JAIST Repository
https://dspace.jaist.ac.jp/

Title	GTTMにおけるcadential retentionに関する理論の再構成
Author(s)	Kodama, Takafumi
Citation	
Issue Date	2015-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/12630
Rights	
Description	Supervisor: 東条敏, 情報科学研究科, 修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

GTTMにおけるcadential retentionに関する理論の再構成

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

小玉 昂史

2015年3月

修 士 論 文

GTTMにおけるcadential retentionに関する理論の再構成

指導教員　東条 敏 教授

審査委員主査　東条 敏 教授
審査委員　Nguyen Minh Le 准教授
審査委員　飯田 弘之 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1310024 小玉 昂史

提出年月: 2015 年 2 月

Copyright © 2015 by Kodama Takafumi

概 要

古くから音楽には文法が存在すると考えられており，音楽に文法を発見することができれば，文法を扱うための様々な手法を音楽に適用することができると考えられる．音楽理論 A Generative Theory of Tonal Music （以下，GTTM）は解析方法の手順をルールとして記述しており，解析の結果として木構造を出力するため，文法発見や計算機への実装に適していると考えられる．しかし，GTTM のルールは手作業での解析を前提としており，その記述は計算機への実装という観点からみると非常に曖昧なものである．これに対し，GTTM を計算機上に実装するために，GTTM に詳細な定義を与え，理論を再構成した exGTTM があるが，この exGTTM もまだ完全ではなく，特に，和声情報（和音の進行による色彩変化）を付加する処理である Cadential Retention の詳細な実装はまだなされていない．この実装を難しくしている要因として，和声学や音楽の曖昧性，さらには GTTM における Cadential Retention 理論の定義が曖昧であることが挙げられる．

Cadential Retention の実装には和声情報の解析が必要不可欠である．この和声を解析する手法としては，GTTM の著者が GTTM を補完するために提案した Tonal Pitch Space (以下，TPS) を利用した手法が，坂本らにより提案されている．TPS は，GTTM とは独立な理論であり，ある和音からある和音への音程間，和音間，調間に定量的な距離を定め，この距離が近ければ近いほど，その部分は安定して心地よく響き，距離が長いほど不自然で違和感がある進行であるとするものである．この方法を用いれば，和声情報の解析が，計算機を用いて定量的に行うことができると考えられるが，この理論は Cadential Retention に適用するためのものにはなっておらず，曖昧な部分もあるため，これを計算機上に実装し，Cadential Retention に応用することは難しくなっている．

本研究では，GTTM に Cadential Retention を実装するための前段階として，Cadential Retention の実装が困難になっている原因や，実装の際に用いることで効果的になる手法を発見するために，Cadential Retention を行う部分である終止部についての探索を行い，その考察を行う．また，Cadential Retention の実装の際に必要となる和声解析を実装するために，坂本らのシステムの方針に従い，Cadential Retention を行うための TPS 理論を再構成することを目的とする．

これらの目的のために，まず構成論的音楽理論について調査を行い，GTTM のデータベースを用いた実験により，GTTM の Cadential Retention 理論の曖昧性およびそれを解消する手法の発見を行う．

Cadential Retention 理論の曖昧性は以下のようなものが考えられる．

曖昧性 1 GTTM における Time-Span 木の役割が不明瞭である．

曖昧性 2 TSRPR7 で終止部に対して操作を行うとあるが，これを決める基準は具体的には示されていない．

曖昧性 3 *egg* を付けるイベントと付けないイベントの具体的な判断基準がない.

曖昧性 4 女性終止の際の木の変更方法や手順が明確に示されていない.

そこで，これらの曖昧性のうち，曖昧性 1 について，GTTM の構成要素をもう一度見直し，リズム構造木 (Rhythmic Structure Tree) の提案を行い，タイムスパン木の性質をわかりやすくするために，和声情報を含むタイムスパン木 (Time-Span Tree with Cadence) という名称への変更を提案する．この和声情報を含むタイムスパン木は，あくまでもプロロンゲーション簡約を行うための過程で生まれる木として捉えることが妥当であると考えられる．また，これらの提案により，各木構造の存在意義が明確になると考えられる．

Cadential Retention は和声の中でも，特に終止部（楽曲が終わるような感じや，区切れるような感じがする部分）に対して行われる．そこで，曖昧性 1 の解消のための新たな木の提案の他に，曖昧性 2 の問題点をさらにはっきりとさせるための終止部の探索実験を行う．この実験により，以下のようなことが明らかになる．

1. グルーピングの大きさによる終止部の判定は妥当ではなく，用いるべきではない.
2. 終止部をある程度特定するためには和声解析が不可欠になる.
3. GTTM 内のルールである TSRPR7(iii) の手法が，和声解析を行った部分に対する終止部の同定に大いに役立つ.
4. 女性終止を発見するためには，和声解析とグルーピング構造解析の結果を用いてその部分を特定する必要がある.
5. Cadential Retention における *egg* が付加されない部分である終止部の持続範囲は，バス声部の簡約および保続音を考慮することにより決定可能である.
6. 終止部のヘッドを選択する際に，グルーピング構造解析の結果を用いることが有効である.
7. 終止部のヘッドを選択する際に，上記の方法を用いる際，先行音に対する処理が必要となる.
8. 和声情報を含むタイムスパン木を構成する際には，木構造を操作するのではなく，ヘッド選択のための手法を確立して一から木構造を構成する必要がある.

またこれに加え，Cadential Retention で木構造に付加される *egg* を TimeSpanXML に付加し，*egg* マーカー付き TimeSpanXML を作成を行う.

さらに，上記の結果を受け，Cadential Retention のための和声解析手法の理論を構成する．GTTM における Cadential Retention は半終止，完全終止，偽終止に対して行うため，重要なのはドミナントを発見するような和声解析を行うことである．また，遠隔の調関係を知る際に和声解析は特に重要になり，そのときに重要視されるのもこのドミナントの発

見である．これらのことから，ドミナントの発見に特化した，Cadential Retention のための和声解析の手法を TPS をもとに提案する．

この理論はドミナントを発見するための手法であり，一般的な和声理論とは多少違う部分があるが，基本的には和声理論に準ずる形で TPS 理論を数学的に再構成したものである．この理論では，和声は音階をもとに発生するという考えのもと和声解析を行い，用いる和音は長音階と和声的短音階に依存する和音に限定している．さらに，この理論は全体としては複雑になっているが，プログラミング言語を用いてそれぞれの関数を計算機上に実装する際には，各関数を定義通りに設定していくことで，非常に作成がしやすいものとなっている．しかし，それぞれの式において複雑になりすぎている部分があるので，今後さらに改良の余地が存在する可能性がある．

また，本研究ではこの和声解析手法の計算機上への実装も行い，理論が正しく構築されていることを確認する．この実装システムを用いて，いくつかの曲を解析してみたところ，以下のような Cadential Retention のために有効であると思われる性質が分かる．

1. 音楽的直観で得られる楽譜上には示されない転調を解析によって発見可能である．
2. バークリーメソッド式の表記ではドミナントとトニックの関係に見える和声であっても，解析により終止部ではないという判断をすることができ，また，これは音楽的直観に一致している．
3. 解析結果で一見終止部に見えるような部分であっても，グルーピング構造解析の結果を用いることで終止部ではないという判断が可能である．

今後の課題として，これらの明らかになった手法や実装されたシステムを用いて，exGTTM の理論内で述べられているような手法を取り入れ，ヘッドとして選択されるための指標となる重みを設けるための検証を行う必要があることが挙げられる．さらなる課題としては，これまでに示した理論による新たな GTTM 理論の再構成を行うこと，大きなレベルでの *egg* 付加の検証を行うこと，多くの楽譜を用いた解析による検証を行うことが挙げられる．

目 次

第 1 章　はじめに **1**
1.1　研究の背景 1
1.2　研究の目的 1
1.3　本稿の構成 2

第 2 章　構成論的音楽理論 **3**
2.1　本研究で使用する音楽理論についての簡単な解説 3
2.1.1　和音記号とバークリーメソッド 3
2.1.2　機能和声と終止 5
2.1.3　近親調と遠隔調 6
2.2　A Generative Theory of Tonal Music (GTTM) 6
2.2.1　Grouping Structure 8
2.2.2　Metrical Structure 16
2.2.3　Time-Span Reduction 18
2.3　exGTTM 31
2.3.1　公開データ 31
2.4　Tonal Pitch Space（TPS） 32
2.4.1　pitch class（ピッチクラス） 32
2.4.2　近親調における和音間距離 33
2.4.3　遠隔調における和音間距離の導出 35
2.5　坂本らによる Tonal Pitch Space を用いた楽曲の和声解析 38

第 3 章　GTTM のデータベースを利用した終止部の探索 **40**
3.1　Cadential Retention 理論の曖昧性 40
3.2　Time-Span tree with cadence と Rhythmic Structure tree 41
3.3　終止部の探索 41
3.3.1　探索方法 43
3.3.2　探索結果の分析 43
3.4　追加実験 47
3.4.1　*egg* マーカー付き TimeSpanXML 48
3.4.2　探索方法 48
3.4.3　探索結果 53

3.5 まとめ 53
3.5.1 和声情報の必要性 55

第 4 章 Cadential Retention のための TPS 理論の再構成と和声解析 57
4.1 基本方針 57
4.2 和声理論との対応 58
4.3 TPS 理論の再構成 61
4.3.1 和音間距離 Δ 62
4.3.2 調間距離 reg 65
4.3.3 和音の五度圏距離 cho 65
4.3.4 ベーシックスペース距離 (basic space distance) bs 66
4.4 実装システム 68
4.5 実験結果と Cadential Retention への利用 70

第 5 章 おわりに 76

図 目 次

2.1 音階上の和音 4
2.2 4 つの終止構造 6
2.3 近親調 7
2.4 GTTM の構成要素 8
2.5 グルーピング構造解析の例 9
2.6 Grouping Structure におけるグループの包含関係 9
2.7 禁止されたグルーピング 1 9
2.8 禁止されたグルーピング 2 9
2.9 GPR1 で禁止されるグルーピング 10
2.10 GPR2a の例 11
2.11 GPR2b の例 12
2.12 GPR3a の説明 13
2.13 GPR4 の例 14
2.14 GPR5 の例 14
2.15 GPR6 の例 15
2.16 GPR6 が無視されるような例 15
2.17 オーバーラップグルーピングの例 15
2.18 拍節構造解析の解析結果例 16
2.19 タイムスパン木の例 19
2.20 タイムスパン木におけるヘッドの役割 19
2.21 タイムスパン簡約の例 20
2.22 タイムスパンの説明 22
2.23 オーバーラップグルーピング時のタイムスパン木とその簡約 22
2.24 セグメンテーションルールの説明 23
2.25 Fusion ルールの例 24
2.26 Transformation ルールの例 25
2.27 Transformation ルールの例 26
2.28 Metrical Stability ルールの説明 29
2.29 終止部と判断されない例 30
2.30 グルーピングによる終止部の判断 30
2.31 調の五度圏 34

2.32 和音の五度圏 35
2.33 ハ長調を基調とするベーシックスペースの例: (a)I/**I**; (b)iv/**vii** 36
2.34 調性空間 37
2.35 和声解釈のグラフ化 39

3.1 新たな木の提案 42
3.2 探索手順 1 の説明 44
3.3 探索手順 2 により除外される木の例 44
3.4 探索手順 3 の説明 45
3.5 探索がうまくいったと思われる例 45
3.6 終止部と同様の形を持つ終止部の可能性のある部分 46
3.7 女性終止の例 46
3.8 探索の結果生成された枝が明らかに間違えている例 47
3.9 ドミナントの範囲を極端に広げた例 47
3.10 *egg* マーカー付き TimeSpanXML の *egg* の位置の検証 49
3.11 ドミナントの範囲を極端に広げた例 50
3.12 保続音を考慮した終止部の範囲 51
3.13 グルーピング構造解析による探索範囲の限定 52
3.14 先行音に関する処理の失敗例 53
3.15 新しい探索方法のプログラムの結果 54
3.16 木構造の生成の手順 54
3.17 大きなレベルでの和声解析の必要性 56

4.1 短調の各音階 59
4.2 音階上に発生する和音とその和声的解釈 60
4.3 Δ 関数の第二式を用いる例 64
4.4 実装システムの構成 69
4.5 入力データの例 69
4.6 作成されたノードの仮想モデル 70
4.7 実際の出力例 71
4.8 実際の実行の様子 72
4.9 解析結果の例 74
4.10 解析結果の部分的なグルーピング 75

表 目 次

2.1　調の表現（ドイツ式）と日本語の対応 4
2.2　ルート音と音名の対応 5
2.3　バークリーメソッド式の和音の構成音 5
2.4　ピッチクラスの値と音名 32

第1章　はじめに

1.1　研究の背景

Bernsteinによれば，古くから音楽には文法が存在すると考えられており [1]，音楽に文法を発見することができれば，文法を扱うための様々な手法を音楽に適用することができると考えられる．音楽理論 A Generative Theory of Tonal Music （[2]，以下 GTTM）は解析方法の手順をルールとして記述しており，解析の結果として木構造を出力するため，文法発見や計算機への実装に適していると考えられる．しかし，GTTM のルールは手作業での解析を前提としており，その記述は計算機への実装という観点からみると非常に曖昧なものである．これに対し，GTTM を計算機に実装するために，GTTM に詳細な定義を与え，理論を再構成した exGTTM [3] があるが，この exGTTM もまだ完全ではなく，特に，和声情報（和音の進行による色彩変化）を付加する処理である Cadential Retention の詳細な実装はまだなされていない．この実装を難しくしている要因として，和声学や音楽の曖昧性，さらには GTTM における Cadential Retention 理論の定義が曖昧であることが挙げられる．

Cadential Retention の実装には和声情報の解析が必要不可欠である．この和声を解析する手法としては，GTTM の著者が GTTM を補完するために提案した Tonal Pitch Space ([4]，以下 TPS) を利用した手法が，坂本らにより提案されている [5]．TPS は，GTTM とは独立な理論であり，ある和音からある和音への音程間，和音間，調間に定量的な距離を定め，この距離が近ければ近いほど，その部分は安定して心地よく響き，距離が長いほど不自然で違和感がある進行であるとするものである．この方法を用いれば，和声情報の解析が，計算機を用いて定量的に行うことができると考えられるが，この理論は Cadential Retention に適用するためのものにはなっておらず，曖昧な部分もあるため，これを計算機上に実装し，Cadential Retention に応用することは難しくなっている．

1.2　研究の目的

そこで本研究では，GTTM に Cadential Retention を実装するための前段階として，Cadential Retention の実装が困難になっている原因や，実装の際に用いることで効果的になる手法を発見するために，Cadential Retention を行う部分である終止部（楽曲が終わるような感じや区切れるような感じがする部分）についての探索を行い，その考察を行う．ま

た，Cadential Retention の実装の際に必要となる和声解析を実装するために，坂本らのシステムの方針に従い，TPS 理論を再構成することを目的とする．

TPS 理論を再構成した理論の定義の際には，終止と呼ばれる Cadential Retention で必要となる和声情報の発見を第一に考え，また，方針として，和声を基準に旋律が発生するのではなく，旋律に和声が付加されていると考えることにし，ある特定の音階をもとにした和声解析となるようにした．また，この定義は，Cadential Retention に TPS を適用するためのものであり，実際の音楽理論と異なる部分も存在するが，実際の音楽理論に準ずる形で定義を行い，音楽理論との乖離ができる限り起こらないようにする．

1.3　本稿の構成

まず，2 章で本研究で必要になる音楽知識の簡単な説明と，本研究に関連する研究について述べる．本研究は構成論的音楽理論をもとにしているため，それらの研究の中で，本研究に特に関連が深いものについて述べる．

次に，3 章で Cadential Retention 理論の曖昧性を示し，その曖昧性を解消する手法を発見するために行った実験について，その方法や考察を述べる．

次に，4 章でこれらの結果を踏まえたうえで，Cadential Retention のための計算機上で実装可能な TPS による和声解析について述べる．

最後に，5 章で本研究のまとめ，および今後の課題について述べる．

第2章　構成論的音楽理論

本研究の大目的として，構成論的音楽理論の計算機上への実装が挙げられる．そのため，本章では，現在提唱されている構成論的音楽理論のうち，本研究に特に関連するものについて解説を行う．具体的には，まず本研究での解説の際に必要となる音楽理論の基礎知識を簡単に述べ，次に本研究の目的となる Cadential Retention の親理論である GTTM について述べ，その理論を計算機上へ実装可能なように理論を再構成した exGTTM について述べ，さらに，Cadential Retention に必要と思われる和声解析の手法として，TPS および TPS を用いた楽曲の和声解析について述べる．各理論や論文等は分量が多いため，特に本研究と関連が深いと思われる部分について重点的に解説を行う．

2.1　本研究で使用する音楽理論についての簡単な解説

本研究では，解説の際に一般的な音楽理論を用いることが多いため，ここでそれらについて簡単に説明を行う．

2.1.1　和音記号とバークリーメソッド

本研究では，ある特定の音を同時に鳴らす「和音」を多く扱うが，その表記方法はさまざまなものが提案されており，使用されるジャンルや地域などによりその表記方法もまちまちである．そこで本節では，本研究で用いる和音記号 [6] とバークリーメソッド [10] について簡単に解説を行う．

和音記号

調とは楽曲の基礎となっている音階の音を統一する和声的関係であり，明るい印象を受ける長調と，暗い印象を受ける短調が存在する．

和音記号による和音の表現は，音階から和音が発生するということを基礎に作られている．図 2.1 は長調の音階（上段）と短調の音階（下段）に三和音 (図の (a)) と四和音 (図の (b)) をそれぞれ重ねたものである．このように，ある調の音階を基準に和音を作るため，これらの和音は $I/\mathbf{C}$ のように表記される．この表記法で，分母にあたる部分がどの調の音階を用いているかを表し，分子にあたる部分が音階の何番目の和音を用いているかをギリシャ数字で表している．また，長調は大文字，短調は小文字で表現され，各調の日本語

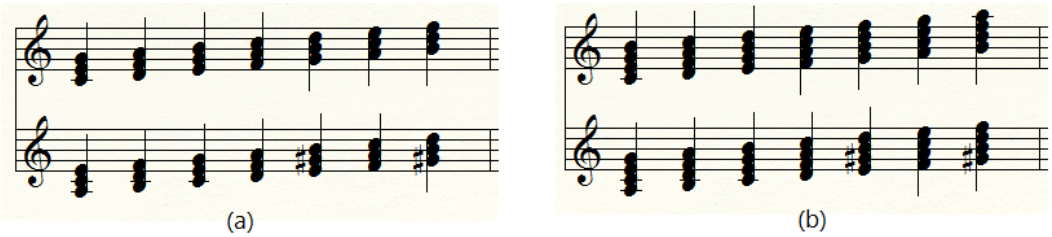


図 2.1 音階上の和音

表 2.1 調の表現（ドイツ式）と日本語の対応

長調	変ハ	変ト	変ニ	変イ	変ホ	変ロ	ヘ	ハ
独式	Ces	Ges	Des	As	Es	B	F	C
長調	ト	ニ	イ	ホ	ロ	嬰ヘ	嬰ハ	
独式	G	D	A	E	H	Fis	Cis	
短調	変イ	変ホ	変ロ	ヘ	ハ	ト	ニ	イ
独式	as	es	b	f	c	g	d	a
短調	ホ	ロ	嬰ヘ	嬰ハ	嬰ト	嬰ニ	嬰イ	
独式	e	h	fis	cis	gis	dis	ais	

表記との対応は表 2.1 のようになっている．さらに，明るい印象を受ける長和音は大文字で，暗い印象を受ける短和音は小文字でギリシャ数字による番号を表記する．

また，和音の主な音はルート音と呼ばれる．さらに，図 2.1 の (b) のような 4 つの音で構成される和音は 4 和音と呼ばれ，その表記には付加された音がルート音から 7 音離れているため，$I_7/\mathbf{C}$ のように，番号の右下に 7 をつけて表現を行う．同様に，5 和音や 6 和音も $I_9/\mathbf{C}$ や $I_{11}/\mathbf{C}$ のように表現される．

バークリーメソッド

和音記号が音階をもとにして作られるのに対して，和音の構成音のみに着目して和音を表現する方法はバークリーメソッドと呼ばれる．このバークリーメソッドによりつけられた和音の名前はコードネームと呼ばれ，単にコードと呼ばれることが多い．

バークリーメソッドでは，表 2.2 で表されるようなアメリカ式のルート音表記に，和音の種類を添えて Daug のように表現が行われる．ルート音にシャープやフラットがつく場合にはルート音の右肩につけて，$C^{\#}$ のように表現が行われる．和音の種類に関しては，ルート音から何半音上の音を重ねるかで表現が行われ，各音の重ね方は表 2.3 で示されるようなものとなっている．これは，ルート音を 0（基準）としたときに，何半音上に音を重ね

表 2.2 ルート音と音名の対応

音名	ド	レ	ミ	ファ	ソ	ラ	シ
米式	C	D	E	F	G	A	B

表 2.3 バークリーメソッド式の和音の構成音

コード記号		(M)	m	m-5	aug	M7+5	M7	7	mM7	m7	m7-5	dim
構成音	7度					11	11	10	11	10	10	9
	5度	7	7	6	8	8	7	7	7	7	6	6
	3度	4	3	3	4	4	4	4	3	3	3	3
	ルート音	0	0	0	0	0	0	0	0	0	0	0

ることでその和音が表現できるかをまとめたものである．例えば，*Cm*7 の和音であれば，ドの音をルート音とし，そこから 3 半音上のミ♭，7 半音上のソ，10 半音上のシ♭がその構成音となる．

バークリーメソッドの利点は，調を決定することなく，和音を表現できることであり，現在の，特にポピュラーミュージックのコード表記にはこのバークリーメソッドが用いられる傾向が強い．

2.1.2　機能和声と終止

和声とは和音の進行により生じる色彩の変化のことであり，機能和声とはそれぞれの和音に，その進行方法に合わせて機能（役割）をつけたものである [6]．本研究では終止と呼ばれる和声進行が非常に重要になっており，機能和声はこの終止構造と非常に関連が深い．また，機能和声の機能は，調に依存して決定されるため，その表現には和音記号が用いられる．

和音の機能はトニック・ドミナント・サブドミナントに分類される．トニックはその調の主とする和音のことであり，*I* や *iv* 等がトニックに分類される．ドミナントはトニックを導く和音であり，*V* などが分類される．このドミナントとトニックを用いて作られるものに終止がある．

終止は，楽曲が区切られるような感じや，終わるような感じを作る和声であり，その性質上よく文章のコンマやピリオドに例えられる．終止には図 2.2 のような 4 つの種類があり，$V-I$ で作られる全終止，$IV-I$ で作られる変終止，$V-vi$ で作られる偽終止，*V* のみで作られる半終止がある．また，これらの和音の *V* や *IV* は機能和声の考え方で，同じ機能を持つ和音に置き換えが可能である．これらの終止の中でも，全終止，偽終止，半終止はドミナントの役割を持つ *V* が必ず存在し，トニックを導くためのドミナントが非常に重要な役割を果たしていることがわかる．

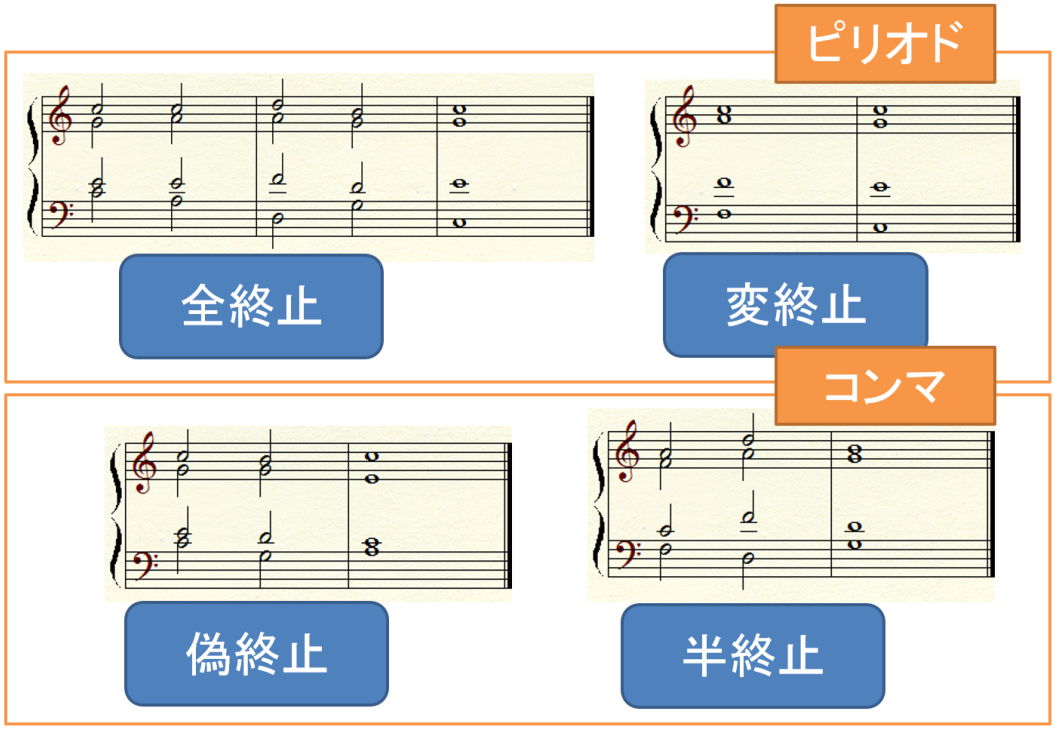


図 2.2 4 つの終止構造

2.1.3　近親調と遠隔調

音楽で用いられる調という概念は互いに関係しあうという性質を持っている．この調の関係の中でも，現在の調（主調）から特に関係の深い調のことを近親調，それ以外の調のことを遠隔調と呼ぶ．

近親調には図 2.3 で示される（主調を除く）6 つのものが存在する．ここでは，ハ長調を主調とした近親調を例として示している．それぞれに図に示したような主調との関係を持ち，楽曲中での転調（本来楽曲が持っていた調から，別の調に調性が移り変わること）にはこの近親調が多く用いられ，音楽理論の中では，主調との関係が非常に深いものとなっている．

これらの音楽理論を踏まえたうえで，本研究で取り扱う構成論的音楽理論について次節以降で述べる．

2.2　A Generative Theory of Tonal Music (GTTM)

GTTM [2] は Chomsky の生成言語文法理論と Schenker の音楽解析理論をもとにした理論であり，和声と一つのメロディで作られる調性音楽（ホモフォニー）のための生成理論となっている．具体的には，GTTM は，生成文法で用いられるツリー構造の解析手法を音

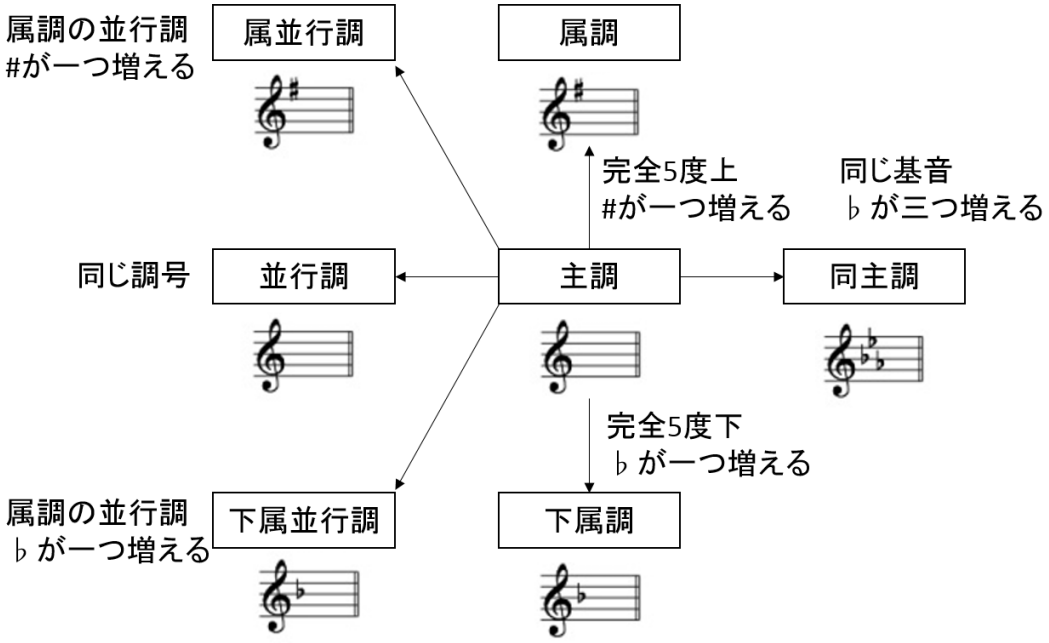


図 2.3 近親調

楽に適用するために，様々なルールを定め，タイムスパン木とプロロンゲーション木と呼ばれる二分木を生成するための理論であり，図 2.4 に示す 4 つの構成要素により成り立っている．GTTM では，楽譜を入力として与え，図 2.4 のように，Grouping Structure 解析と Metrical Structure 解析が行われ，それらの情報を用いることで Time-Span Reduction が行われる．さらに，Time-Span Reduction の情報を用いることで Prolongational Reduction が行われ，これらの結果を手前の解析にフィードバックさせながら，収束させることで解析が完了する．各構成要素は，それぞれ Well-Formedness Rules と Preference Rules と呼ばれる二種類のルールによって記述され，これらのルールは，それぞれ，構成要素内でのルール適用の際の基本ルールと，構成要素内での具体的な解析ルールと言うことができる．

GTTM のルールは，他の音楽理論と比べて比較的厳密なルールで記述されており，音楽知識を形式化するうえで最も有望であると考えられるが，これらのルールは人間が解析することを前提としているため，計算機で行うために適切なルールにはなっていないと考えられる．

GTTM は本研究の背景・目的に深く関連し，3 章で，GTTM 内のルールの 1 つである Time-Span Reduction について扱うため，ここで Time-Span Reduction までの理論についてまでを述べることにする．

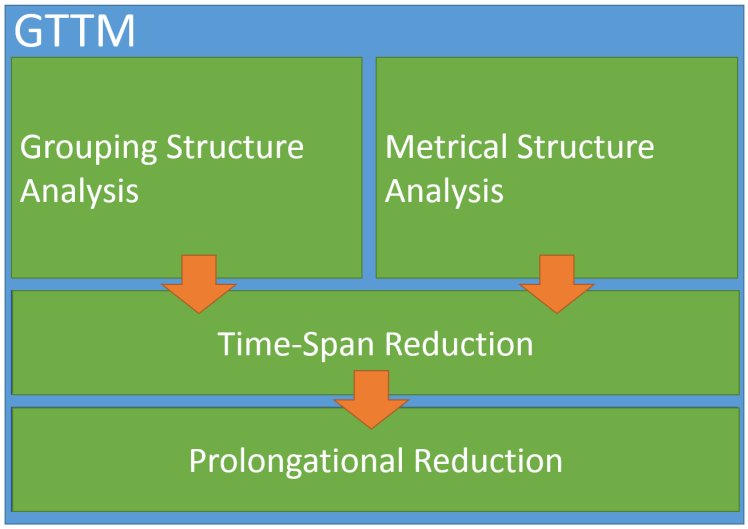


図 2.4 GTTM の構成要素

2.2.1 Grouping Structure

グルーピング構造解析 (Grouping Structure Analysis) は，楽曲を音楽的にまとまりを感じるグループに分割し，各グループの階層構造を決定するための解析である．この解析では，図 2.5 のように Grouping Preference Rules (GPR) によって分割される可能性のある部分をマークし，その可能性の強い部分でグループに分割する．分割の際には図 2.5 の下部に示しているスラーと呼ばれる記号を用いる．

Groupinf Well-Formedness Rules (GWFR)

グルーピング構造解析のルールは Grouping Well-Formedness Rules (GWFR) と GPR で構成され，このうち GWFR は以下の 5 つで構成されている．

1. 連続したピッチイベント、ドラムビート、もしくはそれらのようなもののどんなシーケンスもグループを構成することができる．また，グループを構成することができるのは連続したシーケンスに限る．
2. 一つの楽曲は一つのグループを構成する．
3. グループはより小さなグループを含むことができる．
4. グループ G_1 がグループ G_2 の一部を含む場合，G_1 は G_2 のすべてを含まなくてはならない．
5. グループ G_1 がより小さなグループ G_2 を含むならば，G_1 はより小さなグループに徹底的に分割される必要がある

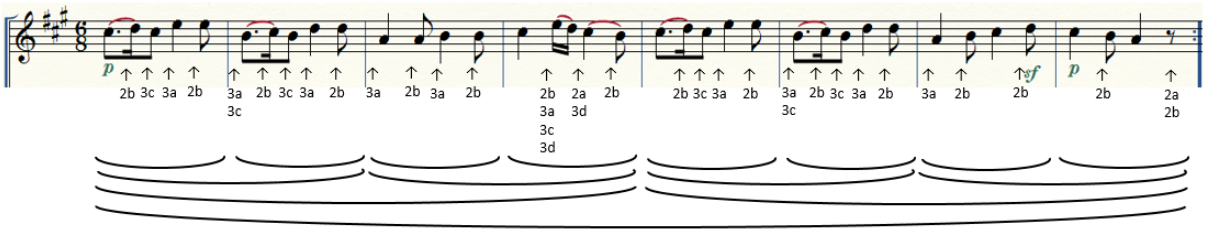


図 2.5 グルーピング構造解析の例
(Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331 より)

ここで，上記のルールについて説明を加える．

GWFR1 は，楽曲内のすべての音やそれに準ずるイベントは連続している場合に限りグループを構成することができるということを示している．また，GWFR2 と 3 は，楽曲は一つのグループであり，その中には図 2.6 のように小さなグループを含むことができるということを示している．しかし，このグルーピングには GWFR4 と 5 で示されるような制限がある．GWFR4 は図 2.7 のようなグルーピングを禁止する．このようなグルーピングの場合には G_2 はそのすべてが G_1 に含まれる必要がある．GWFR5 は図 2.8(a) のようなグルーピングを禁止する．GWFR5 によると，大きなグループの中に小さなグループが含まれている場合，大きなグループの中の音は図 2.8(b) のようにすべてがグループ化されていなくてはならない．

これらのルールにより，グルーピングのための基本的なルールが示される．

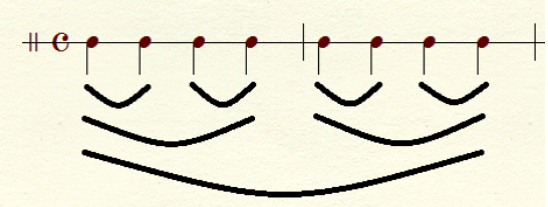


図 2.6 Grouping Structure におけるグループの包含関係

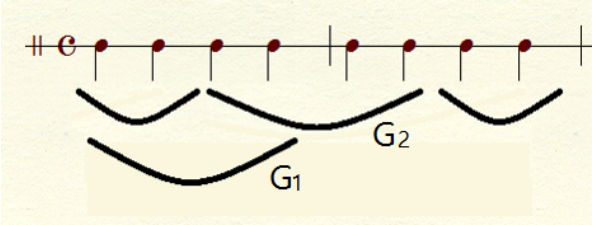
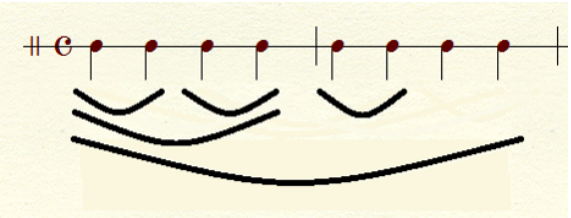
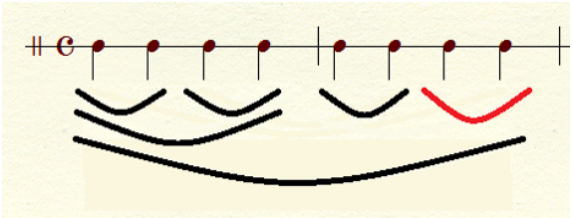


図 2.7 禁止されたグルーピング 1



(a)



(b)

図 2.8 禁止されたグルーピング 2

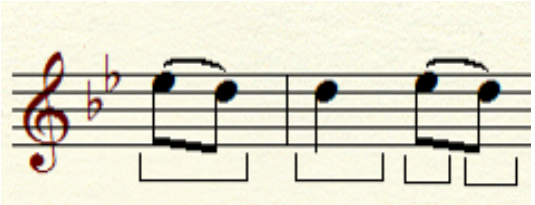


図 2.9 GPR1 で禁止されるグルーピング
(Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331. より)

Groupinf Preference Rules (GPR)

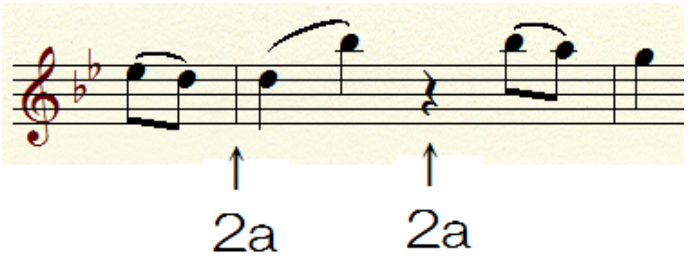
GWFR に対し，具体的にどのようなポイントをマークするのかについては以下で示されるの GPR を用いる．

1. altenative form
2. Proximity
3. Change
4. Intensification
5. Symmetry
6. Parallelism
7. Time-span and Prolongational Stability

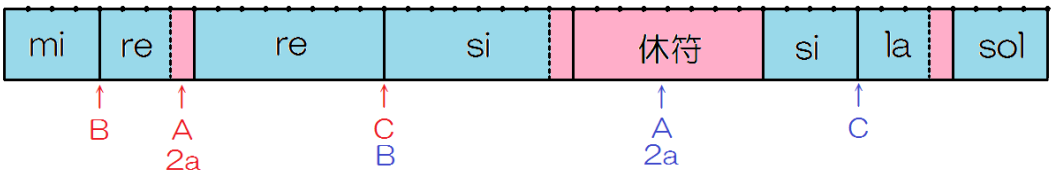
これらのルールのうち，GPR1～3 は Local Detail Rules と呼ばれ，小さなレベル（階層）で使用されるルールである．また，GPR4～7 は Organization of Larger-Level Grouping と呼ばれ，大きなレベルで使用されるルールである．これらのルールの詳細を以下に述べる．

GPR1(alternative form) は，より小さなグルーピングはそれほど好ましくないので，非常に小さなグループへの解析は避けるというものである．これはつまり，非常に強い根拠がある場合および，それよりは弱いがかなり強い根拠がある場合を除いて，図 2.9 のような，単一音のグルーピング，および 2 つの音のグルーピングをそれぞれ禁止するというものである．これは，非常に小規なグルーピングを行うという知覚は，あまり重要ではない傾向があるということを根拠に，グルーピングの区分化があまりにも神経質になるのを防ぐことを目的としたルールである．

GPR2(Proximity) は，音の長さによるグループの境界として可能性がある部分をマークするためのルールである．具体的に，このルールでは，4 つの連続する音 n_1，n_2，n_3，n_4 を考えたとき，以下の a. および b. のいずれかが適用される場合には n_2 と n_3 の間がグループの境界として聴かれる可能性があることを示している．



(a)Mozart, Wolfgang Amadeus, Symphony No.40 in G minor, K.550 の楽譜（抜粋）



(b) 時間軸に音を並べた帯グラフ (青が音, 赤が休符)

図 2.10 GPR2a の例

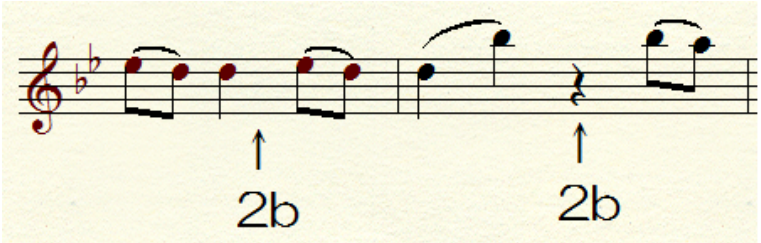
a. (Slur/Rest) n_2 の終わりから n_3 の始まりまでの時間間隔が，n_1 の終わりから n_2 の始まりまでの時間間隔および，n_3 の終わりから n_4 の始まりまでの時間間隔よりも大きい場合

b. (Attack-Point) n_2 と n_3 のアタックポイント (始まりの点) の時間間隔が，n_1 と n_2 のアタックポイント間の時間間隔および，n_3 と n_4 のアタックポイント間の時間間隔よりも大きい場合

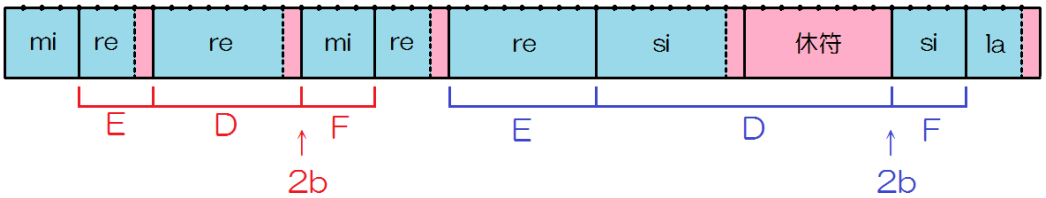
GPR2a を考える際に注意しなくてはならないことは，図 2.10(b) の 2 番目の音 (re) と三番目の音 (re) の間に示しているように，スラーとスラーの間にはごくごく小さな隙間ができるということである．このようにスラーがある音とない音の長さを違うものとして考えることで，GPR2a はスラーとスラーの間の切れ目や，休符の部分に対してマークされる．例として，図 2.10(b) の始めの音から順番に n_1〜n_4 とすると，A の長さが B よりも長く，かつ，A の長さが C よりも長い場合に n_2 と n_3 の間が GPR2a によって図 2.10(a) のようにマークされる．同様に，休符が存在する部分でも同じように GPR2a によるマークが行われる．

GPR2b は，n_2 の音の始まりと n_3 の音の始まりの時間間隔が長い場合に適用される．具体的には，図 2.11(b) の D で示す時間が E および F で示す時間間隔よりも長い場合に，図 2.11(a) のようなマークが行われる．

GPR3(Change) は，音の変化によるグループの境界として可能性がある部分をマークするためのルールである．このルールでも，4 つの連続する音 n_1，n_2，n_3，n_4 を考えたとき，以下の a. から d. のいずれかが適用される場合に，n_2 と n_3 の間がグループの境界として聴



(a) 楽譜 (Mozart, Wolfgang Amadeus, Symphony No.40 in G minor, K.550 より)



(b) 時間軸に音を並べた帯グラフ (青が音, 赤が休符)

図 2.11 GPR2b の例

かれる可能性があることを示している．

a. (Register) n_2-n_3 間の音高が，n_1-n_2 および n_3-n_4 間の音高よりも離れている場合

b. (Dynamics) n_2 と n_3 の間でダイナミクス (音量) の変化があり，n_1 と n_2 および n_3 と n_4 の間で変化がない場合

c. (Articulation) n_2 と n_3 の間でアーティキュレーションの変化があり，n_1 と n_2 および n_3 と n_4 の間で変化がない場合

d. (Length) n_2 と n_3 の間で音長の変化があり，n_1 と n_2 および n_3 と n_4 の間で変化がない場合

GPR3a は，図 2.12 のようにある 4 つの音が並んでいるとき，G の音高差が H と I の音高差よりも大きい場合 n_2 と n_3 の間がマークされるというものである．このルールでは，一音だけ高く（あるいは低く）なるような音と音の間をグループの境界としてマークすることはできず，二音以上の連続した音程の間に大きな音高差がある場合に限りマークを行うことができる．

GPR3b は，*p* (ピアノ) や *f* (フォルテ) といった記号によるダイナミクス (音量) の変化に対するマーキングであるが，これも GPR3a と同様に *sfz*(スフォルツァンド，その音のみを強く奏するという記号) などによる単音の変化ではマークされない．

GPR3c は，アーティキュレーションの変化に対するマーキングである．ここでいうアーティキュレーションとは，スラーやスタッカート，アクセントといった演奏の表情づけを行うために書かれた記号のことであり，GPR3c も GPR3a や 3b と同様に単音の変化ではマークされることはない．

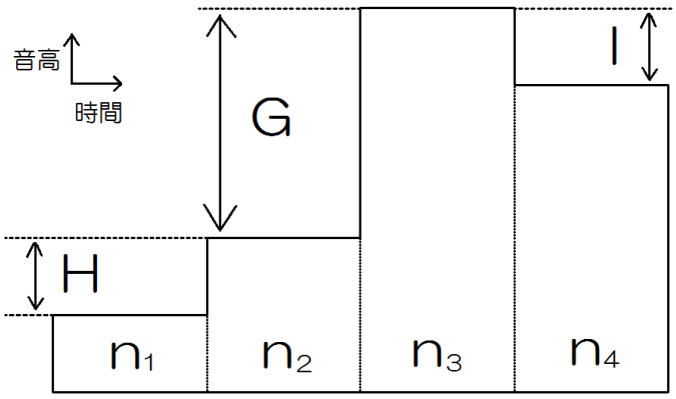


図 2.12 GPR3a の説明

GPR3d は，音長の変化に対するマーキングである．しかし，GPR2 で説明したようなスラーを含めた長さは考慮しておらず，単純に音そのものが持つ長さによってマークされる．また，GPR3d においても，GPR3a～3c と同様に単音の変化によりマークされることはない．

GPR2 と 3 の強さについて，一般的に以下のようなことが言える．

- GPR2a(Slur/Rest) > GPR2b(Attack-Point)
- GPR2a(Slur/Rest) > GPR3a(Register)
- GPR3b(Dynamics) > GPR2a(Slur/Rest)
- GPR2 > GPR3a(Articulation), 3c, 3d(Length)

しかし，これらのことは絶対ではなく，音の長さやビート感などによってどのルールが重要視されるかは変わってくる．

続いて，Organization of Larger-Level Grouping と呼ばれる大きなレベルで使用されるルール GPR4～7 について説明する．

GPR4(Intensification) は，GPR2 と 3 によってマークされている効果が比較的明白な部分は，大きなレベルにおいても境界となる可能性が高いというものである．このルールを図 2.13 のような例を用いて説明すると次のようになる．図 2.13 では，休符の部分には GPR2a と GPR2b がマークされており，ほかの部分 (GPR2a のみ) よりもそのマークの数が多い．さらに休符であるために，グループの境界となる可能性は高いとも言える．このような場合には，その部分を高次レベルでのグループ境界として選択することが望ましいというのが GPR4 の示すところである．

GPR5(Symmetry) は，同じ長さの二つの部分によって大きなレベルのグループが構成されることを理想とし，そうなるようにグループを構成するというものである．図 2.14 を

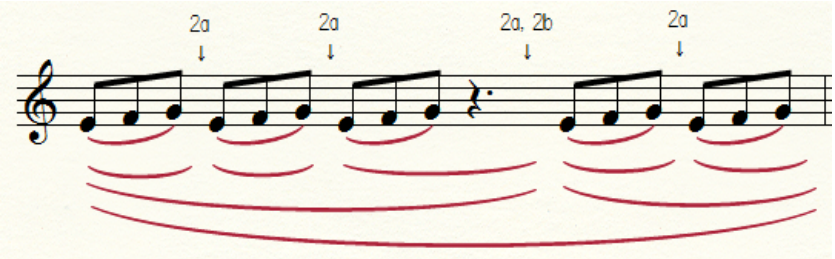


図 2.13 GPR4 の例
([2] の p.49, 図 3.20 を参考に作成)

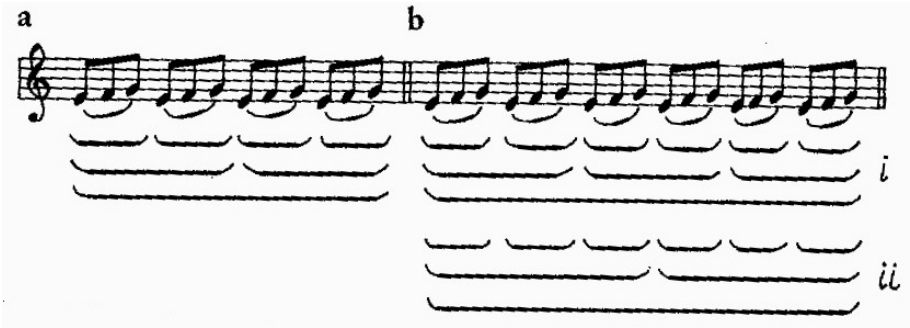
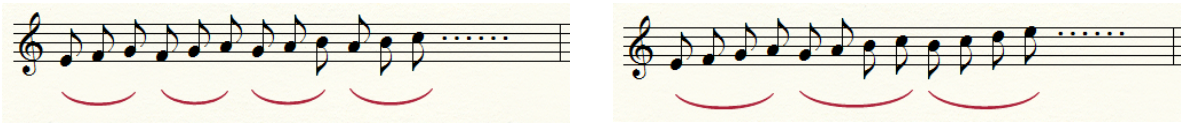


図 2.14 GPR5 の例
([2] の p.50，図 3.21 より転載)

見てみると，a のような例の場合，中間レベルでのグルーピングはこの GPR5 を用いて同じ長さの二つの部分をグループ化することで行われ，これらのグループが同じ長さとなるので，上位レベルでのグルーピングでは，再び GPR5 を用いてこれらをまとめている．しかし，b の例では，*i* では中間レベルで GPR5 を用い，上位レベルでは GPR5 に背いたグルーピングをしているのに対し，*ii* では中間レベルで GPR5 に背いたグルーピングを行い，上位レベルで GPR5 を用いている．これらのどちらが正しいのかについては，この情報のみではわからず，他のルールを参照しつつどちらが正しいかを選択することになる．

GPR6 (Parallelism) は，2 個以上の音楽の部分が並行 (相似) と解釈できる場合，それらを並行にグルーピングすることが望ましいというものである．図 2.2.1 を見てみると，これら (a)，(b) の楽譜には GPR2 や 3 によるマークはできないが，これらの音符は明らかに並行した音の配列をしており，この場合には図の下部に示すようなグルーピングをするべきである．しかし，例えばこれが，図 2.16 のようにスラーがかかり，GPR2a でマークされたような場合には，GPR6 は無視されることになる．このことから，GPR6 はあくまでも補助的なルールであるということが見て取れる．

GPR7 (Time-span and Prolongational Stability) は，タイムスパン簡約やプロロンゲーション簡約もしくはその両方の結果がより安定するようなグルーピング構造を好んで選ぶと



(a) (b)

図 2.15 GPR6 の例
([2] の p.50，図 3.22 を参考に作成)

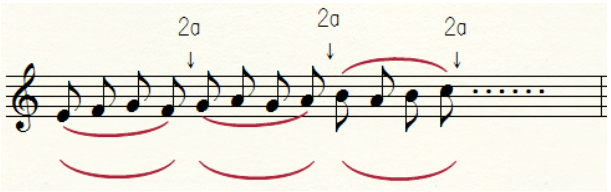


図 2.16 GPR6 が無視されるような例
([2] の p.51，図 3.23 を参考に作成)

いうものである．図 2.4 で示されたような順序で GTTM の各解析が行われた後，このルールによってフィードバック処理が行われることになる．このように，GTTM による楽曲の解析は，各解析がフィードバックを繰り返して，安定になるような形に収束したときに完了する．

これらの GPR に加えて，グルーピング構造を解析する際にはオーバーラップと呼ばれる例外が存在する．これは，グルーピングの最後の音が次のグルーピングの最初の音と共通のものであるときに図 2.17 のような，一つの音が同じレベルでの 2 つのグループにグルーピングされることを例外的に許可するというものである．これは主に，終止部で結ばれた部分と，その次のフレーズの始まりが共通の音になっている場合に起きる現象であり，そのような状況でのグルーピングには注意が必要である．

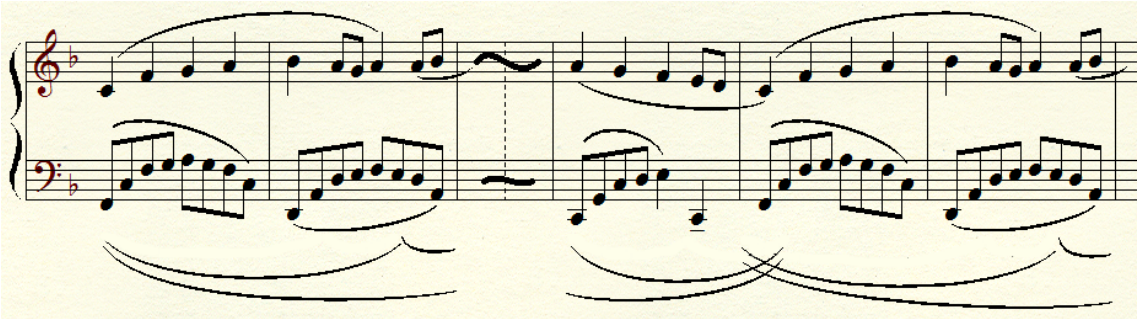


図 2.17 オーバーラップグルーピングの例

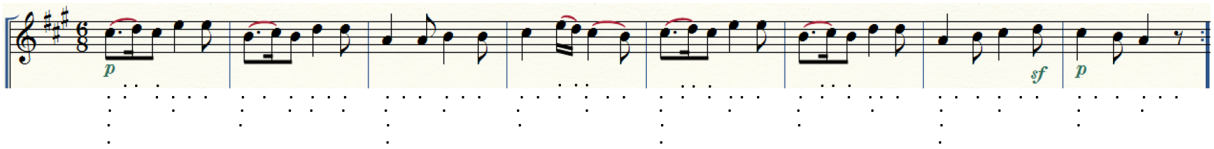


図 2.18 拍節構造解析の解析結果例

2.2.2 Metrical Structure

拍節構造解析 (Metrical Structure Analysis) は，四分音符や二分音符などの各拍節レベルにおける強拍と弱拍を決定するための解析である．強拍とは，頭拍とも呼ばれ，曲に合わせて手拍子をするタイミングや，指揮棒を振るタイミングのようなものである．弱拍とは，強拍以外のもののことで，裏拍とも呼ばれる．拍節構造解析は指揮棒を振るようなタイミングを見つけるための理論であるが，現在楽譜を表すデータ形式として musicXML([9]) が登場したことにより，楽曲の拍子が容易にデータとして扱えるようになっているため，この理論による解析はほとんど重要ではなくなってきている．つまり，楽曲の拍子の情報が分かることで，この理論を用いずとも拍節構造が捉えられるようになっているのである．

しかし，この理論では小節を超えた強拍（つまり，どの小節がより強拍の役割を持つか）である Hyper Measures を解析できるという利点があるため，ここでは各ルールを下記に列挙し，簡単な説明を述べるまでにとどめる．

図 2.18 は拍節構造解析の解析結果の例である．解析の結果はドットで示され，ドットの数が多い拍ほど高次レベルでの強拍になる拍となっている．また，拍節構造解析では標準音価を tactus と呼び，解析の際に，現在のレベルを表す指標としている．例えば，図の一小節目の一拍目の音について，現在の解析レベルが二つ目のドットのレベルであるとすると，tactus は付点四分音符の長さとなる．

Metrical Well-Formedness Rule (MWFR)

拍節構造解析の Well-Formedness Rule は以下のものである．

1. すべてのアタックポイントは曲中の各部分における最小の拍節レベルの拍でなければならない．
2. あるレベルのすべての拍はより小さなレベルでの一つの拍でもある．
3. 各々の拍節レベルで，強拍は 2 または 3 拍の間隔を持つ．
4. tactus や大きな拍節レベルは，同等の間隔を持つ拍によって構成される．

これらのルールが拍節構造解析を行う際の基本となるルールになるが，これらは一般的に拍子における強拍と弱拍のルールに類似しており，このことからも拍子から拍節構造解析と同様の情報が得られることがわかる．

Metrical Preference Rule (MPR)

拍節構造解析の Preference Rule は以下のものである．

1. (parallelism) 複数のグループ，またはグループの各部を並行的と解釈できる場合には，並行的な拍節構造を優先する．
2. (strong beat early) 最も強い拍がグループ内で比較的早く表れる拍節の構造を優先する．
3. (event) 拍のポイントに音符がある拍節構造を優先する．
4. (stress) 強く演奏された拍が強拍である拍節構造を優先する．
5. (length) 以下の各事項にたいして，より長いという条件を満たす拍を強拍とする拍節構造を優先する．
    a. 相対的に長い音
    b. 相対的に長く続く一定の音量
    c. 相対的に長いスラー
    d. 相対的に長い同じアーティキュレーションパターンの繰り返し
    e. タイムスパン簡約による相対的に長く続く 1 つの音高
    f. タイムスパン簡約による相対的に長く続く 1 つの和声
6. (bass) バス音が拍節的に安定した拍節構造を優先する．
7. (cadence) 終止部では拍節的に安定した構造を優先する．つまり，ほかの場合よりも終止部での局所的な MPR のルール違反は避けなければならない．
8. (suspension) 掛留音はその解決よりも開始部を優先する．
9. (time-span interaction) タイムスパン簡約における競合が最小になるような拍節構造を優先する．
10. (binar regularity) 各レベルにおいて，強拍が 1 つおきにくる拍節構造を優先する．

ここで，掛留音とは和音をなす音がその他の和音まで延長保持されている音のことを指す．この掛留音は必ず解決（和声外の音から，和声音に変化すること）される．これらのルールにより拍節構造が解析されるが，特に重要なのは Hyper Measures であり，拍子から解析できる部分は拍子から解析を行い，それ以降はこれらのルールを用いた解析を行うことが好ましい．

2.2.3 Time-Span Reduction

タイムスパン簡約 (Time-Span Reduction) は，メロディの各音を重要なものとそうでないものに分け，重要でない音をその音が従属している音に吸収させ，楽曲の本質となる音を取り出す「簡約」という作業を行うための解析である．解析のためにはグルーピング構造解析と拍節構造解析の結果が用いられ，その出力結果は図 2.19 で示すような優先度を持つ二分木構造を組み合わせたものとなる．この二分木構造は二音を一音で代表する効果を持っており，図 2.20(a) で示すヘッド（ド）に従属する音 (ソ) は，簡約を行うことで (b) のようにヘッドに吸収される．これを繰り返すことで，楽曲の持つ重要な音を知ることができる．また，図 2.19 に示している *egg* と呼ばれる記号は，本研究でも特に重要なものであり，終止部の構造を持つ部分に付ける．

図 2.21 は，図 2.19 の木をもとにタイムスパン簡約を行った例である．上から順に，原曲，八分音符レベルでの簡約，付点四分音符レベルでの簡約，一小節レベルでの簡約…という風に，各レベルでの簡約を行っている．先にも述べたように，簡約のレベルが上がるにつれて音が少なくなり，楽曲中で重要となる音がどの音であるかがはっきりしていくことが見て取れる．また，先には述べなかった *egg* の効果について，赤い四角で囲った終止部の部分を見てもらいたい．この部分では，大きなレベルでの簡約を行っているにも関わらず，その簡約レベルよりも小さな長さを持つ終止部が残っていることがわかる．これが *egg* の効果であり，楽曲構造上重要な役割をもつ終止部を，簡約せずに残すためのものであるといえる．この *egg* を付加するルールは Cadential Retention と呼ばれ，終止部の特殊な簡約方法は cadential reduction と呼ばれる．

ここで，タイムスパンという言葉の意味について説明を行う．

タイムスパンとは時間間隔のことであり，これはある拍節構造のビートではじまり，そして他のビートを含まずにそのビートを上へと拡張する．これをわかりやすく言いかえるために，簡約前の楽譜を上段に，二分音符レベルでの簡約を行った楽譜を下段に記した図 2.22 を用いて説明を行う．図のグルーピング構造を示している部分の下に記している記号がタイムスパンを表している．いま，上段の四分音符のレベルを下段の二分音符のレベルに簡約したとき，まず，タイムスパン木において一番下のレベルにあるミの音（第四音）がファの音（第三音）に簡約される．しかし，この状態ではまだドの音（第一音）とソの音（第二音）が四分音符の長さのタイムスパンとなってしまい，二分音符レベルでの簡約にはならない．そこでファの音をさらにその音へと簡約させる．すると，ソの音は付点二分音符分の長さ（四分音符と二分音符を合わせた長さ）を持つことになるように感じるが，実際には下段の楽譜のように，どちらの音も二分音符分の長さになる．つまり，タイムスパンの長さは拍節構造解析で得られた強拍の長さと一致し，上位レベルへの簡約を行った場合には，簡約された音はタイムスパンの長さと一致することになる．

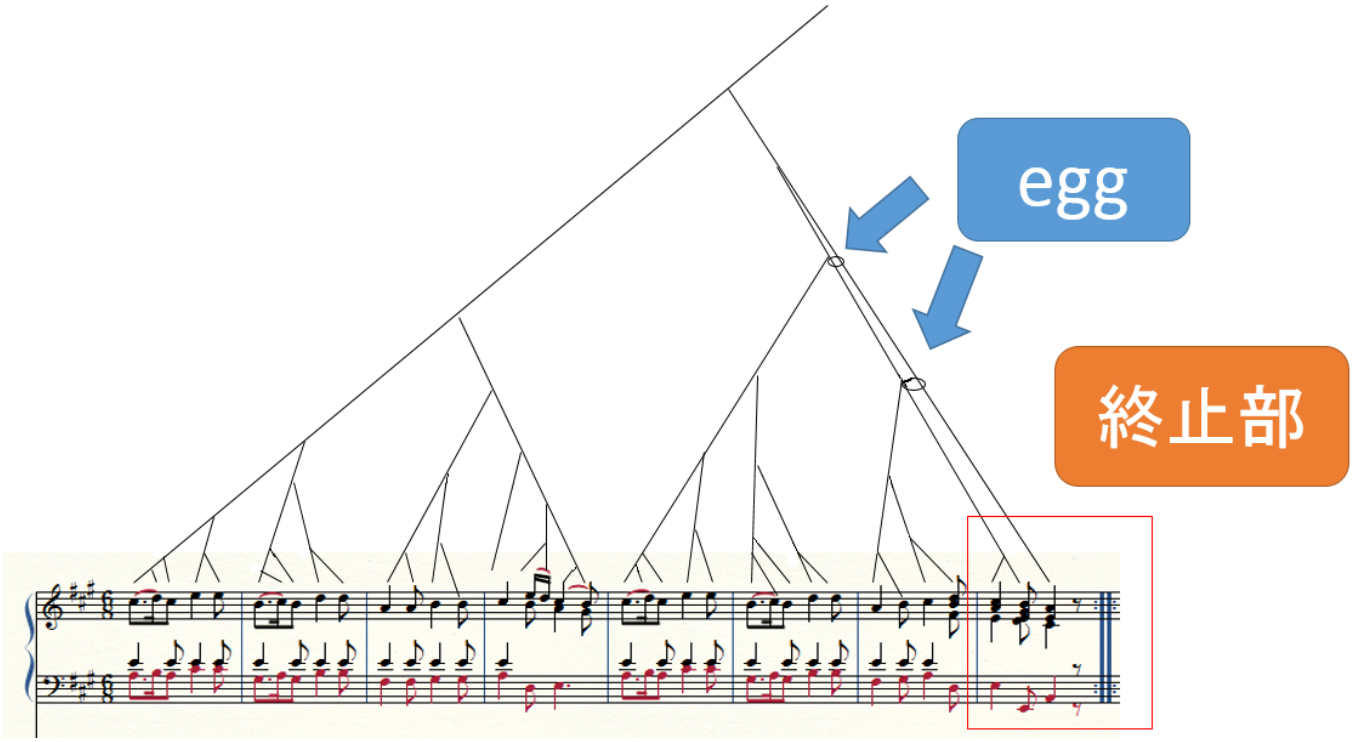


図 2.19 タイムスパン木の例
(Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331. より)

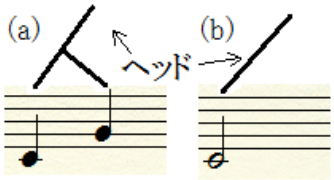


図 2.20 タイムスパン木におけるヘッドの役割

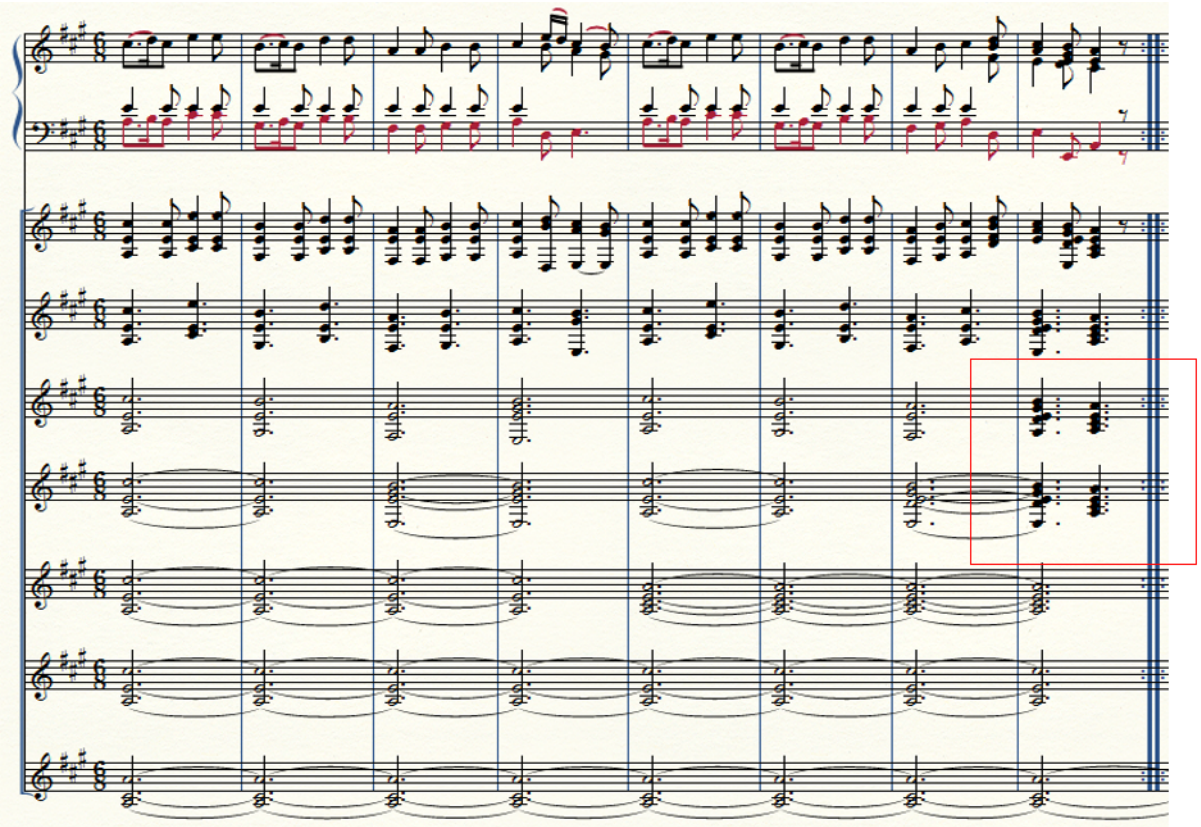


図 2.21 タイムスパン簡約の例
(Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331. より)

Segmentation Rule (SR)

タイムスパン簡約での大きな二つのルールを説明する前に，まずSegmentation Rule(SR）について説明を行う．SR とはタイムスパンの範囲に関するルールであり，下記の二つのルールで構成される．

1. 一つの曲中の全グループはその曲のタイムスパンセグメンテーション中の一つのタイムスパンである
2. グルーピング構造によって以下のタイムスパンが決定される．
   a. 一番小さな拍節レベルの各拍 B は，B から次の拍までの長さを持つタイムスパン T_B を決定する
   b. 拍節レベル L_i の各拍 B は regular time-span T_B を決定する．この T_B は次に小さなレベルである L_{i-1} の B からのすべての拍のタイムスパンの合計である．ただし以下は含まない．
      (i) レベル L_i の次の拍 B'
      (ii) B' よりも早く来るグループの境界
   c. グループの境界 G が B および B と同じレベルの次の拍の間にあるとき，B は augmented time-span T'_B を決定する．また，この T'_B は G から T_B の終わりまでの長さを持つ．

SR1 で述べられているグループとはグルーピング構造解析の解析結果によるグループのことであり，このグループのそれぞれが一つのタイムスパンであることを示している．つまり，楽曲の簡約を行っていったときに，簡約した音が鳴っている時間であるタイムスパンはこのグループに一致することになる．また，このルールは，グルーピングのオーバーラップに非常に深く関係している．図 2.23 のようにグルーピングがオーバーラップしている場合，オーバーラップとして重なっている音（二小節目の始めのドの音）はタイムスパン木でも，それを用いた簡約でもそれぞれ二回用いられる可能性がある．これは，グルーピングがタイムスパンの長さと等しいためであり，グルーピングがオーバーラップしている部分でのタイムスパン木の作成や簡約には注意が必要である．

SR2 については図 2.24 を用いて説明を行う．まず，SR2a が四分音符レベルのタイムスパンを T_B 作る．続いて，二分音符レベルのタイムスパンを作るが，図に G で示したグループの境界があるために，SR2b の (ii) により，12 小節目三音目のタイムスパンは四分音符分の長さしか持たなくなる．これにより四音目の regular time-span T_B は存在しなくなる．また，このレベルにおけるほかの T_B は，SR2b の (i) により二分音符分の長さになる．さらにこの時 SR2c により，12 小節目の四音目から 13 小節目の二音目の終わりまでの長さを持つ augmented time-span T'_B が作られる．このタイムスパンは二分音符と四分音符を合わせた長さを持つが，二分音符のレベルのタイムスパンになる．このように，グループの

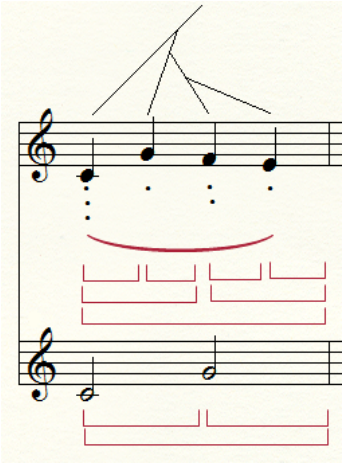


図 2.22 タイムスパンの説明

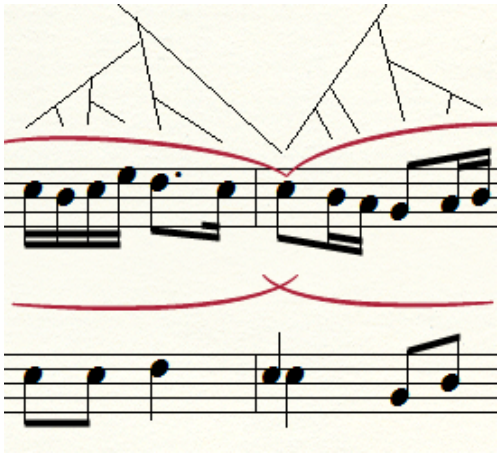


図 2.23 オーバーラップグルーピング時のタイムスパン木とその簡約

境界におけるタイムスパンの長さには注意が必要になる．また，このタイムスパンの作り方は楽曲の始まりに現れるような弱起（アウフタクト，小節の最初以外の音で始まるフレーズの始まりのこと）にも適用される．

Time-Span Reduction Well-Formedness Rules (TSRWFR)

ここからはSRによって説明されたタイムスパンおよびタイムスパン木が，具体的にどのようなルールによって作られていくかを説明する．具体的なタイムスパン簡約のルールはTime-Span Reduction Well-Formedness Rules (TSRWFR) と Time-Span Reduction Preference Rules (TSRPR) で構成され，このうち TSRWFR は以下の 4 つのルールで構成されている．

1. すべてのタイムスパン T にはイベント e，もしくはイベントの連続 e_1,e_2 が存在し，そのイベントは T のヘッドとなる．

2. T が他のいかなるタイムスパンも含んでいない場合（つまり，タイムスパンが最小レベルの場合），e はTに生じるすべてのイベントとなる．

3. T が他のタイムスパンを含んでいる場合，$T_1, ..., T_n$ を T のすぐ下のレベルにある（regular もしくは augmented）タイムスパン，$e_1, ..., e_n$ をそれらの各ヘッドとしたとき以下のことが言える．

    a. (Ordinary Reduction)T のヘッドはイベント $e_1, ..., e_n$ のいずれかになる可能性がある．

    b. (Fusion)$e_1, ..., e_n$ がグループの境界によって分けられていない場合 (“locality” condition)，T のヘッドは $e_1, ..., e_n$ が二個以上重なったものとなる可能性がある．

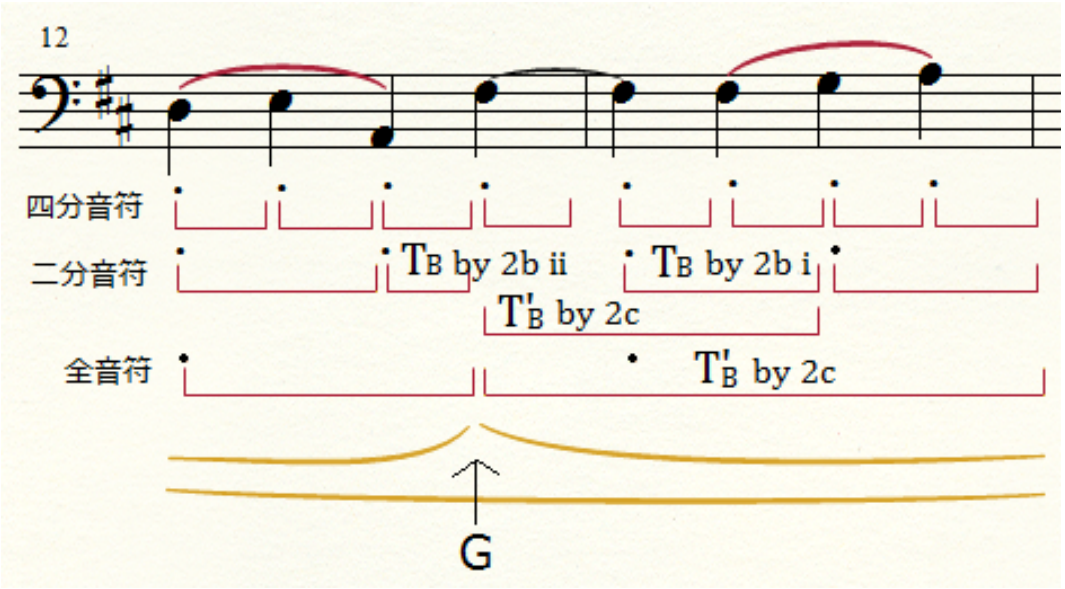


図 2.24 セグメンテーションルールの説明
([2] の p.148，図 7.2 をもとに作成．Beethoven, Ludwig van, Symphony No.9, Op.125. より)

c. (Transformation) $e_1, ..., e_n$ がグループの境界によって分けられていない場合，T のヘッドは $e_1, ..., e_n$ から選ばれたいくつかの相互に調和したピッチの組み合わせとなる可能性がある．

d. (Cadential Retention) T のヘッドは，最後の音 *final* が e_n（そのヘッドは T に直接含まれている最後のタイムスパン T_n）で，そのタイムスパン T_n を直接導くような e_n の一つ手前のタイムスパンのヘッド *penult* が存在しするような終止部 (cadence) となる可能性がある．また，T および T_n は必ずしも同じレベルでなくてもよい．

4. 二つの要素からなる終止部 (cadence) がタイムスパン T のヘッド e に直接従属する場合，*final* は e に直接従属し，*penult* は *final* に直接従属する．

TSRWFR1 はすべてのタイムスパンがヘッドを持つということを示している．ここで言うイベントとは，音程のあるピッチイベントの他に，打楽器なども含み，また T よりも下位のレベルにあるヘッドを持つタイムスパンも含んでいる．

TSRWFR2 は最小レベルのタイムスパンにおいては，そのタイムスパン T に存在するイベント e そのものがタイムスパン T のヘッドとなるというものである．

TSRWFR3 はヘッドを生成するための制約と方法を示したものである．各ルールの具体的な説明については図を用いて説明を行う．

TSRWFR3a は図 2.20(a) のようにタイムスパン内（ここでは二拍の長さのタイムスパンと考える）に存在する音からヘッドを一つ選択する方法である．この方法ではタイムスパ

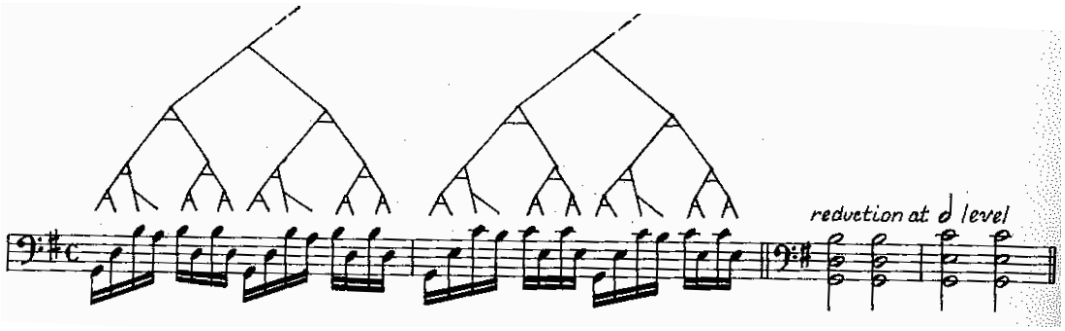


図 2.25 Fusion ルールの例
([2] の p.154，図 7.12 より転載．Bach, Johann Sebastian, Cello Suite No.1 in G major, BWV 1007. より)

ン木の簡約後の結果は (b) のようになるか，もしくは TSRWFR3c を用いた簡約方法になる．この簡約方法が最も一般的なものであるため TSRWFR3a は Ordinary Reduction と呼ばれる．

TSRWFR3b は音の融合により簡約が行われる場合があるというルールである．このルールでのタイムスパン木は図 2.25 のような木の分岐点に三角の印をつけたものとなる．この木を簡約する際には，これらの記号で結ばれた音を融合し，一つの音として考える．具体的に，図の左の楽譜を二分音符のレベルで簡約すると右の楽譜のようになる．この際，TSRWFR3a によって簡約される部分は融合が行われない．この木が使用されるのは，一つの和音を時間軸でずらして演奏する分散和音や，一人でメロディと伴奏になるルート音を交互に奏するような場合である．また，分散和音は一つの和音を分散して演奏しているため，必ず 1 つのローカルなタイムスパン内にそれらの音が存在する．そのため，このルールは完全にローカルな簡約のレベルでのみ使用される．

TSRWFR3c は変換による簡約方法に関するルールである．図 2.26(a) において，二分音符のレベルでの簡約を行う際，一小節目のコードは Em (イーマイナー) となる．Em の構成音はミソシであり，特にルート音であるミの音が重要となる．この楽譜では，上段の楽譜の一拍目のファの音はこのコードの構成音になっていない．これは倚音と呼ばれる和声外の音であり，この解決音（緊張がほぐれて落ち着く音，和声外の音の実際の音は解決音であると判断されることが多い）は次の拍のミの音である．このような場合，簡約を行った音は Em の構成音となる解決音であるミを選択するほうが好ましい．そのため簡約の結果は (b) のようにするのが妥当である．このように，和音の構成音に合わせて和声外の音ではなくその解決音を選ぶ簡約方法を Transformation と呼び，その内容は TSRWFR3c によって定義されている．

TSRWFR3d は終止部における特殊な木についてのルールである．Cadential Retention は，$V-I$ のような和声構造を持つ完全終止や，$V-VI$ のような和声構造を持つ偽終止に対して行われる．図 2.27 は TSRWFR3d (Cadential Retention) のルールの定義に従って楽譜にメモ

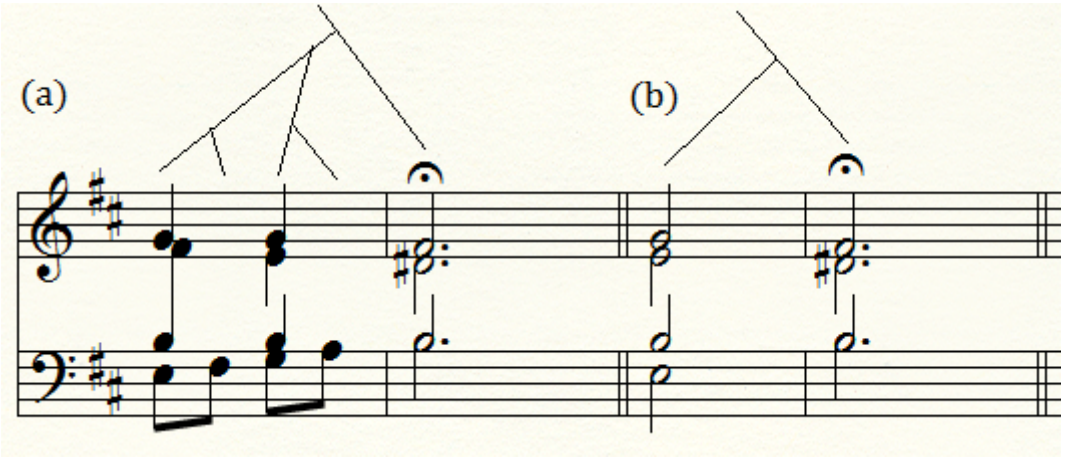


図 2.26 Transformation ルールの例
([2] の p.155，図 7.13 をもとに作成)

を加えたものである．タイムスパン T 内の各三つの音において，この和声構造は $I^2 - V_7 - I$ となる．ここで，I^2 とは I の和音の構成音の一番下の音をオクータヴ上げる操作を 2 回繰り返すことによって得られる和音であることを示す．音楽の和声理論によると，このような和声構造のとき，I^2 は V_7 と結合しドミナントとして扱われるため，第一音は第二音に従属する．そして，このヘッドとなる第二音はドミナントとして扱われるため，この音が T_n の一つ手前の音 *penult* となる．ここで，この *penult* は終止部の I をである *final* の音（第三音）と結合し，終止構造となるのでこれらの音は一つのユニットとして扱われることになる．この終止部に従属するような音は，図 2.27 の一小節目第一音のように，*egg* と呼ばれる記号を付加して枝を結合する．これにより，終止部に従属する音は，二つの音に従属していることが分かるようになる．

TSRWFR4 も終止部に関するルールであり，終止部が他のタイムスパンに従属する際にどのように従属するかを示している．このような場合には，図 2.27 に示す木の上部に示したように，*final* は従属する木に直接従属し，*penult* は *final* に直接従属することになる．

Time-Span Reduction Preference Rules (TSRPR)

TSRWFR に対し，あるタイムスパンにおいて，具体的にどのようにしてヘッドを選ぶかについては以下で示されるの TSRPR を用いる．

1. (Metrical Position) タイムスパン T のヘッドとして選択可能なものとして，比較的強い拍節を持つ部分を優先して選択する．

2. (Local Harmony) タイムスパン T のヘッドとして選択可能なものとして，以下のものを優先して選択する．

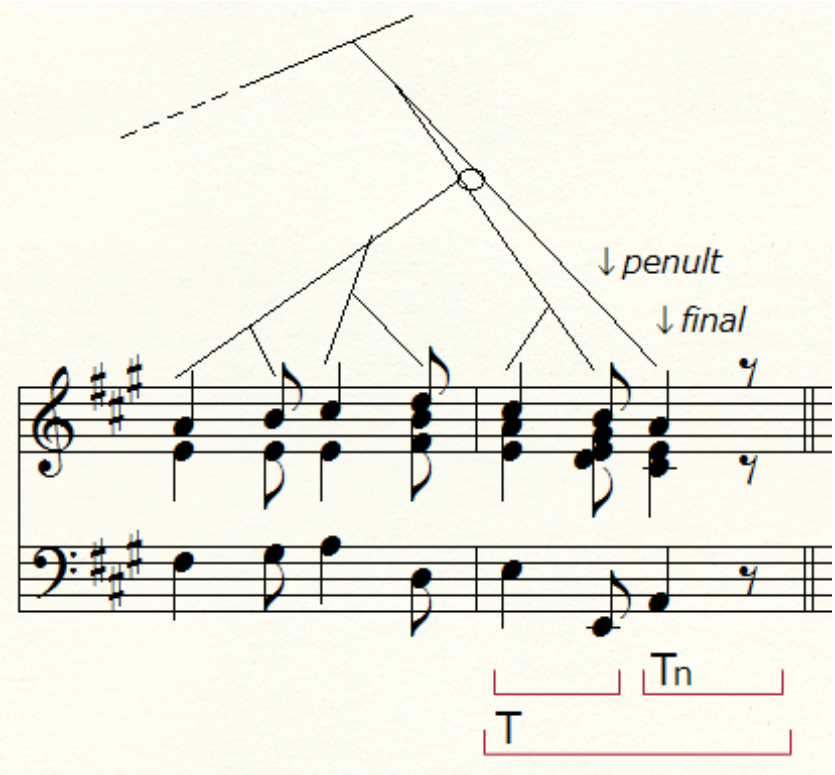


図 2.27 Transformation ルールの例
([2] の p.156，図 7.14 をもとに作成．Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331. より)

a. 比較的本質的に協和するもの

b. ローカルなトニックに比較的密に関係するもの

3. (Registral Extremes) タイムスパン T のヘッドとして選択可能なものとして，以下のものを若干優先して選択する．

   a. より高いメロディの音程

   b. より低いバスの音程

4. (Parallelism) 二個以上のタイムスパンがモチーフとして解釈されるかリズム的に並行であるか，もしくはその両方が満たされる場合，それらのタイムスパンには並行なヘッドを好んで割り当てる．

5. (Metrical Stability) タイムスパン T のヘッドとして選択する際，より安定した拍節構造の選択に帰着するものを好んで選択する．

6. (Prolongational Stability) タイムスパン T のヘッドとして選択する際，より安定したプロロンゲーション簡約の選択に帰着するものを好んで選択する．

7. (Cadential Retention) 次のような状態がタイムスパン T に見受けられるならば，終止部と捉えられる部分にラベルを張り，それをヘッドとして選択することを強く優先する．

   i. 完全終止，半終止，偽終止の進行が形作られている $(e_1)e_2$ の二つのイベントの連続，または一つのイベントが存在する．

   ii. この進行の最後の要素が T の終わりに存在するか T の最後に延長されている．

   iii. その進行が構造の終わりとして機能するための T を含むより大きなグループ G が存在する．

8. (Structural Beginning) 構造の始まりとして機能するタイムスパン T のヘッドを含むより大きなグルーピング G が存在するならば，T のヘッドとして T の始まりに最も関係するイベントを好んで選択する．

9. 楽曲のヘッドを選択する際，構造の始まりよりも構造の終わりを選択することを優先する．

TSRPR1 は拍節構造解析におけるドットが多くついている部分を優先するというものである．

また，TSRPR2a は，和声外の音を含まない，より協和する音を優先して選択するものである．TSRPR2b での「ローカルなトニックに関係するものを優先して選択する」ということは，例えば，ヘッドとして選択される可能性のある音として，I (トニック) の手前

にある *V* (ドミナント) と *IV*（サブドミナント）があるようなときには，よりトニックとの関係が深い *V* の音を優先して選択するということである．

TSRPR3 については特に説明は不要であるが，このルールがあまり優先されるルールではないことと，優先される高い音はメロディに，低い音はバスに限定されていることに注意が必要である．

TSRPR4 は，例えば図 2.19 の一小節目と二小節目のように，モチーフとして解釈されるかリズムが同じもの（この例ではその両方）の場合には，並行したヘッドを割り当てるというものである．実際にこの図においても，一小節目と二小節目，さらに五小節目と六小節目で同じようにヘッドを割り当てていることが分かる．

TSRPR5 については図 2.28 を用いて説明を行う．図の a はモーツァルトのピアノソナタの楽曲を付点四分音符のレベルで簡約したものであり，このレベルでの簡約は確実にこれが正しいといえる．しかし，この次のレベルである一小節のレベルで簡約を行うとき，三小節目の解釈の仕方で図の b と c の候補が考えられる．直感的には vi_7 である三小節目の一音目よりも， V^1（*V* の和音の一番下の音をオクターヴ上げたもの）の方が選ばれる方がより自然であると考えられる．この理由として，図の c の簡約方法をとった場合の和声的律動（和音の変化により生じるリズム）と拍節構造のズレ，さらに下降する線形的なバスの移動がなくなることが挙げられる．図の c の簡約方法をとった場合，その和声的律動は図のように，二小節目と三小節目の和音は同じもので，三小節目と四小節目は転回させただけの関係となり，二小節目以降の和声は変わらなくなってしまう．MPR5f にも示されるように，和声的律動がない場合には，その和音の始まりはより強拍にあることが望ましい．これが和声的律動と拍節構造のズレであり，このズレをが無いようにヘッドを選択するということが TSRPR5 の示すところである．また，もう一つの理由として，図の b の簡約方法をとった場合には，バスが二度ずつ線形的に下がっていっているが，c ではそのような規則性がないということも挙げられる．これらの理由により図 2.28 のような例では b のような簡約方法をとることが望ましい．

TSRPR6 はプロロンゲーション簡約を行った後に，フィードバックをしてさらに木構造を安定させるためのルールである．このルールにより，プロロンゲーション簡約とタイムスパン簡約の両方で，解釈の衝突が最小となるようなヘッドの選択をすることになる．

TSRPR7 は，TSRWFR3d および TSRWFR4 とともに本研究の対象とする部分であり，終止部に関するヘッドの選択方法を示したものである．ルール i は Cadential Retention が完全終止と半終止，および偽終止に適用されることを示している．ルール ii は，終止部と判断されるための和音の位置についてのルールである．例えば図 2.29 のような例では，9,10 小節目が *V* の和音（ドミナント），11 小節目が *I* の和音となっており，一見完全終止であるかのように見えるが，その後に *V*/*ii* が存在するために終止には聴こえない．つまり，*I* がグループの最後にない場合には終止部とは判断されない．そのため，ルール ii でこのような条件を付加している．ルール iii は，終止部と判断されるためのグルーピングについてのルールである．これを図 2.30 を用いて解説する．図 2.30 は簡単化のために簡約後の音としてソプラノとバスのみが表記されている．この楽曲は 1〜2 小節目および 5〜6 小

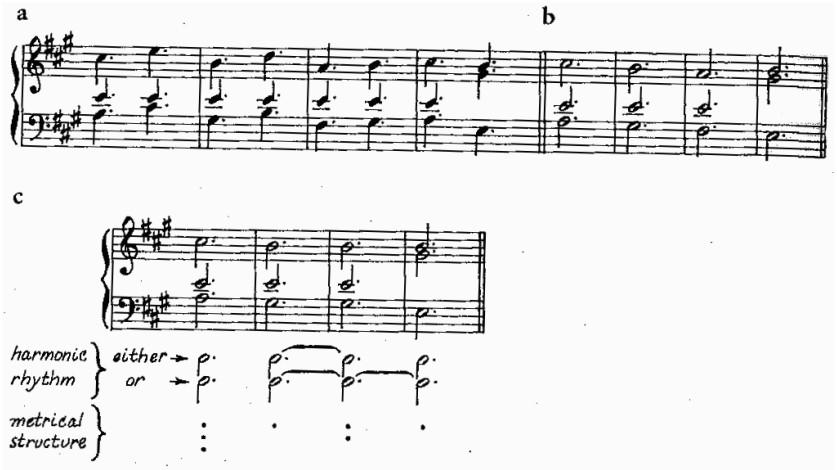


図 2.28 Metrical Stability ルールの説明
([2] の p.164，図 7.20 より転載)

節目が *V*，3〜4 小節目および 7〜8 小節目が *I* となっており，グループの最後に *I* が存在するため，一見どちらも完全終止となるように見える．しかし，実際には 1〜4 小節目は終止部とはならない．この理由として終止部はある程度の構造的な長さが必要であることが挙げられる．具体的には，2 小節のレベルでの簡約を行った際に 1〜4 小節目と 5〜8 小節目の両方が *V* – *I* のみの終止のように見えるグループを持つことになるが，その部分が終止部と判断されるためには，終止部の可能性があるグループの上位のグループにおいても *I* がグループの最後となっている必要があるというものである．このルールにより，1〜4 小節目の最後はこの上位のブルーピングにおける最後の音になっていないので，終止部と判断されなくなり，5〜8 小節目は終止部と判断されるようになる．また，これをさらに簡約した場合の結果は 3 段目の楽譜のようになる．これをルールとして記述したものが TSRPR7(iii) となる．TSRPR7 はこれらをすべて満たすものを終止部と判断し，その部分を *egg* でマークする．また，Cadential Retention を行う際の木構造の作り方について，GTTM では以下のことが述べられている．

- 終止部となる 2 つの音 *penult* および *final* は，1 つのまとまり（ユニット）として機能する．
- 終止部の内部分析は通常のタイムスパン木を作る法則に反し，*V* – *I* の終止の時 *I* は必ず *V* のヘッドとなる．
- *V*，*I* の各和声内の装飾的な音は終止を構成する音に従属する．
- 終止に従属するイベントは，終止の一つの要素にではなく，終止全体に従属する場合がある．

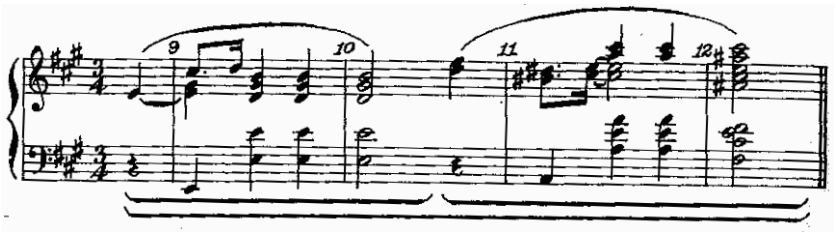


図 2.29 終止部と判断されない例
([2] の p.168，図 7.23 より転載．Chopin, Frdric, Preludes, Op.28, No.7 Prelude in A major. より)

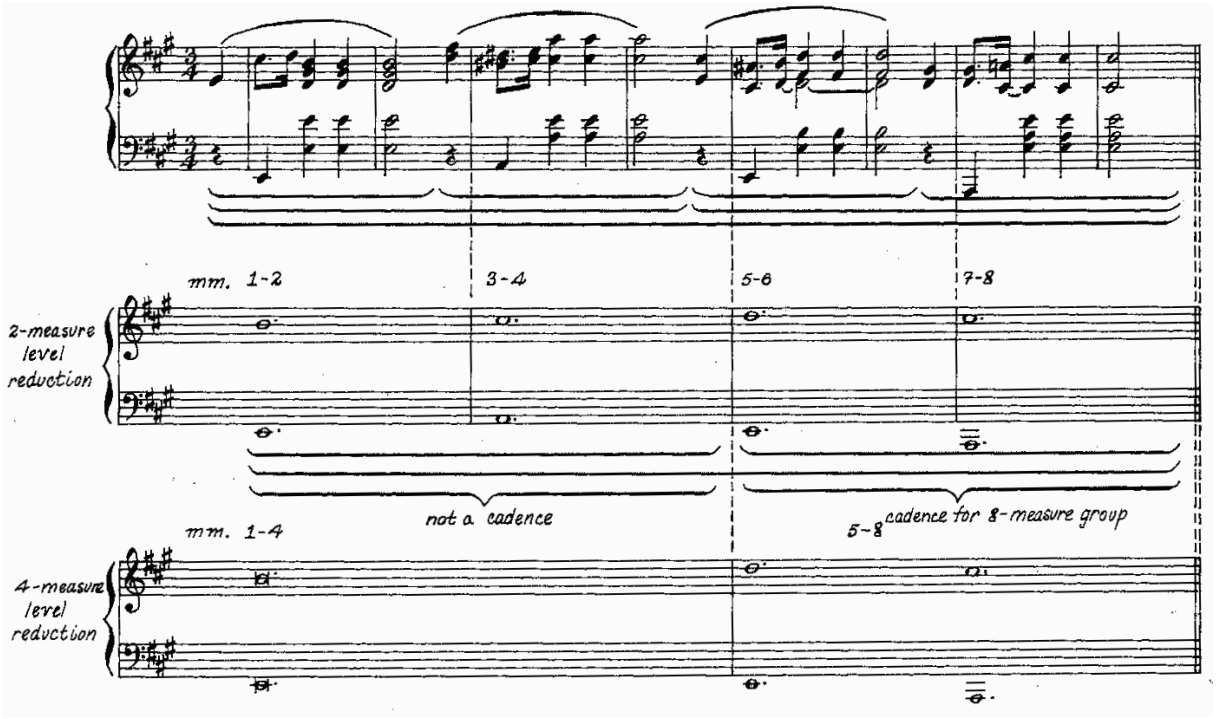


図 2.30 グルーピングによる終止部の判断
([2] の p.167，図 7.24 より転載．Chopin, Frdric, Preludes, Op.28, No.7 Prelude in A major. より)

TSRPR8 も TSRPR7(iii) に類似しており，ヘッドとなりうるタイムスパン T と同じ長さを持つグルーピングよりも大きなグルーピングでもその T がグルーピングの始まりとなる場合には，それをグルーピングのヘッドとして優先して選択するというものである．

TSRPR9 は最後のヘッド（楽曲全体のヘッド）を選択する際には構造の始まりが終わりに従属する形でヘッドを選択するというものである．

2.3 exGTTM

exGTTM [3] は浜中らによって提案された手法であり，GTTM を計算機に適用するために，楽曲の解釈の違いを除いて解析結果が可能な限り一意に決まるように，GTTM の各ルールにパラメータを与え，GTTM のルール記述を具体化したものである．exGTTM では，GTTM の各ルールを実装するために，定性的な記述を定量的に記述しなおすことと，必要なパラメータを補うことを課題としてあげ，パラメータを導入して曖昧さをできる限り排除することと，並列性など詳細な定義がないまま用いられている用語に，直感的でわかりやすい定義を与えるという方針のもと，その解決策を述べている．また，複雑となることを避けるために，GTTM で述べられているホモフォニーではなく，単旋律のみに限定したモノフォニーを対象として扱っている．

exGTTM を実装した ATTA では，musicXML [9] の形式で記述された楽譜を入力として与え，パラメータを手動でコントロールすることで，内部の計算結果により解析結果が変わるという仕様になっている．この際，GTTM の構成要素である Grouping Structure，Metrical Structure，Time-Span Reduction の各解析結果を，それぞれ GroupingXML，MetricalXML，Time-SpanXML という独自の形式で出力する．また，Prolongational Reduction についての理論の拡張は行われていないが，これを記述するためのデータ形式である ProlongationXML は公開されている．

先にも述べたように exGTTM では，Prolongational Reduction の実装が行われておらず，これを実装するためには，和声情報を利用するなどして楽曲の終止部と呼ばれる部分を探索し，現在作成されているタイムスパン木を変更する必要がある．Prolongational Reduction の実装が行われていない原因として，この和声解析の難しさが挙げられると考えられる．

2.3.1 公開データ

exGTTM では，有識者が作成した実験で使用した正解データと現在までに再構築を行った理論を実装したシステムである ATTA がインターネット上で公開されており，そのデータはオープンソースとして誰もが利用可能である．

正解データは，すべて XML のデータ形式で作成されており，グルーピング構造解析を表現するためのデータ形式の GroupingXML，拍節構造解析を表現するためのデータ形式の MetricalXML，タイムスパン簡約を表現するためのデータ形式である TimeSpanXML，

表 2.4 ピッチクラスの値と音名

音名（米国式）	C	$C^{\#}/D^{\flat}$	D	$D^{\#}/E^{\flat}$	E	F	$F^{\#}/G^{\flat}$	G	$G^{\#}/A^{\flat}$	A	$A^{\#}/B^{\flat}$	B
ピッチクラス	0	1	2	3	4	5	6	7	8	9	10	11

プロロンゲーション簡約を表現するためのデータ形式である ProlongationXML，和声を表現するためのデータ形式である HarmonyXML が存在する．

また，これらのデータにおいても，Cadential Retention は完全には行われていない．

本研究では，Cadential Retention を計算機上へ実装するために，どのような処理が必要か，また Cadential Retention 理論がどのように曖昧なのかを検証するために，これらのデータを 3 章で用いている．

2.4 Tonal Pitch Space（TPS）

TPS [4] は 2001 年に F．Lerdal によって提案された音楽理論であり，和声解析を定量的に行うことを目的としている．また，TPS は GTTM を補完するために提案されているが，GTTM とは独立な理論である．この理論では，音程間，和音間，調間に定量的な距離を定め，この距離の和が近ければ近いほどその部分は安定して心地よく響き，距離が長いほど不自然で違和感のある進行であるとしている．

本研究では TPS を用いた和声解析を行ったため，そのために必要となる計算部分について述べることにする．

2.4.1 pitch class（ピッチクラス）

TPS では，ピッチクラスと呼ばれる概念を導入している．ピッチクラスとは新たな音名の概念である．通常の音楽理論では，$C^{\#}4$（ピアノのおおよそ中央にあるド♯の音）と，$D^{\flat}4$（$C^{\#}4$ の異名同音音程）と，$C^{\#}3$（$C^{\#}4$ のオクターヴ下の音）の音を区別することが多いが，TPS ではこれらを区別せず，表 2.4 のように，それぞれの音を 0〜11 の数字で表すとしている．TPS の目的は和声解析であり，和声解析においては，和音を構成する音（構成音）こそが重要であり，オクターブの違いは考慮しない．また，構成音を知る際に，十二平均律（オクターブを 12 等分して音律を作る方法）で各音を考える方が，単純でわかりやすい解析が可能となる．これらの背景から，TPS ではピッチクラスの概念を導入している．

以降，本稿ではピッチクラスの表現には p を頭に付けて，p1，p2 のように表現する．

2.4.2　近親調における和音間距離

TPS では，和音間距離を近親調の場合と遠隔調の場合で分けて算出している．このうち，近親調における和音間距離 δ は，遠隔調における和音間距離 Δ を導出するための基礎となる．近親調における和音間距離は以下の式で表される．

$$\delta(x \to y) = i + j + k \tag{2.1}$$

ここで，x および y は和音を表し，i は調間距離を，j は和音の五度圏距離を，k はベーシックスペース距離をそれぞれ表す．以降，これらのパラメータに関する詳細を述べる．

和音とその表記

式 2.1 における x および y は和音を表している．和音にはバークリーメソッド [10] や和音記号 [6] などいろいろな表記が存在するが，TPS では具体的にどのような表記を行うかは述べていない．しかしながら，本文からその書式は把握できるので，ここでその書式についてまとめることにした．TPS では，和音記号の書式に近い以下のような書式をとっていると考えられる．

- 和音，および調はすべてギリシャ数字で表し，これらの数字は主調の音階に対して何音目に当たるかを表す
- ”和音/調”の形式で表記し，調は太字での表記とする
- 四和音については V^7 や II^7 のように，右肩に 7 をつけて表す
- 長調や長和音は大文字で，短調や短和音は小文字で表記する
- 通常の音階と比べて，半音低いものや半音高いものは，それらの和音記号の左側に♭や#をつけて表現する

この書式を用いて，基調をハ長調としたときの，ハ長調の第一和音からト短調の第五和音の属七への距離を表す $(x \to y)$ は，$(\mathrm{I}/\mathbf{I} \to \mathrm{V}^7/\mathbf{v})$ と表現することができる．以降，本節で和音を表記する際にはこの方法による表記を行う．

調間距離 (regional distance) i

式 2.1 における i は調間距離を表す．これは，以下の r egion circle-of-fifth ルールの最小適用回数により計算される．

> (region circle-of-fifth ルール) ベーシックスペースの level d にあるピッチクラスを level e の上で右か左に動かす．また，この際のベーシックスペースは左右にローテーションする．

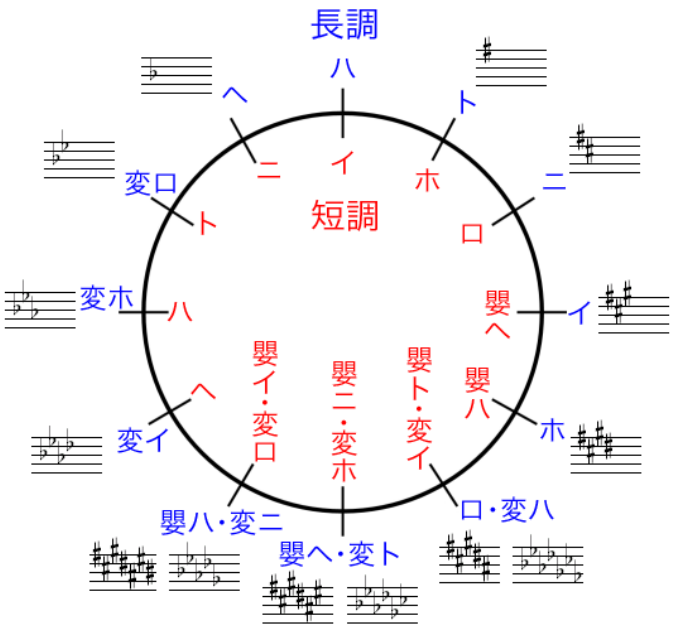


図 2.31 調の五度圏
([11] より)

これは，図 2.31 で示される調の五度圏上で，距離を計算する x の調から y の調へ到達するための最短ステップ数と同義である．例えばハ長調 (**I**) からロ短調 (**vii**) への計算は，ハ長調から右に 2 ステップ移動させるとロ短調になるので，region circle-of-fifth の適用回数は 2 回となり，調間距離 i の値は 2 となる．

和音の五度圏距離 (chordal circle distance) j

式 2.1 における j は和音の五度圏距離を表している．和音の五度圏距離は以下の chordal circle-of-fifth ルールの適用回数により計算される．

> (chordal circle-of-fifth ルール) ベーシックスペースの level a-c にあるピッチクラスを level d の上で数えて右か左に 4 マス動かす．また，この際のベーシックスペースは左右にローテーションする．

これは，図 2.32 で示される和音の五度圏上で，距離を計算する x の和音のルート音から y の和音のルート音へ到達するための最短ステップ数と同義である．例えばハ長調の I である「ドミソ」の和音から，ホ短調の V である「ミソ♯シ」の和音への計算は,「ドミソ」の和音のルート音が p0(ド),「ミソ♯シ」の和音のルート音が p4(ミ) なので，chordal circle-of-fifth の最短適用回数は 3 回となり，調間距離 j の値は 3 となる．

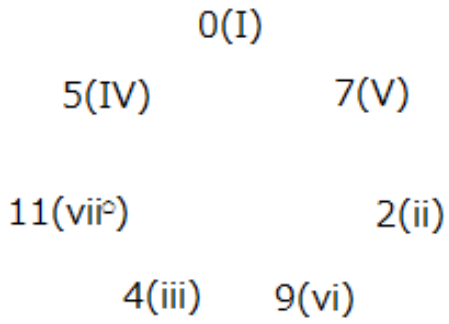


図 2.32 和音の五度圏

ベーシックスペース距離 (basic space distance) k

式 2.1 における k はベーシックスペース距離を表している．ベーシックスペースとは，TPS で定義されている図 2.33 のようなものである．これは，I/**I** のような和音を考えたときに，各音（ピッチクラス）に重要度を表す重みづけを行うようなものである．level a は根音 (root) を，level b は五度音 (fifth space) を，level c は和音の構成音 (triadic space) を，level d はこの調の音階の構成音 (diatonic space) を，level e はすべての音 (chromatic space) をそれぞれ表していて，これらに対応する回数が多いほど重要な音であるといえる．また，四和音について，基本的にクラシック音楽の世界では，これらの音は和声外の音として扱われるので，考慮する必要はない．しかし，これが和声音だと判断される場合に限り，これを triadic space (level c) として扱う．

続いて，ベーシックスペース距離 k の計算方法について述べる．ベーシックスペース距離 k は，和音 x のベーシックスペース内のピッチクラスと比較したときの和音 y 特有のピッチクラスの数である．例として，図 2.33 の (a) から (b)(I/**I** → iv/**vii**) へのベーシックスペース距離を求めながら説明を行う．

図 2.33 の (a) と (b) を比較したとき，(a) のベーシックスペースにない (b) のベーシックスペースのピッチクラスは，p1 が 1 つ，p4 が 2 つ，p6 が 1 つ，p11 が 2 つである（下線部）．これらの数を合計すると，1+2+1+2 で 6 が得られる．これがベーシックスペース距離となる．

2.4.3　遠隔調における和音間距離の導出

ここまでで，近親調における和音間距離 δ の説明は一通り終えたが，この δ の式を遠隔調の関係にある調同士で行うと，それらの値は小さくなってしまい，和声学との乖離が生じてしまう．そこで，遠隔調における和音間距離の計算では以下の別の式 Δ によって計算を行う．この計算では調性空間と呼ばれるものを使用する．

調性空間は近親調の重なりで作られる空間である．TPS において，例えば I/**I** から他調

(a)

level a (root)	0											
level b (fifth)	0							7				
level c (triadic)	0				4			7				
level d (diatonic)	0		2		4	5		7		9		11
level e (chromatic)	0	1	2	3	4	5	6	7	8	9	10	11

(b)

level a (root)					$\underline{4}$							
level b (fifth)					$\underline{4}$							$\underline{11}$
level c (triadic)					4			7				$\underline{11}$
level d (diatonic)		$\underline{1}$	2		4		$\underline{6}$	7		9		11
level e (chromatic)	0	1	2	3	4	5	6	7	8	9	10	11

図 2.33 ハ長調を基調とするベーシックスペースの例:　(a)I/**I**; (b)iv/**vii**

へのトニックへの距離を求め，短い順に並べると近親調が揃うという性質がある．これを属調ならば上に，下属調ならば下に，並行調や同主調は左右に，という風に配置し，発生した新たな調でも同様の操作を繰り返すと図 2.34 のような調性空間が得られる．この調性空間は同じ調が何度も出現しており，それらを重ねると螺旋状のトーラス空間になることがわかる．また，配置の仕方はその性質上長調と短調が左右逆になってしまうことに注意が必要である．

図 2.34 において，矢印で示した部分は，ある調から到達可能な近親調であり，調間距離のはこの調性空間を用いて，以下のような調間距離ルール（式）で決定される．調間距離ルールでは，ある調から移動できる範囲はこの近親調のみであるとし，遠隔調への距離もこの近親調への計算を繰り返すことで到達する距離とするというものである．また，四角で囲んだ部分はピボットと呼ばれる範囲であり，このピボット範囲は同主調に移動する際には青い線を挟んでひっくり返るような性質を持っている．

$$\Delta(\mathbf{I} \rightarrow \mathbf{R}) = [\delta_1(\mathbf{P_1} \rightarrow \mathbf{P_2})] + [\delta_2(\mathbf{P_2} \rightarrow \mathbf{P_3})] + \ldots + [\delta_n(\mathbf{P_n} \rightarrow \mathbf{R})] \tag{2.2}$$

ここで，$\Delta(\mathbf{I} \rightarrow \mathbf{R})$ はある調 **I** から，目標とする調 **R** への距離，$\mathbf{P_1}$～$\mathbf{P_n}$ は各調の近親調である経由調を表す．また，δ は式 2.1 で示される式で，トニック (I，i) を用いてその計算を行う．

なお，このルールでは式 2.2 の値は調性空間での経路の選択方法によって変化し，一意に定まらない．そこで，この経路は距離が最短になるように選択を行うとする．

調間距離は以上のもので計算が可能であるが，これはトニックからトニックへの計算という極めて特殊な例である．そのため，通常の和音間距離の計算は以下の和音/調間距離

c	Es	es	Fis	fis	A	a	C
f	As	gis	H	h	D	d	F
b	Des	cis	E	e	G	g	B
dis	Fis	fis	A	a	C	c	Es
gis	H	h	D	d	F	f	As
cis	E	e	G	g	B	b	Des
fis	A	a	C	c	Es	es	Ges
h	D	d	F	f	As	as	Ces
e	G	g	B	b	Des	des	E
a	C	c	Es	es	Ges	fis	A
d	F	f	As	gis	Ces	h	D
g	B	b	Des	cis	E	e	G

図 2.34 調性空間

ルール（式）を用いて計算される．

$$\Delta(C_1/\mathbf{R_1} \rightarrow C_2/\mathbf{R_2}) = [\delta_1(C_1/\mathbf{R_1} \rightarrow I/\mathbf{P_1})]+[\delta_2(\mathbf{P_1} \rightarrow \mathbf{P_2})]+[\delta_3(\mathbf{P_2} \rightarrow \mathbf{P_3})]+...+[\delta_{n+1}(I/\mathbf{P_n} \rightarrow C_2/\mathbf{R_2})] \tag{2.3}$$

ここで注意が必要なのは，式 2.3 の右辺第一項と最後の項での計算はピボット内のみにしか適用できないということである．つまり，もし，$\Delta(V/\mathbf{I} \rightarrow III/\mathbf{i})$ の計算を行う際には，以下のような計算式を用いる必要がある．

$$\Delta(V/\mathbf{I} \rightarrow III/\mathbf{i}) = [\delta_1(V/\mathbf{I} \rightarrow I/\mathbf{I})] + [\delta_2(\mathbf{I} \rightarrow \mathbf{i})] + [\delta_3(i/\mathbf{i} \rightarrow III/\mathbf{i})] \tag{2.4}$$

これを踏まえたうえで，$\Delta(vi/\mathbf{F} \rightarrow V/\mathbf{h})$ を例として挙げると，以下のような式で計算が可能である．

$$\Delta(vi/\mathbf{F} \rightarrow V/\mathbf{h}) = [\delta_1(vi/\mathbf{F} \rightarrow I/\mathbf{C})] + [\delta_2(\mathbf{C} \rightarrow \mathbf{e})] + [\delta_3(i/\mathbf{e} \rightarrow V/\mathbf{h})] \tag{2.5}$$

さらに，$\Delta(I/\mathbf{C} \rightarrow iv/\mathbf{h})$ のような例では，中間の調間距離ルールを省き，以下のような式で計算が行われる．

$$\Delta(I/\mathbf{C} \rightarrow iv/\mathbf{h}) = [\delta_1(I/\mathbf{C} \rightarrow i/\mathbf{e})] + [\delta_2(i/\mathbf{e} \rightarrow iv/\mathbf{h})] \tag{2.6}$$

2.5　坂本らによる Tonal Pitch Space を用いた楽曲の和声解析

TPS は，一般的な和声学の感覚と近似しており，人間の音楽認知との近似が見られ，また，距離を定量的に計算することができ，いかなる和音間の距離をも計算が可能であり，なおかつシンプルな設計になっている．これらのメリットから，坂本らは [5] で TPS を和声解析に用いた．

具体的な手法としては，大まかには以下のような手順で和声解析が行われる．

1. 入力としてコードネームの配列を受け取る
2. 各コードネームの解釈として可能性のある和音記号をノードとして作成する
3. 各ノードを連結しパスを作成，図 2.35 のようなグラフを生成する
4. ノード間をつなぐパスの距離を TPS によって計算する
5. 楽曲の始まり (S) から終わり (G) までで，最短距離をとるようなノードを選択し，そのノードに記されている和音記号を和声解析結果として出力する

これにより，もっとも自然な解釈と思われる和音記号がコードネームに割り当てられ，和声解析が行われる．

また，坂本らは計算機への実装に際し，以下のような TPS の問題点を挙げている．

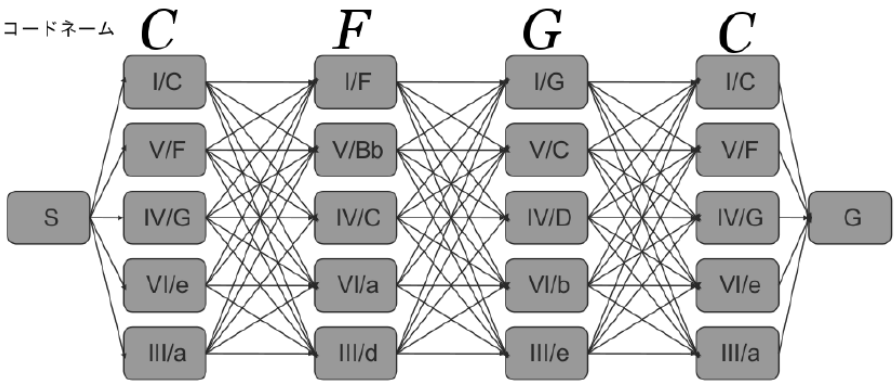


図 2.35 和声解釈のグラフ化
([5] より転載)

- ピッチ・調・和音といった音楽要素の定義が不十分である．特に，調と和音の関係があいまいで，時に混同されているため，ルールの理解が難しくなっている．
- ルール中に示される数式について，一般的ではない表記が用いられている．また，経路の選択など不明瞭な部分がある．$\delta(x \to y)$ や $\Delta(x \to y)$ は経路を参照しており厳密には関数と言えない．
- 計算機上ではより単純な構造により実装できる部分がある．

これらの問題点のために，[5] では，TPS のルールが意図するところを極力そのままに，計算機での実装が容易となるように，TPS の定義に修正を加えている．

本研究では，これらも踏まえたうえで，4 章において，Cadential Retention のための新たな TPS による和声解析の手法を提案する．

第3章　GTTMのデータベースを利用した終止部の探索

2.3節で述べたように，浜中らによって作成されたATTA([3])には，現在Cadential Retention理論が実装されていない．実装が難しい原因として，GTTMで述べられているCadential Retention理論がそもそも曖昧にしか書かれていないことに原因があると考えられる．そこで本章では，まず著者が[12]で明らかにしたCadential Retention理論の曖昧性を示し，その曖昧性を解消する手法を発見するために行った実験の説明をする．

3.1　Cadential Retention理論の曖昧性

Cadential Retention理論を計算機上へ実装することが困難になっている理由は，この理論の曖昧性にある．そのため，まずは現在明らかになっている範囲でこの曖昧性についての考察を行う必要がある．具体的には以下のような曖昧性が考えられる．

曖昧性1 GTTMにおけるTime-Span木の役割が不明瞭である．

曖昧性2 TSRPR7で終止部に対して操作を行うとあるが，これを決める基準は具体的には示されていない．

曖昧性3 *egg*を付けるイベントと付けないイベントの具体的な判断基準がない．

曖昧性4 女性終止の際の木の変更方法や手順が明確に示されていない．

曖昧性1は，GTTMにおける理論の構成についてそもそも曖昧な部分が存在するということを示唆している．GTTMでは楽譜を入力として与えて，そのリズムの情報を用いてグルーピング構造解析と拍節構造解析を行い，それらの情報とCadential Retentionによる和声情報の付加を行うことでタイムスパン簡約が行われる．しかし，本来和声の情報を表すための木はプロロンゲーション簡約で作られるプロロンゲーション木であるはずである．この状態では，タイムスパン木はリズムと和声を表す木となり，その役割が非常に不明瞭に感じられることが問題点として挙げられる．

曖昧性2は，Cadential Retentionを行う上で致命的な問題である．終止部は人間の感覚では認知が可能であるが，そもそも終止感の強さの度合いのようなものが存在し，どの部分を終止部と感じるかは人によってまちまちである．また，GTTMでどの程度の強さの

終止に対し Cadential Retention を行うかが定義されていないために，*egg* の付加をどこに行うのかが曖昧になっていると思われる．

曖昧性 3 は，曖昧性 2 で述べた問題に加え，終止部に従属するときには必ず *egg* をつけるわけではないという問題である．GTTM では，終止部に従属するときに，大きなレベルになると *egg* を付加せず，終止部の *I* のみに従属するような形で木構造が作られている例が示されている．また，終止部の各音に近い音には *egg* の付加が行われず，*penult* と *final* にあたるそれぞれの音に直接従属するような例が示されているが，その判断基準が明確なわけではないという問題がある．後者については，曖昧性 2 の終止部の範囲やヘッドの問題になると考え，曖昧性 2 に含むべきであると判断した．従って，曖昧性 3 は大きなレベルでの *egg* をつける基準についての曖昧性であるとする．

曖昧性 4 は女性終止の問題である．GTTM 上で詳しく述べられているわけではないが，女性終止の際には，通常の木構造を作る方法に従うと，必ず終止部はヘッドとして優先されなくなってしまうという問題がある．この理由として，GTTM では拍節構造の強い点を優先していることが挙げられ，これにより，拍節の弱い部分で終止する女性終止はヘッドとして優先されなくなってしまうという問題がある．

次の節以降では，これらの曖昧性 1 および 2 について，筆者が [12] で明らかにした内容について触れていく．

3.2 Time-Span tree with cadence と Rhythmic Structure tree

曖昧性 1 については，GTTM の構成要素をもう一度見直す必要があると考えられる．タイムスパン木は完全にリズムのみで構成することが好ましく感じられるが，プロロンゲーション簡約を行うためには Cadential Retention による和声情報の付加が不可欠な操作であることもまた事実である．そこで，著者は [12] で新たな木の提案を行った．図 3.1 が著者が考える本来あるべき GTTM の理論構成である．まず，グルーピング構造解析と拍節構造解析の結果のみを用いてリズム構造木 (Rhythmic Structure Tree) を作成する．その後，そこに Cadential Retention による和声情報の付加を行い，和声情報を含むタイムスパン木 (Time-Span Tree with Cadence) を作成し，その解析結果を用いてプロロンゲーション木の作成を行う．この和声情報を含むタイムスパン木は，GTTM で述べられているタイムスパン木と同様のものであり，和声情報を含んでいることを明示するためにこのような名称を付けた．また，この木はあくまでもプロロンゲーション簡約を行うための過程として生まれる木として捉えることが妥当であると考えられる．これにより，各木構造の存在意義が明確になると考えられる．

3.3 終止部の探索

曖昧性 1 の解消のための新たな木の提案の他に，[12] で著者は曖昧性 2 の問題点をさらにはっきりとさせるための終止部の探索実験を行った．実験の前提条件としては以下のこ

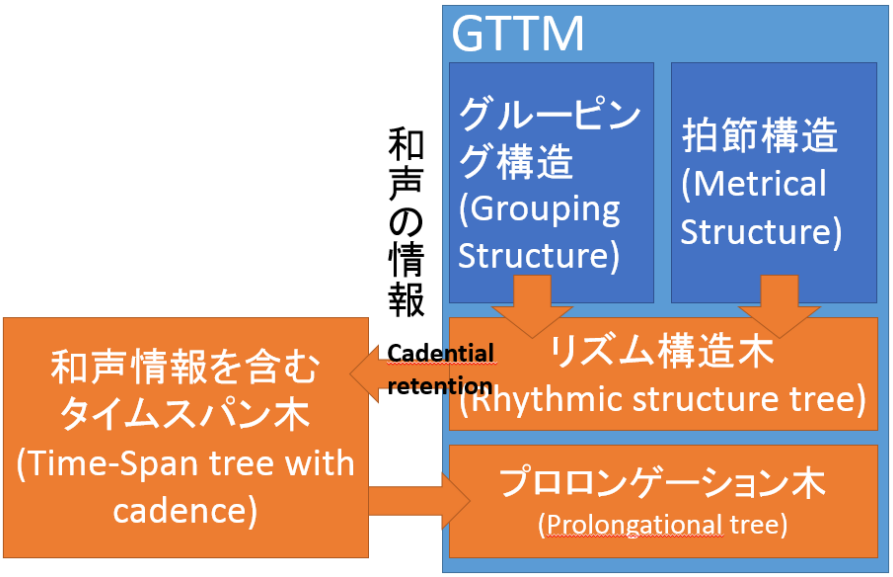


図 3.1 新たな木の提案

とを考えた．

1. $V-I$ もしくは V_7-I の和声に限定して探索を行う．
2. 実験の入力データとして使用するのは [3] で紹介されている GTTM データベースの musicXML，HarmonyXML，TimeSpanXML，GroupingXML の 4 種類のデータとする．
3. 出力は TimeSpanXML と同様もので，終止部のヘッドのみを出力する
4. グルーピング構造解析の結果による 8 小節以上のグルーピングの最後の部分に対して探索を行う．
5. 終止部と思われる部分の I が，探索を行う 8 小節以上のグルーピング内のヘッドでない場合には，その和声を終止と判定しない．

前提条件 1 は，まず，終止で最も重要な全終止の最も基本的な和声パターンに限定することで，基本的な問題点を表面化させるために設定した．

前提条件 2 について，GTTM データベースのデータは Cadential Retention を行う前のデータとなっているため，これを利用し，Cadential Retention のために必要な条件を割り出すために利用した．

前提条件 3 については，GTTM でも述べられているように，全終止の終止は楽曲の大きなまとまりの最後にあるため，そのまとまりの基本的な単位である 8 小節以上という長さの制限を加えた．

前提条件 4 は，次の二つの理由から設定を行った．まず，終止部は曲の終止感を出すために用いるので，終止部の I はリズムにおいても重要な音になるということである．このことから，グループ内でヘッドでない音は終止部ではない可能性が高い．二つ目は終止部と言えるような和声を持つときの木構造は図 3.3 のような人型の木ではなく，入型の木になっているはずであるということである．GTTM では曲の終わりの部分がヘッドとして優先されるので，図 3.3 のような木は曲の終わりを作る終止を行っていないと考えられる．

これらの前提条件のもと，終止部判断のための実験とその考察を行った．

3.3.1 探索方法

行った終止部の探索方法は以下のような手順である．

1. GroupingXML と HarmonyXML より，8 小節以上の長さを持つグループの最後の和声が $V-I$ もしくは V_7-I となっているものを探索し，その範囲の集合を取得する．

2. TimeSpanXML より，人型の木を排除するために 1 で得られた I の中の音で最も優位な音が上記の範囲の音で最も優位でない場合には終止部ではないとみなす．

3. TimeSpanXML より，V の和音と I の和音に属している音の中で，最も優位な音をそれぞれ探索し，それらの音で I が優位になるような二分木を再構成し，TimeSpanXML と同様の形式で出力する．

4. このときの木が終止部を表す木と考える．

手順 1 で図 3.2 のような楽譜があったときに，赤で示したような 8 小節以上のグルーピングを見つけ，その部分の最後が V_7-I や $V-I$ の範囲（図の赤で囲った部分）の音の集合を取得する．次に手順 2 で図 3.3 のような木が排除される．この例においては，8 小節のグルーピング内に，赤線で示した I のヘッドよりも優位な青線で示した音が存在するために終止部ではないと判断されている．手順 3 で，和音 V_7(もしくは V) と I 内の各ヘッドが選択される．このとき，図 3.4 の V_7 内の第一音は装飾的な音として捉えることになる．

3.3.2 探索結果の分析

ここからは，上記の実験で得られた結果の分析を行っていく．

図 3.5 は探索がうまくいったと思われる例である．V に含まれる装飾的な音は排除され，最も重要な音であると考えられる音で終止部の構成ができていることがわかる．

図 3.6 は終止部と同様の形を持つ部分が存在する例である．この例では最後の小節の部分に終止部と思われる部分が存在するが，これと同様の形が 4 小節目にも存在している．これは終止部が 8 小節以上のグルーピングにあると限定したためであり，実際には 4 小節のグループにも存在する可能性がある．このことから，終止部探索に小節数の限定を入れ

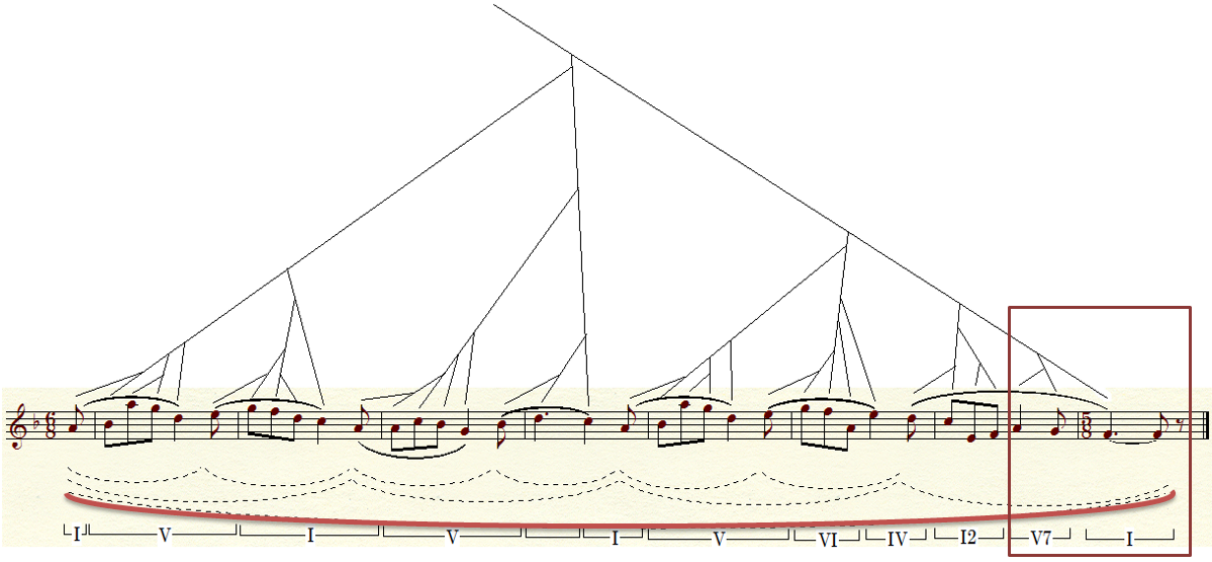


図 3.2 探索手順 1 の説明
(Lange, Gustav, Blumenlied, Op.39. より)

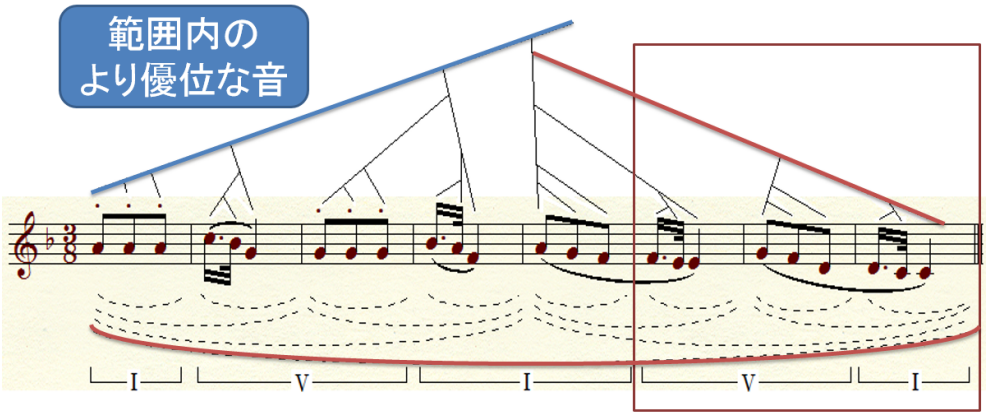


図 3.3 探索手順 2 により除外される木の例
(Verdi, Giuseppe, Rigoletto, Act III, No.17, La donna e mobile. より)

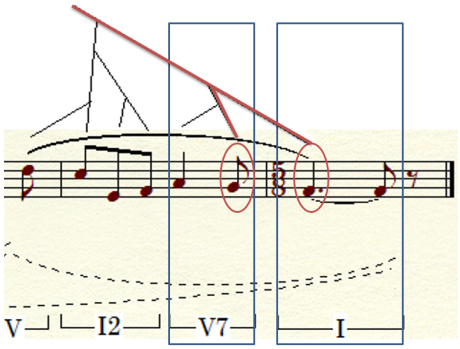


図 3.4 探索手順 3 の説明
(Lange, Gustav, Blumenlied, Op.39. より)

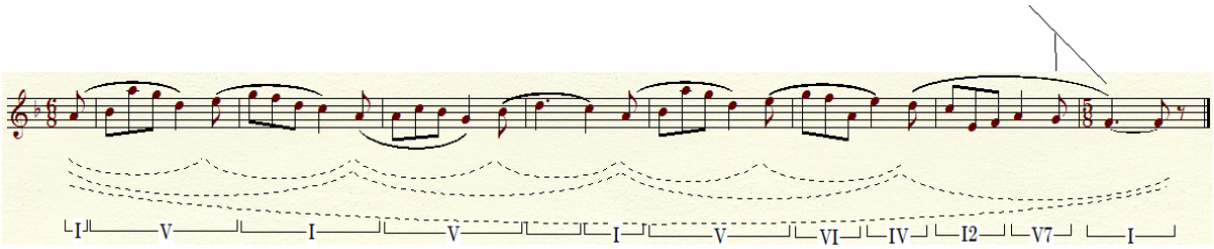


図 3.5 探索がうまくいったと思われる例
(Lange, Gustav, Blumenlied, Op.39. より)

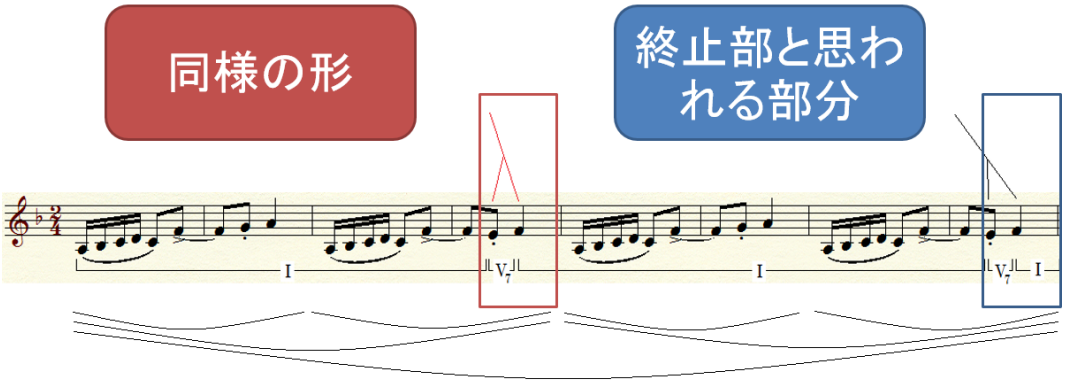


図 3.6 終止部と同様の形を持つ終止部の可能性のある部分
(Albert Ellmenreich, Spinnerlied Op.14 No.4. より)

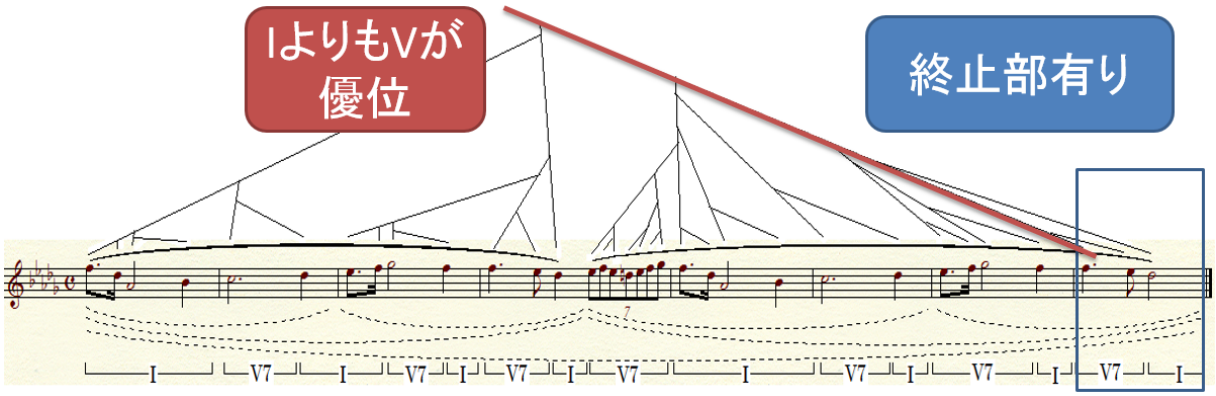


図 3.7 女性終止の例
(Chopin, Frdric, Preludes, Op.28, No.15 Prelude in D♭ major 'Raindrop'. より)

ることは妥当ではなく，純粋に終止部の和声を探索する必要があることがわかる．また，この例では TSRPR7(iii) のような方法で 4 小節目の終止部も探索可能であると思われる．よって，この 4 小節目も GTTM で述べられている終止部であると考えられる．

図 3.7 は女性終止のために終止部があると判断されなかった例である．この例では拍節構造解析の結果により，終止部があるにもかかわらず *I* よりも *V* が優位になってしまっている．このことから，木構造をもとに終止部を発見し，木構造の変更を行うことは難しく，純粋に和声解析による終止部の発見が必要になってくると考えられる．

図 3.8 は探索の結果生成された枝が明らかに間違えている例である．本来ならば，V_7 として取り出すヘッドは破線で示したものが妥当であると考えられるが，探索の範囲内に同じレベルでのヘッドが二つ存在したために，より手前にあったヘッドが選択され，本来とるべき音がとられなくなってしまった．このことから，同じレベルのヘッドの場合にはよ

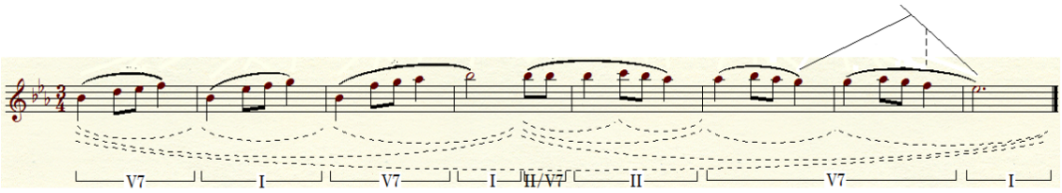


図 3.8 探索の結果生成された枝が明らかに間違えている例
(Chopin, Frdric, Grande valse brillante. より)

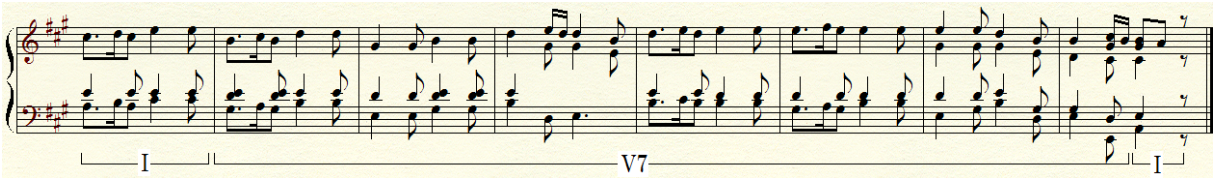


図 3.9 ドミナントの範囲を極端に広げた例
(Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331. を著者が編曲したもの)

り *I* に近いものを選択するのが妥当とも思われるが，それだけでは図 3.9 のような例の場合に，終止部のヘッドよりも優位なヘッドが存在してしまう可能性があり，これは不十分である．このような例ではどの部分が終止部なのか，どの部分に Cadential Retention を適用するべきなのか，その詳細な議論は GTTM ではなされていない．

3.4 追加実験

前節の実験で，以下のような問題点が明らかになった．

1. 終止部がどこになるのかの判断は人の感覚にゆだねられている．
2. 終止部を構成するヘッドとなる音の選択基準が設定されていない．
3. どの範囲を終止部とするかが明確に示されていない．

これらの問題をさらに深く検証するためにまず，TimeSpanXML に *egg* マーカーを付け，上記の方法にいろいろな条件を追加して終止部の探索を行う実験を行った．実験の目的としては，男性終止の終止部の範囲を明確にすることであり，その前提条件として以下の設定を行った．

1. HarmonyXML 内のコードの表記を和音記号ではなく，バークリーメソッドを用いた表記に変更する．

2. バス声部の musixXML および TimeSpanXML データはデータベースのデータをもとに自作する．

前提条件 1 は，終止部の探索には和声解析が必要かもしれないということが分かったため，すでに人の手によって和声解析が行われている HarmonyXML のデータを和声解析が行われていない状態に変更し，その場合の結果を見るためである．前提条件 2 は，上記の問題点 3 を解消する手段として，バス声部を用いるためである．これらの詳しい説明は実験方法の部分で詳細な説明を加えることにする．

3.4.1 *egg* マーカー付き TimeSpanXML

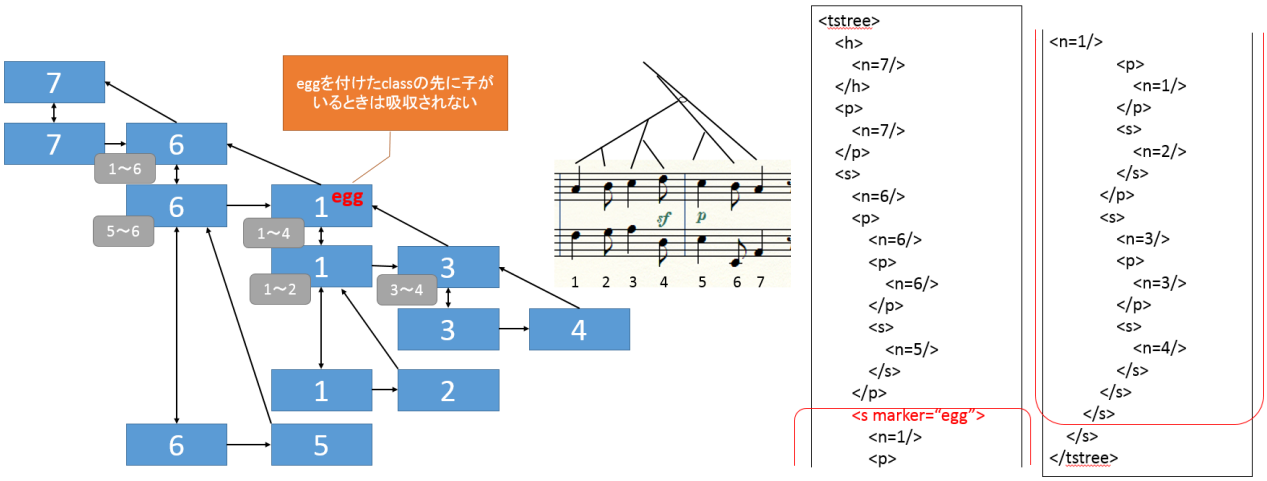
終止部の探索にあたって，終止部に付加する *egg* を TimeSpanXML 中でどのように表現するべきかの検証を行った．

図 3.10(a) は同図の右上に示されている楽譜上のタイムスパン木を連結リストの形式で表現したものであり，青い四角がヘッドのインスタンス，その中の数字が楽譜上の音の番号，そのインスタンスの左下に示している数字がタイムスパンの長さ（タイムスパン内の音の番号）を示している．この連結リストでは，インスタンスを空にすることで簡約を表現することができるようにしている．この方法に従って簡約を行うと，一小節レベルでの簡約を行う際には，通常の場合上から三段目以降のインスタンスは二段目の 6 のインスタンスに吸収されてしまう．しかし，*egg* という表示のあるインスタンスの先に子がいるときにはそのインスタンスは吸収されないという条件を加えると，一小節のレベルでの簡約の結果は上から三段目まで残すことができるようになる．このように，*egg* に従属するヘッドのインスタンスに *egg* のパラメータを付加し，先のような条件を加えることで *egg* の表現が可能となることがわかった．そこで TimeSpanXML に，図 3.10(b) のように，TimeSpanXML の secondary タグに marker=“egg”という属性を付加した *egg* マーカー付き TimeSpanXML を新たに定義することにした．これにより，赤い枠で囲っているように，*egg* に従属している範囲が XML 上でも容易に確認できるようになる。

3.4.2 探索方法

新たに設定した探索方法は以下のようなものである．また，実験の入力データとして使用するのは [3] で紹介されている GTTM データベースの musicXML，HarmonyXML，TimeSpanXML，GroupingXML の 4 種類のデータとし，出力は *egg* マーカー付き TimeSpanXML を用いて木を組み替えて出力した．

1. タイムスパン木をもとに終止部を探索
    a. 入型の木の優位な音のコードとその手前のコードを得る
    b. コード進行が $V - I$ の関係になりうる部分をマーク



(a)TimeSpanXML より得た連結リスト

(b)TimeSpanXML(略記) と *egg* による従属範囲

図 3.10 *egg* マーカー付き TimeSpanXML の *egg* の位置の検証

(Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331. より)

2. 保続音を考慮し終止部の持続範囲 (*egg* のつかない部分) を決定
   a. 簡約のパラメータ R をもとに，バスを簡約
   b. マークした範囲で，*I* と同じ和音を使っているか，バスが *I* の音をとっているか，*I* の音を鳴らした後，次の音が鳴るまで休符になっている範囲を終止部の *I* の範囲とする
   c. マークした範囲で，*V* と同じ和音を使っているか，バスが *V* の音をとっているか，*V* の音を鳴らした後，次の音が鳴るまで休符になっている範囲を終止部の *V* の範囲とする
3. グルーピング構造解析の結果をもとに，終止部のヘッドを探索
4. 表示のために，終止部に属しておらず，終止部に従属している部分に *egg* を追加し，*egg* マーカー付き TimeSpanXML を出力する

まず，図 3.11 に示すように，入型になっている木の音が持つコードをすべて確認して，その進行が $V-I$ の関係になりうるものの範囲の音の集合を得る．ここで言う $V-I$ になりうるものとは，調号に合わせた $V-I$（図の例では $E-A$）のみではなく，とにかく五度の関係になるようなもの（例えば $G-C$ など）はすべて含むということである．これは，人の手によって解析された和声の解析結果を用いずに探索を行うためである．図の例では最後の小節の $E-A$ の部分が集合として取り出される．

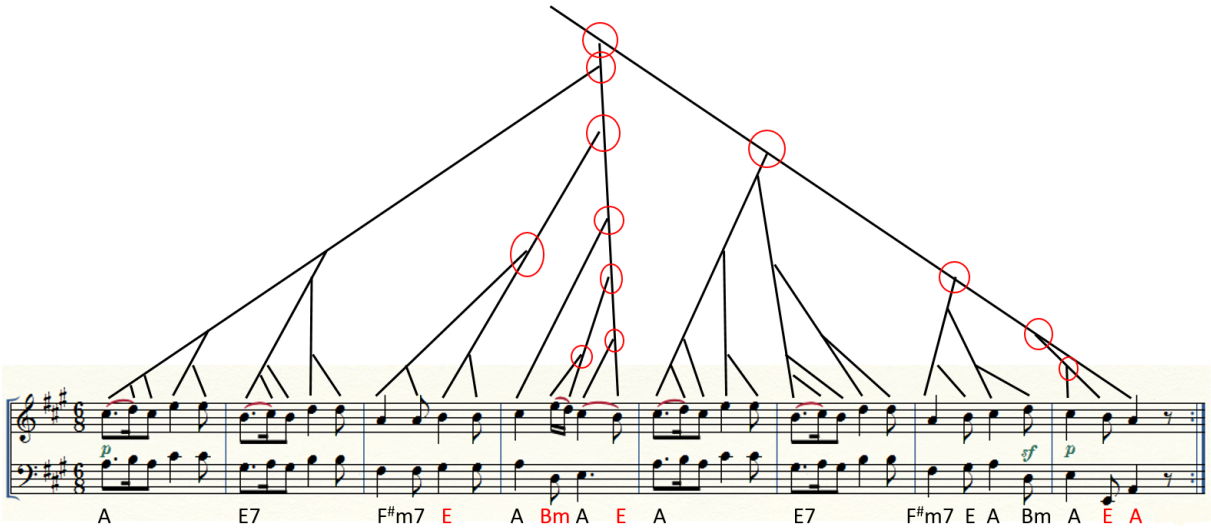


図 3.11 ドミナントの範囲を極端に広げた例
(Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331. より)

次に，保続音を考慮した終止部範囲の決定を行う．図 3.12 は [2] で示されているタイムスパン木の例であり，この終盤に存在する *egg* の部分を見てみると，そのあとに続く部分には *egg* がついていないことがわかる．このことから，バス声部の保続音を考慮して終止部の範囲を特定していることが推察できる．保続音とは，ある一定の音程が連続して鳴らされている状態の音のことである．島岡譲の [7] によると，バス声部に保続音がある場合には，和声構造がどんなに変化しようとも，和音自体の機能は変わらない．つまり，図 3.12 の最後の部分は，同じバスの音を保続しているために，和声が変化しているにも関わらずトニックの機能が変わっていないので，すべてがトニック (I) であると判断され，*egg* が付けられていないのだということがわかる．これらのことから，保続音の範囲を終止部の範囲とすることにし，探索方法 2a〜c で述べた方法をとることにした．また，この範囲のことを終止部の持続範囲と呼ぶことにする．

最後に終止部のヘッドを選択する．これについては，前節の図 3.8 で述べたような問題があるので，ヘッドとなる範囲を終止部の持続範囲とは別に定義し，限定する必要がある．この手法については，以下のようなものを考えた．

- 終止部の可能性のある部分に含まれる 1 番小さなグループを利用する．
- I を含むグループに V も含まれる場合，そのグループのみを探索する．
- I を含むグループに V が含まれない場合，同じレベルのその手前のグループまでを探索する．
- $V - I$ で I の中に複数のグループがある場合は除外する．

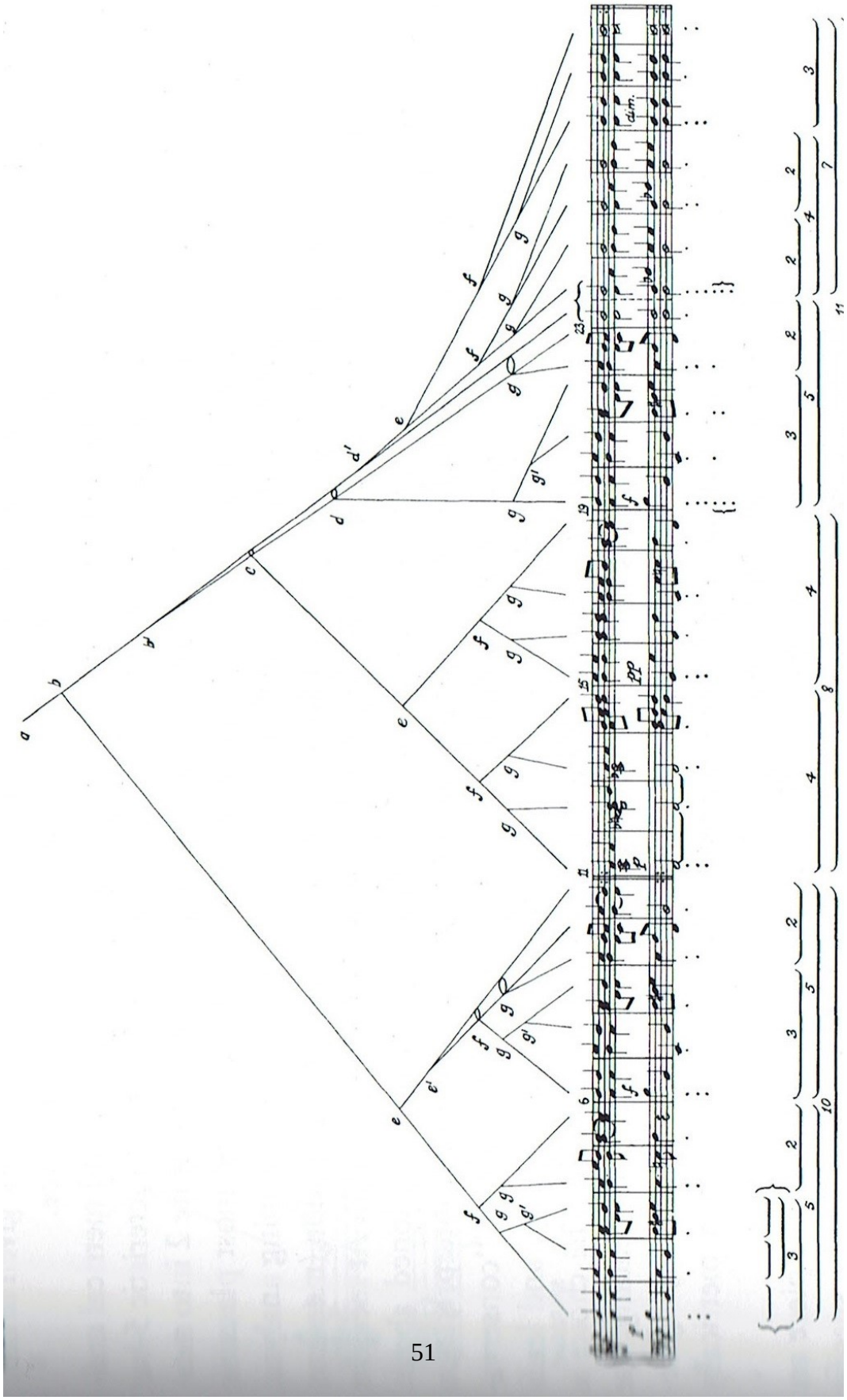


図 3.12 保続音を考慮した終止部の範囲
([2] の p.205 より転載)

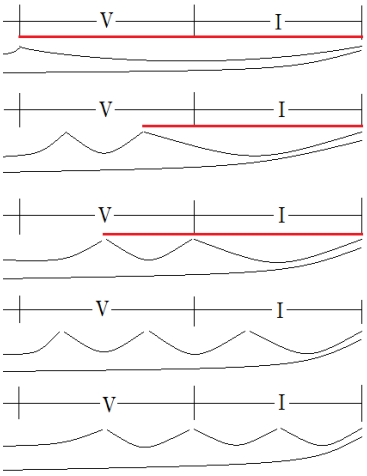


図 3.13 グルーピング構造解析による探索範囲の限定

以上について図 3.13 を用いて説明を行う．方法の一つ目は全体としての方針である．二つ目の方法は図の 1 段目や 2 段目の時の方法であり，この時には赤線で示した範囲内のみからヘッドを選択する．三つ目の方法は図の 3 段目の時の方法であり，このときにも赤線で示した範囲内のみからヘッドを選択する．4 つ目の方法は，図の 4 段目や 5 段目の時の方法である．このような場合には，終止部 $(V - I)$ となる部分が上位グループでもグループの最後になっていないので，終止していないと考えるべきである．そのため，このようなものは終止部ではないとして除外することとした．

また，これだけの条件で探索を行った結果，図 3.14 の上段で示すような解析結果だったものが下段のような解析結果に変わった．しかし，下段に示した四角で囲った音は先行音と呼ばれる和声外の音であり，次に来る音の予備音として，次の音と同じ音を鳴らしているだけの音である．そのため先行音は V の音には含まれず，これを V の音のヘッドと考えることはふさわしくない．また，先行音はその性質上 I と同じグループを作ってしまうため，上記の方法では先行音があったときに正しい音が選ばれなくなってしまう．そこで，先行音に関する対策として，以下のような条件を加えた．

- 以下のような条件に当てはまる場合には，選択される音が先行音であると判断し，その手前のグループのヘッドを終止部の V のヘッドとして選択する．

  a. V の最後の音と I の最初の音が同じ音名 (音程の区別は行わない)

  b. 上記の二つの音が同じ最小グループ内に存在する

これらにより，ヘッドの選択方法を限定することが可能になる．

最後に，これらの方法により得られた範囲やヘッドを用いて，*egg* マーカー付き TimeSpanXML を出力する．

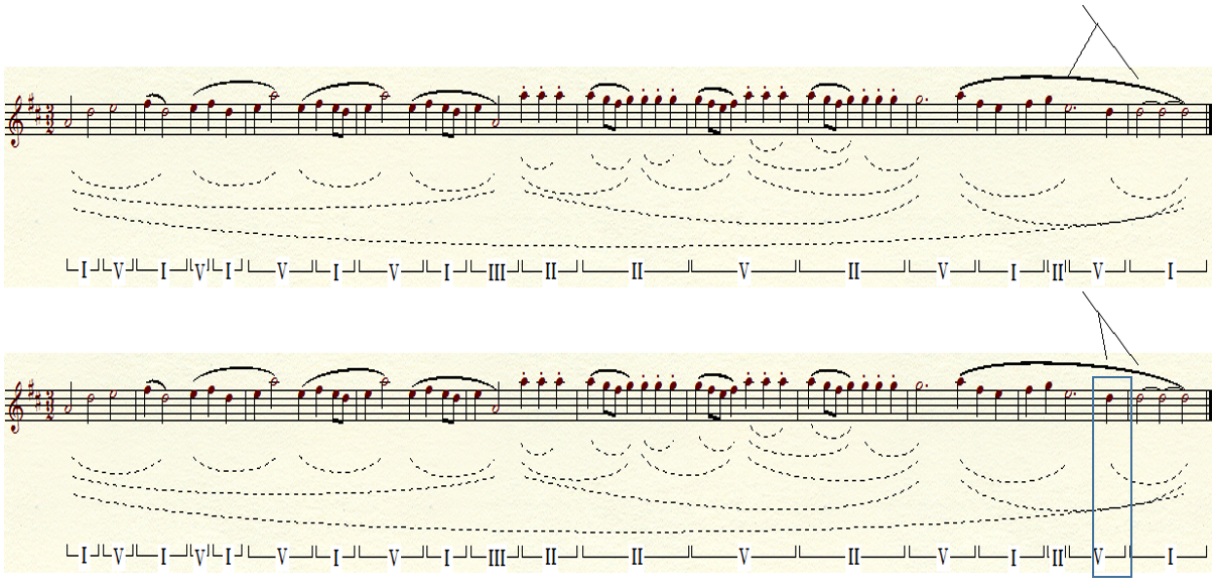


図 3.14 先行音に関する処理の失敗例
(Handel, George Frideric Water Music in D major, HWV 349 No.11 Alla Hornpipe1. より)

3.4.3　探索結果

図 3.15 は，新しい探索のプログラムにより生成された XML データから，タイムスパン木を描いたものである．この結果から，こちらが意図した終止部の範囲とヘッドが選択されていることがわかる．しかし，今回行った実験では，女性終止について，終止部の発見はできなかった．このことから，女性終止の発見には和声の解析が必要不可欠であり，すでに作られた木を変更するのではなく，図 3.16 のように，グルーピング構造解析と拍節構造解析の結果のリズムの情報のみを入力としてリズム構造木を作成し，それらにさらに和声の情報を含めたものを入力として和声情報を含むタイムスパン木を生成する必要があると考えられる．

3.5　まとめ

本章では，Cadential Retention 理論についての曖昧性を示し，その曖昧性を解消する手法を発見するために終止部探索の実験を行った．それらの実験の結果よりわかったことは以下のことである．

1. グルーピングの大きさによる終止部の判定は妥当ではなく，用いるべきではない．
2. 終止部をある程度特定するためには和声解析が不可欠になる．

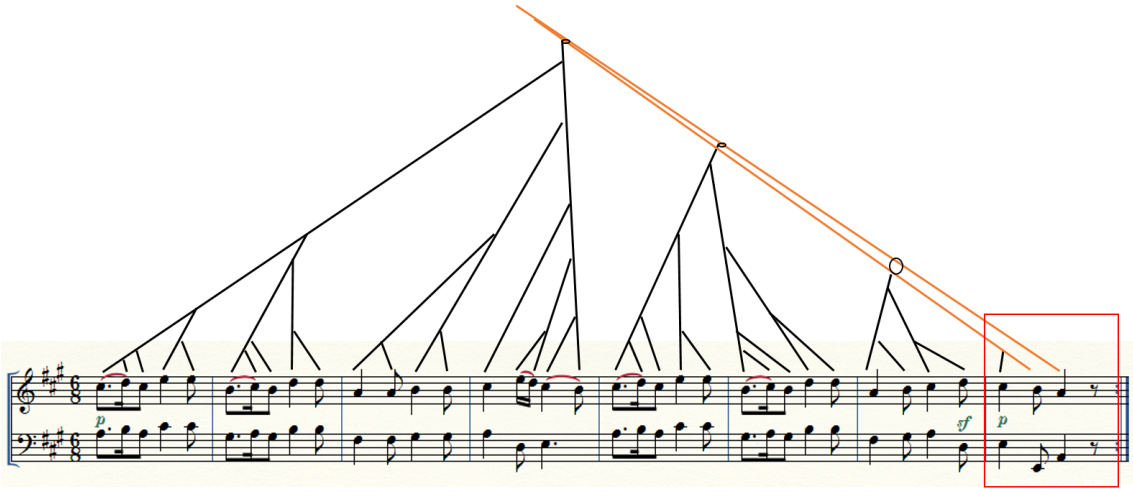


図 3.15 新しい探索方法のプログラムの結果
(Mozart, Wolfgang Amadeus, Piano Sonata No.11 in A major, K.331. より)

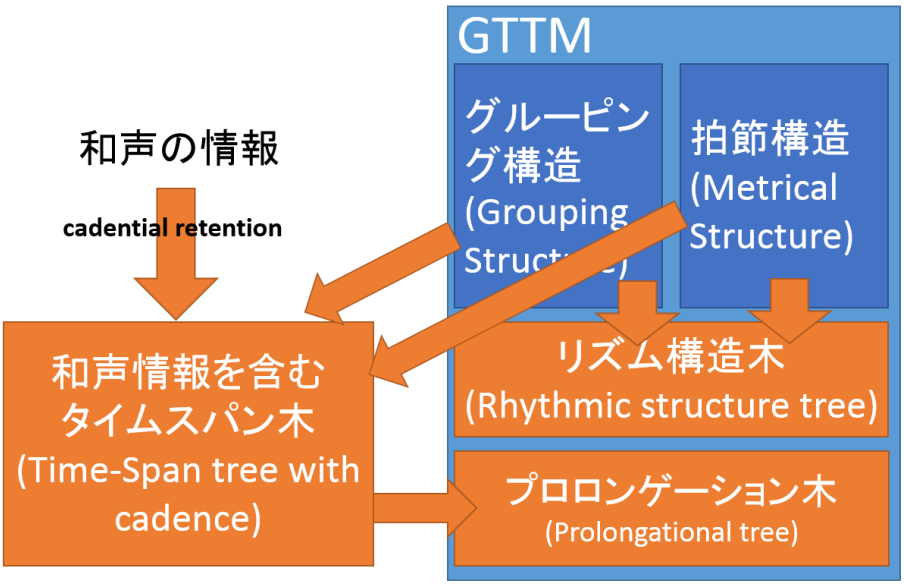


図 3.16 木構造の生成の手順

3. GTTM 内のルールである TSRPR7(iii) の手法が，和声解析を行った部分に対する終止部の同定に大いに役立つ．

4. 女性終止を発見するためには，和声解析とグルーピング構造解析の結果を用いてその部分を特定する必要がある．

5. Cadential Retention における *egg* が付加されない部分である終止部の持続範囲は，バス声部の簡約および保続音を考慮することにより決定可能である．

6. 終止部のヘッドを選択する際に，グルーピング構造解析の結果を用いることが有効である．

7. 終止部のヘッドを選択する際に，上記の方法を用いる際，先行音に対する処理が必要となる．

8. 和声情報を含むタイムスパン木を構成する際には，木構造を操作するのではなく，ヘッド選択のための手法を確立して 1 から木構造を構成する必要がある．

また，本章で行った実験方法の終止部の持続範囲の決定方法や，先行音を考慮したヘッドの選択方法の考え方は，木構造を一から生成する際にも有効であると考えられる．

3.5.1 和声情報の必要性

本章で，Cadential Retention を exGTTM に実装するために重要な項目や，手法が明らかになったが，和声解析を行う必要がある点について解決を行わない限りは Cadential Retention を含めた木構造の生成は困難である．しかし，現在の和声理論は機能和声という立場からみて，機械的に処理できる方法が確立されていない．GTTM における Cadential Retention は半終止，完全終止，偽終止に対して行うため，重要なのはドミナントを発見するような和声解析を行うことである．また，和声解析の必要性は大きなレベルでの Cadential Retention にも非常に重要な項目であると考えられる．

例えば，図 3.17 のような木構造があったときに，バークリーメソッド式のコード表記では中央の G は最後の C と五度の関係であり，そこには Cadential Retention が発生するように感じる．しかし，この *G* の和音が必ずしも *C* の音のドミナントであるかどうかはわからない．もし，この和音が下段に示すような和声解析の結果の場合，この *G* の音は *C* のドミナントではなくなってしまう．このような場合には Cadential Retention を行うのは妥当ではないため，和声解析を行わないと，誤った *egg* の付加を誘発してしまう危険性がある．また，これは半終止に対しても同様のことが言える．半終止は前半の構造の一番最後に作られ，後半の構造の一番最後の *I* の和音に対応する *V* の和音であるため，遠隔の調関係を知る必要があり，和声解析は特に必要になってくると考えられる．

これらのことから，次章ではドミナントを見つけることに特化した，Cadential Retention のための和声解析の手法の実験を行った．

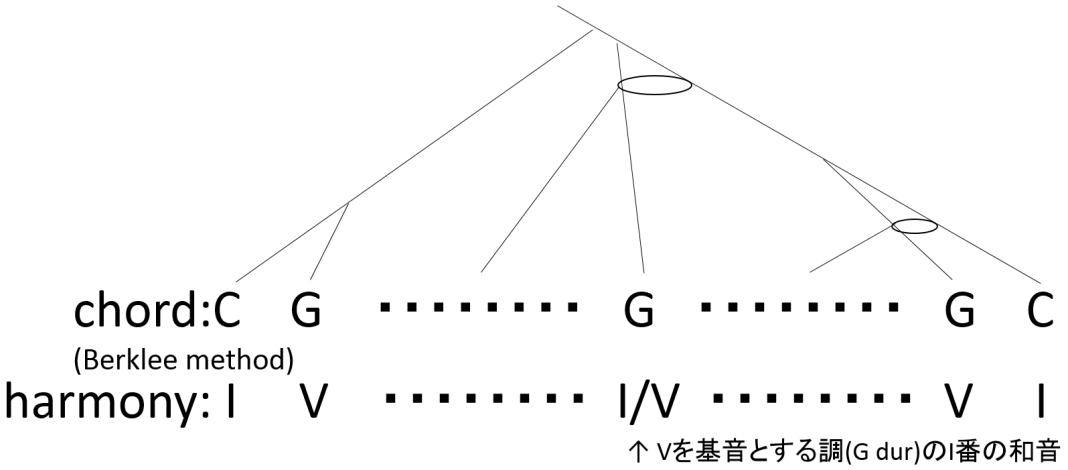


図 3.17 大きなレベルでの和声解析の必要性

第4章　Cadential RetentionのためのTPS理論の再構成と和声解析

本章では，これまでに解説を行った論文や，前章の結果を踏まえたうえで，著者が行ったCadential Retentionのための計算機上で実装可能なTPSによる和声解析について述べる．

4.1　基本方針

3章でも述べたように，Cadential Retentionを計算機上に実装するためには，ドミナントの機能を持つ和音に重きを置いた和声解析が必要となる．一般的な終止部の進行はVの和音が基本になっているが，実際にはこの和音は他のドミナントの役割を持つ和音によって置き換えが可能である．また，楽曲中でしばしば用いられる現在の調性から外れた和音は借用和声と呼ばれるが，坂本らの [5] でも述べられているように，TPSを用いた和声解析時には，その借用和声は短い転調であるとみなすことが妥当であると考えられる．さらに，TPSを用いた際には，ナポリの和音 (♭II) などの特殊な和音でさえも解析が行えてしまうが，本研究ではそのようなものは除外し，ただ単純にドミナントを発見するための手法としてのシステムを提案する．また，TPSを用いた手法では解析が不可能になってしまう部分が存在したため，その部分についても触れる．

これまでに述べてきたことと，これらのことを踏まえたうえで，理論再構成の基本方針を以下に示す.

1. ドミナントを発見するための手法であり，一般的な和声理論とは多少違う部分があるが，基本的には和声理論に準ずる形で理論を再構成する．
2. 上記の方針より，V(ドミナント) の解釈ができる可能性のあるものをVであるとして解析する
3. TPSや [5] では明記されていなかったが，本理論は音階をもとに和声が発生するという考えのもと理論を構成する．
4. コードにはバークリーメソッドを用い，sus4と7sus4以外の基本的な3和音と4和音に限定する．つまり，テンションコード（もとのコードに，ある音を付加したコード）は用いない．また，sus4や7sus4はその前後の関係から，別の基本的な和音に置き換えて解析を行う．

5. 用いる和音は長音階と和声的短音階に限定する．

6. 7th 以上のコードで，ルート音と五度の音に属す音以外は，すべて TPS のベーシックスペースでは level c に属すると仮定する．

7. 和音記号によるコードは調と和音記号により決定されるため，TPS での x や y のようなものではなく，和音記号の変数を x，調を y で表し，距離を測る手前のコードには f，次のコードには t の添え字をつけて x_f や x_t のように表す．

8. 解析するコードは松山祐士の [8] を用いる．

4.2 和声理論との対応

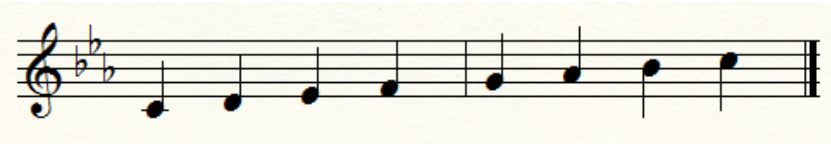
TPS において，ベーシックスペース距離を計算する際には和音によって音階の種類を変更しており，どの音階を用いるかで計算される距離が変わってしまう．このため，本理論では使用する音階の選択を行わなくてもよいこと，そして使用する和音をはっきりとさせることを目的に使用する音階を長調と短調でそれぞれ一種類ずつのみに限定することにした．つまり，扱う和音は調と音階により発生するもののみを使用することになる．

では，次にどの音階を使用するかを決定しなくてはならない．これは，長調については通常の調性音楽で基本とされる長音階を用いることが自然だが，短調については通常使用される音階は三種類存在し，音階を限定しなくてはならない．

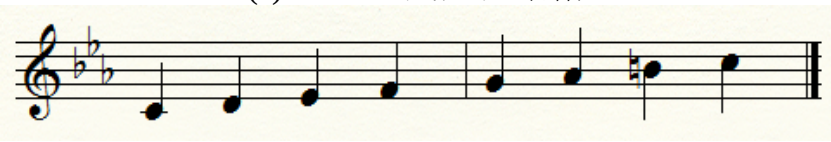
短調の音階には，図 4.1 のような三種類が存在する．このうち (a) 自然的短音階は，長調との違いが表れず，また，終止部で重要になる導音（音階の七音目）が主音（音階の一音目）の半音下になっておらず，これでは和声学における V の表現ができない．つまり，この音階を用いると，短調の楽曲には終止部が存在しなくなってしまう．また，(c) 旋律的短音階は，和声的短音階が歌いにくいために第六音を半音あげたものだが，今回の目的はメロディではなく，和声の解析であるため，この変化は小さな転調であるとみる方がふさわしい．これらを踏まえたうえで，(b) 和声的短音階を見ると，この音階はもともと和声的に正しい導音を短調に設けるために作られた音階であり，これを和声解析に用いるのは非常に妥当であると考えられる．そのため，本研究では（自然的）長音階と和声的短音階を用いて，使用する和音とその解釈を決定する．

図 4.2 は，これらの音階の上に，3 和音および 4 和音を黒玉で並べ，それを和声学での解釈に当てはまるように白玉を付加したものである．また，和声外の音であると判断したものについては×で記した．

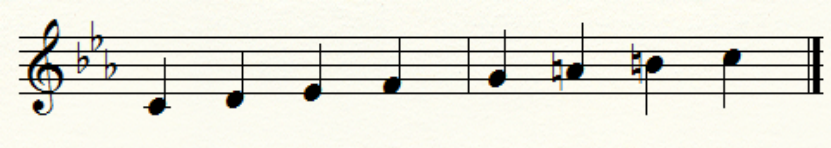
この図を見てみると，赤で囲った部分は和声学には登場しない音であるが，和声学ではドミナントであると解釈される音でもある．そのため，これを$\cancel{\mathrm{V}}$（ルート音が省略された V の和音）と解釈することで和声学との一致をはかり，かつ，終止部のドミナントの可能性がある音として認識するようにした．また，この方法により表れた各コードを見てみると，sus4 と 7sus4 以外のすべての基本的なコード（テンションを含まないコード）が表



(a)c moll の自然的短音階



(b)c moll の和声的短音階



(c)c moll の旋律的短音階 (上昇形のみ)

図 4.1 短調の各音階

れていることがわかる．これもこの方法を用いる利点であり，これにより，バークリーメソッドで表記された和音からそのまま和声解析がおこなえる．

音階を基準として発生する和音を作成し，全調でこれを行うことで，各コードから生成される和音記号のセットの集合が得られる．これを定義するために，まずは，ルート音の集合 (ピッチクラス)$\mathbb{P}$ および，バークリーメソッド式の使用するコードの種類の集合 $\mathbb{B}$，調の集合 $\mathbb{K}$，和音記号の集合 $\mathbb{C}$ を以下のように設定する．

$$
\begin{aligned}
\mathbb{P} &= (p6, p1, p8, p3, p10, p5, p0, p7, p2, p9, p4, p11) \\
\mathbb{B} &= \{M, m, mf5, aug, M7s5, M7, 7th, mM7, m7, m7f5, dim\} \\
\mathbb{K} &= (Ges, Des, As, Es, B, F, C, G, D, A, E, H, \\
&\qquad es, b, f, c, g, d, a, e, h, fis, cis, gis) \\
\mathbb{C} &= \{I, i, ii, iii, IV, iv, V, VI, vi, I7, i7, ii7, iii7, IV7, iv7, V7, VI7, vi7, Vn7, Vn9, Vn11\}
\end{aligned}
\tag{4.1}
$$

ここで，$\mathbb{B}$ における mf5 はバークリーメソッドの m-5 コード，M7s5 は M7+5，m7f5 は m7-5 をそれぞれ表しており，また，$\mathbb{C}$ における Vn7，Vn9，Vn11 はそれぞれ和音記号の $\cancel{\mathrm{V}}_7$，$\cancel{\mathrm{V}}_9$，$\cancel{\mathrm{V}}_{11}$ を表している．これらの和音記号の表記方法はルート音を省略した和音を意味している．調号については，これ以外にも存在し，TPS でもそれらの調の計算が可能であることは明らかになっているが，本来 TPS では，異名同音音程を区別しないようにピッチクラスを導入しているので，異名同音の調についても省くべきと考えた．この方針により，調はピッチクラスと同様の数が長調と短調でそれぞれ存在するので 24 個の調に限定される．

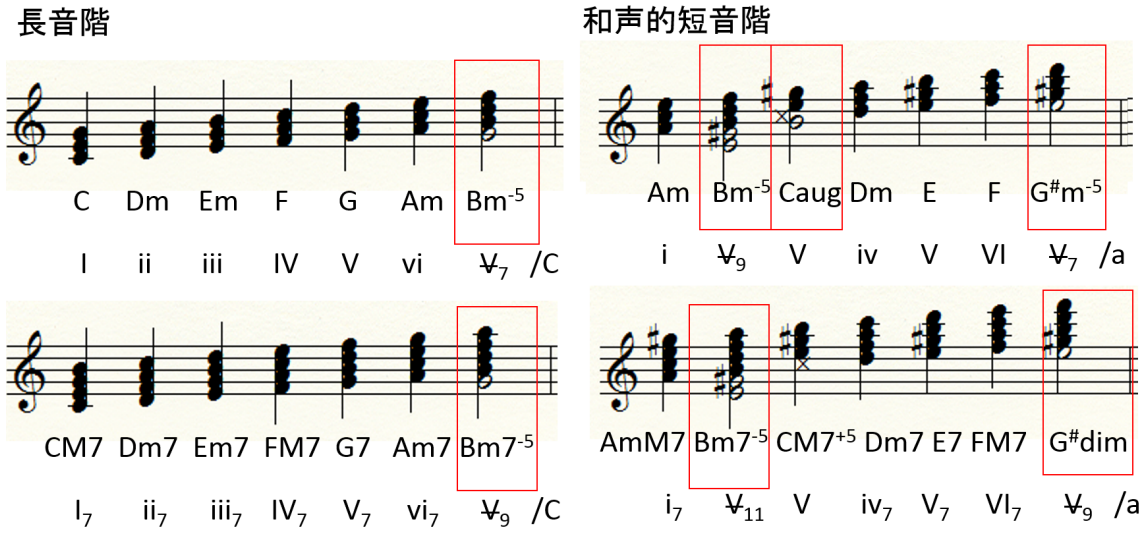


図 4.2 音階上に発生する和音とその和声的解釈

これらの定義で，リストで表現されているものはこの順序に非常に重要な意味がある．このような順序で定義を行うことで，ピッチクラスの変換や調の変換が定数の足し算で表現することが可能になる．例えば，ハ長調 (C) からト長調 (G) への変換はハ長調のインデックス (リストの何番目かを表すもの) に 1 を足し，そのインデックスの要素を参照することで変換可能である．これは，C 言語や C++言語における enum に似ており，これを enum で定義することで，調などの変換が容易に行えるようになる．

以上のことから，計算機上への実装に際し，変換が容易になることがわかるが，この際，インデックスが存在しない要素を参照してはいけないいともいえる．そこで，インデックスが要素数からはみ出してしまった場合の処理（つまり調の循環の表現）として，$\mathbb{Z}$ を整数の集合，$\mathbb{N}$ を自然数の集合として，値 l を 0〜o の範囲内に収めるための関数 $f : \mathbb{Z}\times\mathbb{N} \to \mathbb{N}$ と短調のための同様の関数として，$g : \mathbb{Z} \to \mathbb{Z}$ を以下のように設定する．

$$f(l,o) = \begin{cases} f(l+o,o) & (\text{if } l < 0) \\ l & (\text{if } 0 \le l \le o) \\ f(l-o-1,o) & (\text{otherwise}) \end{cases} \tag{4.2}$$

$$g(l) = \left\{ 12 + l \; mod \; 12 \right. \tag{4.3}$$

ここで，$l \; mod \; 12$ は l を 12 で割った時のあまりを計算する演算子である．

これらを用いて，各コードから生成される和音記号のセットの集合を得る関数 $node$: $\mathbb{P}\times\mathbb{B} \to \mathcal{P}(\mathbb{C}\times\mathbb{K})$ は，以下で表される．

$$
\begin{aligned}
&node(\pi_n(\mathbb{P}), M) = \{\langle I, \pi_{f(n,11)}(\mathbb{K})\rangle, \langle IV, \pi_{h(n+1)}(\mathbb{K})\rangle, \langle V, \pi_{h(n+11)}(\mathbb{K})\rangle, \langle VI, \pi_{g(n+13)}(\mathbb{K})\rangle, \langle \pi_{g(n+20)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), m) = \{\langle iii, \pi_{h(n+8)}(\mathbb{K})\rangle, \langle vi, \pi_{h(n+9)}(\mathbb{K})\rangle, \langle ii, \pi_{h(n+11)}(\mathbb{K})\rangle, \langle i, \pi_{g(n+21)}(\mathbb{K})\rangle, \langle iv, \pi_{g(n+22)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), mf5) = \{\langle Vn7, \pi_{h(n+7)}(\mathbb{K})\rangle, \langle Vn7, \pi_{g(n+16)}(\mathbb{K})\rangle, \langle Vn9, \pi_{g(n+19)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), aug) = \{\langle V, \pi_{g(n+12)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), M7s5) = \{\langle V, \pi_{g(n+12)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), M7) = \{\langle I7, \pi_{f(n,11)}(\mathbb{K})\rangle, \langle IV7, \pi_{h(n+1)}(\mathbb{K})\rangle, \langle VI7, \pi_{g(n+13)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), 7th) = \{\langle V7, \pi_{h(n+11)}(\mathbb{K})\rangle, \langle V7, \pi_{g(n+20)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), mM7) = \{\langle i7, \pi_{g(n+21)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), m7) = \{\langle iii7, \pi_{h(n+8)}(\mathbb{K})\rangle, \langle vi7, \pi_{h(n+9)}(\mathbb{K})\rangle, \langle ii7, \pi_{h(n+10)}(\mathbb{K})\rangle, \langle iv7, \pi_{g(n+22)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), m7f5) = \{\langle Vn9, \pi_{h(n+7)}(\mathbb{K})\rangle, \langle Vn11, \pi_{g(n+19)}(\mathbb{K})\rangle\}\\
&node(\pi_n(\mathbb{P}), dim) = \{\langle Vn9, \pi_{g(n+11)}(\mathbb{K})\rangle, \langle Vn9, \pi_{mair(n+16)}(\mathbb{K})\rangle, \langle Vn9, \pi_{g(n+18)}(\mathbb{K})\rangle, \langle Vn9, \pi_{g(n+22)}(\mathbb{K})\rangle\}
\end{aligned}
\tag{4.4}
$$

ここで，この関数の表記の方法には注意が必要である．先に述べたように，本理論では，要素のインデックスを参照することが非常に多い．この関数もその一つである．具体的に例を示すと，$node(p0, M)$ という引数を入れた場合に，n 番目の要素を得る関数 π_n によって，$p0$ のインデックスである 6 が n に代入される．この 6 に各数 (1,11,13,20) を足し合わせることによって，ピッチクラスの変換が行われ，そのインデックスが表す要素を π_n が取り出す仕組みになっている．以降，このような表現が何度か登場するので，その際にはこの方法を利用しているものとする．

以上のことから，上記の n の範囲は $\mathbb{P}$ の要素数の範囲と同様になり，0〜11 となる．

後に示す実装システムで，この式を用いて各コードに対応するノードの生成を行うことになる．

4.3 TPS 理論の再構成

本節では TPS 理論の再構成を行う．再構成には集合の数式を用いて，[4] で述べられている各式を再定義していく．またその際に，関数名であるとわかりやすくするためうに，名称の変更を行う場合がある．

まず，これまでに定義してきた式に加え，ある調の同主調を返す関数 $para : \mathbb{K} \to \mathbb{K}$ および，ある調の同主調と主調を除く近親調の集合を返す関数 $pipo : \mathbb{K} \to \mathcal{P}(\mathbb{K})$ および，ある調に対応するトニックを返す関数 $toni : \mathbb{K} \to \mathbb{C}$ を以下で定義する．

$$
para(\pi_n(\mathbb{K})) = \begin{cases} \pi_{g(n+21)}(\mathbb{K}) & (\text{if } 0 \le n \le 11) \\ \pi_{h(n-9)}(\mathbb{K}) & (\text{otherwise}) \end{cases} \tag{4.5}
$$

$$pipo(\pi_n(\mathbb{K})) = \begin{cases} \{\pi_{h(n+1)}(\mathbb{K}), \pi_{h(n-1)}(\mathbb{K}), \pi_{g(n+12)}(\mathbb{K}), \pi_{g(n+13)}(\mathbb{K}), \pi_{g(n+11)}(\mathbb{K})\} & (\text{if } 0 \leq n \leq 11) \\ \{\pi_{g(n+1)}(\mathbb{K}), \pi_{g(n-1)}(\mathbb{K}), \pi_{h(n-12)}(\mathbb{K}), \pi_{h(n-11)}(\mathbb{K}), \pi_{h(n-13)}(\mathbb{K})\} & (\text{otherwise}) \end{cases} \tag{4.6}$$

$$toni(\pi_n(\mathbb{K})) = \begin{cases} I & (\text{if } 0 \leq n \leq 11) \\ i & (\text{otherwise}) \end{cases} \tag{4.7}$$

ここで，上記の n の範囲は $\mathbb{K}$ の要素数の範囲と同様であり，0〜23 である．
以降，これらを用いて TPS で示される各関数の定義を行っていく．

4.3.1 和音間距離 Δ

TPS での和音間距離には以下のような問題点がある．

- 同様の距離を計算しているにも関わらず，状況に応じて二種類の関数 (δ および Δ) を使い分けなくてはならない．
- Δ の式については，計算方法が複数あるにも関わらず，その詳細な計算方法の表記が存在せず，計算時に人が判断を行っている状態である．

これらの問題点を解決するために，自然数の集合を $\mathbb{N}$ として，これまで δ および Δ で行っていた計算を一つにまとめ，和音間距離を計算する関数 $\Delta : \mathbb{C} \times \mathbb{K} \times \mathbb{C} \times \mathbb{K} \to \mathbb{N}$ を以下のように定義した．

$$\Delta(x_f, y_f, x_t, y_t) = \begin{cases} \delta(x_f, y_f, x_t, y_t) & (\text{if } F \cap T \neq \emptyset \text{ and } y_t \in F) \\ min\{\delta(x_f, y_f, toni(z), z) + \delta(toni(z), z, x_t, y_t) \mid z \in F \cap T\} & (\text{if } F \cap T \neq \emptyset \text{ and } y_t \notin F) \\ \delta(x_f, y_f, x_t, y_t) & (\text{if } F \cap T = \emptyset \text{ and } y_f = y_t) \\ dic(x_f, y_f, x_t, y_t) & (\text{if } F \cap T = \emptyset \text{ and } y_f \neq y_t) \end{cases} \tag{4.8}$$

ここで，$F = (pipo(y_f) \cup \{para(y_f)\})$，$T = (pipo(y_t) \cup \{para(y_t)\})$ である．また，関数 min は任意の順序集合から最小元を返す関数であり，関数 dic は主調から全調（24 種）への距離を計算しダイクストラ法で最短距離を求める関数である．この dic 関数では，現在の調 w_p から到達できる調は w_p の近親調 w_r のみであるとし，各調への距離の計算は以下の式を用いて行う．

$$
\begin{cases}
\Delta(y_f, w_p, toni(w_r), w_r) & (\text{if } w_p = x_f) \\
\Delta(toni(w_p), w_p, y_t, w_r) & (\text{if } w_r = x_t) \\
\Delta(toni(w_p), w_p, toni(w_r), w_r) & (\text{otherwise})
\end{cases}
\tag{4.9}
$$

ここで，これらを用いた *dic* 関数について，および Δ 関数について説明を行う．

関数 Δ について，その第一式は主調から近親調への計算を表している．また，第三式は距離の計算を行う二つの調が自分自身である場合の計算方法である．第二式は，一つの経由調を挟んで目的の調に到達できる場合の計算方法であり，具体的には図 4.3 の円で囲った調同士の関係にあるような計算の際に用いられる．第四式については，ダイクストラ法を用いる *dic* 関数を用いるが，ダイクストラ法は最短経路を計算するための手法であり，ここの関数における計算のアルゴリズムは以下のようなものである．

1. y_f の調のコストを 0，それ以外の調のコストを大きな値 (例：100 など) に設定する．
2. y_f のコストを確定する．
3. y_f の調（図の **C**）から到達可能な世界 (近親調：**G**，**F**，**a**，**e**，**d**，**c**) への距離を，式 4.9 の第一式を用いて計算し，この値をコストとして各近親調に記録する．
4. コストが確定されていない調の中で最もコストが小さくなるものを選択し，それを確定する．
5. 4 で確定した調のコストが確定されたものを除く近親調に対し，式 4.9 の第二式を用いて計算を行う．その際，到達先の調が y_t と一致する場合には第三式を用いる．
6. 4〜5 を繰り返し，すべての調のコストを確定させる．
7. y_t に記録されているコストを Δ の値として返す．

また，式 4.8 で用いられている $\delta : \mathbb{C} \times \mathbb{K} \times \mathbb{C} \times \mathbb{K} \to \mathbb{N}$ は以下で定義される．

$$
\delta(x_f, y_f, x_t, y_t) = \begin{cases}
reg(y_f, y_t) + cho(x_f, y_f, x_t, y_t) + bs(x_f, y_f, x_t, y_t) \\
\qquad\qquad\qquad\qquad (\text{if } x_t \in pipo(x_f) \text{ or } x_f = x_t) \\
reg(y_f, y_f) + cho(x_f, y_f, toni(y_f), y_f) + bs(x_f, y_f, toni(y_f), y_f) \\
+reg(y_f, y_t) + cho(toni(y_f), y_f, toni(y_t), y_t) + bs(toni(y_f), y_f, toni(y_t), y_t) \\
+reg(y_t, y_t) + cho(x_t, y_t, toni(y_t), y_t) + bs(x_t, y_t, toni(y_t), y_t) \\
\qquad\qquad\qquad\qquad (\text{if } x_t = para(x_t)) \\
100 \qquad\qquad\qquad\qquad (\text{otherwise})
\end{cases}
\tag{4.10}
$$

b	Des	cis	E	e	G	g	B
dis	Fis	fis	A	a	C	c	Es
gis	H	h	D	d	F	f	As
cis	E	e	G	g	B	b	Des
fis	A	a	C	c	Es	es	Ges
h	D	d	F	f	As	as	Ces
e	G	g	B	b	Des	des	E
a	C	c	Es	es	Ges	fis	A
d	F	f	As	gis	Ces	h	D
g	B	b	Des	cis	E	e	G

図 4.3 Δ 関数の第二式を用いる例

これらの式により，これまで人の判断でどの式を用いるかを選択していた TPS での計算式が自動で計算できるようになり，また，Δ のみを用いて距離の計算が行えるようになる．式の三段目は，近親調のみを計算対象にしているため通常到達不可能な選択肢であるが，δ を関数として成り立たせるために大きな値を戻り値として設けているためこのような値になっている．

また，関数 reg，cho，bs は TPS の i，j，k にそれぞれ対応しており，以降でその定義を行う．

4.3.2 調間距離 *reg*

TPS で述べられている調間距離 i と同様の働きをする関数 $reg : \mathbb{K} \times \mathbb{K} \to \mathbb{N}$ は，以下の式で表される．

$$reg(\pi_n(\mathbb{K}), \pi_m(\mathbb{K})) = \begin{cases} 12 - reg'(\pi_n(\mathbb{K}), \pi_m(\mathbb{K})) & (\text{if } reg'(\pi_n(\mathbb{K}), \pi_m(\mathbb{K})) > 6) \\ reg'(\pi_n(\mathbb{K}), \pi_m(\mathbb{K})) & (\text{otherwise}) \end{cases} \tag{4.11}$$

ここで，$reg' : \mathbb{K} \times \mathbb{K} \to \mathbb{N}$ は以下の式である．

$$reg'(\pi_n(\mathbb{K}), \pi_m(\mathbb{K})) = \begin{cases} |n - m| & (\text{if } 0 \leq n \leq 11 \text{ and } 0 \leq m \leq 11) \\ |n - (m - 12)| & (\text{if } 0 \leq n \leq 11 \text{ and } m > 11) \\ |(n - 12) - m| & (\text{if } n > 11 \text{ and } 0 \leq m \leq 11) \\ |n - m| & (\text{if } n > 11 \text{ and } m > 11) \end{cases} \tag{4.12}$$

ここで，上記の n および m の範囲は $\mathbb{K}$ の要素数の範囲と同様であり，0〜23 である．

式 4.12 は，$\mathbb{K}$ を五度の関係で並べたことにより，各要素間の関係がちょうど調の五度圏と同様のものになったためにこのような式で計算が可能である．しかし，このような単純な引き算では最短距離の計算ができない．そこで，reg 関数で大きくなりすぎた距離を調整する処理を入れている．

4.3.3 和音の五度圏距離 *cho*

TPS の和音の五度圏を計算する際，TPS ではピッチクラスを使用して定義をしている．このとき，ハ長調の和音の五度圏を示すと図 2.32 のようになるが，距離を測りたい次の調が，近親調であるヘ長調であったとしても，*IV*/**F** のルート音のピッチクラス p10 はこの和音の五度圏には含まれなくなってしまい，計算を行うことができなくなってしまう．これでは非常にまずいので，このような場合には p11 を選択して計算を行うのが妥当であると考えた．本理論では，この計算を形式的に行うために，まず，以下のダイアトニックを表す集合 $\mathbb{D}$ を用意した．

$$\mathbb{D} = \{0, 1, 2, 3, 4, 5, 6\} \tag{4.13}$$

この$\mathbb{D}$の各音はそれぞれ，ド，ソ，レ，ラ，ミ，シ，ファを表している．また，和音番号のルート音を変ト長調を基準にして，相対的なダイアトニック$\mathbb{D}$に変換する関数$rdia : \mathbb{C} \to \mathbb{D}$を以下のように定義する．

$$\begin{aligned}
&rdia(I) = rdia(i) = rdia(I7) = rdia(i7) = 0 \\
&rdia(ii) = rdia(ii7) = 2 \\
&rdia(iii) = rdia(iii7) = 4 \\
&rdia(IV) = rdia(iv) = rdia(IV7) = rdia(iv7) = 6 \\
&rdia(VI) = rdia(vi) = rdia(VI7) = rdia(vi7) = 3 \\
&rdia(V) = rdia(V7) = rdia(Vn7) = rdia(Vn9) = rdia(Vn11) = 1
\end{aligned} \tag{4.14}$$

さらに，これらを用いて，調との関係から絶対的なダイアトニックの値を得る関数$dia : \mathbb{C} \times \mathbb{K} \to \mathbb{D}$を以下のように定義する．

$$dia(x, \pi_n(\mathbb{K})) = \begin{cases} f(n + root(x), 6) & (\text{if } n \leq 11) \\ f(n - 2 + root(x), 6) & (\text{otherwise}) \end{cases} \tag{4.15}$$

ここで，上記のnの範囲は$\mathbb{K}$の要素数の範囲と同様であり，0〜23である．

ここまでで，絶対的なダイアトニックの音が表現可能となったので，その距離であり，和音の五度圏距離jと同様の値となる$cho : \mathbb{C} \times \mathbb{K} \times \mathbb{C} \times \mathbb{K} \to \mathbb{N}$を以下で定義する．

$$cho(x_f, \pi_n(\mathbb{K}), x_t, \pi_m(\mathbb{K})) = \begin{cases} 6 - cho'(x_f, \pi_n(\mathbb{K}), x_t, \pi_m(\mathbb{K})) & (\text{if } cho'(x_f, \pi_n(\mathbb{K}), x_t, \pi_m(\mathbb{K}) > 3) \\ cho'(x_f, \pi_n(\mathbb{K}), x_t, \pi_m(\mathbb{K})) & (\text{otherwise}) \end{cases} \tag{4.16}$$

ここで，$cho' : \mathbb{C} \times \mathbb{K} \times \mathbb{C} \times \mathbb{K} \to \mathbb{N}$は以下の式である．

$$cho'(x_f, \pi_n(\mathbb{K}), x_t, \pi_m(\mathbb{K})) = |dia(x_f, \pi_n(\mathbb{K})) - dia(x_t, \pi_n(\mathbb{K}))| \tag{4.17}$$

ここで，上記のnおよびmの範囲は$\mathbb{K}$の要素数の範囲と同様であり，0〜23である．

このように，各関数を参照しながら和音の五度圏距離が計算される．ここでもregの関数と同様に大きくなりすぎた距離を調整する処理を入れている．

4.3.4 ベーシックスペース距離 (basic space distance) bs

ここからはベーシックスペース距離を定義していく．ベーシックスペースは基本となるベーシックスペースを調ごとにローテーションさせて作られる．そこで，各ベーシックス

ペースを得るために，まず長調におけるベーシックスペースの基本となる関数 $bsma$ と短調におけるベーシックスペースの基本となる関数 $bsmi$ を定義する．これらの関数は $\mathbb{C}$ を得て，12 個の要素からなる自然数 $\mathbb{N}$ の列を返す.

$$
\begin{aligned}
&bsma(I) = (0,3,0,1,0,1,4,0,1,0,2,1)\\
&bsma(I7) = (0,3,0,1,0,2,4,0,1,0,2,1)\\
&bsma(ii) = (0,1,0,3,0,1,1,0,4,0,1,2)\\
&bsma(ii7) = (0,1,0,3,0,1,2,0,4,0,1,2)\\
&bsma(iii) = (0,2,0,1,0,3,1,0,1,0,4,1)\\
&bsma(iii7) = (0,2,0,1,0,3,1,0,2,0,4,1)\\
&bsma(IV) = (0,1,0,2,0,1,3,0,1,0,1,4)\\
&bsma(IV7) = (0,1,0,2,0,1,3,0,1,0,2,4)\\
&bsma(V) = (0,4,0,1,0,2,1,0,3,0,1,1)\\
&bsma(V7) = bsma(Vn7) = (0,4,0,1,0,2,1,0,3,0,1,2)\\
&bsma(Vn9) = (0,4,0,2,0,2,1,0,3,0,1,2)\\
&bsma(vi) = (0,1,0,4,0,1,2,0,1,0,3,1)\\
&bsma(vi7) = (0,2,0,4,0,1,2,0,1,0,3,1)\\
&bsma(i) = bsma(iv) = bsma(VI) = bsma(i7)\\
&\quad = bsma(iv7) = bsma(VI7) = bsma(Vn11) = zero
\end{aligned}
\tag{4.18}
$$

$$
\begin{aligned}
&bsmi(i) = (0,0,1,4,0,1,2,0,1,0,3,1)\\
&bsmi(i7) = (0,0,2,4,0,1,2,0,1,0,3,1)\\
&bsmi(iv) = (0,0,1,3,0,1,1,0,4,0,1,2)\\
&bsmi(iv7) = (0,0,1,3,0,1,2,0,4,0,1,2)\\
&bsmi(V) = (0,0,2,1,0,3,1,0,1,0,4,1)\\
&bsmi(V7) = bsma(Vn7) = (0,0,2,1,0,3,1,0,2,0,4,1)\\
&bsmi(Vn9) = (0,0,2,1,0,3,1,0,2,0,4,2)\\
&bsmi(Vn11) = (0,0,2,2,0,3,1,0,2,0,4,2)\\
&bsmi(VI) = (0,0,1,2,0,1,3,0,1,0,1,4)\\
&bsmi(VI7) = (0,0,1,2,0,1,3,0,1,0,2,4)\\
&bsmi(I) = bsmi(ii) = bsmi(iii) = bsmi(IV) = bsmi(vi)\\
&\quad = bsmi(I7) = bsmi(ii7) = bsmi(iii7) = bsmi(IV7) = bsmi(vi7) = zero
\end{aligned}
\tag{4.19}
$$

ここで，$zero$ は 0 を 12 個集めた列である．これを用いて，各調に依存しローテーションさせたベーシックスペースのリストを得る関数 bsl を以下のように定義する.

$$bsl(x, \pi_n(\mathbb{K})) = \begin{cases} (\pi_{f(n+7y,11)}(bsma(x)))_{0 \le y \le 11} & (\text{if } n \le 11) \\ (\pi_{f(n+7(y-12),11)}(bsmi(x)))_{0 \le y \le 11} & (\text{otherwise}) \end{cases} \tag{4.20}$$

これにより，ベーシックスペース距離を得る関数 $bs : \mathbb{C} \times \mathbb{K} \times \mathbb{C} \times \mathbb{K} \to \mathbb{N}$ は，以下のように定義される．

$$bs(x_f, y_f, x_t, y_t) = \sum_{n=0}^{11} \{rem(x_f, y_f, x_t, y_t, n) | n \in 11\} \tag{4.21}$$

ここで，関数 $rem : \mathbb{C} \times \mathbb{K} \times \mathbb{C} \times \mathbb{K} \times \mathbb{N} \to \mathbb{N}$ は，ベーシックスペースのリスト間の距離を測る関数であり，以下で定義される．

$$rem(x_f, y_f, x_t, y_t, n) = \begin{cases} \pi_n(bsl(x_t, y_t)) - \pi_n(bsl(x_f, x_t)) & (\text{if } \pi_n(bsl(x_t, y_t)) - \pi_n(bsl(x_f, x_t)) > 0) \\ 0 & (\text{otherwise}) \end{cases} \tag{4.22}$$

以上が，TPS を数学的に再定義した理論となる．この理論は，全体としては複雑になっているが，それぞれの関数をプログラミング言語を用いて計算機上に実装する際には，それぞれの関数を定義通りに設定していくことで，非常に作成がしやすいものとなっている．しかし，それぞれの式が複雑になりすぎている部分はあるので，今後さらに改良の余地が存在する可能性がある．

4.4 実装システム

実装には C++言語を用いて，TPS に関しては前節のものに即した実装を行った．大まかな構成は図 4.4 のようなものである．

まず，入力部で，図 4.5 のようなデータを入力し，パーサーによって解析を行う．このパーサーでは，図の二段目に示すようにコードに対応する和音記号の集合を *node* 関数を用いて生成し，各ノードを繋ぐ経路の設定を行っている．

次に TPS 部によって，前節で定義した各式を用いて各ノード間の距離を設定する．

最後に出力部で，作成されたグラフの最短経路を探索し，最短経路の和音記号を出力することで和声解析が完了する．

出力部における最短経路の探索は，このような探索で一般的なものと考えられるビタビアルゴリズムを利用した．ビタビアルゴリズムをどのように利用したかについて図 4.6 を用いて簡単に解説する．

基本的には，通常のビタビアルゴリズムと変わらないが，スタートのノード (図の S) とゴールのノード (図の G) から延びる経路のコストを 0 にすることで，ノードの終端と始まりにからのノードを連結し，経路が一本に絞られるよにした．また，グラフは有効グラフであり，各ノードには対応する和音記号 (例：I/**C**) が記録されている．その各コストは関

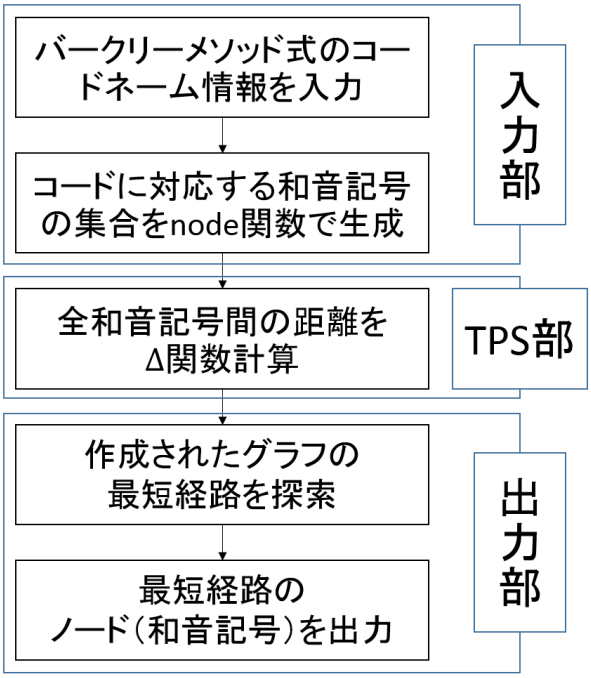


図 4.4 実装システムの構成

```
@これはコメント
@W.A.Mozart
@交響曲40番第一楽章
@p.256
@♭2個(B)
Gm
Am7-5
D7
Gm
Cdim
Gm
Cdim
…
```

図 4.5 入力データの例

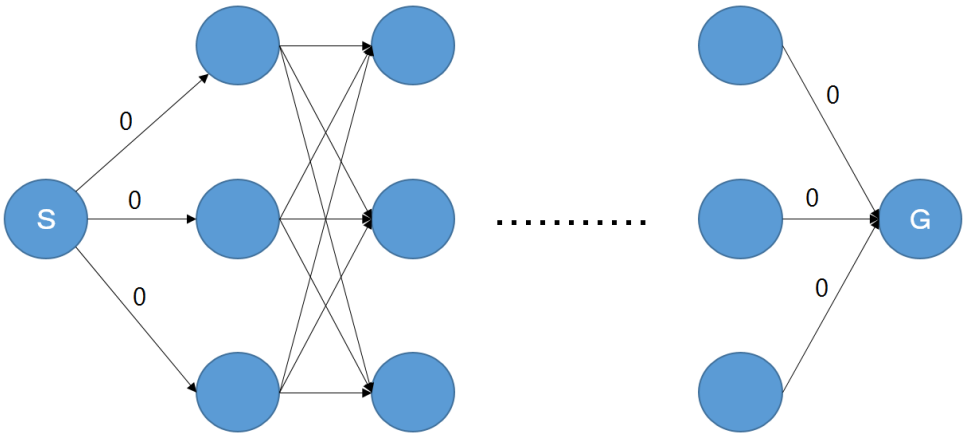


図 4.6 作成されたノードの仮想モデル

数 Δ により計算され，すべてのコストが設定された時点で，終端ノード (G) から，経路の探索が行われる．経路探索の際に，同様のコストを記録するノードが生成された際には，上位（図の上のノード）から順に探索を行う．

このような方法によりビタビアルゴリズムを適用し，最短経路の探索を行った．

また，TPS 部において，計算の効率を上げるために，一度計算を行った値は記録しておき，同じ和音記号同士への計算が行われる際にはそれを使用した．

4.5 実験結果と Cadential Retention への利用

前節で説明を行った実装システムを用いて，[8] からいくつかの曲を選曲して実行を行った．この出力例を図 4.7 に，実行の様子を図 4.8 に示す．

実行時間は 10 曲ほどで数秒程度で，特に問題にはならないと思われる．また，ランダムに選出された楽曲の解析結果は，違和感を感じるものではなかった．今後，これを Cadential Retention に応用する必要があるが，現在はその実装まで完了していないため，図 4.7 で示した解析結果を楽譜に写し，それを検証することにした．

図 4.9 はモーツァルトのト短調のシンフォニーの第一楽章前半部分の解析結果である．終止部の進行と判断されるものについては，その *I* の和音を円で囲んでいる．また，同主調のドミナントから *I* に進行するものについては四角で囲んだ．

解析結果を見てみると，前半では全体的に g moll（ト短調）の和音が多いのに対し，後半では B dur(変ロ長調) の和音が多くなっている．実際に，この楽曲はそのような転調をしている楽曲であるといわれ，これは音楽的直観にあっていることがわかる．

また，例えば 4 段目の 2 小節目から 3 小節目にかけての和音を見てみると，バークリーメソッド式の表記では五度の移動になっており，両者がドミナントとトニックの関係に見えることから，一見すると終止部を作る部分のように見えるが，解析結果では同じ調のド

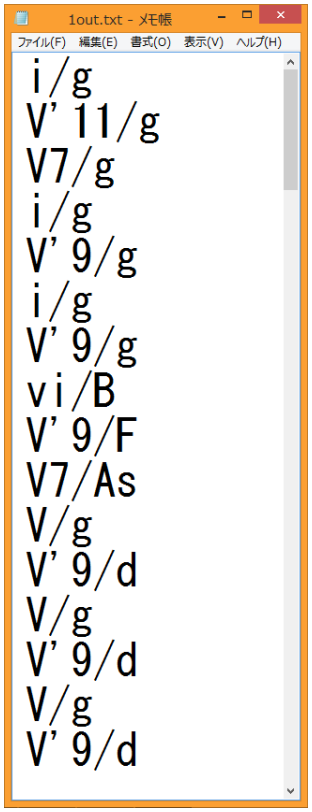


図 4.7 実際の出力例

([8] より． Mozart, Wolfgang Amadeus, Symphony No.40 in G minor, K.550 より)

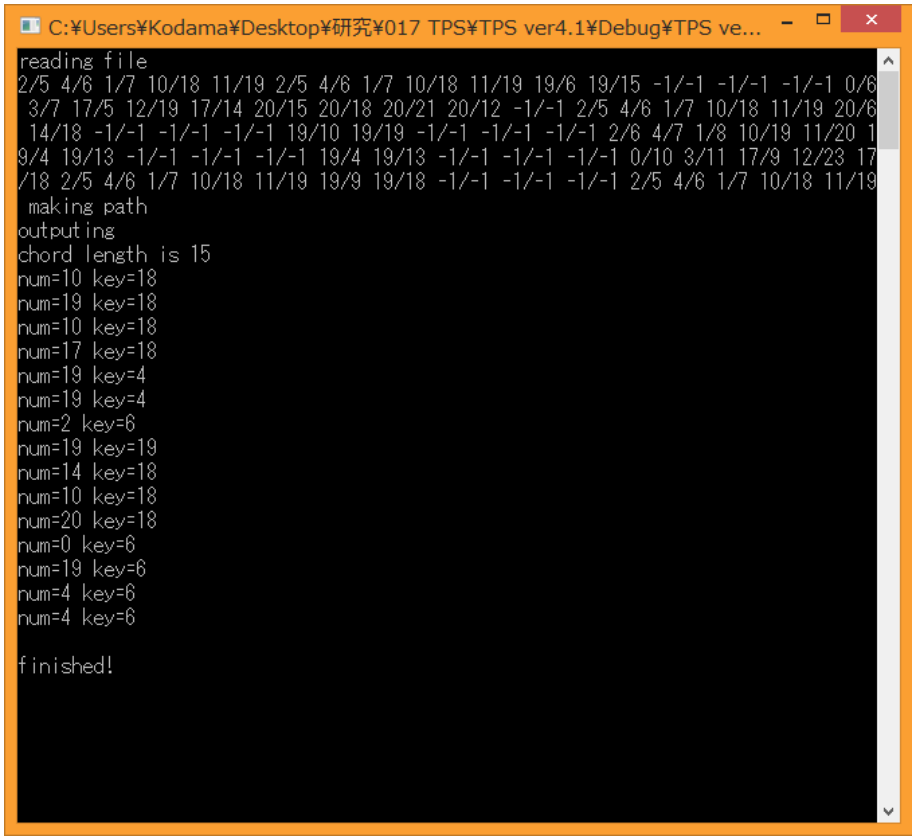


図 4.8 実際の実行の様子

ミナントとトニックの関係にはなっておらず，本システムではこの部分を終止部ではないと判断していることがわかる．実際に楽曲を聴いてみると，この部分には終止感は感じず，これも音楽的直観に即していることがわかる．

また，8 段目の 5 小節目においては終止部と思われる部分が連続しているが，実際に楽曲を聴いてみるとこの部分に終止感は感じない．これは，GTTM の TSRPR7iii で説明が可能であり，この部分のグルーピングを行うと，図 4.10 のようになり，この部分を終止部と判断するためには，上位のグルーピングの結果が不可欠であることがわかる．

これらの結果より，3 章で明らかになったような方法を組み合わせると Cadential Retention の実装が可能になるように思われる．

実際に和声情報を含むタイムスパン木を生成するためには，[3] で述べられているような手法を取り入れ，ヘッドとして選択されるための指標となる重みを設ける必要がある．これらの結果を用いて，どのような重みづけを設定するかは検証の必要があるが，これらの手法を用いて [3] で述べられているような方針で式の設定やモデリングを行えば Cadential Retention の計算機上への実装，つまり，和声情報を含むタイムスパン木の生成の計算機上への実装が可能となる可能性は非常に高いように思われる．

今後の課題としては，以下のようなものが挙げられると考えられる．

- これまでに示した理論によるあらたな GTTM 理論の再構成
- ヘッド選択のための重みづけパラメータの検証
- 大きなレベルでの *egg* 付加の検証
- 多くの楽譜を用いた解析による検証

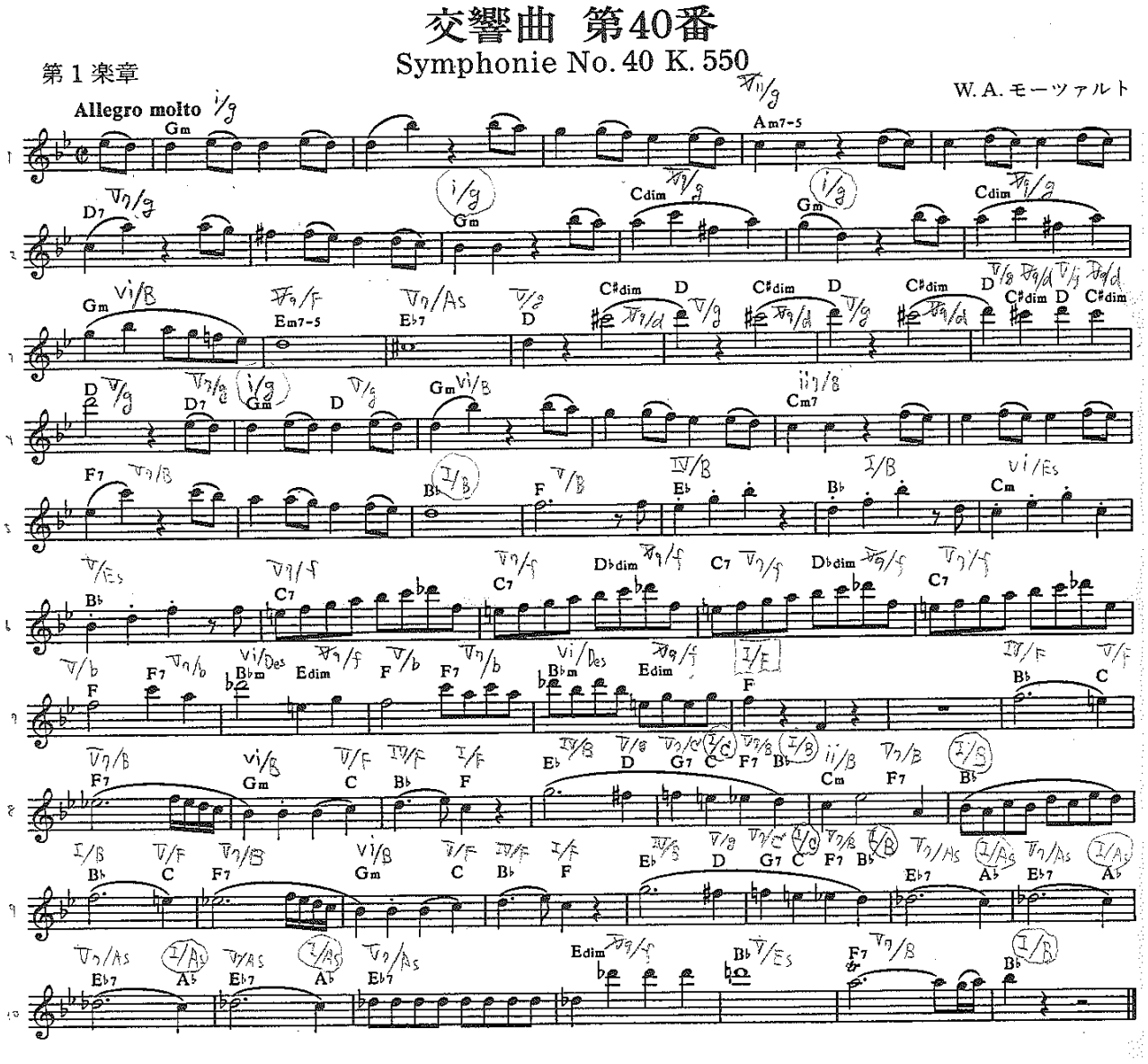


図 4.9 解析結果の例
([8] に書き込みを加えたもの．Mozart, Wolfgang Amadeus, Symphony No.40 in G minor, K.550 より)

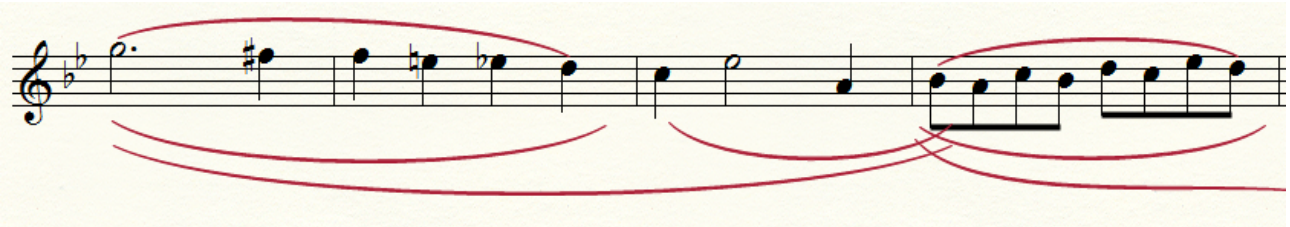


図 4.10 解析結果の部分的なグルーピング
(Mozart, Wolfgang Amadeus, Symphony No.40 in G minor, K.550 より)

第5章　おわりに

本研究では，GTTM に Cadential Retention を実装するための前段階として，Cadential Retention の実装が困難になっている原因や，実装の際に用いることで効果的になる手法を発見するために，Cadential Retention を行う部分である終止部についての探索を行い，その考察を行うこと．また，Cadential Retention の実装の際に必要になる和声解析を実装するために，坂本らのシステムの方針に従い，Cadential Retention を行うための TPS 理論を再構成することを目的としていた．

この目的のために，まず構成論的音楽理論について調査を行い，GTTM のデータベースを用いた実験により，GTTM の Cadential Retention 理論の曖昧性およびそれを解消する手法の発見を行った．Cadential Retention 理論の曖昧性は以下のようなものが考えられた．

曖昧性 1 GTTM における Time-Span 木の役割が不明瞭である．

曖昧性 2 TSRPR7 で終止部に対して操作を行うとあるが，これを決める基準は具体的には示されていない．

曖昧性 3 *egg* を付けるイベントと付けないイベントの具体的な判断基準がない．

曖昧性 4 女性終止の際の木の変更方法や手順が明確に示されていない．

そこで，これらの曖昧性のうち，曖昧性 1 について，GTTM の構成要素をもう一度見直し，リズム構造木 (Rhythmic Structure Tree) の提案を行い，タイムスパン木の性質をわかりやすくするために，和声情報を含むタイムスパン木 (Time-Span Tree with Cadence) という名称への変更を提案した．この和声情報を含むタイムスパン木はあくまでもプロロンゲーション簡約を行うための過程として生まれる木として捉えることが妥当であると考えられる．また，これらの提案により，各木構造の存在意義が明確になると考えられる．

曖昧性 1 の解消のための新たな木の提案の他に，曖昧性 2 の問題点をさらにはっきりとさせるための終止部の探索実験を行った．この実験により，以下のようなことが明らかになった．

1. グルーピングの大きさによる終止部の判定は妥当ではなく，用いるべきではない．
2. 終止部をある程度特定するためには和声解析が不可欠になる．
3. GTTM 内のルールである TSRPR7(iii) の手法が，和声解析を行った部分に対する終止部の同定に大いに役立つ．

4. 女性終止を発見するためには，和声解析とグルーピング構造解析の結果を用いてその部分を特定する必要がある．

5. Cadential Retention における *egg* が付加されない部分である終止部の持続範囲は，バス声部の簡約および保続音を考慮することにより決定可能である．

6. 終止部のヘッドを選択する際に，グルーピング構造解析の結果を用いることが有効である．

7. 終止部のヘッドを選択する際に，上記の方法を用いる際，先行音に対する処理が必要となる．

8. 和声情報を含むタイムスパン木を構成する際には，木構造を操作するのではなく，ヘッド選択のための手法を確立して一から木構造を構成する必要がある．

またこれに加え，Cadential Retention で木構造に付加される *egg* を TimeSpanXML に付加し，*egg* マーカー付き TimeSpanXML を作成した．

さらに，上記の結果を受け，Cadential Retention のための和声解析手法の理論を構成した．GTTM における Cadential Retention は半終止，完全終止，偽終止に対して行うため，重要なのはドミナントを発見するような和声解析を行うことである．また，遠隔の調関係を知る際に和声解析は特に重要になり，そのときに重要視されるのもこのドミナントの発見である．これらのことから，ドミナントの発見に特化した，Cadential Retention のための和声解析の手法を TPS をもとに提案した．

この理論はドミナントを発見するための手法であり，一般的な和声理論とは多少違う部分があるが，基本的には和声理論に準ずる形で TPS 理論を数学的に再構成したものである．この理論では，和声は音階をもとに発生するという考えのもと和声解析を行い，用いる和音は長音階と和声的短音階に依存する和音に限定している．さらに，この理論は全体としては複雑になっているが，プログラミング言語を用いてそれぞれの関数を計算機上に実装する際には，各そ関数を定義通りに設定していくことで，非常に作成がしやすいものとなっている．しかし，それぞれの式が複雑になりすぎている部分はあるので，今後さらに改良の余地が存在する可能性がある．

また，本研究ではこの和声解析手法の計算機上への実装も行い，理論が正しく構築されていることを確認した．この実装システムを用いて，いくつかの曲を解析してみたところ，以下のような Cadential Retention のために有効であると思われる性質が分かった．

1. 音楽的直観で得られる楽譜上には示されない転調を解析によって発見可能である．

2. バークリーメソッド式の表記ではドミナントとトニックの関係に見える和声であっても，解析により終止部ではないという判断をすることができ，また，これは音楽的直観に一致している．

3. 解析結果で一見終止部に見えるような部分であっても，グルーピング構造解析の結果を用いることで終止部ではないという判断が可能である．

今後の課題として，これらの明らかになった手法や実装されたシステムを用いて，[3] で述べられているような手法を取り入れ，ヘッドとして選択されるための指標となる重みを設ける指標を検証する必要があることが挙げられる．さらに，これまでに示した理論によるあらたな GTTM 理論の再構成を行うこと，大きなレベルでの *egg* 付加の検証を行うこと，多くの楽譜を用いた解析による検証を行うことが挙げられる．

参考文献

[1] Bernstein, L., “Unanswered Question”, Lecture II, Musical Syntax (DVD), Norton Lectures, 1973.

[2] Fred Lerdahl, Ray Jackendoff, *A Generative Theory of Tonal Music*, The Massachusetts Institute of Technology, 1983, 368p.

[3] Masatoshi Hamanaka, Keiji Hirata, Satoshi Tojo: *ATTA: Automatic Time-span Tree Analyzer based on Extended GTTM*, Proceedings of the 6th International Conference on Music Information Retrievalconference (ISMIR2005), pp.358-365, September 2005.

[4] F. Ledahl, *Tonal Pitch Space*, New York, Oxford University Press, 2001, 411p.

[5] 坂本鐘期, 東条敏: 「Tonal Pitch Space を用いた楽曲の和声解析」, 情報処理学会研究報告 (IPSJ2009), Vol.2009-MUS-80 No.9, May 2009.

[6] 島岡譲 他, 『和声』, 東京, 株式会社　音楽之友社, 1964, 166p, (理論と実習, *I*).

[7] 島岡譲, 『和声と楽式のアナリーゼ＝バイエルからソナタアルバムまで＝』, 東京, 株式会社　音楽之友社, 1964.

[8] 松山祐士, 『プロフェッショナル・ユース コード付　クラシック・メロディ・ファイル』, 東京, 株式会社ドレミ楽譜出版社, 2001.

[9] makemusic., musicXML, http://www.musicxml.com, 2015 年 1 月 21 日アクセス

[10] Rochel All Rights Reserved., 音楽理論　作曲　DTM 専門音楽学校　Rochel(ロシェル）オンラインミュージックスクール, http://www.rochelm.com/riron.html, 2015 年 2 月 2 日アクセス

[11] Wikipedia(日本語版), 調, http://ja.wikipedia.org/wiki/調, 2014 年 11 月 11 日アクセス

[12] 小玉昂史, 東条敏, 「exGTTM を用いたカデンツ部の探索」, JSAI2014 人工知能学会全国大会（第 28 回）, 1K5-OS-07b-2, 松山, 2014 年 5 月.

謝辞

本研究を進めるにあたり，熱心なご指導・ご助言をいただきました本学　東条敏教授に深く感謝いたします.

また，ご指導をいただきました本学　飯田弘之教授，Nguyen, Minh Le 准教授，白井清昭 准教授，また，同研究室に所属しており，多くの助言をくださった佐野勝彦 助教に深く感謝いたします.

最後になりますが，助言・協力・ご指導をいただきました秦野亮氏，野村尚新氏，新井規広氏をはじめとする東条研究室の皆様に厚くお礼申し上げます．特に，野村尚新氏には式の定義の相談や英語についての相談など大変お世話になりました．この場を借りて深くお礼申しあげます.