1.1 抽象化による構造

例えば [図1] の上段にある、童謡「ちょうちょ」の冒頭の2小節について考える。図の下段の2小節は、上段の各音を1オクターブ高くしたものである。図中で青文字で示したのは上段、下段それぞれの音の階名である。これを見ると、上段と下段で音の高さは異なっているが、音の高さを抽象化した両者の階名は一致したままであることが分かる。また、音の高さの情報を無視し、リズムにのみ焦点を当てると、1小節目と2小節目は4分音符が2つ並んで「タタ」というリズムのあと、2分音符の「ター」という長さの音が続いているという点で共通性が見られる。このようにして、余分な情報を切り捨てることによって浮かび上がる共通の性質が、抽象化による構造である。





図1 抽象化による構造の例：階名とリズム

抽象化においては、これを段階的に行うことで異なる性質や特徴を得ることがある。例えば、[図1]の例では音の階名が一致していると述べたが、この音列をさらに抽象化することを考える。抽象化の1つの観点として、NarmourのIRM理論<sup>3)</sup>では、変化の向きと大きさに着目している。変化の向きとは、ある音と次の音の高さを比較した時に、次の音のほうが高いのか低いのか、それとも同じ音の高さであるのかということ、変化の大きさとは次の音との差が大きいのか小さいのか、それとも等しいのかということを表す。これに関連して、Walterらの”Melodic Similarity”<sup>4)</sup>でも、単旋律の音高のデータ表現の1手法として音高の変化する方向をU(次の音の方が高い) / R(次の音と同じ高さ) / D(次の音の方が低い)で表すことについて述べている。[図1]の各小節ごとに、隣合う音の音高の変化する方向をこれに倣って表すと、[図2]のようになる。[図1]で着目した階名の特徴は、上段全体と下段全体で共通する特徴



図2 抽象化による構造の例：変化の方向

であり、1小節ごとに着目した際は、1小節目同士、2小節目同士では階名が一致するものの、1小節目と2小節目では一致しなかった。しかし[図2]では、各小節の3つの音の変化はどれも「D,R」で一致しており、この特徴は全てに共通している。このように、抽象化のレベルを変えることで、異なる性質を得ることができる。

### 2.1.2 パターンによる構造と音高変化

パターンによる構造とは、抽象化などの構造の結果として得られる特徴的な性質を指す。顕著なパターンの例として、『音楽とコンピュータの世界』<sup>5)</sup>ではベートーヴェンの交響曲第5番「運命」の第1楽章冒頭（[図3]）が挙げられている。この例では、「4つの音のリズムのパターン」と、「最初の3つの音が同じ高さの音であり、最後の1音がそれよりも低い音であるというパターン」が表れている。特にこのリズムのパターンに関しては、このリズムを聞いたらこの曲以外は連想できないほどの強烈な印象を与える、特徴的なものである。この曲の第1楽章でもこのリズムパターンは300回近く登場し、さらに全曲を通じて用いられて曲全体のまとまった雰囲気演出している。



図3 顕著なパターンの例

このような例の他にも、例えば曲のアレンジを変えたり、長調を短調に変えたりといったデフォルメを加えても元の曲が認識できるための手がかりになるものがここでいう「パターン」である。パターンは独立した構造である分割、制約、抽象化による構造とは異なり、実際には抽象化や分割などの結果として得られる構造である。例えば[図3]で見られる、「4つの音のリズムのパターン」とは音の高さという情報を切り捨てることによって浮かび上がる共通の性質、つまり抽象化による構造の結果であると言える。逆に、抽象化による構造の説明で用いた[図1]の「タタター」という共通のリズムも、一つのパターンと言える。また、[図3]の「最初の3つの音が同じ高さの音であり、最後の1音がそれよりも3度（ピアノの白鍵2つ分）低い音である」という性質は、個々の音の具体的な高さ（ソとミ、ファとレ）は考慮せず、4音の中での相対的な高さに着目することで生まれるものである。

パターンによる構造とは他の構造の結果として得られる構造ではあるが、曲のアイデンティティを示し、照合・識別といった操作の対象となる点に独自性があると言える。例えば、分割による構造の例として挙げた「ソナタ形式」のような楽曲形式は、多くの楽曲に共通する構造的な特徴であり、特定の楽曲を指し示しているのではない。これに対しパターンによる構造は、様々な楽曲の中からある特定の楽曲を特徴付けるものである。文字認識に例えると、分割、制約、抽象化による構造を分析することが文字の内容や言語を認識することにあたるのに対し、パターンによる構造の分析は文字の書き手の識別することにあたる。このことから、楽曲におけるパターンを理解することは、その楽曲そのものの特徴を理解することと深く関係すると言える。

パターンという視点からの楽曲構造に関連するものとして、保科の『生きた音楽表現へのアプローチ』<sup>7)</sup>では、作曲家が印象づけたい音群をパターン化し、それを何らかの方法で繰り返すことは最も基本的な

楽曲の構成法だと述べている。また、その結果として生まれた楽曲を分析すると、作曲家が自身の意図を聴衆に伝えるため、いかに特徴的なパターンを生み出すかという点に腐心した跡が見受けられるとの記述もある。保科はパターン化のために作曲家が最も普遍的に活用してきた方法を、「音の長短（音長）」「音の高低（音高）」「音の集積（音積）」という「音符の三機能」に基いて以下のように分類し、考察を行っている。

- 「音長」の組み合わせを活用する方法：「リズム・パターン」
- 「音高」の組み合わせを活用する方法：「フィギュア・パターン」
- 「音積」（和音）の組み合わせを活用する方法：「コード・パターン」

以下では、保科の文献に沿ってそれぞれのパターンについて説明する。まず、リズム・パターンは先述した [図3] に表れる「4つの音のリズムのパターン」のようなものである。次に、フィギュア・パターンとは音高の異なる複数の音を特長ある音型に組み合わせてできるパターンのことを指し、これを旋律において繰り返し提示することは最も常套的な作曲技法である。さらに、一般的にフィギュア・パターンが同じであれば、たとえリズム・パターンが変化しても同じ旋律として知覚される場合が多い。フィギュア・パターンの例としては、[図3] の「最初の3つの音が同じ高さの音であり、最後の1音がそれよりも3度低い音である」というものが当てはまる。コード・パターンとは、響きの異なる複数の和音を組み合わせてできるパターン（和声）のことであり、楽曲内の緊張⇄弛緩という特定の流れを生み出すものである。

保科は3章3節で、「音長」「音高」「音積」の「音符の三機能」のうち、実際の音楽表現における「音高」による表現力は絶大なものであり、音高の変化とは作曲家の情緒や感情の「起伏の軌跡」を表現するための最も普遍的な手段であるため、作曲家の意図をうかがい知るには重要かつわかりやすいヒントであると述べている。また、パターンの繰り返しという技法も、作曲家が自身の意図を表現するために用いられている。よって、楽曲中の音高の変化によるパターンを分析することは、聴衆や演奏者が作曲者の意図を読み取る手がかりの1つを理解することであると言える。

### 2.1.3 音高変化のパターンと簡約化

ここまで、音高変化によるパターンは楽曲を特徴づける重要な要素であることを述べた。しかし、この他にも楽曲そのものを特徴づける要素として、楽曲の骨格のようなものが存在する。楽曲の骨格とは、楽曲中の重要な音の集合であり、音列から重要な音を取り出すことを「音列の簡約化」と呼ぶ。『音楽とコンピュータの世界』4章6節「音楽の構造解析とその構造」<sup>6)</sup>では、簡約（還元）について次のように説明されている。

”例えば、変奏曲では1つのテーマを様々な方法で変形するが、人間はそれらを別々の音楽として聴くのではなく、それぞれが1つのテーマから派生していったものだと認知する。つまり、変形された音楽に共通する抽象的な構造を作って相互に関連付けているのである。還元(reduction)とは、音楽の表層構造からこのような抽象的な構造を抽出することを意味している。言い換えれば、装飾や変形が理解できるというのは、構造的に重要な音というものを直感的に理解しているからであり、ある音符が他の音符を装飾していると感じたなら、その音符は他の音符よりも構造的にあまり重要ではないと認識するということである。”

つまり、音高変化によるパターンは変形された音列の「変形の特徴」を表すことに対し、音列を簡約化することは変形された音列から「変形の軸」となる部分を抜き出すことである。従って両者を関連付けて分析を行うことで、その音列が持つ固有の特徴が表現できると考えられる。また、簡約化された音列をさらに簡約化するという形で階層的な簡約を行うことで、音列や楽曲の特徴を段階的に抽象化し、分析することもできる可能性がある。

以上の事柄を踏まえて、本研究では、主に音高変化によるパターン(フィギュア・パターン)の構造分析及び簡約化という視点から楽曲を自動的に分析する手法の検討を行う。まず2.2では、コンピュータで音楽を扱う形式の1つである、MIDI形式について述べる。その後、2.3では分析する対象の楽曲形式である、変奏曲について説明する。対象として変奏曲を用いる理由は、音高変化によるパターンが用いられる部分が多いことと、各パターンに単純でわかりやすいものが多いこと、さらに音高変化によるパターンの分析及び簡約化が、楽曲そのものを理解する上で他の楽曲よりも重要であることが挙げられる。『音楽とコンピュータの世界』<sup>6)</sup>からの引用部分でも変奏曲が例として挙げられており、変奏の派生の基となる1つのテーマとはそのまま主題のことを指すことを踏まえると、この文章から「変奏の簡約は主題(または主題を簡約したもの)である」という、変奏曲における重要な関係が表れている。よって本研究では、変奏の音高変化列からパターンを抽出し、それぞれのパターン内での重要な音を決定する手法について検討を行う。音高変化のパターン分析の具体的な例や、変奏曲の分析に音高変化のパターンと簡約化という視点で取り組む意義について、詳しくは3.3で述べる。

## 2.2 楽曲のデータ形式

楽曲構造の自動分析など音楽を扱う研究では、MIDI形式と呼ばれる形式に基づいたデータを用いることが多い。MIDI形式とは、楽曲情報を楽譜のように符号化した形で表記するファイルである。本節では、本研究でも使用するMIDI形式のデータについて説明する。

MIDI形式のファイルは、ヘッダチャンクとトラックチャンクから構成される([表1])。ヘッダチャンクにはトラック数などのファイル全体の形式に関する情報が格納されている。一方、実際の楽譜データはトラックチャンクとして各トラック毎に格納されている。トラックチャンクはそのトラックのデータ長の情報と、多数のイベントから構成されている。

イベントには次の3つの種類がある。

|            |   |
|------------|---|
| ヘッダチャンク    | トラック数<br>全体のデータ長 など                                 |
| トラックチャンク 1 | トラック 1 のデータ長<br>トラック 1 のイベント<br>トラック 1 のイベント<br>... |
| トラックチャンク 2 | トラック 2 のデータ長<br>トラック 2 のイベント<br>トラック 2 のイベント<br>... |

表 1 トラック数が 2 の MIDI 形式のファイルの概要

- MIDI イベント

個々の音の発音・消音や楽器音指定、演奏制御情報などを指定する  
(例: note on <発音>、note off <消音>)

- SysEx イベント

音源固有のパラメータをコントロールする

- メタイベント

楽曲のタイトルや拍子 (例: 4 分の 4 拍子)、調 (例: ハ長調)、テンポなどの情報を扱う

各イベントには、前のイベントとそのイベントとの開始タイミングの差分時間 (デルタタイム) の情報が含まれている。よって最初のイベントからデルタタイムを合計していくことで、そのイベントの開始時刻が計算できる。

これらのイベントの中で、1 つ 1 つの音符の情報を表すのが、MIDI イベントに属する note on、note off である。note on は発音、note off は消音を意味し、それぞれのイベントは音高を表す数値 (MIDI ノート番号) と、音の強さを表す数値 (velocity) を含んでいる。以下では MIDI ノート番号について説明する。

MIDI ノート番号とは、半音単位を 1 として各音の高さに割り当てられた 0~127 までの番号である。88 鍵のピアノの鍵盤と照らし合わせると、一番低い音が 21、一番高い音が 108 となる。中央ハの音は 60 にあたり、この音から次に高いハの音まで (1 オクターブ分) の MIDI ノート番号と鍵盤の対応を、[図 4] に示す。

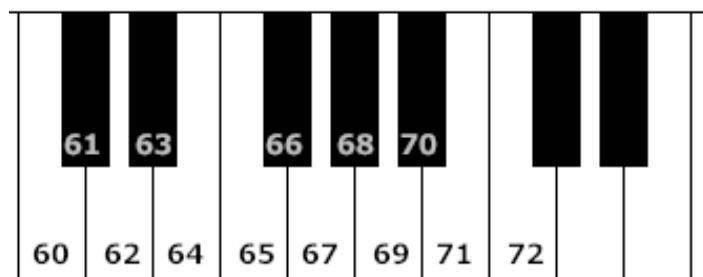


図4 MIDI ノート番号と鍵盤の対応

例えば、モーツァルトによる「きらきら星変奏曲」の第1変奏の冒頭、右手パートのそれぞれの音の高さを MIDI ノート番号で表すと、[図5] のようになる。



MIDIノート番号 74 72 71 72 71 72 71 72 81 79 78 79 78 79 78 79 80 81 84 83 86 84 83 81

図5 「きらきら星変奏曲」第1変奏（右手）冒頭の MIDI ノート番号

コンピュータで楽曲を扱う際は、このような MIDI 形式のデータから必要な情報を取り出す作業を行う。実際に本研究で必要な情報を得る手順については、第5章で述べる。

## 2.3 変奏曲

本節では、本研究で対象とする楽曲の形式である「変奏曲」とその分類、変奏曲の楽曲構造分析とは具体的にどのようなことを行うのかについて述べる。

### 2.3.1 変奏曲とは

変奏曲とは、主題と変奏から構成される楽曲を指す。16世紀には楽曲形式として確立され、17～18世紀には、J.S. バッハやモーツァルト、ベートーヴェンなどの作曲家により多くの作品が作曲された。その後、19世紀や20世紀、そして現代音楽においてもさまざまな変奏曲の例がある。変奏曲の中でもよく知られたものとしては、モーツァルトによる「きらきら星変奏曲」や、バッハによる「ゴルトベルク変奏曲」などが挙げられる。

以下では音楽事典<sup>8)</sup>の「変奏曲」の項に基いて変奏曲とその分類について説明する。

変奏曲を構成する主題は短いもので4～8小節、長いもので16～32小節程度の長さであり、音高や音積の変化や、構造などの特徴が単純で記憶されやすいものとなっている。一方変奏は、主題に音楽的な変化を施したものであり、1つの変奏曲に少ないもので4個、多いもので30個以上もの変奏が含まれている。変奏は主題に変化を加えて作られたものであることから、一般に主題と変奏の間には共通の性質が存在しており、聴衆がその共通点を認識し、主題と変奏が似ている（類似関係にある）と感ずることができる。

変奏曲にはさまざまなものがあり、その分類にも複数の観点からの方法が存在する。例えば、全体的な形式という観点からは以下のように分類される。

- 区分的変奏曲
  - 連続的変奏曲
- 区分的変奏曲 は、主題自体が曲としてのまとまりをもつものであり、各変奏もそれぞれ独立した形をもつ。「主題と変奏」と題する大部分の曲がこれに属し、通常、主題に続いて複数の変奏が演奏され、各変奏にはその演奏順の番号が付けられている（主題、第1変奏、第2変奏、…のような形）。最後の変奏が拡大されてコーダ（変奏曲全体の終始部）となるものや、最後の変奏の後再び主題が提示されるものもある。主題と第1変奏や、各変奏の間には明確な区切りがあり、楽譜上でも段の切り替えなどで表現されている。[図6]は区分的変奏曲に分類される楽曲の例で、モーツァルトによる「きらきら星変奏曲」の主題と第1変奏である。
- 連続的変奏曲 は、主題やそれぞれの変奏に楽曲としての区切りがないものを指す。主題は数小節の短い和声や旋律の断片にすぎないものであることが多く、曲を通して断続的に繰り返される主題に基いて変奏が連続的に作られる。[図7]は連続的変奏曲に分類される楽曲の例で、J.S. バッハによる「パッサカリアとフーガハ短調 BWV582」の冒頭である。この曲の場合は、冒頭の8小節で低音の主題が提示され（図の赤枠）、その後20回にわたって低音部で繰り返される主題の上に、変奏として旋律が加えられている。図の青枠、緑枠はそれぞれ主題の1回目の繰り返しと2回目の繰り返しの冒頭部分を示しており、区分的変奏曲における第1変奏、第2変奏と区別して、変奏(1)、変奏(2)と表記している。

また、変奏曲全体において、主題と各変奏の間にどのような共通性があるかという観点から、次の3種に分類する方法もある。今日最も一般的であるのがこの分類である。

- 装飾的変奏曲
- 性格的変奏曲
- 対位的変奏曲

装飾的変奏曲は、主題の構造（小節数など）と和声の骨格を保存するもので、古典派（18～19世紀初期）までの大部分の変奏曲がこれに分類される。全体として主題と変奏の間の共通性が強く、これを「厳格変奏」と呼ぶこともある。対して性格的変奏曲は、主題の旋律の大まかな動き、特徴的なリズムや和声

**TEMA. 主題**

**VAR. I. 第1变奏**

The image displays two musical sections. The top section, labeled 'TEMA. 主題', is in 3/4 time and marked 'mf'. It features a melody in the right hand and a bass line in the left hand. The bottom section, labeled 'VAR. I. 第1变奏', is also in 3/4 time and marked 'legato'. It features a more complex melody in the right hand with many sixteenth notes and a bass line in the left hand. The first system of the variation includes first and second endings.

图6 区分的变奏曲の例



図7 連続的変奏曲の例

など、主題の一部のみを保存して自由に独自の変化を行うものを指す。特にベートーヴェンの中期以後ロマン派（主に19世紀）に好んで用いられた。この種の変奏曲ではあらゆる変奏技法が併用され、小節数などの構造にも縛られないため、複雑な性格変奏の中には主題との関連が分かりづらいものもあり、「自由変奏」と呼ばれる。対位法的変奏曲とは、主題の繰り返しに際して、他の特徴的な旋律と一緒に提示するという対位法的な変奏技法によるものを総称する。これには、全体が対位法的変奏によるものと、特定の変奏にのみこの種の特徴をもつものがある。前者に関しては、バロック時代（17～18世紀半ば）に特に多く見られ、[図7]もその例である。

### 2.3.2 変奏の技法

2.3.1で挙げた分類の他にも、変奏及び変奏曲には様々な分類法があり、様々な変化の仕方、変化の度合いを持つ変奏曲が存在している。また、主題から変奏に至るまでのその変化の技法は変奏曲のみならず、多くの楽曲に部分的に取り入れられている。以下では、その変化の技法について説明する。

主題や動機（旋律の断片）、音型（特長のある形を持った音の一連）が繰り返される時に加えられる変化の技法を「広義の変奏」と呼ぶ。これは作曲の最も基本的な技法の1つであり、変奏曲だけでなく多くの楽曲で用いられており、広義の変奏を含まない曲はないとも言える。次に広義の変奏の様々な技法を挙げる。

- a) 和声の変化
- b) 動機の延長
- c) 音価の変化
- d) 反行、逆行

- e) 対位的変化
- f) 旋律の装飾
- g) リズムの変化
- h) 音色の変化
- f) 音量、表情の変化
  
- a) 和声の変化



図8 a) 和声の変化の例（ベートーヴェン：ピアノソナタ Op.2 の1）

同一の動機、音型などを異なった和声で反復するもので、変奏技法の中で楽句（4小節の構造を持つ区切り）の構成に最も数多く用いられる。[図8]の例では、後半部分は前半部分の音型をそのままに、高さの違う音へ移行することで、2つのまとまりに異なった響きを持たせている。

- b) 動機の延長



図9 b) 動機の延長の例（ベートーヴェン：弦楽四重奏曲 第一番 Op.18 の1）

動機である旋律が繰り返される際、その旋律の一部が変化するもので、ベートーヴェンなどが好んでこの技法を用いた。[図9]はその例であり、冒頭2小節が少しずつ変化しながら繰り返されていることが分かる。

- c) 音価の変化  
 主題や動機の音価を拡大または縮小するものであり、相対的なリズムは保存したまま、全体的に音の長さを変化させる技法である。これに類するものにテンポの変化による変奏がある。
  
- d) 反行、逆行  
 反行とは音の高さの上下関係を逆に置き換える変形のことを、逆行とは旋律の元の旋律を逆向きに

並べ替え、終の音から最初の音へと至るような変形のことをいう。

- e) 対位法的变化  
「対位法的変奏曲」で先述したように、主題の反復の際に他の対位旋律が提示されることである。
- f) 旋律の装飾



図 10 f) 旋律の装飾 の例 (ベートーヴェン：ピアノソナタ Op.53)

元の旋律に新たな音を加えて変化させることを指し、最も多い変奏技法の1つである。装飾の方法も様々な種類が存在する。例として参考文献では、[図 10] が挙げられており、点線枠の部分は、実線枠の各音に新たな音を加えていることがわかる。この項目については、前節で述べた「パターンによる構造」との関連が深く、本研究における重要な要素であるため、詳しくは後述する。

- g) リズムの変化
- c) 音価の変化 に対して相対的なリズムを変化させるものであり、他の変奏技法と併用されることが多い。
- h) 音色の変化
- f) 音量、表情の変化  
これらは主題そのものの変形という意味での変奏技法とは異なるが、演奏する楽器や演奏の仕方によって変化を生み出すことを意味している。

## 第3章 研究概要・目的

本章では、変奏曲について音高変化のパターン分析及び簡約化を行うことについての概要と目的を述べる。

### 3.1 本研究で対象とする楽曲

ここまで、参考文献に沿って変奏曲や変化の技法について述べた。これを踏まえて、本研究では、主題と変奏の間にある構造的な関係を自動解析することについて考える。その際、主題と変奏の区切りや、変奏中のある部分に対応するのは主題中のどの部分なのかといった点がコンピュータにとって明白である方が解析の手法を考えやすい。よって、まずは変奏曲の中でも比較的主题と変奏の関係が把握しやすい、区分的変奏曲であり装飾的変奏曲に分類される曲を対象とする。主題と変奏が独立している区分的変奏曲や、主題と変奏の間に共通性の多い装飾的変奏曲について分析を行うことができれば、その手法や結果をより自由度の高い他の分類の変奏曲の分析に応用できると考えられる。また、区分的変奏曲かつ装飾的変奏曲に分類される曲の中にはよく知られているものが多く、演奏家や聴衆による構造分析も盛んに行われていることも、このような楽曲を対象とする理由の1つである。

### 3.2 変奏曲の構造解析

2.3 では変奏曲およびその分類や技法について、3.1 では本研究で扱う楽曲の種類について述べたが、このような曲の主題と変奏から、どの変奏技法がどのように用いられているか、主題と変奏のそれぞれの音の関係を分析することが本研究の目的である。

主題と変奏の関係や共通点の分析は、変奏曲を作曲・演奏する上で欠かせない要素である。名曲事典<sup>9)</sup>の「きらきら星変奏曲」の項にも、いくつかの変奏の解説で主題との関係について述べられている。以下はその部分の抜粋である。

- ”変奏1は16分音符による装飾的演奏。たえず動いていく音のなかに、かすかに主題旋律がとらえられる。”
- ”変奏2ではこの動きが左手に移り、上声部は主題旋律を和音的に扱い、…”
- ”変奏6は、しだいにずり上がっていき、しだいに下がってくる音の動きを低声部において、上声部は主題旋律を和音で出していく。この関係は中間部では反対になる。”

これらを先述した変化の技法に沿って言い換えると、まず変奏1は主題旋律を、旋律の装飾という技法を用いて16分音符の並びに変化させていると言える。変奏2と変奏6の、「主題旋律を和音として表現する」という部分は、主題旋律の音に新たな音を加えて和音にすること、つまりこれも旋律の装飾と

いう技法を用いている。このようにして、参考文献では変奏は主題をどのような技法を用いてどの程度変化させたものなのかという点について解説されている。これは他の変奏曲についても同様であり、変奏曲という楽曲形式において、主題と変奏の関係を理解することが、音楽家や聴衆にとって重要なものであることがわかる。この主題と変奏の関係を自動化することで、音楽家や聴衆の支援につながるとともに、人間がどのような手がかりを基に、どのような手順で主題と変奏の共通点を見出しているのかを考察する基礎となりうる。よって本研究では、主題と変奏の関係、中でも様々な変化の方法があり、多くの変奏曲や他の楽曲で用いられる変奏技法である「旋律の装飾」という点に着目した関係について、分析の自動化および考察を行う。

### 3.3 音高変化のパターンの分析と簡約という視点からの主題変奏関係

旋律の装飾に着目し主題と変奏の関係を分析すると言っても、音楽事典<sup>8)</sup>の変奏曲の項で述べられていたように旋律の装飾として新たに音を加える方法には様々な例が存在する。具体的には、名曲事典<sup>9)</sup>から取り上げた「きらきら星変奏曲」の第1変奏では主題の音の前後に16分音符が加えられている、第2、第6変奏では主題の音と同じタイミングで、和音となるように音が加えられているといったものが挙げられる。このように、加える音の高さや長さによって種々の装飾が可能となる。本研究ではこの装飾後の旋律の、音高の変化をパターン化することで、装飾としてどの音にどのような高さを持つ音を加えられているのかという観点から変奏曲の自動分析を行う。今回はフィギュア・パターンに着目するため、先の第2、第6変奏の例のような、基となる音と同じ時間に発音する和音に関しては対象外とする。

[図11]は、装飾的变化がわかりやすい「きらきら星変奏曲」の第1変奏と主題の右手を比較し、音高変化のパターンの例と主題との関係性を手作業で分析したものである。図の上段が主題、下段が第1変奏であり、第1変奏中の赤い音符は、主題の対応する小節内に同じ高さの音があることを示している。この



図11 きらきら星変奏曲 主題と変奏の冒頭（右手）の比較と分析

楽譜を見ると、1小節目の前半は4つの音の中の2番目と4番目に主題と同じ音があり、1番目はそれより1音高い音、3番目は1音低い音がある。最初の音を基準として言い換えると、「(1音下がる) - (1

音下がる) — (1 音上がる) という音の変化」をしていて、かつ「2 番目と 4 番目が主題に含まれる音である」と言える。

まず、「(1 音下がる) — (1 音下がる) — (1 音上がる) などの音の変化」に着目する。この音の並びを A というパターンだとすると、これは 2 小節目の前半にも当てはまる。このようにして 1 小節目を 2 分割し、4 つの音の組と対応する主題の音を照らしあわせると、B、C、D という音の変化のパターンが図の 8 小節目の中で複数回現れている。さらに、1 小節目と 2 小節目は A の後に B のパターンがくるという共通点があり、この部分は A と B という個々のパターンから成る AB という複合パターンとしても捉えることができる。D、C についても同様で、4 小節目から CD というパターンが 3 回続く、または 3 小節目の後半から DC というパターンが 3 回続くという捉え方ができる。

次に、[図 11] 中の赤い音の位置に着目する。1 小節目の前半部分では、変奏の「2 番目と 4 番目の音が含まれる音である」ことを表していると先述したが、この特徴は同じ音の変化パターン A である 2 小節目の前半にも共通している。他の音高変化パターンである C や D に関しても同様である。変奏とは主題に変化を加えたものであることを考えると、A~D のパターンは、主題の音とそれを修飾する音から構成されていると言える。つまり、A~D 中での主題の音は各パターン内の他の音に比べて重要な音であり、パターン内でその位置が固定されている。このことから、他の変奏や変奏曲においても、パターン内での位置という情報から各パターンを簡約化し、主題の音を取り出すことができると予測される。また、簡約化された音列がさらにパターンを形成している場合もある。例えば、[図 11] の下段がそれぞれ赤で示した音に簡約されたとき、6 小節目までの音は「同じ高さの音が 2 回続く」パターンとなる。ここで、同じ高さの音を後ろの音に簡約すると、矢印で指した音を取り出される。この時、3~6 小節目の音は、D と同じ「1 音ずつ下がる」というパターンであることが分かる。

本研究では [図 11] で示したような分析を自動で行うことを考える。音高変化をパターン化するということは、音高の相対的な変化を考えることであり、旋律の装飾という観点以外にも、楽曲中にどのような変化のパターンが多く用いられているのかということ調べることもつながる。楽曲中の変化のパターンとしてどのようなものが頻出するか、その中の重要な音がどれかという点を調べることであれば、2.3.1 で述べた変化の技法の中の、a) 和声の変化 や b) 動機の延長、e) 対位法的变化 についての分析の足がかりとなるため、変奏曲のみならず様々な楽曲の分析に利用が期待できる。さらに、また、2.1 でも述べたように、パターンによる構造的特徴や簡約化された音列は、その楽曲のアイデンティティを示し、楽曲を照合・識別といった操作の対象となる。よって、音高変化のパターンとその簡約を調べることで、楽曲や各変奏の特徴をコンピュータが自動的に認識することにつながると考えられる。これは、楽曲分析や音楽学分野への還元というだけでなく、楽曲を人間が認知するメカニズムを解明する手がかりになる可能性や、楽曲の識別という点から楽曲検索などへの応用も期待できる。

### 3.4 第 4 章以降の流れ

第 4 章では、パターンに着目した変奏曲の構造分析という点で関連の深い先行研究を紹介する。そして、第 5 章では実装したプログラムに共通する前処理について説明し、第 6 章以降で本研究での取り組み

について述べる。

先行研究では、変奏曲の音高変化を含むパターンを抽出、分類している。本研究ではこれを受けて、先行研究で分析に至らなかった点について考察および実装を行う。具体的には、音高変化パターンを抽出し、パターン内の重要な位置にある音が主題の音になりうるかどうかを分析するためのプログラムの実装を目指す。第6章ではまず、先行研究や装飾記号に倣った複数の音高変化パターンに対して、変奏曲中から各パターンの抽出と簡約を行った。しかし、ここで対象とした音高変化パターン以外にも、楽曲中には様々な音高変化パターンが存在する。よって、第7章及び第8章では、楽曲に頻出する音高変化パターンを分析し、そのパターンを抽出の対象としてパターン抽出と簡約を行うことを目指す。第7章ではそのための準備の一環として作成した、抽出対象とする音高変化パターンが記述された辞書ファイルを読み込み、その辞書ファイル内のパターンについての抽出および簡約を行うプログラムについて述べる。第8章では第7章で読み込めるようにした辞書ファイルを楽曲から自動生成するプログラムについて説明し、第9章にてまとめを行う。

## 第4章 先行研究

本章では、音高変化パターンに着目して変奏曲の構造分析を行うことについての先行研究として、山崎の論文<sup>10)</sup>を紹介する。

### 4.1 音列パターン分類に基づく楽曲の主題・変奏関係の解析

山崎は、変奏曲における変奏を1小節ごと、あるいは1拍ごとに区切り、その音高変化のパターン进行分类する(音列パターン分類)アルゴリズムの検討・実装を行った。アルゴリズムの大まかな流れは以下のようになっている。

#### 音列パターン分類のアルゴリズム

- a) MIDI形式のファイルから音高データを取り出す
- b) a)で取り出したデータを分割し、音列リストとして格納する(音列リスト集合Aの生成)  
分割単位:4分音符、2分音符、半小節、1小節、2小節、4小節
- c) 分割したそれぞれのブロックに含まれる音高データに対して、隣の音との音高差が正か負どちらになるかを計算し、音列リストとして格納する(音列リスト集合Aの生成)
- d) 音列リスト集合Aが空であれば終了
- e) 音列リスト集合Aの最初のリストを取り出す
- f) e)で取り出した音列リストに対して、予め定義したそれぞれのパターン[図12]と照合を行う  
ターンか?  
トリルか?  
下行か?  
上行か?  
...
- g) 照合結果を表として標準出力で表示
- h) 結果をファイルに書き込み
- i) 音列リスト集合Aから最初の音列リストを取り除き、d)に戻る

なお、上記のf)で用いられる予め定義したパターンとは、[図12]に掲載された14のパターンのことを指す。

例えば、「きらきら星変奏曲」の第1変奏冒頭(右手)が入力データである場合を考える。まず、a)の処理によって音高データを取り出す。音高データはMIDIノート番号で表されており、「きらきら星変奏曲」の第1変奏冒頭(右手)のMIDIノート番号は[図13]のようになっている([図5]の再掲)。



|        | 番号     | パターン    |                        | 例 |
|--------|--------|---------|------------------------|---|
|        | 1      | ターン     | ターン1                   |   |
|        |        |         | ターン2                   |   |
|        | 2      | トリル     | トリル1                   |   |
|        |        |         | トリル2                   |   |
|        | 3      | 下行      | 全音階上                   |   |
|        |        |         | 半音階上                   |   |
|        |        |         | 和音の構成音                 |   |
|        |        |         | ○↓○↓○(3音単位, 音間の差は自由)   |   |
|        |        |         | ○↓○↓○↓○(4音単位, 音間の差は自由) |   |
|        | 4      | 上行      | 全音階上                   |   |
|        |        |         | 半音階上                   |   |
|        |        |         | 和音の構成音                 |   |
|        |        |         | ○↑○↑○(3音単位, 音間の差は自由)   |   |
|        |        |         | ○↑○↑○↑○(4音単位, 音間の差は自由) |   |
| 5      | モルデント  |         |                        |   |
| 6      | 和音     | 同時に鳴るもの |                        |   |
|        |        | 分散和音    |                        |   |
| 7      | オクターブ差 | ○↑○     |                        |   |
|        |        | ○↓○     |                        |   |
| (4音単位) | 8      |         | ○↑○↑○↓○                |   |
|        |        |         | ○↓○↓○↑○                |   |
|        | 9      |         | ○↑○↓○↑○                |   |
|        |        |         | ○↓○↑○↓○                |   |
|        | 10     |         | ○↑○-○↑○                |   |
|        |        |         | ○↓○-○↓○                |   |
|        | 11     |         | ○↑○-○↓○                |   |
|        |        |         | ○↓○-○↑○                |   |
|        | 12     |         | ○↑○↑○↓○                |   |
|        |        |         | ○↓○↓○↑○                |   |
| (3音単位) | 13     |         | ○↑○↓                   |   |
|        |        |         | ○↓○↑                   |   |
|        | 14     |         | ○↑-○                   |   |
|        |        |         | ○↓-○                   |   |

○は任意の1音を、↑、↓および-は2音間の音程の上下関係を表す。

図 12 音列パターンの定義



MIDIノート番号 74 72 71 72 71 72 71 72 81 79 78 79 78 79 78 79 80 81 84 83 86 84 83 81

図 13 「きらきら星変奏曲」第1変奏（右手）冒頭のMIDIノート番号（[図5]の再掲）

次に、b)によって音高データの分割を行う。分割の単位は実行時に指定でき、今回は4分音符単位とする。分割したデータを音列リストに追加した後、さらにc)では各音列に対して隣の音との音高差を計算する。隣の音との音高差は、各音のMIDIノート番号を次の音のMIDIノート番号から引き算することで得られる。結果が正の数になる場合は"+", 負の数になる場合は"-", 0になる場合は"s"と表現し、これも音列リストへ追加する（音程上下レベルのリスト）。この場合の音列リスト集合Aは以下ようになる。左側が単純な音列リスト、右側が音程上下レベルの音列リストである。

- (74 72 71 72) (- - +)
- (71 72 71 72) (+ - +)
- (81 79 78 79) (- - +)
- (78 79 78 79) (+ - +)
- (80 81 84 83) (+ + -)
- (86 84 83 81) (- - -)

これらの音列に対して、f)~i)で各パターンとの照合及び結果の書き出しが行われる。照合の際に単純な音列リストと音程上下レベルのリストのどちらを用いるかは、各パターンの定義の仕方によって自動的に判断される。

上記の処理を「きらきら星変奏曲」「サリエリのオペラ」の各変奏、右手左手それぞれに対して行った結果を、表計算ソフトを用いて視覚的に表示した。[図14]は、点線より上が手作業で解析した結果、下がプログラムの結果である。なお、この図において、左側のvar.1やvar.2という表記は変奏番号を、(R)と(L)はそれぞれ右手と左手を表しており、色ついていないセルの数字は小節番号を意味している。色をついたセルが音列パターンが抽出されたことを、セルの数値が抽出された音列パターンの番号を表している。この結果を比較すると、全体としておよそ全曲の50%から音列パターンが抽出できたという結果が出た。

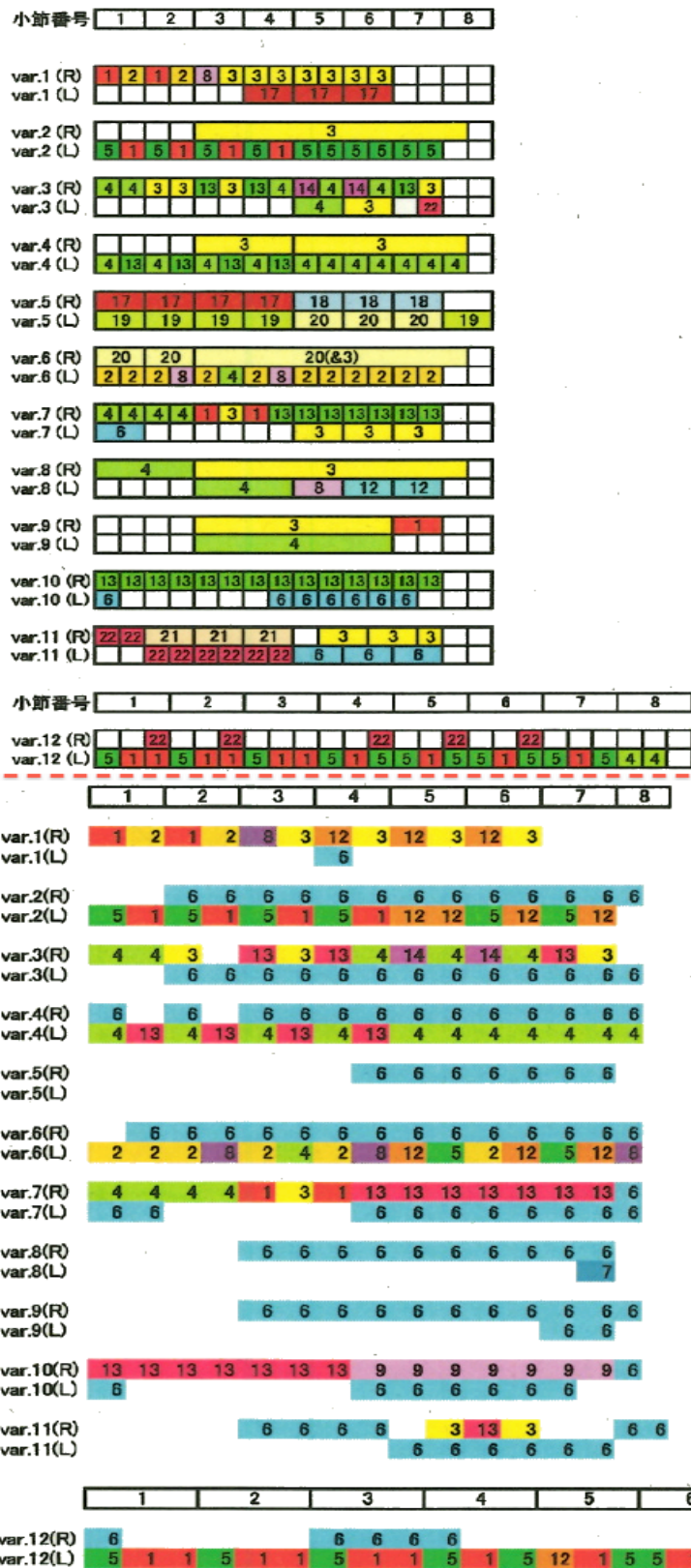


図 14 「きらきら星変奏曲」 処理結果

## 第5章 前処理

本研究におけるプログラム作成等は、Mac OS X 10.6.8 を用いた。解析のためのプログラムは C 言語を、解析結果の表示のためのプログラムは Processing を用いて作成した。Processing とは、電子アートとビジュアルデザインのためのプログラミング言語及び統合開発環境である。

解析に必要なデータは MIDI 形式のファイルから取り出すこととした。使用する MIDI 形式のファイルは簡易音楽記述言語 (MML) を用いて記述し、変換システム<sup>11)</sup>にて生成したものをを用いた。また、右手パートと左手パートの分離や、和音や複数の旋律を含むデータを単音・単旋律のデータへ変換する必要があるファイルについては、楽譜作成ソフトである finale を用いて変換を行い、新たな MIDI 形式のファイルを生成した。以下では MIDI 形式のファイルから必要なデータを取り出す手順について述べる。

2.2 で述べたようにイベントには MIDI イベント、SysEx イベント、メタイベントの3種類がある。解析に必要なデータは、MIDI イベントとメタイベントから取り出す。まず、MIDI イベントからは発音に関する情報を取り出す。発音情報を持っている”note on”という MIDI イベントには、2.2 で述べたように音の高さを表す数値である MIDI ノート番号と、音の強さを表す数値である velocity が含まれている。また、それぞれに対して消音情報を扱う MIDI イベントである”note off”が存在する。つまり、MIDI ファイルの中の1つの音は、note on によって発音され、note off によって消音される。各イベントにはデルタタイムが付与されているので、各音の note on と note off のデルタタイムを調べることで音の長さを得ることができる。さらに、各イベントをイベント開始時刻順に並べ替えることで、時系列に沿った各音の情報(楽譜のような情報)を得ることができる。

次に、メタイベントから楽曲の拍子の情報を得る。これにより、基準となる音の長さと、1小節が何拍子分であるかが明らかになるので、各音の発音時刻が何小節目の何拍子目であるかを計算することができる。例えば、メタイベントからある楽曲が8分の6拍子であることがわかった場合、基準となる音の長さは8分音符1つ分であり、1小節は基準音6つ分の長さである。このことから、基準音の長さが480で表されるとき、1小節の長さは $480 \times 6 = 2880$ であると言える。このとき、発音時間3840は $(2880 \times 1) + 480 \times 2$ で表されるので、楽譜上では1小節と2拍が終わったところ(2小節目の3拍目)を指すといったように計算できる。

このような処理を行い、MIDI 形式のファイルから解析に必要なデータを取り出す解析プログラムの出力は以下ようになる。これは、入力ファイルとしてモーツァルトの「きらきら星変奏曲」主題の右手部の MIDI ファイルを与えた場合である。1行が1音に対応しており、通算発音時刻、何小節目か、何拍目か、拍子未満の時間(拍と拍の間のどこにあたるか)、MIDI ノート番号、音の長さをそれぞれ表している。例えば、以下の出力の1行目は、通算発音時刻0、1小節目、1拍目、拍びつたりの時刻に、72の高さの音が長さ480で発音されるということを表している。なお、「きらきら星変奏曲」は4分の2拍子であり、本プログラムでは基準音の長さを480に設定している。

MIDI 解析プログラム出力 (一部)

00000,001,1,000,72,480,  
 00480,001,2,000,72,480,  
 00960,002,1,000,79,480,  
 01440,002,2,000,79,480,  
 01920,003,1,000,81,480,  
 02400,003,2,000,81,480,  
 02880,004,1,000,79,480,  
 03360,004,2,000,79,480,  
 03840,005,1,000,77,480,  
 04320,005,2,000,77,480,  
 04800,006,1,000,76,480,  
 05280,006,2,000,76,480,  
 05760,007,1,000,74,480,  
 06240,007,2,000,74,360,  
 06720,008,1,000,72,960,

本研究の各プログラムに対する共通の前処理として、このデータから音高変化の数値列を生成する処理がある。具体的には、MIDI 解析プログラムの結果の中から MIDI ノート番号の情報を取り出し、次の音の MIDI ノート番号との差を計算する。先ほどの解析プログラムの結果から MIDI ノート番号の部分を取り出すと、順に 72,72,79,79,81,81,79,79,... となっているので、これに対する隣との音高変化の数値列は、0,7,0,2,0,-2,0,... となる ([図 15])。

|    |    |    |    |    |    |    |    |     |
|----|----|----|----|----|----|----|----|-----|
| 72 | 72 | 79 | 79 | 81 | 81 | 79 | 79 | ... |
|    | 0  | 7  | 0  | 2  | 0  | -2 | 0  | ... |

図 15 音高変化の数値列計算例

また、「きらきら星変奏曲」の第 1 変奏の右手パート、冒頭 3 小節分に関しては [図 16] のようになる。次章からの解析は、この隣の音との音高変化の数値列 (以降音高変化列) を基に行う。



MIDIノート番号 74 72 71 72 71 72 71 72 81 79 78 79 78 79 78 79 80 81 84 83 86 84 83 81  
 隣との音高差 -2 -1 1 -1 1 -1 1 3 -2 -1 1 -1 1 -1 1 1 1 3 -1 3 -2 -1 -2

図 16 「きらきら星変奏曲」第1変奏（右手）冒頭のMIDIノート番号と音高変化

## 第 6 章 処理 1:特定の音高変化パターンの抽出と簡約化

### 6.1 概要

4.1 では、変奏曲の変奏を指定した分割単位に沿って分割し、用意した辞書内のパターンと同一の変化をする箇所を探す先行研究を紹介した。結果として、変奏から辞書内のパターンを抽出し、変奏曲の分析の手がかりを得ることができた。実際に変奏曲の分析を行うためには、これに加えて以下に対する考察が必要である。

- 先行研究によって変奏から抽出されたパターンと主題との間の関係性
- 先行研究で用いた辞書パターン以外のパターンの抽出

パターンと主題との間の関係性の例としては、3.3 で述べたように、変奏中のパターン内で重要な音が主題においても重要な音であるというものが挙げられる。また、楽曲中には先行研究で用いた辞書パターン以外にも、楽曲や変奏によって様々なパターンが存在する。本研究では、その中でも特に音高変化の様々なパターンについて、パターン内の位置から重要な音を決定し、その音と主題との関連について分析することを目指す。本章ではまず、「抽出されたパターンと主題との間の関係性」について考える。具体的には、先行研究を参考にして選出した辞書パターンについて、それぞれ何番目の音が重要であるかという情報を加えることによって、入力音列中で抽出されたパターン内での重要な音を割り出し、入力音列全体の簡約を行う処理について述べる。

### 6.2 手法

6.1 を踏まえて、本研究で作成した特定の音高変化パターンによる簡約プログラムでは、主に以下の処理を行った。

- a) 入力音列の中で各辞書パターン [表 2] と一致する箇所を抽出
- b) 各辞書パターンについて重要な音を抽出
- c) 抽出されたパターン内で、重要な音を親、そうでない音を子として関係を記述する
- d) 重要な音のみを集めた音列を生成 (簡約音列)
- e) 生成した簡約音列について、a)~d) を行う (階層的な簡約)

a) の、「1 つの音列の中で辞書パターン [表 2] と一致する箇所を抽出」する処理について、本研究では単パートから成る 1 つの音列からパターン抽出を行う。先行研究では入力音列を音楽的に意味のある単位で分割し、その中で音高変化を対象にしてパターン抽出を行っていた。しかし、その場合は分割地点をまたがったパターンは抽出することができない。よって本研究では分割を行わず、音列中のどこ


| パターン名        | 音高変化の条件                                  | 重要な音 | 例（ハ長調の場合）   |
|--------------|--|------|---|
| flat         | 0 0 0....（半音階上、2音以上）                     | 先頭   |    |
| up1          | 1 1 1....（半音階上、4音以上）                     | 末尾   |    |
| up2          | 1 1 1....（全音階上、4音以上）                     | 末尾   |    |
| up3          | ++ +....（変化の方向が上昇、5音以上）                  | 末尾   |    |
| down1        | 1 1 1....（半音階上、4音以上）                     | 末尾   |   |
| down2        | 1 1 1....（全音階上、4音以上）                     | 末尾   |  |
| down3        | ++ +....（変化の方向が下降、5音以上）                  | 末尾   |  |
| trill        | 1 -1 1 -1....or -1 1 1 -1....（全音階上、5音以上） | 先頭   |  |
| turnA+       | 1 1 -1（全音階上、4音）                          | 2音目  |  |
| turnA-       | -1 -1 1（全音階上、4音）                         | 2音目  |  |
| turnB+       | 1 1 -2（全音階上、4音）                          | 先頭   |  |
| turnB-       | -1 -1 2（全音階上、4音）                         | 先頭   |  |
| mordent      | -1 1（全音階上、3音）                            | 先頭   |  |
| pralltriller | 1 -1（全音階上、3音）                            | 先頭   |  |

表2 辞書パターンとして用いたパターン一覧



に位置するパターンについても抽出対象とする。b) の、「各辞書パターンについて重要な音を抽出」する処理については、まず予め辞書内の各パターン内で一番重要だと考えられる音を決めておく。a) の処理によって入力音列中のパターン一致箇所が抽出された際、抽出されたパターン内での重要な音は、辞書内での重要な音と同じ位置にある音であると考え、これを音列中での重要な音とする。例えば、[図 17] という入力音列について、[表 2] の turnA- というパターンを抽出する場合、a) の処理で青枠部分が turnA- であるという結果を得る。次に、[表 2] turnA- の行を見ると、このパターンにおける重要な音は 2 番目であると分かる。よって、青枠部分の 2 番目の音符である赤枠部分が入力音列における重要な音であると判断する。これが b) の処理である。すべての辞書パターンについてこれを調べることで、音列中での重要な音を割り出す。c) の、「抽出されたパターン内で、重要な音を親、そうでない音を子として関係を記述」する処理では、入力音列のどの箇所でもどのパターンが抽出され、どの音が重要な音と判断されたのかという抽出された個々のパターンに関する情報を記述する。入力音列全体、すべてのパターンを対象に a)~c) の処理を終えると、d) の「重要な音のみを集めた音列を生成」する処理に入る。また、e) の「生成した簡約音列について、a)~d) を行う」ことについては、d) で生成した簡約音列について a)~d) の一連の処理を繰り返すことにより、簡約音列をさらに簡約する、つまり階層的な簡約を行う。



図 17 パターン抽出と重要な音判定の例

[図 18] は、前章で述べた前処理と、先述した a)~e) の処理が実際にプログラム内でどのような手順で行われているかを示したものである。本プログラムではこの図から分かるように、入力音列をそれぞれの辞書パターンと一致する部分を割り出す関数へ渡し、一致する部分が見つかる度に、重要な音の判定と親子関係の構築を行っている。

本プログラムにおいて入力音列がある変奏曲の変奏である場合には、変化後の音列からパターンを取り出すという処理を行うこととなる。そして d) の処理の結果である簡約された音列と主題の対応を調べることで、パターンという観点で変奏から主題の音をどの程度割り出せるのかを考察する。

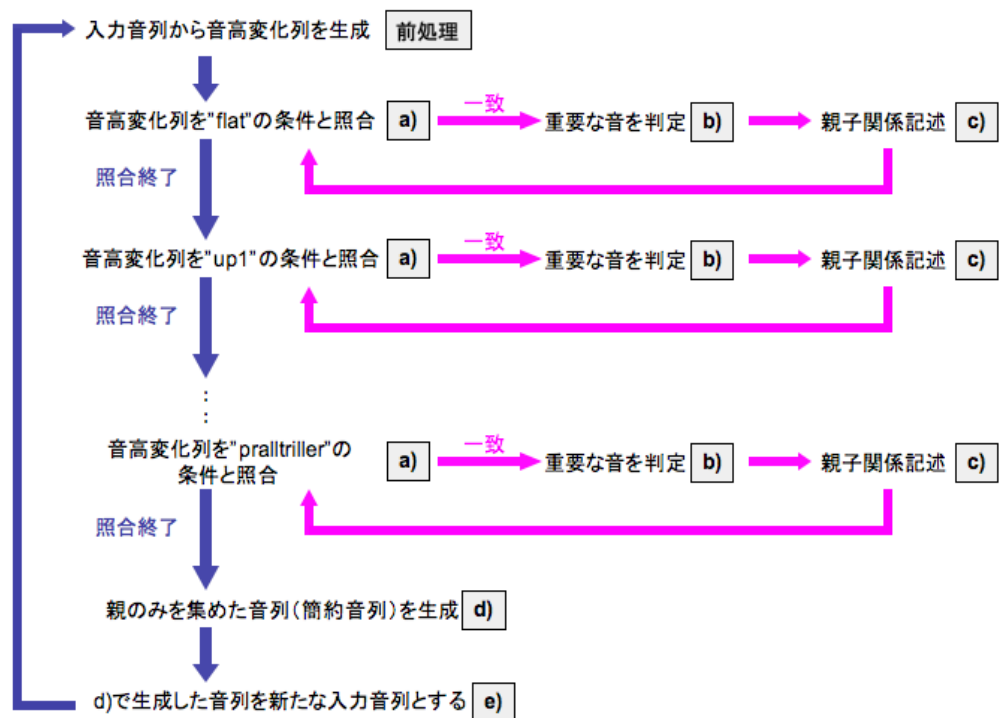


図 18 パターンによる簡約プログラムの流れ

### 6.2.1 辞書パターンと重要だと考えられる音

本章で辞書パターンとして用いた [表 2] の変化パターンについて、それぞれのパターン名は先行研究や音楽記号に倣った。各パターンについて何番目の音が重要であるかという項目に関しては、装飾音の楽譜上の表記を参考にし設定した。例えば turnA-の場合、楽譜上ではしばしば [図 19] のように装飾音として表記されることがある。これは、2番目の音が他の音を装飾していることを表現していると考えられるため、turnA-の重要な音は2番目の音であると推測できる。このようにして判断した重要な音は、[表 2] 中の例では赤枠で示している。



図 19 turnA-を表す装飾記号

### 6.2.2 抽象化を用いた一致判定

各辞書パターンと入力音列の一部のマッチング (a) の処理) においては、単純に音高変化列が同じ数値であることのみで一致を判断するのは不十分である。先の辞書パターン [表 2] 中の up ([表 3]) は、半音階上、全音階上、上昇という変化の向きのみという 3 つの種類を設定した。まず、半音階

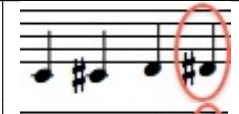


| パターン名 | 音高変化の条件                  | 重要な音 | 例 (ハ長調の場合)  |
|-------|--------------------------|------|---|
| up1   | 1 1 1.... (半音階上、4 音以上)   | 末尾   |  |
| up2   | 1 1 1.... (全音階上、4 音以上)   | 末尾   |  |
| up3   | +++.... (変化の方向が上昇、5 音以上) | 末尾   |  |

表 3 辞書パターン up ([表 2] の一部を再掲)

上での変化である up1 の判定について述べる。これは、入力音列の音高変化の数値は半音上がることが 1 として表現されているため、音高変化列の中で 1 が連続する箇所がパターンに一致する箇所であると判断できる。しかし、調に基づいた全音階上での変化を意味する up2 の判定はこのように単純には行えない。全音階上で 1 音高が上がると言うことは、音高変化の数値としては 1 の場合と 2 の場合があるためである。例えば、[図 20] の上段はハ長調の音階を示しているが、これにおいてド→レという変化とミ→ファという変化は、どちらも全音階上で 1 音上がっていると言える。しかし、音高変化の数値は半音単位であるため、前者はド→ド#→レなので 2、後者はミ→ファなので 1 となり両者は異なっている。これらの違いを考慮するために、調に基づいた全音階の音高変化の数値 ([図 20] の数値) を予めデータとして持っておき、up2 ではこれと入力音列の音高変化列を照らし合わせることで一致判定を行っている。なお、[図 20] の下段はイ長調の音階を示しているが、上下段の音高変化を見れば分かるように、調の違いによって音階の音高変化の数値に変わりはない。up3 の判定に関しては、単純に音高変化列の中で正の数値が連続することを基準にした。このように、up という 1 つのパターンについて、半音階、全音階、変化の方向の一致という複数の視点で抽出を行う処理は、2.1 で述べた抽象化による構造にあたる。

階名 ド レ ミ ファ ソ ラ シ ド レ ミ ファ ソ ラ シ ド

音高変化 2 2 1 2 2 2 1 2 2 1 2 2 2 1

音高変化 2 2 1 2 2 2 1 2 2 1 2 2 2 1

図 20 ハ長調とイ長調の音階と音高変化

### 6.2.3 辞書パターン同士の関係

本章で用いた 14 個の辞書パターンの中には、互いに関連するものが存在する。例えば、6.2.2 で述べたように up1、up2、up3 は抽象化のレベルの違いによって生まれるパターンであり、up3 で表されるパターンの中の特定的ものが up1、up2 である。よって、up1 あるいは up2 に当てはまる箇所はすべて up3 にも当てはまる。この関係は down1、2 と down3 でも同様に成り立つ。また、mordent、pralltriller というパターンは trill や turnA+、turnA- というパターンの一部であり、例えば trill が抽出された部分には必ず mordent と pralltriller が交互に抽出される。これらのように関連するパターンの抽出に関しては、パターン別にどう処理するのかを設定した。具体的には、up や down のように抽象化のレベルの違いがパターンの違いになっている場合には、他の抽象度の抽出結果を考慮せず、独立して抽出を行うようにした。これは、up1 が抽出される部分は必ず up3 が抽出されるが、それを理由に up1 が抽出された部分は up3 を適用しないという処理を行うと、up3 の一部が up1 である場合、どこからどこまでが up3 で、そのうちの部分が up1 であるかの判断が困難となるからである。これに対し、mordent と pralltriller がパターンの一部となる trill や turnA などの場合には、trill や turnA のどの部分が mordent や pralltriller になるのかが明確に決まっているため、両方を抽出する意義はない。そこでこれらについては trill や turnA が抽出された部分には、mordent や pralltriller を適用しないという処理を行う。

## 6.3 出力例と考察

[図 21] は「きらきら星変奏曲」第 1 変奏の右手パートを入力音列として上記の処理を行った結果を、Processing を用いて可視化したものである。下段が入力音列、上段が簡約化後の音列であり、それぞれの音符には通算時刻を添えている。なお、この音列表示は通常の楽譜とは異なり、音符の長さは次の音符ま

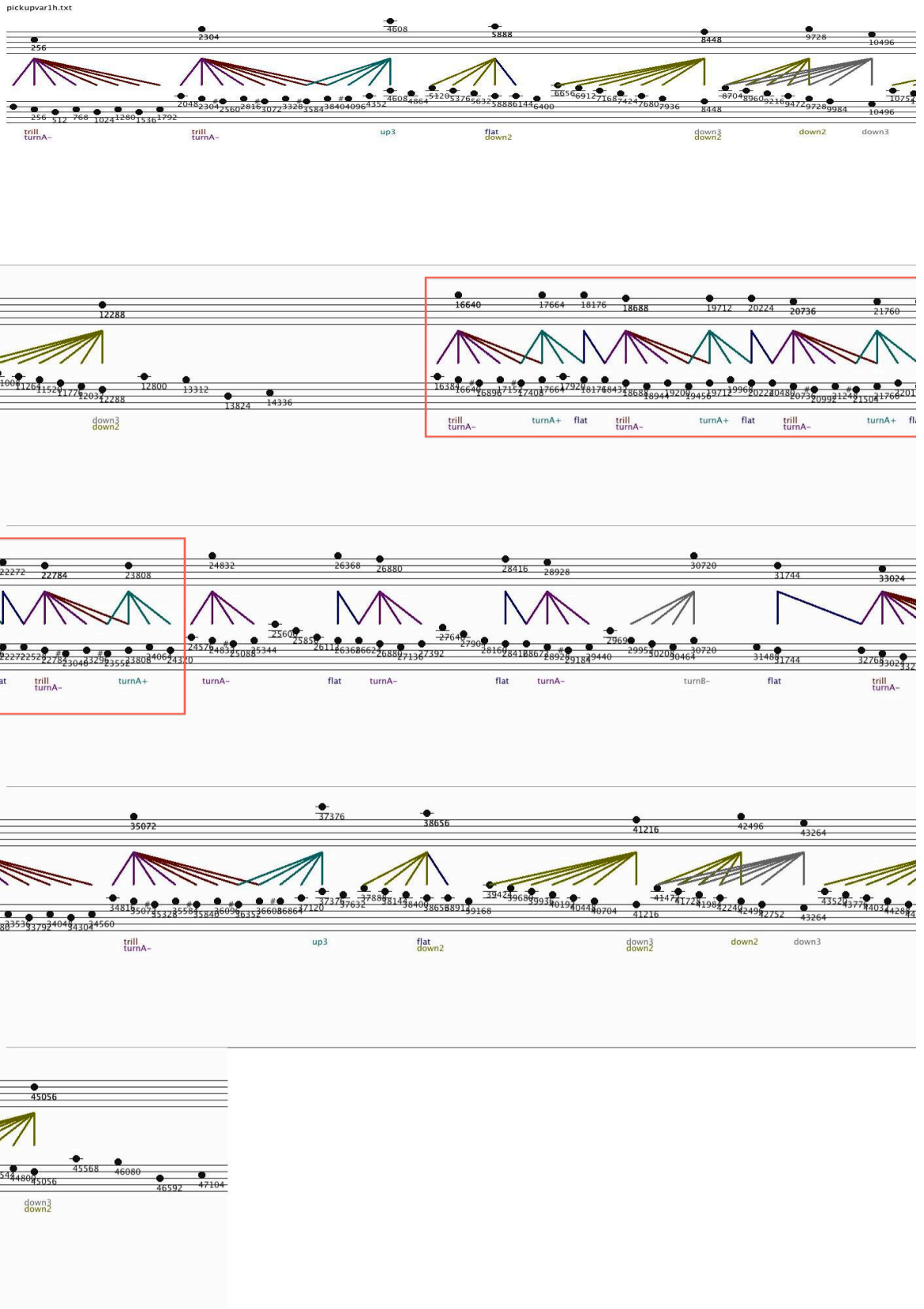


図 21 「きらきら星変奏曲」第 1 変奏右手について抽出を行った結果

での距離で表現している。上段と下段の間の音符同士を結ぶ直線は、パターン抽出時に生成された親子関係を表している。下段の音列より下に表示されているパターン名と、それによって生成された親子関係を表す直線の色は同じ物を使用している。また、[図 21] の赤枠部分を拡大したものが [図 22]、その上段の音列を入力音列として同様の処理を行った結果が [図 23] である。

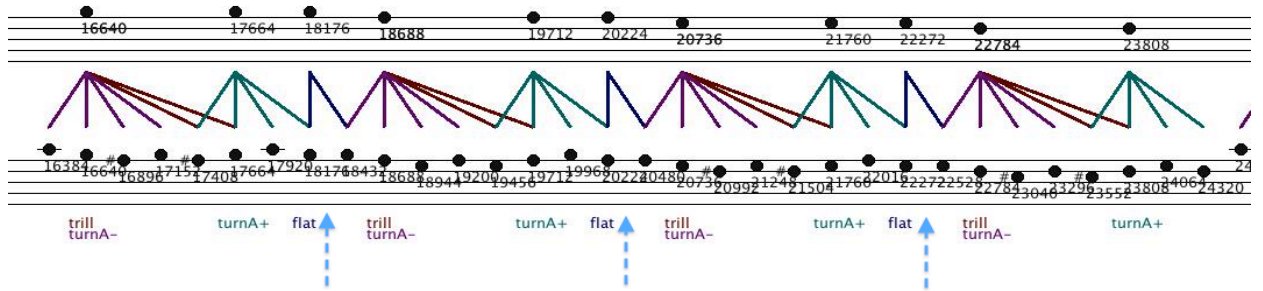


図 22 「きらきら星変奏曲」第1変奏右手について抽出を行った結果の一部（1回目）

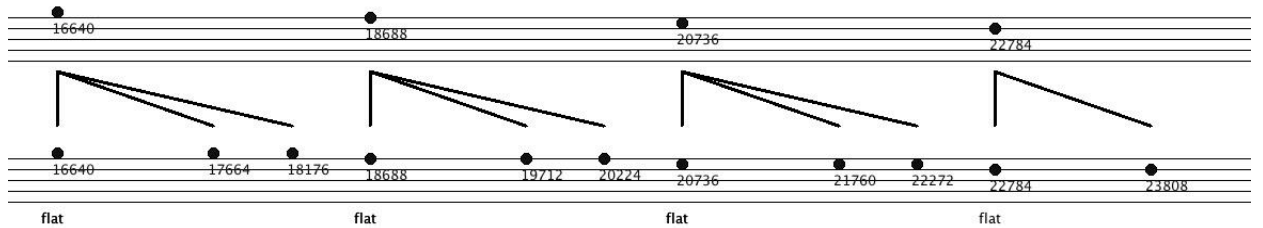


図 23 「きらきら星変奏曲」第1変奏右手について抽出を行った結果の一部（2回目）

[図 22] では、変奏を簡約化して残った上段の音は、主題の同じ箇所に見える音と同じ高さである。また、抽出されたパターンの種類と位置から、turnA- → trill → turnA+ → flat というパターンの組み合わせが繰り返されていることも見て取れる。さらに、旋律のまとまりとまとまりの境界にまたがったパターンも存在することが表れている。この変奏の場合には、[図 22] での点線の矢印（説明のために図に補足したもの）が旋律のまとまりの境界を表すが、これらにまたがる部分では共通して flat というパターンが抽出されている。この部分の他にも、変奏を通して旋律のまとまりの境界に flat が抽出されている部分があり、これは入力音列を区切ってパターンを抽出していた先行研究では得ることができなかった結果である。

簡約化した音列を入力音列とし、再度簡約化を行った [図 23] では、下段の音列がさらに上段の4音に簡約され、[図 22] での入力音列の骨格がより明確に表れている。このようにして、先述した辞書パターンから変奏の簡約を行った結果、主題と共通する音や主題をさらに簡約した音を取り出せる場合がある。しかし、簡約した結果がこのように適切である部分は、変奏曲としては比較的単純な構造だと言われる「きらきら星変奏曲」の中でも一部に限られた。具体的には、第1変奏の右手や第2変奏の左手には辞書パターンとして抽出されるものが多く見られたが、変奏番号が大きくなるにつれて辞書パターンとして抽

出される部分自体が少なくなり、適切な簡約が行えないという傾向があった。

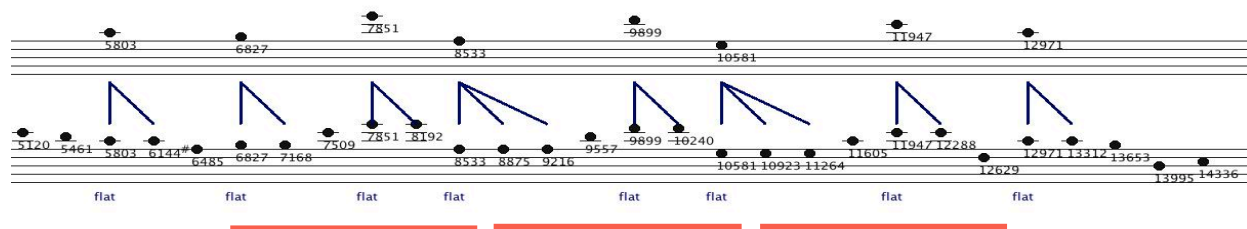


図24 「きらきら星変奏曲」第3変奏右手について抽出を行った結果の一部

特に、一見してパターン性が強いように思われる部分でも、そのパターンが[表2]の辞書パターンには当てはまらないために抽出できないケースが目立った。その例が[図24]である。赤い直線は補足のために図に加えたものであり、この部分は特定の音型のパターンが繰り返されているが、この結果ではそれが表現されていない。

このように、今回設定した辞書パターンは先行研究や装飾記号を基にしたもののみであり、実際の楽曲中には他にも様々なパターンが存在する。そこで以降の章では、楽曲に頻出するパターンを辞書として設定し、抽出や簡約を行う方法について考える。

## 第7章 処理 2:正規表現を用いたパターン辞書への対応

### 7.1 概要

第6章では、予め設定した辞書パターンについて抽出および簡約を行った。それを踏まえて、本章と次章では楽曲に頻出するパターンを辞書として設定し、抽出や簡約を行う手法について述べる。まず本章では、抽出すべき辞書パターンを記載したテキストファイル（辞書ファイル）を用意し、このファイル内のパターンについて抽出・簡約を行うプログラムについて説明する。そして第8章では、辞書ファイルに記載するパターンを楽曲から自動生成する手法について述べる。

第6章の、「予め設定した辞書パターンについて抽出と簡約を行う」ことについてシステムの観点から述べると、入力音列から計算した音高変化列の一部を、各辞書パターンに一致するかどうかを判定する関数に渡すという処理を繰り返し行うことでこれを実装した。しかし、新たに判定したい辞書パターンを追加する場合などには、プログラム中に判定用の関数を1から実装しなくてはならない。楽曲に頻出するパターンについて抽出を行うためには、辞書パターン1つ1つについて関数を用意するなど、抽出プログラム自体に手を加えるよりは、パターンの概形などの情報を外部のファイルから設定できる方が望ましい。そこで本章の「辞書ファイルに記載されたパターンについて抽出と簡約を行う」プログラムを作成した。具体的には、第6章では音高変化列は `int` 型の配列として扱い、各判定用関数では数値の比較を行っていたが、本章のプログラムでは音高変化列を文字型の配列に変換し、読み込んだ辞書パターンと文字列の比較を行う。

しかし、単純な文字列の照合でパターンを抽出する方法では、パターンを構成する音の数の一致と、半音単位での音高変化が一致している必要がある。例えば、第6章で用いた辞書パターン（[表2]）中の `up`（[表3]、これを [表4] に再掲）のパターンを取り出したいとする。まず `up1` に関して、半音単位の上




| パターン名            | 音高変化の条件                  | 重要な音 | 例（ハ長調の場合）   |
|------------------|--------------------------|------|---|
| <code>up1</code> | 1 1 1....（半音階上、4音以上）     | 末尾   |  |
| <code>up2</code> | 1 1 1....（全音階上、4音以上）     | 末尾   |  |
| <code>up3</code> | + + +....（変化の方向が上昇、5音以上） | 末尾   |  |

表4 辞書パターン `up`（[表3] の再掲）

昇に対応する音高変化列は、`1,1,1,...` である。しかし、単純な文字列照合でこれを抽出する場合、辞書パターンとしては4音のパターンの `1,1,1`、5音のパターンの `1,1,1,1` という形で、パターンを構成する音の数まで考慮したものを用意する必要がある。このような場合、すべての音の数について辞書を用意するこ



とは困難である上、実際にはこのような手間のかかる準備を行うことなく、4音以上の連続であればすべて up1 として判断したい。また、up2 や up3 に関しては、半音単位のみでの判断ではなく、音高変化にある程度の幅を持たせ、抽象化して判断する必要がある。そこで、音の数や変化量を柔軟に指定する方法として、正規表現<sup>12)</sup>を用いる。本章では正規表現を用いた辞書パターンの表現と、その辞書パターンの抽出について述べる。

## 7.2 手法

本処理で対応した正規表現を以下に示す。

### 正規表現パターン

- 任意の1文字： それ自身とマッチ（ただし下記のパターン表記中の文字は除く）
- .: 任意の1文字とマッチ
- [abc0-9]: 1文字の分岐（否定は未対応）
- {abc|xyz|123}: 複数文字（単語）の分岐
- P\*: パターン P の0回以上の繰り返し
- P+: パターン P の1回以上の繰り返し
- \: 直後の文字を正規表現としてではなく、本来の文字として扱う

例えば、[表5]の左の正規表現に対しては右のような文字列がマッチングする。

| 正規表現の例             | 該当する文字列の例                            |
|--------------------|--------------------------------------|
| "a[bc]{de ef fg}*" | "ab", "acefef", "abfgfgfg"           |
| "{\+1, \+2,}+"     | "+1,+1,", "+2,+2,+2,", "+1,+2,+2,+1" |
| "0,[\+\-].,0"      | "0,+4,0", "0,-7,0"                   |

表5 正規表現パターンと該当する文字列の例

まず、[表5]の左の項目のような正規表現文字列と、探索対象となる文字列を受け取ると、探索対象文字列中に正規表現文字列が含まれる場合にはその位置を返すプログラムを用意した。次に、第6章で用いた音高変化列を数値列から文字列化し、これを探索対象文字列とすることで、正規表現文字列として表現されたパターンと一致する箇所を音高変化列中から抽出した。なお、表現の都合上、音高変化列を文字列化する際、正の数値の直前には+の符号を加えている。さらに、[表2]（[表6]に再掲）の内容を正規表現で表した[表7]のような情報を持つ辞書ファイルを読み込み、それぞれのパターンについて抽出を行えるようにした。

| パターン名        | 音高変化の条件                                   | 重要な音 | 例（ハ長調の場合）   |
|--------------|---|------|---|
| flat         | 0 0 0....（半音階上、2 音以上）                     | 先頭   |    |
| up1          | 1 1 1....（半音階上、4 音以上）                     | 末尾   |    |
| up2          | 1 1 1....（全音階上、4 音以上）                     | 末尾   |    |
| up3          | ++ +....（変化の方向が上昇、5 音以上）                  | 末尾   |    |
| down1        | 1 1 1....（半音階上、4 音以上）                     | 末尾   |   |
| down2        | 1 1 1....（全音階上、4 音以上）                     | 末尾   |  |
| down3        | ++ +....（変化の方向が下降、5 音以上）                  | 末尾   |  |
| trill        | 1 -1 1 -1....or -1 1 1 -1....（全音階上、5 音以上） | 先頭   |  |
| turnA+       | 1 1 -1（全音階上、4 音）                          | 2 音目 |  |
| turnA-       | -1 -1 1（全音階上、4 音）                         | 2 音目 |  |
| turnB+       | 1 1 -2（全音階上、4 音）                          | 先頭   |  |
| turnB-       | -1 -1 2（全音階上、4 音）                         | 先頭   |  |
| mordent      | -1 1（全音階上、3 音）                            | 先頭   |  |
| pralltriller | 1 -1（全音階上、3 音）                            | 先頭   |  |

表6 辞書パターンとして用いたパターン一覧（[表2]の再掲）

| パターン名        | 正規表現パターン  | 重要な音 |
|--------------|---|------|
| flat         | {0,}+   | f    |
| up1          | \+1, \+1, {\+1,}+   | 1    |
| down1        | \-1, \-1, {\-1,}+   | 1    |
| up3          | \+[1-9], \+[1-9], \+[1-9], \+[1-9], {\+[1-9],}+                 | 1    |
| down3        | \-[1-9], \-[1-9], \-[1-9], \-[1-9], {\-[1-9],}+                 | 1    |
| trill        | \-1, \+1, {\-1, \+1,}+  | f    |
| trill        | \-2, \+2, {\-2, \+2,}+  | f    |
| turnA+       | {\+[12], \+1, \-1   \+[12], \+2, \-2}                           | 2    |
| turnA-       | {\-[12], \-1, \+1   \-[12], \-2, \+2}                           | 2    |
| turnB+       | {\+1, \+1, \-2   \+1, \+2, \-3   \+2, \+1, \-3   \+2, \+2, \-4} | 2    |
| turnB-       | {\-1, \-1, \+2   \-1, \-2, \+3   \-2, \-1, \+3   \-2, \-2, \+4} | 2    |
| mordent      | {\-1, \+1   \-2, \+2}   | f    |
| pralltriller | {\+1, \-1   \+2, \-2}   | f    |

表7 [表6] に対応する正規表現辞書パターン

この表の「重要な音」という項目においては、fが先頭、1が末尾、数値が何音目かを表している。また、[表6]におけるup2、down2については、音階上の推移が正規表現を用いた場合に表現しづらいため、今回は除外した(7.3で後述)。本章のシステムについては、音高変化列の中でパターンと一致する箇所があれば、第6章と同様に重要な音と他の音との親子関係を記述し、すべての抽出が終わった後、重要な音のみを集めた簡約音列を生成する。このような流れを第6章同様に図として表したものが、[図25]である(図中の赤字は、第6章にはない処理を表している)。例えば辞書ファイルとして[表7]を用いた場合は、[図25]中の「辞書パターン(1)」がflatに、「辞書パターン(N)」がpralltrillerに対応する。辞書パターンと音高変化列の照合は、[図25]中の吹き出し部分のように、辞書パターンと比較する箇所を音高変化列の先頭から1文字ずつずらしながら、それぞれの箇所について両者が一致するかどうかを調べる。一致した場合には、音高変化列のどの部分からどの部分までが一致しているのかを記録する。1つの辞書パターンについて音高変化列中の一致する部分をすべて記録した後、各部分についての重要な音を判定し、親子関係の記述を行う。これを辞書ファイル中のすべてのパターンについて繰り返し行った後、親のみを集めて簡約音列を生成し、階層的な簡約を行う場合にはこれを入力音列として繰り返す。

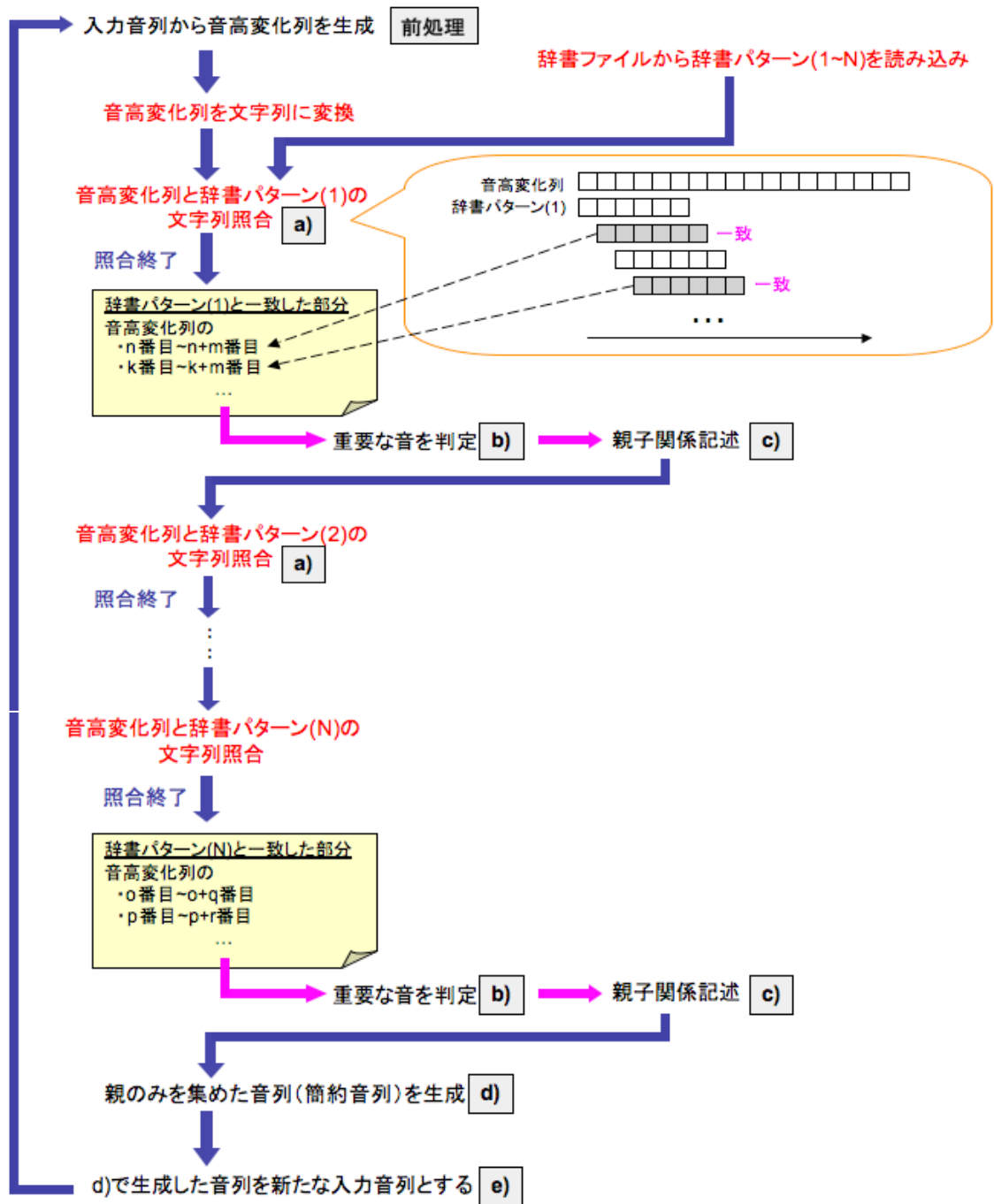


図 25 正規表現パターンによる簡約プログラムの流れ

### 7.3 出力例と考察

[図 26] は 6.3 と同様に、「きらきら星変奏曲」第 1 変奏の右手パートを入力音列として、7.2 で述べた処理を行った結果を Processing を用いて可視化したものである。結果の見方も 6.3 と同様に、下段の入力音列を上段に簡約した形となっている。[表 6] の各パターンについて判定用関数を実装した処理 1 の結果の [図 21] と、正規表現を用いてパターンを表現した [表 7] のようなファイルを読み込み、文字列マッチングでパターンを抽出した処理 2 の結果の [図 26] とで比較を行い、両者の結果の違いについて検討する。

まず、一見して把握できる違いとして、処理 2 の結果である [図 26] の方が上段に残っている音が多い、つまり抽出されたパターンが多いことが挙げられる。このような結果の原因として、処理 2 で行う辞書パターンファイルを読み込むという手法では、各パターンが独立して抽出されることが考えられる。処理 1 では 6.2.3 で述べたように、trill が抽出された場合は mordent と pralltriller は抽出しない、のような互いに関係のあるパターン同士の抽出について特別に判定の条件を与えていた。しかし、処理 2 の各パターンについては抽出を独立に行い、パターン同士の関係は考慮していないため、このようにパターンの抽出が重複した結果が出る場合が生まれた。また、処理 2 の結果の冒頭には、2 音目から始まるものと 4 音目から始まる 2 つの trill が抽出されているが、処理 1 では最初のものだけ抽出されている。これは、処理 2 では各辞書パターン同士が独立であるだけでなく、抽出されたパターン同士も独立であるためである。

処理 1 の手法から処理 2 の手法にすることで起こる抽出の違いは、他にも存在する。それは 7.2 の中でも触れたように、[表 6] における up2 および down2 のような、音階上の推移を表現しづらい点である。音階上で音が 1 つずつ高くなる場合には、音高差列は基本的に +1+2 の組み合わせで表現できる（和声的短音階の場合のみ +3 も用いる）。しかし、[図 20] で示すような音階の音高変化の一部であるという条件を正規表現を用いて表すには、パターンの音数や開始位置の違いを別のパターンとして扱う必要があり、困難である。このような、音階上でのパターンを考えることができない点は処理 2 のデメリットである。

しかし、処理 1 と比較してそのような欠点はあるにせよ、このような処理を行うことで、外部からの辞書パターンを読み込み、柔軟なパターン抽出を行うことができるようになった。次章ではこれを利用して、与えられた音列の中で頻出のパターンを調べ、自動的にその曲に合った辞書を生成する処理について述べる。また、本処理の応用としてこの他にも、次のような辞書を生成し、パターン抽出を行うことによる構造分析が考えられる。

- 階層的な簡約に適した辞書
- 右手と左手それぞれに適した辞書
- 和音を対象とした辞書

「階層的な簡約に適した辞書」とは、[図 22] の結果を出し（1 回目の分析）、その結果得られた簡約音列を入力として [図 23] の処理を行う（2 回目の分析）といった処理を行う際に用いることを想定した辞書である。例えば、1 回目用の辞書と 2 回目用の辞書をそれぞれに合ったものにするというものが考えら

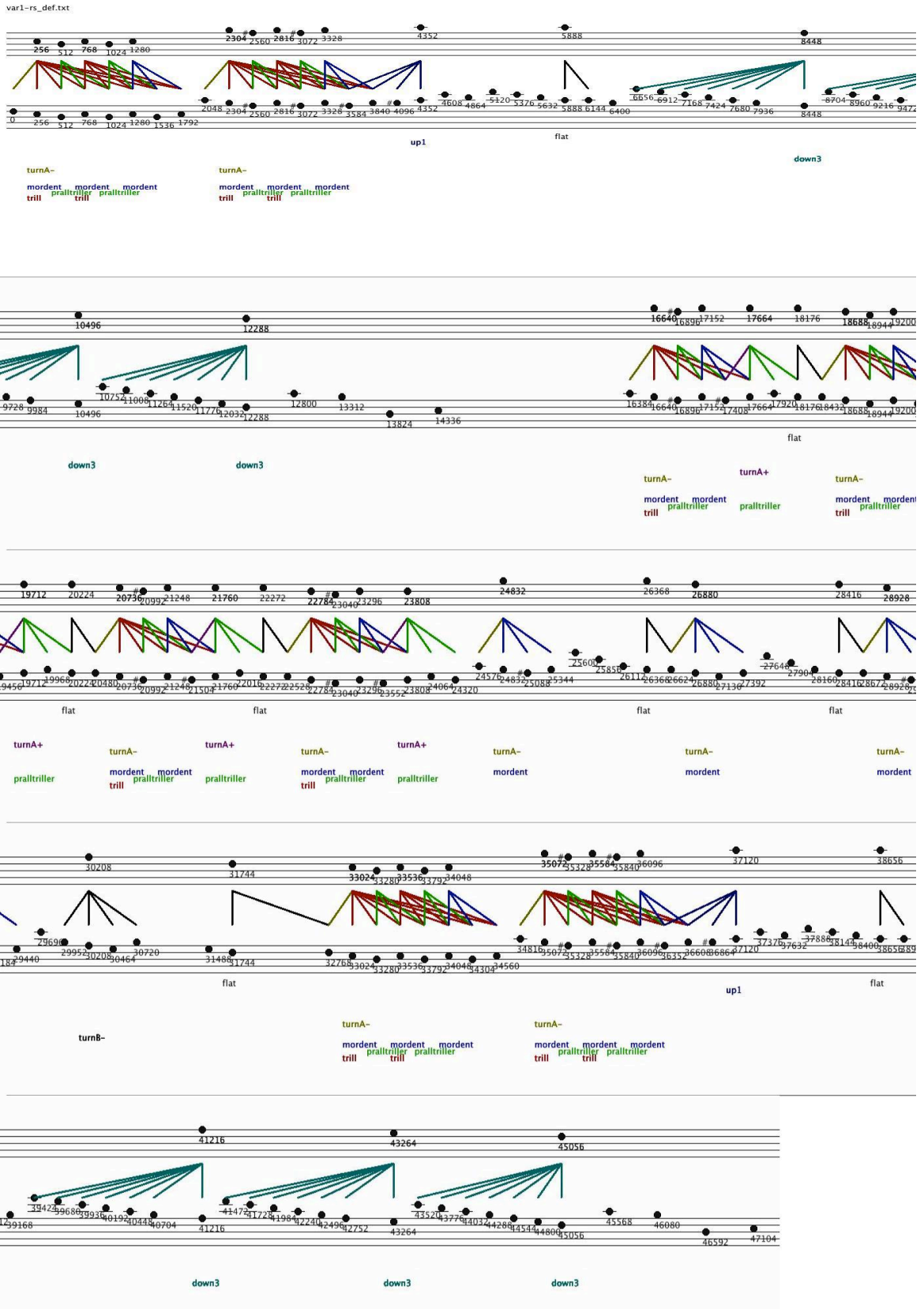


図 26 「きらきら星変奏曲」第1変奏右手について抽出を行った結果

れる。

「右手と左手それぞれに適した辞書」について、変奏曲に限らず一般のピアノ曲では、右手が旋律を奏で左手が伴奏をするという場合が多いため、旋律の簡約を想定した辞書と、伴奏の簡約を想定した辞書を別に生成して適用した場合の分析も考えられるということを表している。

「和音や複数旋律を対象とした辞書」について、今回は音高変化のみに着目し、単旋律、単音の音列を対象としていた。しかし、音高変化のパターンが多く見られる楽曲中にも和音が存在する場合が多く、さらには和音の中の音が新たな旋律を形成し、音高変化のパターンの一部となっていることもある。今回は入力音列中の和音については、前処理にて予め和音の中から重要な音を1つ取り出して単旋律に変換しているため、和音の他の音の情報は抽出や簡約に含まれていない。このような、和音を構成するそれぞれの音に対してもパターン抽出や簡約を行うことができるよう、抽出のアルゴリズムや和音を対象にした辞書の設定も考える必要がある。

## 第 8 章 処理 3: 楽曲に頻出するパターンの辞書化

### 8.1 概要

第 7 章では、抽出すべきパターンを辞書ファイルとして読み込み、音高変化列と辞書パターンの文字列マッチングを行うことでパターンの抽出と簡約を行った。結果として辞書ファイルへのパターンの追加や変更が容易に行えるようになった。そこで次のステップとして、辞書パターンとしてどのようなものを設定するのが適切であるかという点が課題となる。第 6 章や第 7 章では、先行研究や装飾記号を参考にして辞書パターンを設定した。しかし、実際の楽曲にはそれらのように多くの楽曲に出現する汎用的な音型パターンの他に、その楽曲内でのみ出現する音型パターンが存在する。そこで本章では、入力された音列の中で頻出するパターンを取り出し、辞書パターンとして追加する、つまり第 6 章で用いたような辞書ファイルを楽曲から自動的に生成する手法について述べる。実際に存在する楽曲を基に辞書パターンを生成することで、楽曲固有のパターンを含む音型パターンを取り出すことができる。さらに異なる楽曲から生成した辞書パターンを比較することで、特定の楽曲に固有に現れるパターンと、幅広い楽曲に現れる汎用的なパターンを分類することにも利用しうる。

### 8.2 手法

本研究で作成した、楽曲に頻出するパターンの辞書化プログラムの構成を [図 27] に示す。入力として音列ファイルと統合用の辞書ファイルを受け取り、2つの辞書ファイルを出力する。統合用の辞書ファイルが入力された場合、出力する辞書ファイルの内容は、統合用辞書ファイルの内容に新たに辞書化されたパターンが加えられたものとなる。統合用辞書ファイルの入力がない場合は、プログラムで生成された辞書パターンのみが出力する辞書ファイルに書き込まれる。出力する辞書ファイルは最低音数のパターンに関するものと、最低音数以上最高音数以下のすべてのパターンに関するものと 2 種類生成した。ここで最低音数とは、音高変化パターンとして扱う最も少ない音数のことを指す。この数字はプログラム内で指定可能であり、今回は 4 とした。最低音数以上の音数を持つパターンは、最低音数のパターンの組み合わせで表現できる。例えば、最低音数が 4 のとき、4 音パターンとして音高変化が -2 -1 1 というパターン A と、-1 1 -1 というパターン B があるとする。この場合、5 音パターンである -2 -1 1 -1 というパターンは、4 音パターンである A と B の重ねあわせである。このように本プログラムでは、最低音数のパターンを基本としてそれ以上の音数のパターンを考えており、出力する辞書ファイルとしても最低音数以上最高音数以下のすべてのパターンを含むものの他に、基本となる最低音数パターンのみを記載したのも生成し、比較や分析の材料とする。最高音数とはパターンとして扱う最も長い音数のことを指す。今回はこれを 9 とし 4 音～9 音のパターンを分析対象とした。

プログラム内部の主な処理は [図 27] の点線内で示したように、まず入力音列から音高変化列を生成し、次にその変化列中のパターンを分析、その後頻出パターンを辞書の形式に整えて出力するという段階



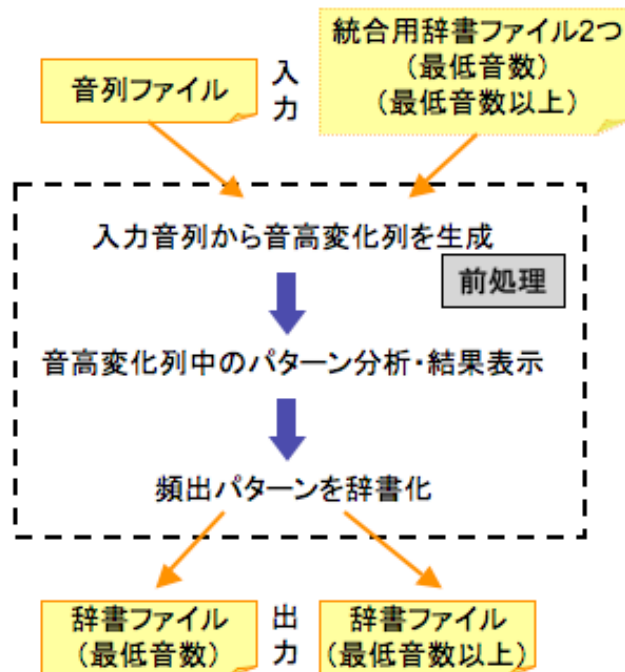


図 27 頻出パターン辞書化プログラムの動作概要

に分かれている。音高変化列の生成については第 5 章で述べた通りであり、第 6 章、第 7 章内の処理と同様である。以降はその後の、音高変化列中のパターン出現数の計算と、パターンの辞書化について説明する。

### 8.2.1 音高変化列中のパターン出現数の計算

[図 28] は、音列変化中のパターン出現数の計算の流れと例を示している。ここでは、音高変化列として -2 -1 1 -1 1 -1 1 3 が得られたとして説明する。まず、最低音数のパターンに関する分析をする。ここでは最低音数を 4 音としているので、4 音パターンを音高変化列で表すと 3 つの差分値の並びになる。音高変化列の先頭から 3 つずつの数字を 1 組として数字の組を作ると、[図 28] 中 1. の表のようになる。これは音列中の 4 音の音高変化を表すため、[図 28] では Group4 として表している。次に、各組と類似の変化をしている箇所が同じ音列の中にいくつ存在するかを数え、1 つでも存在する場合にはそのパターンに名付けを行う。なお、2 つの音高変化パターンが類似しているということは、音高変化のそれぞれの数値が一致しているか、一致はしていないがそれぞれの数値が近いということである。類似するパターンを判断する基準については 8.2.2 で述べる。今回の例では、表中の Group4[0] は他の組で類似するものがないため # で表し、Group4[1] は Group4[3] と同じ数値から成る組であるので、Group4[1] のパターンには A0 という名前をつけ、Group4[3] も同じ名前です。各パターンについてこの処理を行うと、この音高変化列は名付けたパターン名を用いて #,A0,B0,A0,B0,# と表すことができ、A0 のパターンは Group4[1]

生成した音高変化列:-2 -1 1 -1 1 -1 1 3、統合用辞書の入力なしの場合

1. 先頭から4音パターンの組を作る

|           |         |
|-----------|---------|
| Group4[0] | -2 -1 1 |
| Group4[1] | -1 1 -1 |
| Group4[2] | 1 -1 1  |
| Group4[3] | -1 1 -1 |
| Group4[4] | 1 -1 1  |
| Group4[5] | -1 1 3  |

2. Group[0]から順に、他の組で類似するものの数を数え、  
1つでも存在すればA0~z9の名前を付ける  
(類似するパターンには同じ名前を付ける)  
存在しない場合は#という名前で表す

|           |         |    |
|-----------|---------|----|
| Group4[0] | -2 -1 1 | #  |
| Group4[1] | -1 1 -1 | A0 |
| Group4[2] | 1 -1 1  | B0 |
| Group4[3] | -1 1 -1 | A0 |
| Group4[4] | 1 -1 1  | B0 |
| Group4[5] | -1 1 3  | #  |

|    |    |
|----|----|
| A0 | 2回 |
| B0 | 2回 |

3. 先頭から(前のパターン数+1)音パターンの組を作る

|           |            |
|-----------|------------|
| Group5[0] | -2 -1 1 -1 |
| Group5[1] | -1 1 -1 1  |
| Group5[2] | 1 -1 1 -1  |
| Group5[3] | -1 1 -1 1  |
| Group5[4] | 1 -1 1 3   |

4. Group[0]から順に、他の組で類似するものの数を数える  
1つでも存在すれば最低音数パターンの名前を  
組み合わせて名前を付ける  
(類似するパターンには同じ名前を付ける)

|           |            |      |
|-----------|------------|------|
| Group5[0] | -2 -1 1 -1 | #    |
| Group5[1] | -1 1 -1 1  | A0B0 |
| Group5[2] | 1 -1 1 -1  | B0A0 |
| Group5[3] | -1 1 -1 1  | A0B0 |
| Group5[4] | 1 -1 1 3   | #    |

|      |    |
|------|----|
| A0B0 | 2回 |
| B0A0 | 1回 |

最高音数パターンを超えるまで繰り返し

図 28 パターン出現数計算の流れ

と Group4[3] の 2 回、B0 のパターンは Group4[2] と Group4[4] の 2 回含まれるということが分かる（[図 28] 中の 2. の表）。ここまでの最低音数のパターン出現数の計算である。

次に、音数を増やしたパターンの出現数を調べる。大まかな流れは最低音数の場合と同様で、まず先頭から対象となる音数の音高変化列の組を作る。その後他の組で類似するものを数え、見つかった場合は名付けを行う。ここで、名付けの際に最低音数のパターン名を用いるのが、最低音数の場合との違いである。例えば Group5[1] の場合、前半 3 つの -1 1 -1 というパターンは A0、後半 3 つの 1 -1 1 というパターンは B0 と先の処理で名付けられている。これを踏まえて、Group5[1] の名前は A0B0 という形で表現する。これは、パターンの組み合わせからさらにパターンを作っていることにあたる。こうしてパターンの音数が最高音数になるまで計算を行うと、最低音数から最高音数までのそれぞれの数のパターンが、音列中に何回現れるかを計算することができる。

### 8.2.2 類似するパターンの判断

2 つの組の音高変化が類似していると判断する基準として、それぞれの数値が完全に一致する場合の他に、それぞれの数値の差が ± 1 以内である場合も含めている。このような場合の ± 1 にあたるものを、以降変化の幅と呼ぶ。それぞれの数値が完全一致するという事は、音高変化が半音単位で一致するという事を意味する。しかし、実際には第 7 章における正規表現辞書パターンのように、音高変化によるパターンは半音単位での厳密な一致ではなく、ある程度のずれを含めて考える方が自然である。理由としては 6.2.2 で述べたように、特定の調に基づいた楽曲においては、音の変化を半音階上だけではなく、全音階上で考える場合も多いことが挙げられる。全音階上の変化を半音単位で表すと、その変化の絶対値は 1 か 2 のどちらかであることを踏まえて、今回はこのような変化の幅を持たせる。ただし、音高変化が 0 である部分に関しては、他の音高変化とは性質が異なるため完全一致のみを条件とし、1 や -1 の変化の幅としては 0 を含まないこととする。例えば、この条件の下では、0 1 -2 という組と類似するものの中には 0 1 -2 のようにすべての数値が一致するもの他に、0 2 -3 なども含める一方で、1 1 -2 というものは 0 の部分が異なるため含めないと判断する。

### 8.2.3 パターン出現数の計算結果

[図 29] は、「きらきら星変奏曲」第 1 変奏右手を入力音列とし、パターンの出現数を計算した結果の一部である（付録にて全体の結果を掲載）。なお、出力結果の左側の数値は行数を表している。結果はこのように、最低音数のパターンから最高音数のパターンまでの出現数の推移を、左から右へ並べることで表している。それぞれ括弧の直前の数値が各パターンの出現回数を、括弧の中身はパターンの内容を示している。例えば 4 行目は A0 という 4 音のパターン（実際の変化列の数値が -2 -1 1）は音列中に 11 回登場し、その右側を見ると A0B0 という 5 音のパターンは 8 回出現していることを表している。図中の +-----> で表した矢印は、+ の真上のパターンに続いて新たなパターンの組み合わせがあることを示している。4 行目と 5 行目の 9 音のパターンを見ると、最後のパターン以外の箇所は一致しており、この一致した部分は 4 行目の 8 音のパターンであることが分かる。このようにして、A0 から始まるパターンは 4 行目から 7 行目まで、B0 から始まるパターンは 9 行目から 13 行目までといった形で、関連するパターンをまとめて表している。

| 1  | 繰り返し回数(該当パターン)   |                     |                        |                           |                              |                              |  |  |  |
|----|------------------|---------------------|------------------------|---------------------------|------------------------------|------------------------------|--|--|--|
| 2  | 4音               | 5音                  | 6音                     | 7音                        | 8音                           | 9音                           |  |  |  |
| 3  |                  |                     |                        |                           |                              |                              |  |  |  |
| 4  | 11(A0:-2, -1, 1) | -> 8(A0,B0)         | -> 8(A0,B0,C0)         | -> 4(A0,B0,C0,B0)         | -> 4(A0,B0,C0,B0,C0)         | -> 2(A0,B0,C0,B0,C0,D0)      |  |  |  |
| 5  |                  |                     |                        |                           |                              | +-----> 2(A0,B0,C0,B0,C0,G0) |  |  |  |
| 6  |                  |                     | +-----> 3(A0,D0)       | -> 2(A0,D0,Z0)            | -> 2(A0,D0,Z0,a0)            | -> 2(A0,D0,Z0,a0,b0)         |  |  |  |
| 7  |                  |                     |                        |                           |                              |                              |  |  |  |
| 8  |                  |                     |                        |                           |                              |                              |  |  |  |
| 9  | 12(B0:-1, 1, -1) | ->12(B0,C0)         | -> 4(B0,C0,B0)         | -> 4(B0,C0,B0,C0)         | -> 2(B0,C0,B0,C0,D0)         | -> 2(B0,C0,B0,C0,D0,E0)      |  |  |  |
| 10 |                  |                     |                        | +-----> 2(B0,C0,B0,C0,G0) | -> 2(B0,C0,B0,C0,G0,H0)      |                              |  |  |  |
| 11 |                  | +-----> 2(B0,C0,D0) | -> 2(B0,C0,D0,E0)      | -> 2(B0,C0,D0,E0,F0)      | -> 2(B0,C0,D0,E0,F0,A0)      |                              |  |  |  |
| 12 |                  | +-----> 6(B0,C0,G0) | -> 2(B0,C0,G0,H0)      | -> 2(B0,C0,G0,H0,I0)      | -> 2(B0,C0,G0,H0,I0,J0)      |                              |  |  |  |
| 13 |                  |                     | +-----> 3(B0,C0,G0,J0) | -> 2(B0,C0,G0,J0,X0)      | -> 2(B0,C0,G0,J0,X0,P0)      |                              |  |  |  |
| 14 |                  |                     |                        |                           |                              |                              |  |  |  |
| 15 | 12(C0: 1, -1, 1) | -> 4(C0,B0)         | -> 4(C0,B0,C0)         | -> 2(C0,B0,C0,D0)         | -> 2(C0,B0,C0,D0,E0)         | -> 2(C0,B0,C0,D0,E0,F0)      |  |  |  |
| 16 |                  |                     | +-----> 2(C0,B0,C0,G0) | -> 2(C0,B0,C0,G0,H0)      | -> 2(C0,B0,C0,G0,H0,I0)      |                              |  |  |  |
| 17 |                  | +-----> 2(C0,D0)    | -> 2(C0,D0,E0)         | -> 2(C0,D0,E0,F0)         | -> 2(C0,D0,E0,F0,A0)         | -> 2(C0,D0,E0,F0,A0,B0)      |  |  |  |
| 18 |                  | +-----> 6(C0,G0)    | -> 2(C0,G0,H0)         | -> 2(C0,G0,H0,I0)         | -> 2(C0,G0,H0,I0,J0)         | -> 2(C0,G0,H0,I0,J0,K0)      |  |  |  |
| 19 |                  |                     | +-----> 3(C0,G0,J0)    | -> 2(C0,G0,J0,X0)         | -> 2(C0,G0,J0,X0,P0)         | -> 2(C0,G0,J0,X0,P0,Y0)      |  |  |  |
| 20 |                  |                     |                        |                           |                              |                              |  |  |  |
| 21 | 5(D0:-1, 1, 9)   | -> 3(D0,E0)         | -> 2(D0,E0,F0)         | -> 2(D0,E0,F0,A0)         | -> 2(D0,E0,F0,A0,B0)         | -> 2(D0,E0,F0,A0,B0,C0)      |  |  |  |
| 22 |                  | +-----> 2(D0,Z0)    | -> 2(D0,Z0,a0)         | -> 2(D0,Z0,a0,b0)         |                              |                              |  |  |  |
| 23 |                  |                     |                        |                           |                              |                              |  |  |  |
| 24 | 4(E0: 1, 9, -2)  | -> 3(E0,F0)         | -> 3(E0,F0,A0)         | -> 3(E0,F0,A0,B0)         | -> 3(E0,F0,A0,B0,C0)         | -> 2(E0,F0,A0,B0,C0,B0)      |  |  |  |
| 25 |                  |                     |                        |                           |                              |                              |  |  |  |
| 26 | 9(F0: 9, -2, -1) | -> 3(F0,A0)         | -> 3(F0,A0,B0)         | -> 3(F0,A0,B0,C0)         | -> 2(F0,A0,B0,C0,B0)         | -> 2(F0,A0,B0,C0,B0,C0)      |  |  |  |
| 27 |                  | +-----> 6(F0,N0)    | -> 6(F0,N0,N0)         | -> 6(F0,N0,N0,N0)         | -> 6(F0,N0,N0,N0,N0)         | -> 4(F0,N0,N0,N0,N0,S0)      |  |  |  |
| 28 |                  |                     |                        |                           | +-----> 2(F0,N0,N0,N0,N0,T0) |                              |  |  |  |
| 29 |                  |                     |                        |                           |                              |                              |  |  |  |

図 29 パターン出現数計算結果の一部

この結果からは、各パターンの出現数が分かるだけでなく、4音のパターンが5音、6音と増えるにつれ、どのような種類のパターンにつながるのかということも分析できる。21行目や24行目を見ると、D0、E0のパターンは音数が増えても多くの種類のパターンが見られるわけではない。これは、D0やE0のパターンは特定のフレーズの中で多く扱われていることを示している。一方で、A0やB0、C0については、音数が増えるに従って様々なパターンの組み合わせが見られる。このようなパターンは、特定のフレーズのみならず多様な場面で出現していると言える。

#### 8.2.4 辞書化するパターンの条件

[図 29] では2回以上出現するすべてのパターンを表示しているが、辞書パターンとして扱うことを考えると、単純に出現数の多いものをそのまま辞書化すればよいとは言えない。[図 27] では出力として、最低音数パターンの辞書ファイルと最低音数以上パターンの辞書ファイルを生成することを示している。最低音数のパターンは、最低音数以上のパターンの基礎となるため、比較やチェックの材料として [図 29] の4音パターンに当てはまる部分をファイルとしてそのまま書き出す。一方最低音数以上のパターンに関しては、パターン同士の関係も考慮し、以下のような条件を満たすものを辞書化することとした。

- 条件 1) そのパターンよりも1音多いパターンとの出現数の差が2回以上である
- 条件 2) すでに辞書化されたパターンとの重複分を除外した出現数が2回以上である

また、[図 30] は [図 29] の中から、条件 1 に当てはまるパターンに黄の背景をつけ、条件 1 かつ条件 2 に当てはまるもの（辞書化の条件に即すもの）にはさらに赤の点線枠を用いて表し、条件に当てはまるパターンを示したものである。ここからは各条件について説明する。

| 1  | 繰り返し回数(該当パターン)        |                            |                        |                         |                            |                              |                        |
|----|-----------------------|----------------------------|------------------------|-------------------------|----------------------------|------------------------------|------------------------|
| 2  | 4音                    | 5音                         | 6音                     | 7音                      | 8音                         | 9音                           |                        |
| 3  |                       |                            |                        |                         |                            |                              |                        |
| 4  | <b>11(A0:-2,-1,1)</b> | → 8(A0,B0)                 | → <b>8(A0,B0,C0)</b>   | → 4(A0,B0,C0,B0)        | → <b>4(A0,B0,C0,B0,C0)</b> | → 2(A0,B0,C0,B0,C0,D0)       |                        |
| 5  |                       |                            |                        |                         |                            | +-----> 2(A0,B0,C0,B0,C0,G0) |                        |
| 6  |                       |                            | +-----> 4(A0,B0,C0,G0) | → 3(A0,B0,C0,G0,J0)     | → 2(A0,B0,C0,G0,J0,X0)     |                              |                        |
| 7  |                       | +-----> 3(A0,D0)           | → 2(A0,D0,Z0)          | → 2(A0,D0,Z0,a0)        | → 2(A0,D0,Z0,a0,b0)        |                              |                        |
| 8  |                       |                            |                        |                         |                            |                              |                        |
| 9  | <b>12(B0:-1,1,-1)</b> | → <b>12(B0,C0)</b>         | → 4(B0,C0,B0)          | → <b>4(B0,C0,B0,C0)</b> | → 2(B0,C0,B0,C0,D0)        | → 2(B0,C0,B0,C0,D0,E0)       |                        |
| 10 |                       |                            |                        |                         | +-----> 2(B0,C0,B0,C0,G0)  | → 2(B0,C0,B0,C0,G0,H0)       |                        |
| 11 |                       | +-----> 2(B0,C0,D0)        | → 2(B0,C0,D0,E0)       | → 2(B0,C0,D0,E0,F0)     | → 2(B0,C0,D0,E0,F0,A0)     |                              |                        |
| 12 |                       | +-----> <b>6(B0,C0,G0)</b> | → 2(B0,C0,G0,H0)       | → 2(B0,C0,G0,H0,I0)     | → 2(B0,C0,G0,H0,I0,J0)     |                              |                        |
| 13 |                       |                            | +-----> 3(B0,C0,G0,J0) | → 2(B0,C0,G0,J0,X0)     | → 2(B0,C0,G0,J0,X0,P0)     |                              |                        |
| 14 |                       |                            |                        |                         |                            |                              |                        |
| 15 | <b>12(C0:1,-1,1)</b>  | → 4(C0,B0)                 | → <b>4(C0,B0,C0)</b>   | → 2(C0,B0,C0,D0)        | → 2(C0,B0,C0,D0,E0)        | → 2(C0,B0,C0,D0,E0,F0)       |                        |
| 16 |                       |                            |                        |                         | +-----> 2(C0,B0,C0,G0)     | → 2(C0,B0,C0,G0,H0)          | → 2(C0,B0,C0,G0,H0,I0) |
| 17 |                       | +-----> 2(C0,D0)           | → 2(C0,D0,E0)          | → 2(C0,D0,E0,F0)        | → 2(C0,D0,E0,F0,A0)        | → 2(C0,D0,E0,F0,A0,B0)       |                        |
| 18 |                       | +-----> <b>6(C0,G0)</b>    | → 2(C0,G0,H0)          | → 2(C0,G0,H0,I0)        | → 2(C0,G0,H0,I0,J0)        | → 2(C0,G0,H0,I0,J0,K0)       |                        |
| 19 |                       |                            | +-----> 3(C0,G0,J0)    | → 2(C0,G0,J0,X0)        | → 2(C0,G0,J0,X0,P0)        | → 2(C0,G0,J0,X0,P0,Y0)       |                        |
| 20 |                       |                            |                        |                         |                            |                              |                        |
| 21 | <b>5(D0:-1,1,9)</b>   | → 3(D0,E0)                 | → 2(D0,E0,F0)          | → 2(D0,E0,F0,A0)        | → 2(D0,E0,F0,A0,B0)        | → 2(D0,E0,F0,A0,B0,C0)       |                        |
| 22 |                       | +-----> 2(D0,Z0)           | → 2(D0,Z0,a0)          | → 2(D0,Z0,a0,b0)        |                            |                              |                        |
| 23 |                       |                            |                        |                         |                            |                              |                        |
| 24 | 4(E0:1,9,-2)          | → 3(E0,F0)                 | → 3(E0,F0,A0)          | → 3(E0,F0,A0,B0)        | → 3(E0,F0,A0,B0,C0)        | → 2(E0,F0,A0,B0,C0,B0)       |                        |
| 25 |                       |                            |                        |                         |                            |                              |                        |
| 26 | <b>9(F0:9,-2,-1)</b>  | → 3(F0,A0)                 | → 3(F0,A0,B0)          | → 3(F0,A0,B0,C0)        | → 2(F0,A0,B0,C0,B0)        | → 2(F0,A0,B0,C0,B0,C0)       |                        |
| 27 |                       | +-----> 6(F0,N0)           | → 6(F0,N0,N0)          | → 6(F0,N0,N0,N0)        | → <b>6(F0,N0,N0,N0,N0)</b> | → 4(F0,N0,N0,N0,N0,S0)       |                        |
| 28 |                       |                            |                        |                         |                            | +-----> 2(F0,N0,N0,N0,N0,T0) |                        |
| 29 |                       |                            |                        |                         |                            |                              |                        |

図 30 [図 29] のうち辞書化する条件に適合するパターン

「条件 1) そのパターンよりも 1 音少ないパターンとの出現数の差が 2 回以上である」については、8.2.3 で述べたように、図中の D0、E0 のような、音数が増えてもパターンの出現数の変化が小さいパターンは、特定のフレーズの中で出現している可能性が高いという理由からである。今回は 9 音のパターンまでを対象に出現数を計算したが、実際には 9 音もの音高変化が一致している部分は、特定のパターンと言うよりは同一旋律の繰り返しである可能性が高い。よって、1 音多いパターンとの出現回数の差を条件とすることで、同一旋律の繰り返しではなく、変化のパターンとして適切である可能性が高いパターンを厳選し辞書化する。

「条件 2) すでに辞書化されたパターンとの重複分を除外した出現数が 2 回以上である」については、例に沿って説明する。[図 30] の 4 行目の 5 音パターン (A0,B0) と 6 音パターン (A0,B0,C0) に着目するとどちらのパターンも 8 回出現しているが、これは両者が音列中の同じ 8 箇所を指しているということである。つまり、5 音パターンの 8 回は 6 音パターンの 8 回と重複しており、これらの両方を辞書化することの意義は薄い。また、結果の中には「前半の重複」「後半の重複」の 2 種類の重複について考える必要がある。「前半の重複」とは先述したような、(A0,B0) と (A0,B0,C0) との間での重複、「後半の重複」とはこれに対し、9 行目の 5 音パターンと 4 行目の 6 音パターン、つまり (B0,C0) と (A0,B0,C0) との間での重複を指す。このような重複を除外するため、プログラム中での辞書化の手順を以下のように

設定した。

- a) 条件 1) に当てはまるパターンをすべて取り出す
- b) 取り出されたパターンのうち、音数の多いものから辞書化する  
ただし、自身より先に辞書化されたパターンの中で重複するものがある場合には、重複するパターンの出現数を自身の出現数から差し引き、1 回以下である場合には辞書化を行わない

例えば [図 30] では条件 1) に当てはまるパターンとして黄色で示した部分が取り出される。これらのうち、まず音数の最も多い 8 音パターンを辞書化する。ここではまず (A0,B0,C0,B0,C0) が辞書化される。次に同じ 8 音パターンである (F0,N0,N0,N0,N0) も、先に辞書化された (A0,B0,C0,B0,C0) とは重複しないためそのまま辞書化する。これで 8 音パターンの辞書化が終わったため、次は 7 音パターンである (B0,C0,B0,C0) について考える。すると、このパターンは先に辞書化された (A0,B0,C0,B0,C0) と後半部分が重複している。よって、(B0,C0,B0,C0) の出現数 4 回から (A0,B0,C0,B0,C0) の出現数 4 回を差し引き、残りが 0 となるためこのパターンは辞書化しない。このようにして、条件 1)、条件 2) に当てはまるパターンを判断する。

### 8.2.5 パターンの辞書化

これまでの処理によって、音列ファイルから生成した音高変化列中の各パターンの出現数を計算し、その結果を基に辞書化するパターンを決定することができた。ここからは、それらのパターンを辞書化し、辞書ファイルとして書き出す処理について述べる。

パターンの辞書化の具体的な例として、[図 30] 中で赤の点線枠で囲まれたパターンを用いながら説明する。これら 7 つのパターンは先述したように、辞書化のための条件を満たしているものであるため、これらを [表 8] ([表 7] の一部) のような形の辞書ファイルに適合するよう処理を行う。

| パターン名 | 正規表現パターン           | 重要な音 |
|-------|--------------------|------|
| flat  | {0, }+             | f    |
| up1   | \+1, \+1, {\+1, }+ | l    |

表 8 正規表現辞書パターンの例 ([表 7] の一部)

例えば、[図 30] 4 行目の 4 音パターンである A0 は、-2 -1 1 という音高変化のパターンである。これを [表 8] のように「パターン名 正規表現パターン 重要な音」という形で表すと、「A0 -2,-1,+1,1」のようになる。なお、重要な音については、現段階では仮として 1 (最後の音) を用いている。同様に 4 行目の 6 音パターンに関しては「A0B0C0 -2,-1,+1,-1,+1,1」となる。さらに、複数の辞書を統合した際にもどの楽曲から得られた辞書パターンなのかが分かるように、各パターン名の前半には楽曲の識別用の名前を付けるよう設定した。よって先の 2 つの例は「第 1 変奏の右手」であることを「v1r」という名で表し、実際には「v1r\_A0 -2,-1,1,1」「v1r\_A0B0C0 -2,-1,1,-1,1,1」として辞書ファイルに記述される。このようにして「きらきら星変奏曲」第 1 変奏右手から生成された辞書ファイルが [図 31] である。また、最低音数のパターンについて同様の方法で書き込みを行った結果が [図 32] である。

[図 31] [図 32] のようなパターンの記述では、半音と全音の細かい数値まで一致しなければパターンに当てはまるという判断にならず、同じ音型であってもその音型の基となる和声や調が異なる場合は抽出されない。このようなことを避け、全音と半音の細かい違いは無視するために、第 7 章で用いた正規表現を利用して [図 31] [図 32] を [図 33] [図 34] のように、0 以外の音高変化は±1 まで許容する、つまり音高変化に幅をもたせるよう出力の切り替えも実装した。なお変化の幅は基本的に±1 に設定しているが、変化 0 は「同じ高さの音が続く」という性質であり、変化 +1、変化-1 の「音の高さが変化する」性質とは異なるため、変化の幅として 0 は含めない（例えば-1 の変化で許容するのは-1 と-2 のみで 0 は含まない）よう処理した。これは、8.2.2 で述べたことと同義であり、音列中のパターン出現数を分析する際に用いた音高変化の幅の考え方を、各パターンの辞書にも適用させたこととなる。

```
v1r_A0B0C0B0C0 -2,-1,+1,-1,+1,-1,+1, l
v1r_F0N0N0N0N0 +9,-2,-2,-1,-2,-2,-2, l
v1r_A0B0C0 -2,-1,+1,-1,+1, l
v1r_B0C0G0 -1,+1,-1,+1,+1, l
v1r_P0Y0A0 -2,0,-2,-1,+1, l
v1r_A0 -2,-1,+1, l
v1r_D0 -1,+1,+9, l
v1r_F0 +9,-2,-1, l
v1r_J0 +1,+3,-1, l
v1r_N0 -2,-1,-2, l
v1r_P0 -2,0,-2, l
```

図 31 生成した辞書ファイルの例（最低音数以上、変化の幅なし）

```
v1r_A0 -2,-1,+1, l
v1r_B0 -1,+1,-1, l
v1r_C0 +1,-1,+1, l
v1r_D0 -1,+1,+9, l
v1r_E0 +1,+9,-2, l
v1r_F0 +9,-2,-1, l
v1r_G0 -1,+1,+1, l
v1r_J0 +1,+3,-1, l
v1r_L0 -1,+3,-2, l
v1r_M0 +3,-2,-1, l
v1r_N0 -2,-1,-2, l
v1r_P0 -2,0,-2, l
v1r_R0 -2,+9,-2, l
v1r_S0 -2,-2,+9, l
v1r_U0 -2,+7,-2, l
v1r_X0 +2,-2,0, l
v1r_Y0 0,-2,-1, l
v1r_a0 +9,-4,-3, l
v1r_b0 -4,-3,-2, l
v1r_c0 -3,-2,0, l
```

図 32 生成した辞書ファイルの例（最低音数、変化の幅なし）

```

v1r_A0B0C0B0C0 \-{3|2|1},\-{2|1},\+{1|2},\-{2|1},\+{1|2},\-{2|1},\+{1|2}, l
v1r_F0N0N0N0N0 \+{8|9|10},\-{3|2|1},\-{3|2|1},\-{2|1},\-{3|2|1},\-{3|2|1},\-{3|2|1}, l
v1r_A0B0C0 \-{3|2|1},\-{2|1},\+{1|2},\-{2|1},\+{1|2}, l
v1r_B0C0G0 \-{2|1},\+{1|2},\-{2|1},\+{1|2},\+{1|2}, l
v1r_P0Y0A0 \-{3|2|1},0,\-{3|2|1},\-{2|1},\+{1|2}, l
v1r_A0 \-{3|2|1},\-{2|1},\+{1|2}, l
v1r_D0 \-{2|1},\+{1|2},\+{8|9|10}, l
v1r_F0 \+{8|9|10},\-{3|2|1},\-{2|1}, l
v1r_J0 \+{1|2},\+{2|3|4},\-{2|1}, l
v1r_N0 \-{3|2|1},\-{2|1},\-{3|2|1}, l
v1r_P0 \-{3|2|1},0,\-{3|2|1}, l

```

図 33 生成した辞書ファイルの例（最低音数以上、変化の幅あり）

```

v1r_A0 \-{3|2|1},\-{2|1},\+{1|2}, l
v1r_B0 \-{2|1},\+{1|2},\-{2|1}, l
v1r_C0 \+{1|2},\-{2|1},\+{1|2}, l
v1r_D0 \-{2|1},\+{1|2},\+{8|9|10}, l
v1r_E0 \+{1|2},\+{8|9|10},\-{3|2|1}, l
v1r_F0 \+{8|9|10},\-{3|2|1},\-{2|1}, l
v1r_G0 \-{2|1},\+{1|2},\+{1|2}, l
v1r_J0 \+{1|2},\+{2|3|4},\-{2|1}, l
v1r_L0 \-{2|1},\+{2|3|4},\-{3|2|1}, l
v1r_M0 \+{2|3|4},\-{3|2|1},\-{2|1}, l
v1r_N0 \-{3|2|1},\-{2|1},\-{3|2|1}, l
v1r_P0 \-{3|2|1},0,\-{3|2|1}, l
v1r_R0 \-{3|2|1},\+{8|9|10},\-{3|2|1}, l
v1r_S0 \-{3|2|1},\-{3|2|1},\+{8|9|10}, l
v1r_U0 \-{3|2|1},\+{6|7|8},\-{3|2|1}, l
v1r_X0 \+{1|2|3},\-{3|2|1},0, l
v1r_Y0 0,\-{3|2|1},\-{2|1}, l
v1r_a0 \+{8|9|10},\-{5|4|3},\-{4|3|2}, l
v1r_b0 \-{5|4|3},\-{4|3|2},\-{3|2|1}, l
v1r_c0 \-{4|3|2},\-{3|2|1},0, l

```

図 34 生成した辞書ファイルの例（最低音数、変化の幅あり）

## 8.2.6 パターン辞書の統合

これまでの例は、[図 27] の入力として統合用ファイルが与えられない場合、つまり完全に新規で辞書を生成する場合である。統合用ファイルが与えられた場合も基本的には同様に動作するが、2点ほど異なる点があるため補足する。

1点目は、パターン名の付け方である。新規に辞書を生成する場合は、A0 から順に名前をつけ、Z0 に至るとその次は a0 になる。これが z0 に達すると次は A1 からという順で、最終的には z9 までの名前が割り当て可能である。統合用ファイルが与えられた場合にも終点が z9 という点是不変だが、新たに



A0 から名前を割り当てると、統合前の辞書内の A0 と区別が付きにくくなってしまうため、統合用辞書内のパターン名と重ならない名前を割り当てる。具体的には、統合用辞書（最低音数）の最後の行のパターン名が v1r\_D1 である場合、追加するパターンは v2r\_E1 から名前をつけ、v2r\_F1、v2r\_G1 と続くよう処理を行う。最低音数以上の辞書パターンの名前は最低音数のパターンとの名前と関連するため、最低音数のパターン名の名付けの際にこのように対応する。

2 点目は、パターンを辞書化する際に、全く同じパターンがすでに辞書として登録されている場合には、パターンの登録をしないという条件が加えられている。これは、パターンを辞書化する際の「条件 2) すでに辞書化されたパターンとの重複分を除外した出現数が 2 回以上である」とは意味が異なる。条件 2) ではパターン内の一部の重複（例えば A0B0C0 と B0C0）について考慮するという条件であるのに対し、「全く同じパターンは辞書にしない」という条件は、例えば音高変化-2 -1 1 というパターンが A0 という名前と B0 という名前で重複しないように設定している。これらの点が、統合用辞書が入力されない場合との違いである。

## 8.3 出力例と考察

ここまで、入力音列中に頻出する音高変化パターンから辞書を自動生成する手法について述べた。結果として、[図 31] [図 32] [図 33] [図 34] のようなパターン辞書を得ることができ、また複数の辞書ファイルの統合にも対応できた。そこで本節では、「きらきら星変奏曲」の変奏の中から、特に音高変化のパターンによる装飾が多く見られる変奏・パートを基に辞書を生成・統合し、その辞書の内容の分析およびその辞書を用いたパターン抽出を行う。なお、実際にこれらの変奏から生成した辞書について、4 音パターンの辞書ファイル内には 227 個、4 音以上のパターンの辞書ファイル内には 147 個がパターンとして記述された。なお、この辞書の内容は付録に掲載する。

### 辞書生成に用いた変奏、パート

|             |                  |
|-------------|------------------|
| 第 1 変奏 (右手) | 第 7 変奏 (右手)      |
| 第 2 変奏 (左手) | 第 11 変奏 (右手)     |
| 第 3 変奏 (右手) | 第 11 変奏 (左手)     |
| 第 4 変奏 (左手) | 第 12 変奏 (右手)     |
| 第 6 変奏 (右手) | 第 12 変奏 (左手)     |
| 第 6 変奏 (左手) | (すべて単音・単旋律化したもの) |

### 8.3.1 生成した辞書パターンの分析

以下では、生成した辞書内のパターンの分類について述べる。複数の音列（ここでは特定楽曲中の複数の変奏）から自動生成した辞書内のパターンを分類することで、どのようなパターンが辞書化されたのかを調査するとともに、辞書生成プログラム中では類似（重複）しないと判断されたパターン同士を、場

合によってはさらに統合する必要があるかを検討する材料となる。そこで、辞書生成プログラムの結果として得た、変化の幅を持たないパターン辞書ファイル（[図 31]、[図 32] の形をとるもの）を読み込み、いくつかの観点で分類するプログラムを実装した。分類の方法は主に以下の 2 つである。

- a) 音高変化の向きと大きさ
- b) 音高変化の大きさの順

- a) 変化の向きと大きさ

辞書パターン内では音高変化を-1、+8、0のように数値で表していたが、この音高変化1つ1つを、変化の向きと大きさという観点に置き換え、次のように分類する。

- 変化の向き：上がる (U) / 等しい (R) / 下がる (D)
- 変化の大きさ：大きい (L) / 小さい (S)

変化の向きと大きさは、2.1.1 で述べたように音高変化を基にした抽象化の1つの観点である。ここでは、音高変化が正の数であれば U、負の数であれば D、0 であれば R と置き換え、音高変化の絶対値が 1~6 であれば S、7 以上であれば L をに置き換える。例えば音高変化が-1 の場合には SD (小さい+下がる)、+8 の場合には LU (大きい+上がる) と表現する。1つの音高変化に対しては SU,LU,R,SD,LD の 5 通りの分類があり、最低音数が 4 の場合には音高変化が 3 つであるので、最大で  $5 \times 5 \times 5=125$  通りに分類される。「きらきら星変奏曲」による生成辞書パターン (最低音数、変化の幅なし) をこれらの組み合わせによって表現し、同じものをまとめて分類する処理を行うと、[図 35] のような出力が得られる。

|  |  |
|--|--|
| <ul style="list-style-type: none"> <li>★ 0:SD,SD,LD,(1) [D,D,D,]<br/>v11l_e5,</li> <li>★ 1:SD,LD,SD,(1) [D,D,D,]<br/>v11l_q4,</li> <li>★ 2:SD,SD,SD,(10) [D,D,D,]<br/>v1r_N0, v1r_b0, v3r_A1, v3r_E1, v7r_c3, v11r_Q4, v11r_R4, v11r_Y4, v11l_t4, v11l_D5,</li> <li>★ 3:LD,SD,R,(1) [D,D,R,]<br/>v3r_t1,</li> <li>★ 4:SD,SD,R,(3) [D,D,R,]<br/>v1r_c0, v3r_L1, v3r_h1,</li> <li>★ 5:LD,SD,LU,(2) [D,D,U,]<br/>v7r_o3, v11l_r4,</li> <li>★ 6:SD,LD,LU,(1) [D,D,U,]<br/>v4l_D2,</li> <li>★ 7:SD,LD,SU,(2) [D,D,U,]<br/>v3r_a1, v4l_y1,</li> <li>★ 8:SD,SD,LU,(4) [D,D,U,]<br/>v1r_S0, v7r_x3, v11r_V4, v11l_u4,</li> <li>★ 9:SD,SD,SU,(9) [D,D,U,]<br/>v1r_A0, v6r_h2, v6l_L3, v6l_R3, v11r_S4, v11r_Z4, v12r_Y6, v12r_b6, v12r_o6,</li> <li>★ 10:LD,R,SD,(1) [D,R,D,]<br/>v12r_x6,</li> <li>★ 11:SD,R,SD,(3) [D,R,D,]<br/>v1r_P0, v3r_M1, v3r_i1,</li> <li>★ 12:SD,R,R,(1) [D,R,R,]<br/>v11r_L4,</li> <li>★ 13:LD,R,R,(1) [D,R,R,]<br/>v3r_U1,</li> </ul> | <p style="text-align: center;">★番号:パターン分類 (該当パターン数) [音の向きのみ分類]<br/>該当パターン名</p> |
|--|--|

図 35 生成辞書内のパターン分類 a) の出力例

この図では 14 種類のパターン分類の結果部分しか示していないが、実際には 82 種類にパターン分類された。この出力では音の変化の向きのみ分類 ([D,D,D] など) ごとにまとめて表示しており、変化の向きが同じパターンの中での変化の大きさの組み合わせを調べることができる。例えば [図 35] の先頭から 6 行目までで、[D,D,D] というひたすら下行をするパターン内でも、SD,SD,SD という少しずつ下行するパターンには多くの種類があることがわかる。これを該当パターン数順に並び替えたものが [図 36] である。この図からは 82 個の分類のうち、パターンの種類が多い 16 個の分類について表している。

```

★ 0:SU,SU,SD,(13) [U,U,D,]
v1r_j0, v2l_f0, v3r_y0, v4l_w1, v4l_P2, v4l_V2, v6r_r2, v6r_v2, v7r_f3, v7r_w3, v7r_A4, v11r_f4, v12r_P7,
★ 1:SD,SU,SD,(12) [D,U,D,]
v1r_B0, v1r_L0, v6r_t2, v6r_x2, v6l_I3, v6l_M3, v11r_T4, v11r_a4, v12l_x5, v12l_K6, v12r_K7, v12r_M7,
★ 2:SU,SD,SD,(12) [U,D,D,]
v1r_M0, v2l_g0, v3r_z0, v3r_D1, v3r_K1, v3r_o1, v6l_J3, v6l_Q3, v7r_g3, v7r_B4, v11r_U4, v11l_C5,
★ 3:SU,SD,SU,(10) [U,D,U,]
v1r_C0, v4l_Q2, v6r_s2, v6r_u2, v6r_w2, v11r_n4, v12r_g6, v12r_J7, v12r_L7, v12r_Q7,
★ 4:SD,SD,SD,(10) [D,D,D,]
v1r_N0, v1r_b0, v3r_A1, v3r_E1, v7r_c3, v11r_Q4, v11r_R4, v11r_Y4, v11l_t4, v11l_D5,
★ 5:SD,SD,SU,(9) [D,D,U,]
v1r_A0, v6r_h2, v6l_L3, v6l_R3, v11r_S4, v11r_Z4, v12r_Y6, v12r_b6, v12r_o6,
★ 6:SU,LD,LU,(7) [U,D,U,]
v2l_j0, v2l_n0, v2l_q0, v2l_t0, v4l_G2, v4l_J2, v12l_B6,
★ 7:SD,LU,SD,(6) [D,U,D,]
v1r_R0, v1r_U0, v3r_G1, v11r_C4, v11r_W4, v12l_z5,
★ 8:SU,SU,SU,(6) [U,U,U,]
v3r_v0, v3r_w0, v3r_x0, v4l_U2, v4l_Z2, v6r_q2,
★ 9:SU,LD,SU,(6) [U,D,U,]
v4l_L2, v4l_S2, v4l_X2, v6l_O3, v12l_o5, v12r_P6,
★ 10:LU,SD,SD,(5) [U,D,D,]
v1r_F0, v1r_a0, v3r_g1, v11r_P4, v11r_X4,
★ 11:R,SD,SD,(5) [R,D,D,]
v1r_Y0, v6r_g2, v12r_X6, v12r_a6, v12r_n6,
★ 12:SD,SU,LD,(5) [D,U,D,]
v2l_i0, v2l_m0, v2l_p0, v2l_s0, v6l_S3,
★ 13:LD,LU,SD,(5) [D,U,D,]
v2l_k0, v2l_o0, v2l_r0, v2l_u0, v12l_k5,
★ 14:LD,SU,SU,(5) [D,U,U,]
v4l_z1, v4l_M2, v4l_T2, v4l_Y2, v11l_A5,
★ 15:SD,SU,SU,(4) [D,U,U,]
v1r_G0, v2l_e0, v3r_I1, v7r_e3,

```

図 36 [図 35] をパターン数順に並べ替えたもの

この結果から次のようなことを知ることができる。

- S から構成される組み合わせの種類が多い

これは、変化の大きさが 6 以下の音高変化から構成されるパターンの種類が多いことを示しており、逆に言うと大きな変化をするパターンの種類は少ないことを表している。特に、図の赤線で示した分類は 4 音パターンのすべてが小さな変化量であり、楽曲中では音高の小さな変化を組み合わせた多くの種類のパターンが用いられていることが分かる。また、大きな音高変化をする部分は旋律の区切りとなる部分が多いため、パターンとして辞書化されにくかったことが理由として考えられる。

- S から構成される組み合わせの中で、SD,SD,SD や SD,SU,SU のパターンは種類が少ない

この結果から考えられることは、「きらきら星変奏曲」の変奏を通して、このようなパターンの中で特に出現しやすい変化の大きさがあるか、もともとこのように分類されるパターンの出現数が少ないかということである。これについては、パターンの種類と合わせて、このように分類されるパターンが楽曲中に何回出現したかを調べることで検討できる可能性がある。

#### b) 変化の大きさの順

a) では変化の向きと大きさについて分類したが、変化の大きさについては 6 半音より大きいかさ

うでないかという分類しかなかった。そこでこれについてより詳しく見るために、パターン内での変化の大きさの順で分類し、どの位置で大きな変化をするのかといったことを調べる。例えば、-1,+2,0 というパターンであれば、これらの変化の大きさはそれぞれ 1,2,0 である。この 3 つの数値の中では 1 は 2 番目に大きく、2 は 1 番目、0 は 3 番目となるので、このパターンの分類は 2 番,1 番,3 番となる。なお、同じ変化量の場合は同じ数字で表す。このように分類した場合、4 音パターンの辞書について分類の種類を数え上げたところ、最大で 13 通りとなる。

```

★ 0:2番,2番,1番,(29)
v1r_D0, v1r_S0, v2l_e0, v2l_i0, v2l_m0, v2l_p0, v2l_s0, v3r_D1, v3r_I1, v3r_V1, v4l_x1, v4l_C2, v6r_r2,
v11r_G4, v11r_J4, v11l_u4, v11l_D5, v11l_J5, v11l_V5, v12l_x5, v12r_N6, v12r_R6, v12r_e6, v12r_p6,
★ 1:3番,1番,2番,(26)
v1r_E0, v1r_L0, v1r_Y0, v2l_q0, v3r_w0, v3r_N1, v3r_Q1, v3r_W1, v3r_f1, v3r_r1, v4l_D2, v4l_J2, v4l_L2,
v11r_O4, v11r_T4, v11l_q4, v11l_z4, v12l_o5, v12l_B6, v12r_X6, v12r_J7,
★ 2:1番,2番,3番,(26)
v1r_F0, v1r_M0, v1r_a0, v1r_b0, v1r_c0, v2l_r0, v3r_x0, v3r_A1, v3r_O1, v3r_R1, v3r_X1, v3r_s1, v3r_t1,
v6l_I3, v7r_g3, v7r_w3, v11l_x4, v12l_k5, v12l_v5, v12r_Q6, v12r_c6,
★ 3:1番,2番,2番,(21)
v1r_A0, v2l_d0, v2l_l0, v3r_H1, v3r_U1, v4l_w1, v4l_Q2, v4l_T2, v4l_V2, v6r_u2, v6l_P3, v6l_R3, v7r_z3,
v11l_U5, v12l_A6, v12r_Q7,
★ 4:2番,1番,3番,(19)
v2l_u0, v3r_G1, v3r_L1, v3r_a1, v3r_h1, v4l_H2, v6r_h2, v6r_t2, v11r_C4, v11r_E4, v11r_R4, v11r_W4, v11r
v12r_g6, v12r_L7,
★ 5:1番,3番,2番,(19)
v3r_K1, v3r_Y1, v3r_g1, v3r_i1, v4l_F2, v4l_K2, v4l_Z2, v6r_b2, v6r_x2, v7r_r3, v11r_D4, v11r_P4, v11r_S
v12r_x6, v12r_K7,
★ 6:2番,1番,2番,(15)
v1r_J0, v1r_R0, v1r_U0, v3r_T1, v4l_y1, v6r_a2, v6r_c2, v6r_e2, v6r_w2, v6l_J3, v6l_Y3, v7r_c3, v7r_y3,
★ 7:2番,3番,1番,(15)
v3r_v0, v3r_M1, v3r_P1, v3r_S1, v3r_j1, v4l_I2, v7r_l3, v7r_o3, v7r_x3, v11r_N4, v11l_r4, v11l_v4, v11l
★ 8:1番,1番,1番,(14)
v1r_B0, v1r_C0, v1r_G0, v3r_o1, v6r_v2, v11r_M4, v11r_Z4, v11r_n4, v11l_C5, v12l_D6, v12l_E6, v12l_G6, v
★ 9:2番,1番,1番,(14)
v2l_f0, v2l_j0, v2l_n0, v3r_y0, v3r_E1, v6r_q2, v6l_L3, v7r_m3, v7r_p3, v11r_Y4, v11l_t4, v11l_K5, v12r
★ 10:1番,1番,2番,(13)
v1r_X0, v2l_g0, v2l_k0, v2l_o0, v3r_z0, v4l_U2, v6l_Q3, v7r_k3, v7r_n3, v7r_q3, v11r_a4, v11l_L5, v12r_w
★ 11:3番,2番,1番,(11)
v2l_t0, v3r_Z1, v4l_G2, v6r_g2, v6r_s2, v11r_Q4, v11r_V4, v11l_E5, v12r_O6, v12r_a6, v12r_f6,
★ 12:1番,2番,1番,(5)
v1r_N0, v1r_P0, v6r_d2, v12r_W6, v12r_M7,

```

図 37 生成辞書内のパターン分類 b) の出力例 (パターン数の多い順)

[図 37] は [図 35] に用いたものと同じ辞書についての分類結果を、分類されるパターン数の多い順に並べ替えたものである。この結果から、次のようなことを読み取ることができる。

- 同じ変化量がパターン内で複数回出てくる (2 番,2 番,1 番など) パターンの種類は多い傾向にある

これについて、例えば音高変化の変化量が 2 で表される場合、これと同じ変化量であるものは音高変化-2 と +2 の 2 通りである。これに対し、異なる変化量であるものは +1、-1、+3、-3、+4、-4…など様々な場合が考えられることから、4 音パターンにおける 3 つの音高変化のうちどれかが同じ変化量であるというパターンは、3 つの音高変化がすべて異なるパターンよりも少ないように思われる。しかし、「きらきら星変奏曲」においては [図 37] の赤線で示したように、3 つの音高変化のうち 2 つの変化量が同じであるパターンの種類が少ないとは言え

ず、逆に 2 番,2 番,1 番の分類にいたっては他の分類よりも多くのパターンの種類をもつことが示された。これは、音高の変化量の中で、特にパターンとして出現しやすい数値があることが考えられる。可能性としては、全音階上での 1 音分の変化を表す 1 と 2 がパターンに用いられやすく、4 音パターンの 3 つの音高変化の中でも複数回現れていることが考えられる。これは、「a) 音高変化の向きと大きさ」の分類において音高変化の変化量が小さいものの種類が多く存在することの理由の 1 つとしても同様のことが言える。実際により深く分析を行うためには、やはり他の楽曲の結果との比較や、辞書パターンのうち音高の変化量として 1 や 2 を含むものがどの程度の割合で存在するかという分類を行う必要がある。

### 8.3.2 生成した辞書を用いたパターン抽出と考察

生成した辞書についての考察を行うため、8.3.1 では辞書パターンの内容を分析した。さらに以下では、第 7 章のプログラムに生成した辞書ファイルを渡し、実際にパターン抽出を行うことについて述べる。用いた辞書ファイルは 8.3.1 と同じ「きらきら星変奏曲」の特定の変奏から生成した、最低音数以上の辞書ファイルである。これを、まずは「きらきら星変奏曲」第 1 変奏右手の音列に適用し、辞書パターンの抽出を行った。[図 38] はその結果を Processing で可視化したものであり、この図の見方は [図 26] と同様である。

自動生成した辞書を適用した [図 38] と、先行研究などに倣った装飾の特定パターンを辞書とした [図 39] ([図 26] の再掲) を比較すると、[図 38] では明らかに抽出されたパターンの数が多い。原因として、主に以下の 2 点が考えられる。

- 辞書パターン数自体が自動生成した辞書の方が圧倒的に多い  
装飾の特定パターンは 14 通りであるのに対し、今回生成したパターンは 227 通りと、網羅しているパターンの種類の差がそのまま抽出数の差になったと考えられる。

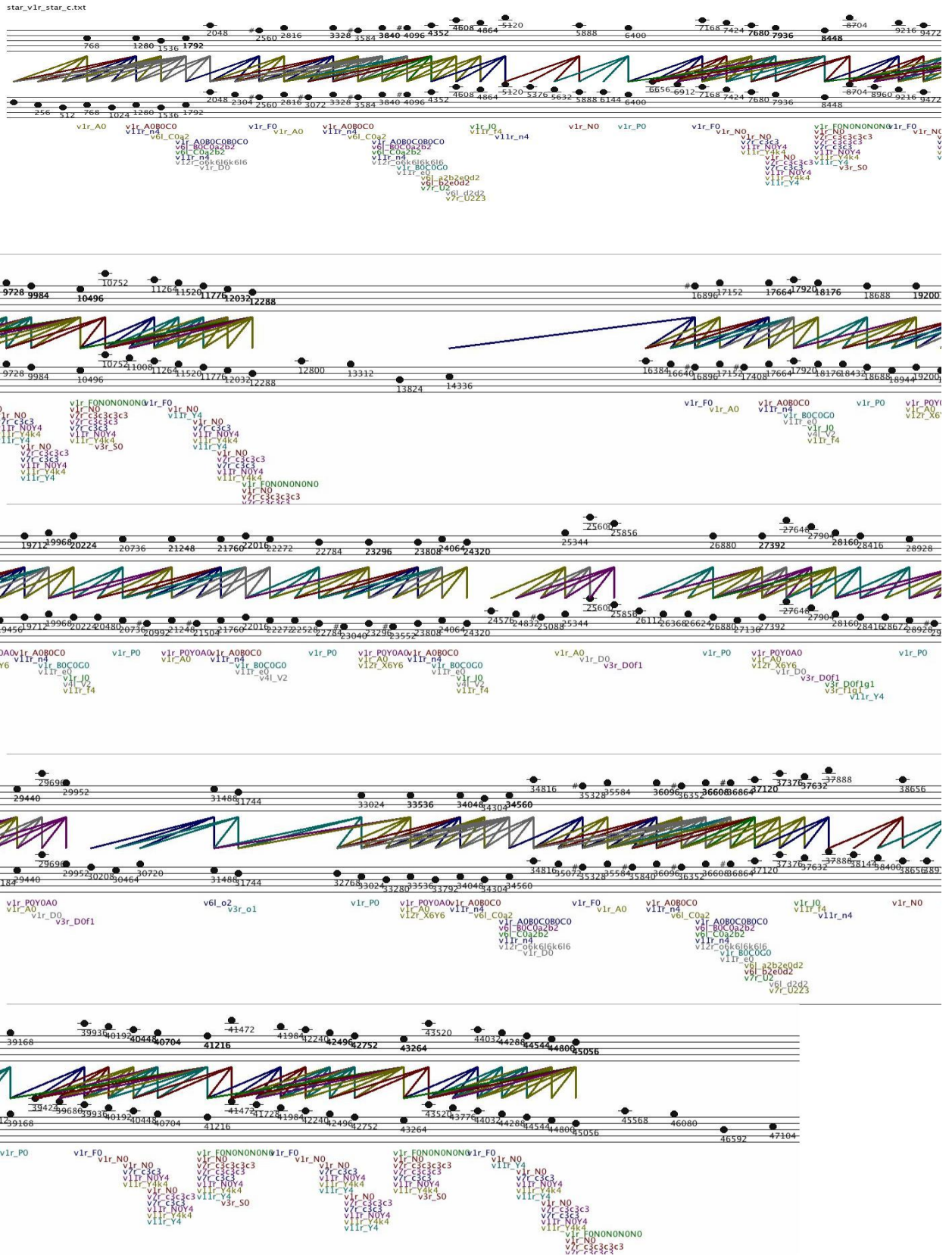


図 38 「きらきら星変奏曲」第 1 変奏右手から、生成した辞書パターンを抽出した結果

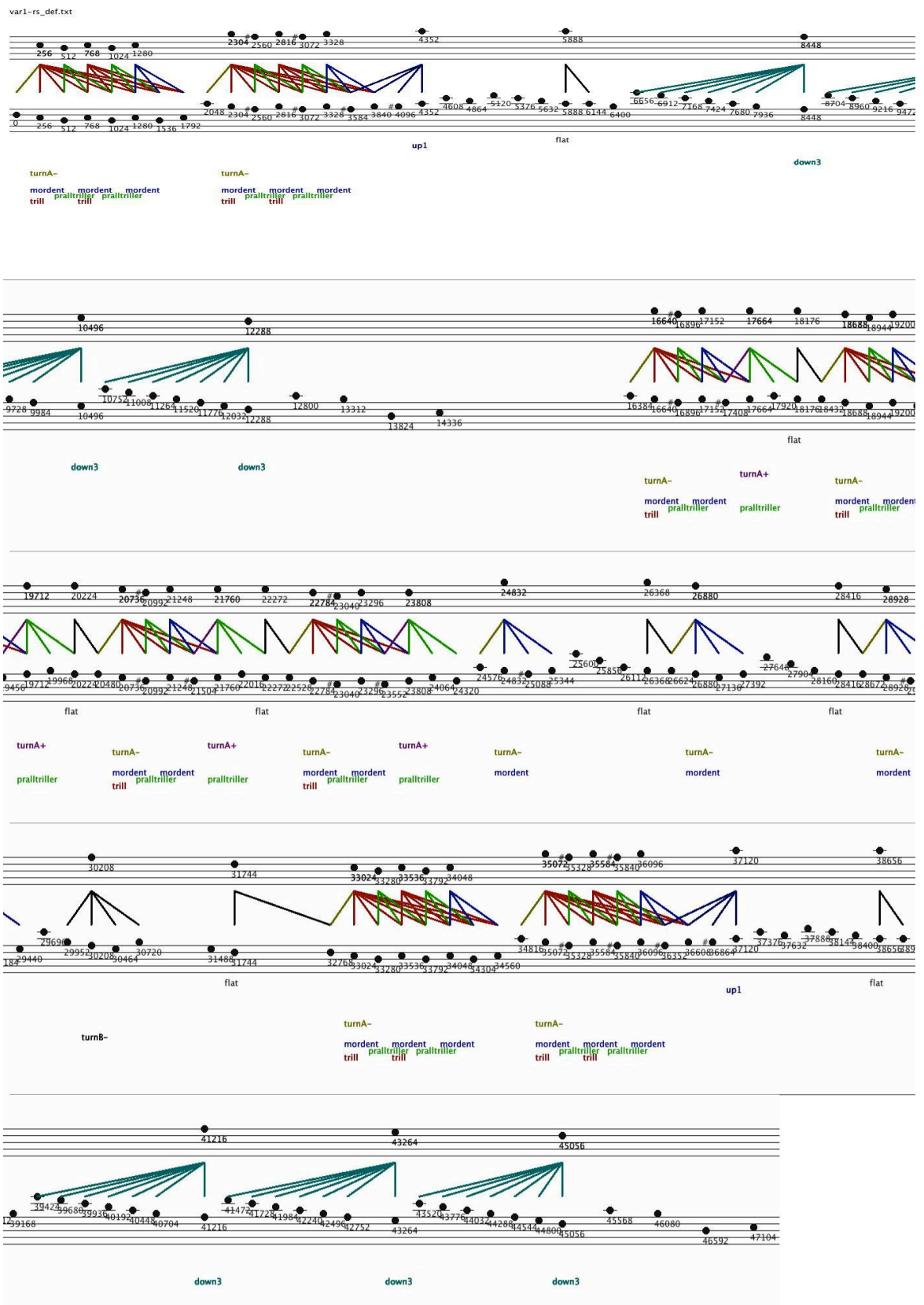


図 39 「きらきら星変奏曲」第 1 変奏右手から、特定パターンの抽出を行った結果（[図 26] の再掲）



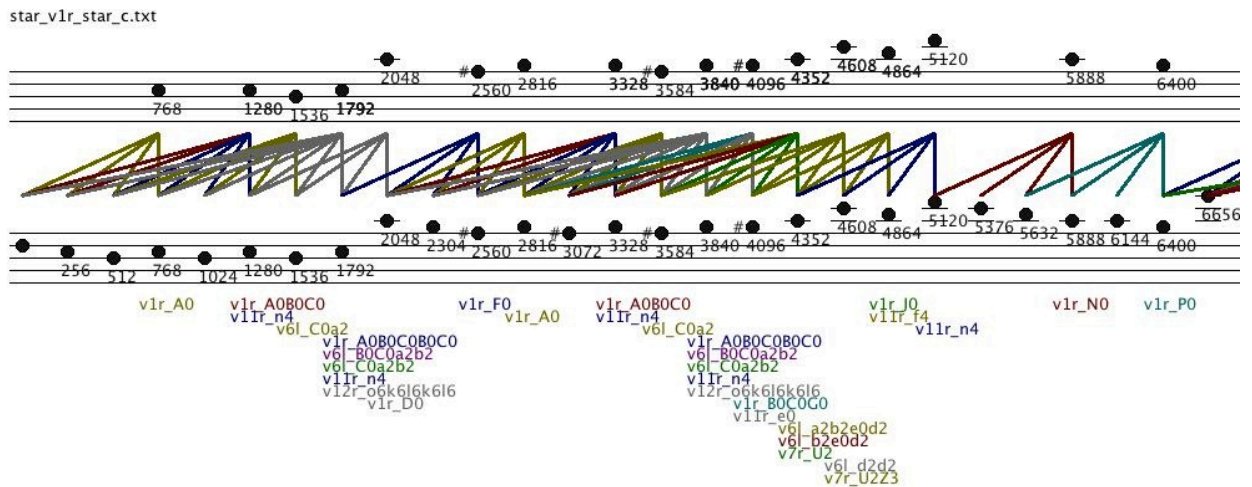


図 40 「きらきら星変奏曲」第 1 変奏右手から、生成した辞書パターンを抽出した結果（一部）

- 変化の幅の設定によるパターンの重複

[図 40] は [図 38] の冒頭部分を拡大した図である。この図でのパターン名の記述位置は、親となる音の下になるように設定している。加えて、今回は辞書を自動で生成する際に、親はすべてパターンの最後の音となるよう記述を行った。つまり、パターン名の記述位置が同一の音の下にある場合は親が同一ということ、そして同一の音が親であるパターンのうち、パターンを構成する音の数も一致するものは、同じ音列を別のパターンとして抽出していることになる。

例えば、[図 40] の 1792 の音は「v1r\_A0B0C0B0C0」「v6l\_B0C0a2b2」「v6l\_C0a2b2」「v1r\_n4」「v12r\_o6k6l6k6l6」の 5 つのパターンの親となっていると言える。このうち、「v1r\_A0B0C0B0C0」と「v12r\_o6k6l6k6l6」はどちらも 8 音パターンであり、この曲の最初の音から 1792 の音までがこれに当てはまることを表している。このようなことが起きるのは、辞書ファイルを統合する場合の辞書化の条件と、辞書に登録する際に変化の幅を持たせることがうまく噛み合っていないことに起因する。

具体的には、同一音列内から頻出パターンを調べる際には、8.2.2 で述べたように、01-2 と 02-3 のようなそれぞれの変化量の差が ± 1 までの場合、これら 2 つのパターンは類似すると判断し、1 つのパターンに集約するよう設定している。しかし、辞書の統合を行う際は、8.2.6 で述べたとおり、全く同じパターンは辞書にしないという条件を用いている。例えば統合する前の辞書に、他の音列から生成した 01-2 というパターンが存在した上で、新たに 02-3 というパターンを辞書化する際は、これを類似すると判断せず、完全一致のみを類似の判断としているため、どちらも辞書パターンとして登録されてしまう。さらに、辞書化する際にはパターンに変化の幅を持たせ、± 1 までの変化量の差は無視して抽出を行うように指定している。このような「パターンの類似」の判断が場合によって異なることが原因で、同じ音列から同じ音数の複数のパターンが抽出されるということが起きたと考えられる。

### 音高変化列:221-22223の場合

|           |        |    |
|-----------|--------|----|
| Group4[0] | 2 2 1  | A0 |
| Group4[1] | 2 1 -2 | #  |
| Group4[2] | 1 -2 2 | #  |
| Group4[3] | -2 2 2 | #  |
| Group4[4] | 2 2 2  | A0 |
| Group4[5] | 2 2 3  | #  |




図 41 変化の幅の設定により矛盾が生じる例

変化の幅の設定に関連して、パターンの重複の他にも生じる問題が存在する。例えば [図 41] のように、ある音列内から頻出パターンを調べる際に、まず 2 2 1 という 4 音パターンに着目したとする。このような場合、2 2 1 に類似するパターンとして Group4[4] の 2 2 2 が当てはまり、Group4[0] 及び Group4[4] に A0 という名前が付く。続いて Group4[1] のパターンが、名前の付いた Group4[0] と Group4[4] を除くパターンのうち類似するものがないか計算を行う。これを Group4[5] まで繰り返すと、[図 41] の右のセルのように、A0, #, #, #, A0, # という結果になり、最初に出現した A0、つまり 2 2 1 が辞書化の対象となる。ここで、Group4[5] の 2 2 3 については 2 2 1 とは類似しないとされている。これは、音高変化 2 2 1 の 3 番目の数字である 1 と、2 2 3 の 3 番目の数字である 3 の差が 2 であり、変化の幅として含まれないということに基づく処理の結果である。しかし、Group4[4] と Group4[5] に着目すると、この 2 つのパターンは本プログラムの基準に基づくとは類似していると言える。つまり、仮に 2 2 2 というパターンと 2 2 1 のパターンの位置を入れ替えた場合には、Group4[4] である 2 2 1 と Group4[5] である 2 2 3 はどちらも A0 として名付けられ、この場合は 2 2 2 が辞書化の対象となる。このように、音高変化列内の位置によって出現数や辞書化されるパターンが異なる結果となるケースが存在する。このことから、変化の幅という考え方について、どこまでを類似するパターンとして扱うのか、どの段階で類似するパターンを統合するのか、また類似するパターンをどう表すのかということについては検討・実装し直す必要がある。アプローチの視点としては、「変化の幅」という考えを用いている部分のアルゴリズムを検討し直す他に、現段階で出力された辞書に記載されたパターンについて、さらなる統合と整理を行うという方法がある。

また、変化の幅が関連する問題点以外にも、今後さらなる検討や分析を行う視点として、次のようなものが挙げられる。

- 他の楽曲への適用

「きらきら星変奏曲」から生成した辞書を、モーツァルトのピアノソナタ第 11 番 K.331 の第 1 楽章、第 1 変奏右手に適用した結果が [図 42] である。ピアノソナタ第 11 番 K.331 の第 1 楽章は、ソナタ曲の第 1 楽章には珍しい変奏曲形式であり、主題に続く 6 つの変奏で構成されている。[図 42] を見ても、[図 38] と同様、パターンの類似判定のズレから生じる、同じパターンを別パターンとして抽出しているケースが見られる。また、パターンの重複されない部分に目を向ける

と、この曲の中では頻出のパターンでも、「きらきら星変奏曲」を基にした辞書には登録されていないケースもいくつかあることが分かる（1段目の2304の音から3328の音までなど）。このような、他の音列を基にした辞書にはないパターンや、辞書に登録されているパターンの中でも、他の曲とは出現数に大きな違いがあるパターンなどが、その音列を特徴づけるパターンである可能性が高い。よって、同一変奏曲内の変奏同士で、あるいは異なる変奏曲同士で生成した辞書を適用し、その抽出パターンの違いや抽出数の分布を調べることは、それぞれのパターンが特定の楽曲固有のものであるのか、また幅広い楽曲に出現する汎用的なものであるのかということの分析につながる。

k331\_v1r\_star\_c.txt

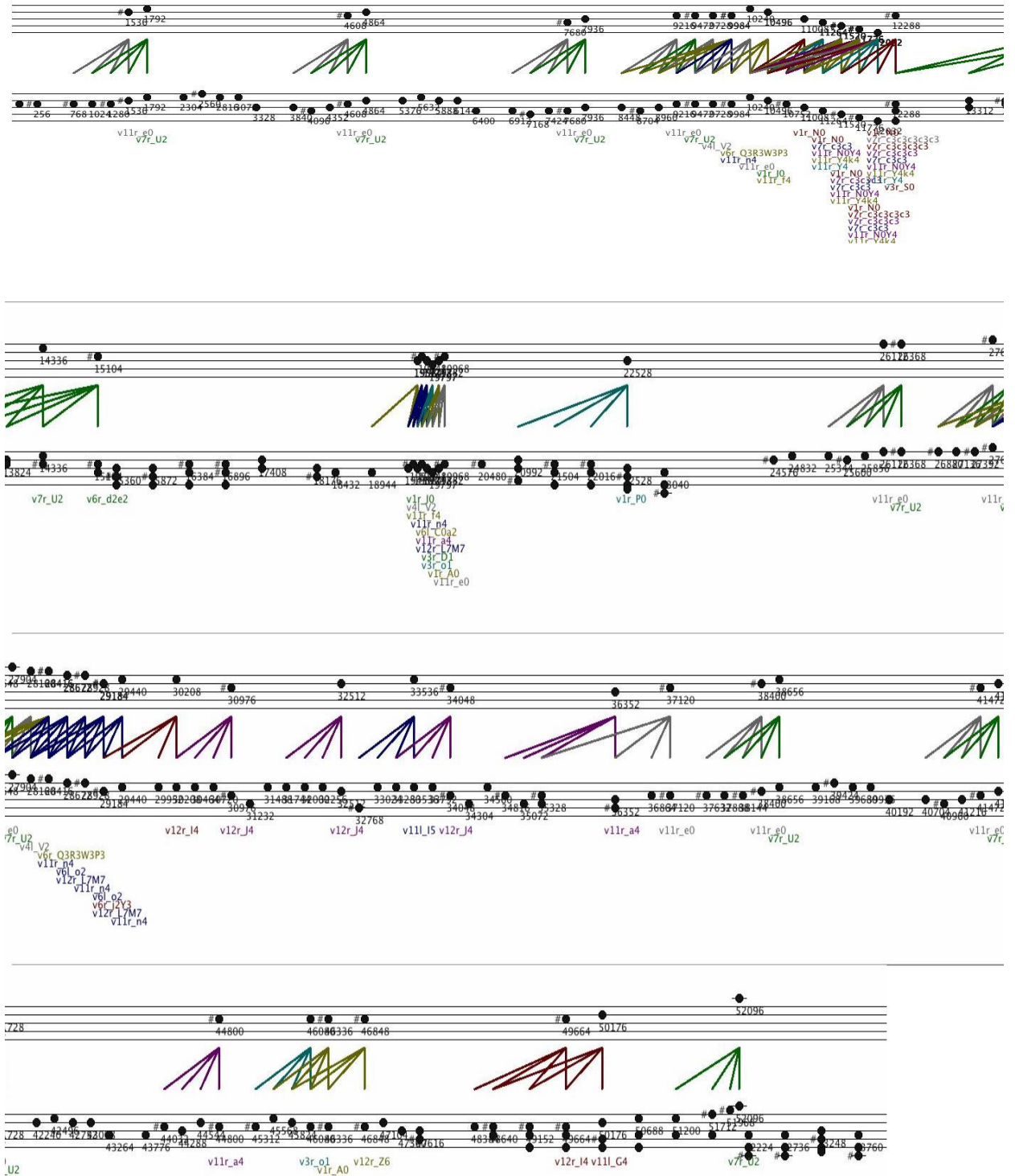


図 42 ピアノソナタ K.331 第 1 楽章 第 1 変奏右手に生成した辞書を適用した結果

- パターン名の再検討

複数の楽曲へ適用し、抽出されるパターンの種類や数を分析することにあたっては、本プログラムでのパターンの名前の付け方についても再検討が必要である。現在は楽曲の頻出パターンに名前を付ける際、辞書の基になった楽曲の識別用文字列（例：v1r）と、アルファベットと数字を合わせたパターンの名前（例：A0）やその組み合わせ（例：A0B0）を結合したものをを用いている。しかし、複数の楽曲から複数の辞書を生成し、その内容や抽出数を分析する際、現在用いている名前では辞書をまたがった分析を行うことが困難である。例えば、あるパターンが辞書1と辞書2の両方に記載されている場合、辞書1では dicA\_A0、辞書2では dicB\_B2 のように、全く別の名前で表現されることとなる。よって、複数の辞書にまたがる分析に向けて、このような名付けの方法を変更する必要がある。例えば、パターンの音型を基に名前を付けると、パターン名から異なる辞書内の類似パターンについての情報を取り出すことができる。

- 重要な音の位置を自動推定

本章においては、楽曲に頻出するパターンの辞書化を行ったが、その際に指定する「重要な音の位置」についてはパターンの内容に関わらず「1（最後の音）」を設定していた。しかし、第7章で用いた辞書パターンのように、パターン中の重要な音の位置はそれぞれのパターンによって異なる。そのため、本章においては音高変化をパターン化し、その特徴を分析するということに重点を置いた一方で、それを適切な簡約へ活かす手法については検討が至らなかった。本研究の内容に加え、「抽出されたパターン内で重要な音の位置の自動推定」を行うアルゴリズムの検討・実装を行うことができれば、楽曲の骨格も考慮したより深い分析につながると考えられる。例として、

- ある楽曲について複数の辞書についてパターンを抽出・簡約を行った場合の、辞書の違いによる簡約音列の相違点の分析
- 簡約音列を基にした辞書の生成を行うことで、階層的な簡約に適したパターン抽出法の検討などが挙げられる。

- 和音や複数旋律への対応

本研究全体を通して、今回はシステムの入力対象を単旋律・単音のデータに制限した。しかし、7.3で述べたように、音高変化のパターンは和音の一部としても存在し、さらには和音の一部が複数の旋律を成す中にパターンが見られる場合もある。例えば、[図43]は「きらきら星変奏曲」の第8変奏の一部であるが、この部分においては色付きの音符で示したとおり、同じ音型が右手パートに並行して現れる。またこの音型は左手パートにもオクターブの和音として出現している。本研究で作成したシステムでは、これらのように複数の旋律が現れる場合、それぞれを独立した旋律として分離しなければ処理を行うことができない。よって入力の一部に和音が存在する場合は、和音の中での重要な音を選択する手法や、和音の一部が独立した旋律を成すとき、これを分離する手法の検討が望まれる。



図 43 複数の旋律でパターンが見られる例

## 第9章 まとめ

本研究では、変奏曲の楽曲構造解析の一環として、変奏から音高変化のパターンを抽出する、及びその結果を基にした簡約を行い、主題との関係を分析することを目的とした。具体的に取り組んだこととして、まず第5章ではMIDI形式のファイルから必要な情報を取り出す前処理を行った。その後第6章では先行研究や装飾記号を参考に設定した辞書パターンの抽出と、各パターンの重要な音の位置を与えることによる簡約化を行うプログラムを作成した。このプログラムに入力音列として旋律を与えた場合にはその旋律の簡約を行うことができるが、その簡約の結果生成された音列をさらに入力音列とすることで、階層的な簡約をすることも可能である。実際にこの抽出・簡約プログラムに「きらきら星変奏曲」のいくつかの変奏を与え、パターン抽出及び簡約を行った出力から、次のようなことが得られた。

- 設定した各辞書パターンを楽曲中から抽出し、簡約を行うことができた
- 単純なパターンから構成される変奏については、階層的な簡約を含め適切な簡約を行える場合も存在する
- 旋律のまとまりの境界にまたがるパターンが存在する

しかし、一見してパターン性が強いように思われる部分でも、辞書パターンに登録されていないためパターンとして抽出できないケースが目立った。そこで、実際の楽曲に出現するパターンを基に辞書を生成することによって、楽曲固有のパターンを含む多様なパターンについての抽出を行うこととした。多様なパターンが得られるというメリットの他にも、複数の楽曲から辞書を生成し、辞書自体の内容や辞書を適用した抽出結果などを比較することで、各パターンが他の楽曲にも汎用的に用いられるものであるのか、逆にその楽曲に現れる特徴的なものであるのかを分類することもできる。第7章では楽曲の頻出パターンを辞書化するための準備として、第6章で利用したパターン抽出・簡約のプログラムに外部のテキストファイルを辞書として読み込ませ、そのパターンについての抽出と簡約を行う手法やプログラムについて述べた。ここでは、辞書パターンをプログラムの外部で表現する際に正規表現を用いることによって、ある程度抽象的なパターンの指定ができるようにした。第7章で述べたプログラムに第6章の辞書パターンを表現した辞書ファイルを読み込ませ、第6章のプログラムと第7章のプログラムの出力を比較した。結果、両者には次のような違いがあることが考えられる。

- 第6章のプログラムではパターン同士の関係を考慮した抽出を行うことができる  
(第7章のプログラムでは、現段階ではパターン同士は独立に抽出することしかできない)
- 第7章の辞書パターンの表現方法では、音階に沿った変化を厳密に表現することはできない

このように、手法について再検討の余地はあるものの、第7章で目的としていた「外部の辞書内に含まれる辞書パターンの抽出・簡約」は達成できた。辞書パターンが柔軟に設定できるこのプログラムは、変奏曲の分析を目的とした辞書の他に次のような辞書を生成し、パターン抽出を行うことで構造解析に活かすことができる。

- 階層的な簡約に適した辞書
- 右手と左手（旋律と伴奏）それぞれに適した辞書
- 和音を対象とした辞書

第8章では、第7章に与える辞書ファイルとして、入力音列中で頻出のパターンを書き出すプログラムについて述べた。このプログラムではまず入力音列内のパターンの分析として、最低音数のパターンと、それを組み合わせた様々な音数のパターンの出現数を計算した。また、各パターンの出現数を、パターン同士の関係性によって並べ替えて表示を行うことにより、それぞれのパターンが楽曲内でどのような場面で用いられているかを知ることができた。このような情報を基に、各パターンが辞書化する条件に当てはまるかどうかを判定し、条件を満たすものを辞書として書き出した。加えて、辞書を生成する際、既に存在する辞書ファイルを与えることで、既存の辞書と新たに生成された辞書の統合する機能も実装した。結果として、辞書ファイルという形で頻出パターンを書き出すことができたため、「きらきら星変奏曲」の複数の変奏から1つの辞書ファイルを生成し、その内容を分析した。

生成した辞書内のパターンを「音の変化の向きと大きさ」「変化の大きさの順」という2つの観点から分類すると、「きらきら星変奏曲」のいくつかの変奏から得られる4音パターンの特徴として、次のようなものが得られた。

- 音高変化の小さいものが中心となって構成されるパターンが多い
- パターン中に同じ変化の大きさをする部分があるパターンが多い

これらの理由として考えられるのは、辞書化されるパターンを構成する音高変化の中で、出現しやすい数値があり、その数値が変化量として小さいことを表している可能性がある。しかし、これらパターンの種類の特徴について考察するためには、複数の楽曲から得た辞書パターンとの比較が必要である。

また、辞書の内容の分析の他に、第8章で生成した辞書を第7章のプログラムに与え、「きらきら星変奏曲」からのパターン抽出と簡約を行った。すると、第8章で生成した辞書の問題点が明らかになった。特に重大なものとして、パターン抽出においてある程度の抽象化を行えるよう、「変化の幅」を設定したことによる問題が挙げられる。「変化の幅」という考え方は、楽曲中から頻出パターンを取り出す、他のパターンとの重複を調べ、調整を行うといった処理において、「2つのパターンが類似する」と判断する際に用いた。「変化の幅」を指定することによって、半音単位の完全な音高変化の一致以外にも、少々音高変化は異なってもその概型が同じであるものについても一部類似しているという判断ができるようにした。しかしこれが原因で、パターン化される順序によって類似するパターンの種類に差が生まれる、辞書の統合を行う際の類似条件との整合性がとれないといった問題が生まれた。よって今後は、このような考え方をしている部分のアルゴリズムの再検討を行う、また出力された辞書についてのチェックをするプログラムを実装し、類似するパターンがあればさらなる統合を行うなどの観点からこの問題点の解消に向けて取り組む必要がある。

全体として本研究では、「音高変化パターン内の重要な音の位置を指定することによる簡約」および「辞書内のパターンを抽出」、そして「楽曲に頻出するパターンの分析と辞書化」についての手法の検討、プログラムの実装を行った。結果、楽曲を基に辞書を生成し、その辞書のパターンを抽出することで簡約



化を行うという一連の処理を実装することができたが、現段階では適切な抽出や簡約に向けてさらなる改善の余地が存在する。加えて、目的である主題-変奏関係の分析に向けて本研究をさらに深めるためには、以下の様な取り組みが必要である。

- 他の変奏曲からの辞書生成と、生成した辞書の比較
- パターン名の再検討
- 重要な音の位置を自動推定
- 和音や複数旋律への対応

## 第 10 章 謝辞

本研究を進めるにあたって、多くの方々のご協力をいただきました。特に、研究の機会や、有益なご助言および幅広い知識を与えてくださった本研究科の平賀譲教授に深く感謝いたします。また、ゼミなどで研究に関する議論にお付き合いいただき、時に悩みを相談し合い、時に刺激を与えてくださった平賀研究室の皆様に、この場を借りて感謝の意を表します。

## 第 11 章 参考

### 参考文献・URL

- 1) Lerdahl.F, Jackendoff.R (1983) "A Generative Theory of Tonal Music" MIT Press
- 2) 浜中、平田、東条 (2007) 「音楽理論 GTTM に基づくグルーピング構造獲得システム」情報処理学会論文誌 Vol.48 No.1 284-299
- 3) Naurmor.E (1992) "The analysis and Cognition of Melodic Coplexity -The Implication-Realization Model" Univ. of Chicago Press
- 4) Walter B Hewlett,Eleanor Selfridge-Field (1998) "Melodic similarity : concepts, procedures, and applications" MIT Press
- 5) 長嶋、橋本、平賀、平田編 (1999) 『コンピュータと音楽の世界-基礎からフロンティアまで』 共立出版社 4章1節「音楽分析・認知研究への招待」(平賀)
- 6) 長嶋、橋本、平賀、平田編 (1999) 『コンピュータと音楽の世界-基礎からフロンティアまで』 共立出版社 4章6節「音楽の構造解析とその構造」(竹内)
- 7) 保科洋 (1998) 『生きた音楽表現へのアプローチ エネルギー思考に基づく演奏解釈法』4章3節 音楽之友社
- 8) 岸辺成雄編 (1983) 『音楽大事典』項目「変奏曲」 平凡社
- 9) 千蔵八郎 (1971) 『名曲事典 ピアノ・オルガン編』項目「きらきら星変奏曲」 音楽之友社
- 10) 山崎奈央子 (2011) 「音列パターン分類に基づく楽曲の主題・変奏関係の解析」筑波大学卒業論文
- 11) 「簡易音楽記述言語 (MML)」 <http://www.slis.tsukuba.ac.jp/hiraga/mml/> (2013/1/15 アクセス)
- 12) 「Wikipedia- 正規表現」 <http://ja.wikipedia.org/wiki/正規表現> (2013/1/5 アクセス)

### 楽譜出典元

「IMSLP」 <http://imslp.org/> (2013/1/6 アクセス)

## 付録

次ページ以降は、付録として以下を掲載する。

- 「きらきら星変奏曲」第1変奏右手に現れるパターンと出現数
- 「きらきら星変奏曲」第1変奏右手から生成したパターン辞書  
辞書 1) 4音パターン、変化の幅なし  
辞書 2) 4音以上のパターン、変化の幅なし

● 「きらきら星変奏曲」第1変奏右手に現れるパターンと出現数

| 繰り返し回数(該当パターン)    |                       |                       |                               |                                   |                                   |  |
|-------------------|-----------------------|-----------------------|-------------------------------|-----------------------------------|-----------------------------------|--|
| 4音                | 5音                    | 6音                    | 7音                            | 8音                                | 9音                                |  |
| 11(A0:-2, -1, 1)  | -> 8(A0, B0)          | -> 8(A0, B0, C0)      | -> 4(A0, B0, C0, B0)          | -> 4(A0, B0, C0, B0, C0)          | -> 2(A0, B0, C0, B0, C0, D0)      |  |
|                   |                       |                       |                               |                                   | +-----> 2(A0, B0, C0, B0, C0, G0) |  |
|                   | +-----> 3(A0, D0)     | -> 2(A0, D0, Z0)      | -> 2(A0, D0, Z0, a0)          | -> 2(A0, D0, Z0, a0, b0)          | -> 2(A0, B0, C0, G0, J0, X0)      |  |
| 12(B0:-1, 1, -1)  | -> 12(B0, C0)         | -> 4(B0, C0, B0)      | -> 4(B0, C0, B0, C0)          | -> 2(B0, C0, B0, C0, D0)          | -> 2(B0, C0, B0, C0, D0, E0)      |  |
|                   |                       |                       |                               | +-----> 2(B0, C0, B0, C0, G0)     | -> 2(B0, C0, B0, C0, G0, H0)      |  |
|                   | +-----> 2(B0, C0, D0) | -> 2(B0, C0, D0, E0)  | -> 2(B0, C0, D0, E0, F0)      | -> 2(B0, C0, D0, E0, F0, G0)      | -> 2(B0, C0, D0, E0, F0, A0)      |  |
|                   | +-----> 6(B0, C0, G0) | -> 2(B0, C0, G0, H0)  | -> 2(B0, C0, G0, H0, I0)      | -> 2(B0, C0, G0, H0, I0, J0)      | -> 2(B0, C0, G0, H0, I0, J0, K0)  |  |
|                   |                       |                       | +-----> 3(B0, C0, G0, J0)     | -> 2(B0, C0, G0, J0, X0)          | -> 2(B0, C0, G0, J0, X0, P0)      |  |
| 12(C0: 1, -1, 1)  | -> 4(C0, B0)          | -> 4(C0, B0, C0)      | -> 2(C0, B0, C0, D0)          | -> 2(C0, B0, C0, D0, E0)          | -> 2(C0, B0, C0, D0, E0, F0)      |  |
|                   |                       |                       | +-----> 2(C0, B0, C0, G0)     | -> 2(C0, B0, C0, G0, H0)          | -> 2(C0, B0, C0, G0, H0, I0)      |  |
|                   | +-----> 2(C0, D0)     | -> 2(C0, D0, E0)      | -> 2(C0, D0, E0, F0)          | -> 2(C0, D0, E0, F0, A0)          | -> 2(C0, D0, E0, F0, A0, B0)      |  |
|                   | +-----> 6(C0, G0)     | -> 2(C0, G0, H0)      | -> 2(C0, G0, H0, I0)          | -> 2(C0, G0, H0, I0, J0)          | -> 2(C0, G0, H0, I0, J0, K0)      |  |
|                   |                       | +-----> 3(C0, G0, J0) | -> 2(C0, G0, J0, X0)          | -> 2(C0, G0, J0, X0, P0)          | -> 2(C0, G0, J0, X0, P0, Y0)      |  |
| 5(D0:-1, 1, 9)    | -> 3(D0, E0)          | -> 2(D0, E0, F0)      | -> 2(D0, E0, F0, A0)          | -> 2(D0, E0, F0, A0, B0)          | -> 2(D0, E0, F0, A0, B0, C0)      |  |
|                   | +-----> 2(D0, Z0)     | -> 2(D0, Z0, a0)      | -> 2(D0, Z0, a0, b0)          |                                   |                                   |  |
| 4(E0: 1, 9, -2)   | -> 3(E0, F0)          | -> 3(E0, F0, A0)      | -> 3(E0, F0, A0, B0)          | -> 3(E0, F0, A0, B0, C0)          | -> 2(E0, F0, A0, B0, C0, B0)      |  |
| 9(F0: 9, -2, -1)  | -> 3(F0, A0)          | -> 3(F0, A0, B0)      | -> 3(F0, A0, B0, C0)          | -> 2(F0, A0, B0, C0, B0)          | -> 2(F0, A0, B0, C0, B0, C0)      |  |
|                   | +-----> 6(F0, N0)     | -> 6(F0, N0, N0)      | -> 6(F0, N0, N0, N0)          | -> 6(F0, N0, N0, N0, N0)          | -> 4(F0, N0, N0, N0, N0, S0)      |  |
|                   |                       |                       |                               | +-----> 2(F0, N0, N0, N0, N0, T0) |                                   |  |
| 6(G0:-1, 1, 1)    | -> 2(G0, H0)          | -> 2(G0, H0, I0)      | -> 2(G0, H0, I0, J0)          | -> 2(G0, H0, I0, J0, K0)          | -> 2(G0, H0, I0, J0, K0, L0)      |  |
|                   | +-----> 3(G0, J0)     | -> 2(G0, J0, X0)      | -> 2(G0, J0, X0, P0)          | -> 2(G0, J0, X0, P0, Y0)          | -> 2(G0, J0, X0, P0, Y0, A0)      |  |
| 2(H0: 1, 1, 1)    | -> 2(H0, I0)          | -> 2(H0, I0, J0)      | -> 2(H0, I0, J0, K0)          | -> 2(H0, I0, J0, K0, L0)          | -> 2(H0, I0, J0, K0, L0, M0)      |  |
| 2(I0: 1, 1, 3)    | -> 2(I0, J0)          | -> 2(I0, J0, K0)      | -> 2(I0, J0, K0, L0)          | -> 2(I0, J0, K0, L0, M0)          | -> 2(I0, J0, K0, L0, M0, N0)      |  |
| 5(J0: 1, 3, -1)   | -> 2(J0, K0)          | -> 2(J0, K0, L0)      | -> 2(J0, K0, L0, M0)          | -> 2(J0, K0, L0, M0, N0)          | -> 2(J0, K0, L0, M0, N0, O0)      |  |
|                   | +-----> 2(J0, X0)     | -> 2(J0, X0, P0)      | -> 2(J0, X0, P0, Y0)          | -> 2(J0, X0, P0, Y0, A0)          | -> 2(J0, X0, P0, Y0, A0, B0)      |  |
| 2(K0: 3, -1, 3)   | -> 2(K0, L0)          | -> 2(K0, L0, M0)      | -> 2(K0, L0, M0, N0)          | -> 2(K0, L0, M0, N0, O0)          | -> 2(K0, L0, M0, N0, O0, P0)      |  |
| 3(L0:-1, 3, -2)   | -> 3(L0, M0)          | -> 2(L0, M0, N0)      | -> 2(L0, M0, N0, O0)          | -> 2(L0, M0, N0, O0, P0)          | -> 2(L0, M0, N0, O0, P0, Q0)      |  |
| 3(M0: 3, -2, -1)  | -> 2(M0, N0)          | -> 2(M0, N0, O0)      | -> 2(M0, N0, O0, P0)          | -> 2(M0, N0, O0, P0, Q0)          | -> 2(M0, N0, O0, P0, Q0, R0)      |  |
| 26(N0:-2, -1, -2) | -> 2(N0, O0)          | -> 2(N0, O0, P0)      | -> 2(N0, O0, P0, Q0)          | -> 2(N0, O0, P0, Q0, R0)          | -> 2(N0, O0, P0, Q0, R0, F0)      |  |
|                   | +-----> 18(N0, N0)    | -> 12(N0, N0, N0)     | -> 6(N0, N0, N0, N0)          | -> 4(N0, N0, N0, N0, S0)          | -> 4(N0, N0, N0, N0, S0, R0)      |  |
|                   |                       |                       | +-----> 2(N0, N0, N0, N0, T0) | -> 2(N0, N0, N0, N0, T0, U0)      | -> 2(N0, N0, N0, N0, T0, U0, V0)  |  |
|                   |                       |                       | +-----> 4(N0, N0, N0, S0)     | -> 4(N0, N0, N0, S0, R0)          | -> 4(N0, N0, N0, S0, R0, F0)      |  |
|                   |                       | +-----> 4(N0, N0, S0) | -> 4(N0, N0, S0, R0)          | -> 4(N0, N0, S0, R0, F0)          | -> 4(N0, N0, S0, R0, F0, N0)      |  |
|                   | +-----> 4(N0, S0)     | -> 4(N0, S0, R0)      | -> 4(N0, S0, R0, F0)          | -> 4(N0, S0, R0, F0, N0)          | -> 4(N0, S0, R0, F0, N0, N0)      |  |
|                   | +-----> 2(N0, T0)     | -> 2(N0, T0, U0)      | -> 2(N0, T0, U0, V0)          | -> 2(N0, T0, U0, V0, W0)          |                                   |  |
| 2(O0:-1, -2, 0)   | -> 2(O0, P0)          | -> 2(O0, P0, Q0)      | -> 2(O0, P0, Q0, R0)          | -> 2(O0, P0, Q0, R0, F0)          | -> 2(O0, P0, Q0, R0, F0, N0)      |  |
| 8(P0:-2, 0, -2)   | -> 2(P0, Q0)          | -> 2(P0, Q0, R0)      | -> 2(P0, Q0, R0, F0)          | -> 2(P0, Q0, R0, F0, N0)          | -> 2(P0, Q0, R0, F0, N0, N0)      |  |
|                   | +-----> 6(P0, Y0)     | -> 6(P0, Y0, A0)      | -> 4(P0, Y0, A0, B0)          | -> 4(P0, Y0, A0, B0, C0)          | -> 3(P0, Y0, A0, B0, C0, G0)      |  |
|                   |                       |                       | +-----> 2(P0, Y0, A0, D0)     |                                   |                                   |  |
| 2(Q0: 0, -2, 9)   | -> 2(Q0, R0)          | -> 2(Q0, R0, F0)      | -> 2(Q0, R0, F0, N0)          | -> 2(Q0, R0, F0, N0, N0)          | -> 2(Q0, R0, F0, N0, N0, N0)      |  |
| 6(R0:-2, 9, -2)   | -> 6(R0, F0)          | -> 6(R0, F0, N0)      | -> 6(R0, F0, N0, N0)          | -> 6(R0, F0, N0, N0, N0)          | -> 6(R0, F0, N0, N0, N0, N0)      |  |
| 4(S0:-2, -2, 9)   | -> 4(S0, R0)          | -> 4(S0, R0, F0)      | -> 4(S0, R0, F0, N0)          | -> 4(S0, R0, F0, N0, N0)          | -> 4(S0, R0, F0, N0, N0, N0)      |  |
| 2(T0:-1, -2, 7)   | -> 2(T0, U0)          | -> 2(T0, U0, V0)      | -> 2(T0, U0, V0, W0)          |                                   |                                   |  |
| 3(U0:-2, 7, -2)   | -> 2(U0, V0)          | -> 2(U0, V0, W0)      |                               |                                   |                                   |  |
| 2(V0: 7, -2, -8)  | -> 2(V0, W0)          |                       |                               |                                   |                                   |  |
| 2(W0:-2, -8, 1)   |                       |                       |                               |                                   |                                   |  |
| 3(X0: 2, -2, 0)   | -> 3(X0, P0)          | -> 3(X0, P0, Y0)      | -> 3(X0, P0, Y0, A0)          | -> 3(X0, P0, Y0, A0, B0)          | -> 3(X0, P0, Y0, A0, B0, C0)      |  |
| 6(Y0: 0, -2, -1)  | -> 6(Y0, A0)          | -> 4(Y0, A0, B0)      | -> 4(Y0, A0, B0, C0)          | -> 3(Y0, A0, B0, C0, G0)          | -> 2(Y0, A0, B0, C0, G0, J0)      |  |
|                   |                       | +-----> 2(Y0, A0, D0) |                               |                                   |                                   |  |
| 2(Z0: 1, 9, -4)   | -> 2(Z0, a0)          | -> 2(Z0, a0, b0)      |                               |                                   |                                   |  |
| 3(a0: 9, -4, -3)  | -> 3(a0, b0)          | -> 2(a0, b0, c0)      | -> 2(a0, b0, c0, P0)          | -> 2(a0, b0, c0, P0, Y0)          | -> 2(a0, b0, c0, P0, Y0, A0)      |  |
| 3(b0:-4, -3, -2)  | -> 2(b0, c0)          | -> 2(b0, c0, P0)      | -> 2(b0, c0, P0, Y0)          | -> 2(b0, c0, P0, Y0, A0)          | -> 2(b0, c0, P0, Y0, A0, D0)      |  |
| 3(c0:-3, -2, 0)   | -> 3(c0, P0)          | -> 3(c0, P0, Y0)      | -> 3(c0, P0, Y0, A0)          | -> 2(c0, P0, Y0, A0, D0)          |                                   |  |

- 「きらきら星変奏曲」 第 1 変奏右手から生成したパターン辞書  
辞書 1) 4 音パターン、変化の幅なし

```

v1r_A0 -2,-1,+1, l
v1r_B0 -1,+1,-1, l
v1r_C0 +1,-1,+1, l
v1r_D0 -1,+1,+9, l
v1r_E0 +1,+9,-2, l
v1r_F0 +9,-2,-1, l
v1r_G0 -1,+1,+1, l
v1r_J0 +1,+3,-1, l
v1r_L0 -1,+3,-2, l
v1r_M0 +3,-2,-1, l
v1r_N0 -2,-1,-2, l
v1r_P0 -2,0,-2, l
v1r_R0 -2,+9,-2, l
v1r_S0 -2,-2,+9, l
v1r_U0 -2,+7,-2, l
v1r_X0 +2,-2,0, l
v1r_Y0 0,-2,-1, l
v1r_a0 +9,-4,-3, l
v1r_b0 -4,-3,-2, l
v1r_c0 -3,-2,0, l
v2l_d0 +12,-1,+1, l
v2l_e0 -1,+1,+2, l
v2l_f0 +1,+2,-2, l
v2l_g0 +2,-2,-1, l
v2l_i0 -1,+1,-8, l
v2l_j0 +1,-8,+8, l
v2l_k0 -8,+8,-1, l
v2l_l0 +8,-1,+1, l
v2l_m0 -1,+1,-12, l
v2l_n0 +1,-12,+12, l
v2l_o0 -12,+12,-1, l
v2l_p0 -1,+1,-15, l
v2l_q0 +1,-15,+12, l
v2l_r0 -15,+12,-1, l
v2l_s0 -1,+1,-10, l
v2l_t0 +1,-10,+12, l
v2l_u0 -10,+12,-1, l
v3r_v0 +4,+3,+5, l
v3r_w0 +3,+5,+4, l
v3r_x0 +5,+4,+3, l
v3r_y0 +3,+5,-5, l
v3r_z0 +5,-5,-2, l
v3r_A1 -5,-2,-1, l
v3r_D1 +1,-1,-2, l
v3r_E1 -1,-2,-2, l
v3r_G1 -2,+9,-1, l
v3r_H1 +9,-1,+1, l
v3r_I1 -1,+1,+3, l
v3r_K1 +3,-1,-2, l
v3r_L1 -1,-2,0, l
v3r_M1 -2,0,-3, l
v3r_N1 0,-3,+1, l
v3r_O1 -3,+1,0, l
v3r_P1 +1,0,+5, l
v3r_Q1 0,+5,+4, l
v3r_R1 +5,+4,0, l
v3r_S1 +4,0,-11, l
v3r_T1 0,-11,0, l
v3r_U1 -11,0,0, l

```

v3r\_V1 0,0,+6, l  
v3r\_W1 0,-10,+7, l  
v3r\_X1 -10,+7,0, l  
v3r\_Y1 +7,0,-2, l  
v3r\_Z1 0,-2,-8, l  
v3r\_a1 -2,-8,+1, l  
v3r\_f1 +1,+9,-4, l  
v3r\_g1 +9,-4,-5, l  
v3r\_h1 -4,-5,0, l  
v3r\_i1 -5,0,-3, l  
v3r\_j1 -3,+1,+9, l  
v3r\_o1 +3,-3,-3, l  
v3r\_r1 +1,+9,-7, l  
v3r\_s1 +9,-7,-2, l  
v3r\_t1 -7,-2,0, l  
v4l\_w1 +5,+4,-4, l  
v4l\_x1 +4,-4,-12, l  
v4l\_y1 -4,-12,+4, l  
v4l\_z1 -12,+4,+3, l  
v4l\_C2 +4,-4,-15, l  
v4l\_D2 -4,-15,+11, l  
v4l\_E2 -15,+11,+1, l  
v4l\_F2 +11,+1,-10, l  
v4l\_G2 +1,-10,+11, l  
v4l\_H2 -10,+11,+1, l  
v4l\_I2 +11,+1,-15, l  
v4l\_J2 +1,-15,+11, l  
v4l\_K2 +11,+1,-7, l  
v4l\_L2 +1,-7,+4, l  
v4l\_M2 -7,+4,+3, l  
v4l\_P2 +4,+3,-1, l  
v4l\_Q2 +3,-1,+1, l  
v4l\_S2 +1,-12,+2, l  
v4l\_T2 -12,+2,+2, l  
v4l\_U2 +2,+2,+1, l  
v4l\_V2 +2,+1,-1, l  
v4l\_X2 +1,-12,+4, l  
v4l\_Y2 -12,+4,+1, l  
v4l\_Z2 +4,+1,+2, l  
v6r\_a2 0,+7,0, l  
v6r\_b2 +7,0,+2, l  
v6r\_c2 0,+2,0, l  
v6r\_d2 +2,0,-2, l  
v6r\_e2 0,-2,0, l  
v6r\_g2 0,-2,-3, l  
v6r\_h2 -2,-3,+1, l  
v6r\_q2 +1,+2,+2, l  
v6r\_r2 +2,+2,-4, l  
v6r\_s2 +2,-4,+5, l  
v6r\_t2 -4,+5,-1, l  
v6r\_u2 +5,-1,+1, l  
v6r\_v2 +2,+2,-2, l  
v6r\_w2 +2,-4,+2, l  
v6r\_x2 -4,+2,-3, l  
v6l\_I3 -4,+2,-1, l  
v6l\_J3 +1,-2,-1, l  
v6l\_L3 -1,-2,+2, l  
v6l\_M3 -2,+2,-4, l  
v6l\_O3 +1,-12,+5, l

v6l\_P3 -12,+5,-5, l  
v6l\_Q3 +5,-5,-3, l  
v6l\_R3 -5,-3,+3, l  
v6l\_S3 -3,+3,-7, l  
v6l\_Y3 0,-1,0, l  
v7r\_c3 -1,-2,-1, l  
v7r\_e3 -1,+1,+5, l  
v7r\_f3 +1,+5,-2, l  
v7r\_g3 +5,-2,-1, l  
v7r\_k3 -1,+1,0, l  
v7r\_l3 +1,0,+9, l  
v7r\_m3 0,+9,-9, l  
v7r\_n3 +9,-9,-2, l  
v7r\_o3 -9,-2,+11, l  
v7r\_p3 -2,+11,-11, l  
v7r\_q3 +11,-11,0, l  
v7r\_r3 -11,0,+9, l  
v7r\_s3 +9,-9,+10, l  
v7r\_w3 +5,+4,-2, l  
v7r\_x3 -2,-1,+12, l  
v7r\_y3 -1,+12,+1, l  
v7r\_z3 +12,+1,+1, l  
v7r\_A4 +1,+1,-3, l  
v7r\_B4 +1,-3,-2, l  
v11r\_C4 -5,+9,-4, l  
v11r\_D4 +9,-4,+7, l  
v11r\_E4 -4,+7,0, l  
v11r\_F4 +7,0,0, l  
v11r\_G4 0,0,+2, l  
v11r\_I4 +2,0,0, l  
v11r\_J4 0,0,-2, l  
v11r\_L4 -2,0,0, l  
v11r\_M4 0,0,0, l  
v11r\_N4 -2,0,+11, l  
v11r\_O4 0,+11,-2, l  
v11r\_P4 +11,-2,-3, l  
v11r\_Q4 -2,-3,-6, l  
v11r\_R4 -3,-6,-1, l  
v11r\_S4 -6,-1,+5, l  
v11r\_T4 -1,+5,-2, l  
v11r\_U4 +5,-2,-3, l  
v11r\_V4 -3,-5,+8, l  
v11r\_W4 -5,+8,-2, l  
v11r\_X4 +8,-2,-3, l  
v11r\_Y4 -2,-3,-3, l  
v11r\_Z4 -3,-3,+3, l  
v11r\_a4 -3,+3,-2, l  
v11r\_f4 +2,+3,-1, l  
v11r\_n4 +2,-2,+2, l  
v11l\_p4 +9,-4,-7, l  
v11l\_q4 -4,-7,-5, l  
v11l\_r4 -7,-5,+9, l  
v11l\_t4 -4,-5,-5, l  
v11l\_u4 -5,-5,+9, l  
v11l\_v4 +9,-4,+17, l  
v11l\_w4 -4,+17,+2, l  
v11l\_x4 +17,+2,0, l  
v11l\_y4 +2,0,-7, l  
v11l\_z4 0,-7,+2, l



v11l\_A5 -7,+2,+2, l  
v11l\_C5 +2,-2,-2, l  
v11l\_D5 -2,-2,-5, l  
v11l\_E5 -2,-5,-7, l  
v11l\_I5 +9,0,0, l  
v11l\_J5 0,0,-9, l  
v11l\_K5 0,-9,+9, l  
v11l\_L5 -9,+9,0, l  
v11l\_U5 +12,0,0, l  
v11l\_V5 0,0,-12, l  
v12r\_Y5 0,0,+5, l  
v12r\_Z5 0,+5,-7, l  
v12r\_a5 +5,-7,+2, l  
v12r\_b5 -7,+2,0, l  
v12r\_c5 0,0,+7, l  
v12r\_h5 -2,0,+2, l  
v12r\_i5 0,-3,-1, l  
v12r\_j5 -3,-1,+2, l  
v12r\_k5 -1,+2,0, l  
v12r\_l5 0,-2,-5, l  
v12r\_m5 -2,-5,+2, l  
v12r\_n5 -5,+2,0, l  
v12r\_p5 0,0,+3, l  
v12r\_q5 0,+3,-6, l  
v12r\_r5 +3,-6,+1, l  
v12r\_x5 +1,0,-2, l  
v12r\_y5 0,-1,-1, l  
v12r\_z5 -1,-1,+1, l  
v12r\_A6 -1,+1,+7, l  
v12r\_E6 +1,+9,-9, l  
v12r\_F6 +9,-9,+9, l  
v12r\_G6 -9,+9,-9, l  
v12r\_H6 +9,-9,0, l  
v12r\_I6 -9,0,-2, l  
v12r\_U6 +1,-5,+2, l  
v12r\_V6 -5,+2,-4, l  
v12r\_W6 +2,-3,+1, l  
v12r\_X6 -3,+1,-3, l  
v12r\_a6 +1,+4,-1, l  
v12r\_b6 +4,-1,+1, l  
v12l\_q6 -15,+12,-2, l  
v12l\_u6 +1,-7,+2, l  
v12l\_x6 +2,+1,-12, l  
v12l\_B7 -7,+5,-1, l  
v12l\_D7 -1,+1,-5, l  
v12l\_F7 -5,+10,-1, l  
v12l\_G7 +10,-1,+1, l  
v12l\_H7 +1,-10,+9, l  
v12l\_M7 -7,+7,-7, l  
v12l\_Q7 -5,+5,-5, l

辞書 2) 4 音以上のパターン、変化の幅なし

v1r\_A0B0C0B0C0 -2,-1,+1,-1,+1,-1,+1, l  
v1r\_F0N0N0N0N0 +9,-2,-2,-1,-2,-2,-2, l  
v1r\_A0B0C0 -2,-1,+1,-1,+1, l  
v1r\_B0C0G0 -1,+1,-1,+1,+1, l  
v1r\_P0Y0A0 -2,0,-2,-1,+1, l  
v1r\_A0 -2,-1,+1, l  
v1r\_D0 -1,+1,+9, l  
v1r\_F0 +9,-2,-1, l  
v1r\_J0 +1,+3,-1, l  
v1r\_N0 -2,-1,-2, l  
v1r\_P0 -2,0,-2, l  
v2l\_d0e0f0g0A0 +12,-1,+1,+2,-2,-1,+1, l  
v2l\_l0e0f0g0h0 +8,-1,+1,+2,-2,-1,+1, l  
v2l\_d0s0t0u0d0 +12,-1,+1,-10,+12,-1,+1, l  
v2l\_d0p0q0r0d0 +12,-1,+1,-15,+12,-1,+1, l  
v2l\_m0n0o0d0 -1,+1,-12,+12,-1,+1, l  
v2l\_p0q0r0d0 -1,+1,-15,+12,-1,+1, l  
v3r\_U1V1Q1R1S1 -11,0,0,+6,+3,0,-10, l  
v3r\_W1X1Y1Z1a1 0,-10,+7,0,-2,-8,+1, l  
v3r\_f1g1h1i1N1 +1,+9,-4,-5,0,-3,+1, l  
v3r\_r1s1t1M1N1 +1,+9,-7,-2,0,-3,+1, l  
v3r\_Q1R1S1 0,+5,+4,0,-11, l  
v3r\_D0f1g1 -1,+1,+9,-4,-5, l  
v3r\_N1e1f1 0,-2,+1,+8,-5, l  
v3r\_M1N1 -2,0,-3,+1, l  
v3r\_D0f1 -1,+1,+9,-4, l  
v3r\_f1g1 +1,+9,-4,-5, l  
v3r\_N1j1 0,-3,+1,+9, l  
v3r\_v0 +4,+3,+5, l  
v3r\_D1 +1,-1,-2, l  
v3r\_S0 -2,-2,+9, l  
v3r\_o1 +3,-3,-3, l  
v4l\_w0w1x1y1z1 +3,+5,+4,-4,-12,+4,+3, l  
v4l\_y1z1u1v1w1 -4,-12,+4,+3,+5,+4,-4, l  
v4l\_H2K2L2M2u1 -10,+11,+1,-7,+4,+3,+5, l  
v4l\_Y2Z2V2Q2R2 -12,+4,+1,+2,-1,+1,-12, l  
v4l\_E2F2G2H2 -15,+11,+1,-10,+11,+1, l  
v4l\_v0w0w1 +4,+3,+5,+4,-4, l  
v4l\_x1y1z1 +4,-4,-12,+4,+3, l  
v4l\_F2G2H2 +11,+1,-10,+11,+1, l  
v4l\_I2J2E2 +11,+1,-15,+11,+1, l  
v4l\_v0w0 +4,+3,+5,+4, l  
v4l\_x0u1 +5,+4,+3,+5, l  
v4l\_u1y0 +4,+3,+5,-5, l  
v4l\_M2u1 -7,+4,+3,+5, l  
v4l\_Q2m0 +3,-1,+1,-12, l  
v4l\_x0 +5,+4,+3, l  
v4l\_E2 -15,+11,+1, l  
v4l\_V2 +2,+1,-1, l  
v6r\_f2e2f2g2h2 -2,0,-1,0,-2,-3,+1, l  
v6r\_C0n2o2n2o2 +1,-1,+1,-1,+1,-1,+1, l  
v6r\_q2v2o2p2q2 +1,+2,+2,-2,+2,+2,+1, l  
v6r\_e2P0e2f2 0,-2,0,-2,0,-1, l  
v6r\_B0C0n2o2 -1,+1,-1,+1,-1,+1, l  
v6r\_p2q2v2o2 -1,+1,+2,+2,-2,+2, l  
v6r\_w2x2w2x2 +2,-4,+2,-3,+1,-3, l  
v6r\_C0n2o2 +1,-1,+1,-1,+1, l  
v6r\_o2e0q2 +1,-1,+1,+2,+2, l  
v6r\_e2P0 0,-2,0,-2, l

v6r\_q2r2 +1,+2,+2,-4, l  
v6r\_w2x2 +2,-4,+2,-3, l  
v6l\_y2z2e0q2r2 -1,+1,-1,+1,+2,+2,-4, l  
v6l\_L3M3D3E3F3 -1,-2,+2,-4,+5,-1,+1, l  
v6l\_O3P3Q3R3S3 +1,-12,+5,-5,-3,+3,-7, l  
v6l\_y2z2e0q2 -1,+1,-1,+1,+2,+2, l  
v6l\_s2t2u2y2 +2,-4,+5,-1,+1,-1, l  
v6l\_y2J3N0L3 -1,+1,-2,-1,-2,+2, l  
v6l\_s2t2u2 +2,-4,+5,-1,+1, l  
v6l\_K3L3y2 -1,-2,-1,+1,-2, l  
v6l\_C0y2 +1,-1,+1,-1, l  
v6l\_B3B3 +1,+2,+2,+2, l  
v6l\_M3 -2,+2,-4, l  
v6l\_Y3 0,-1,0, l  
v7r\_U2Z3Z3Z3Z3 +2,+2,+1,+2,+2,+2,+1, l  
v7r\_o3p3q3r3m3 -9,-2,+11,-11,0,+9,-9, l  
v7r\_p3q3r3m3s3 -2,+10,-10,0,+9,-9,+10, l  
v7r\_c3c3c3c3c3 -2,-2,-1,-2,-2,-2,-1, l  
v7r\_y3z3Z3A4B4 -1,+12,+1,+1,+1,-3,-2, l  
v7r\_U2Z3Z3Z3 +2,+2,+1,+2,+2,+2, l  
v7r\_c3c3c3c3 -2,-2,-1,-2,-2,-2, l  
v7r\_U2Z3Z3 +2,+2,+1,+2,+2, l  
v7r\_c3c3c3 -2,-2,-1,-2,-2, l  
v7r\_U2Z3 +2,+2,+1,+2, l  
v7r\_g3c3 +5,-2,-1,-2, l  
v7r\_c3c3 -2,-2,-1,-2, l  
v7r\_U2 +2,+2,+1, l  
v11r\_W4X4Y4Z4a4 -5,+8,-2,-3,-3,+3,-2, l  
v11r\_J4e2 0,0,-2,0, l  
v11r\_N0Y4 -2,-1,-2,-2, l  
v11r\_Y4k4 -1,-2,-2,-1, l  
v11r\_P4 +11,-2,-3, l  
v11r\_Q4 -2,-3,-6, l  
v11r\_R4 -3,-6,-1, l  
v11r\_U4 +5,-2,-3, l  
v11r\_Y4 -2,-3,-3, l  
v11r\_a4 -3,+3,-2, l  
v11r\_f4 +2,+3,-1, l  
v11r\_e0 -1,+1,+2, l  
v11r\_n4 +2,-2,+2, l  
v11l\_A5v2C5D5E5 -7,+2,+2,-2,-2,-5,-7, l  
v11l\_I5J5K5L5I5 +9,0,0,-9,+9,0,0, l  
v11l\_I5J5 +9,0,0,-9, l  
v11l\_U5V5 +12,0,0,-12, l  
v11l\_C4 -5,+9,-4, l  
v11l\_D5 -2,-2,-5, l  
v11l\_I5 +9,0,0, l  
v11l\_J5 0,0,-9, l  
v11l\_G4 0,0,+2, l  
v12r\_X5M4p5q5r5 +2,0,0,+3,-6,+1, l  
v12r\_w5v5w5k5x5 +1,-1,+1,-1,+1,0,-2, l  
v12r\_z5v5w5v5w5 -1,-1,+1,-1,+1,-1,+1, l  
v12r\_v5w5v5w5A6 -1,+1,-1,+1,-1,+1,+7, l  
v12r\_E6F6G6H6I6 +1,+9,-9,+9,-9,0,-2, l  
v12r\_I1a6b6 -1,+1,+3,-1,+1, l  
v12r\_c2I4 0,+2,0,0, l  
v12r\_X5f5 +2,0,0,-3, l  
v12r\_i5j5 0,-3,-1,+2, l  
v12r\_U6V6 +1,-5,+2,-4, l

v12r\_W6X6 +2,-3,+1,-3, l  
v12r\_a6b6 +1,+4,-1,+1, l  
v12r\_I4 +2,0,0, l  
v12r\_b5 -7,+2,0, l  
v12r\_J4 0,0,-2, l  
v12r\_k5 -1,+2,0, l  
v12r\_G6 -9,+9,-9, l  
v12r\_e3 -1,+1,+5, l  
v12l\_A0d6e6f6g6 -2,-1,+1,+2,-2,-1,+1, l  
v12l\_g6m0n0o0c6 -2,-1,+1,-12,+12,-1,+1, l  
v12l\_g6p0q0r6c6 -2,-1,+1,-15,+12,-2,+2, l  
v12l\_w6w6w6w6x6 +2,+1,+2,+2,+2,+1,-12, l  
v12l\_k6B0C0y6z6 +9,-1,+1,-1,+1,-1,+1, l  
v12l\_y6z6y6z6h6 -1,+1,-1,+1,-1,+1,-9, l  
v12l\_z6y6z6h6i6 +1,-1,+1,-1,+1,-9,+7, l  
v12l\_u2y6z6y6z6 +5,-1,+1,-1,+1,-1,+1, l  
v12l\_y6z6y6z6D7 -1,+1,-1,+1,-1,+1,-5, l  
v12l\_G7y6z6y6z6 +10,-1,+1,-1,+1,-1,+1, l  
v12l\_z6y6z6r6H7 +1,-1,+1,-1,+1,-10,+9, l  
v12l\_e0f0g0A0 -1,+1,+2,-2,-1,+1, l  
v12l\_i0j0k0l0 -1,+1,-8,+8,-1,+1, l  
v12l\_p0q0r6c6 -1,+1,-15,+12,-2,+2, l  
v12l\_i0j0 -1,+1,-8,+8, l  
v12l\_k0l0 -8,+8,-1,+1, l  
v12l\_G6F6 -9,+9,-9,+9, l  
v12l\_i0 -1,+1,-8, l  
v12l\_s0 -1,+1,-10, l  
v12l\_M7 -7,+7,-7, l  
v12l\_Q7 -5,+5,-5, l