

Title	HPSG を用いた楽曲の和声解析
Author(s)	西田, 昌史
Citation	
Issue Date	2003-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1681
Rights	
Description	Supervisor:東条 敏, 情報科学研究科, 修士

修 士 論 文

HPSG を用いた楽曲の和声解析

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

西田 昌史

2003年3月

修 士 論 文

HPSG を用いた楽曲の和声解析

指導教官 東条 敏 教授

審査委員主査 東条 敏 教授
審査委員 烏澤 健太郎 助教授
審査委員 島津 明 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

110098 西田 昌史

提出年月: 2003 年 2 月

概要

本稿では、和音進行解析（カデンツ解析）を中心に据えた楽曲の和声解析手法を提案し、これを用いた楽曲の調・和音・カデンツの総合的な解析について述べる。

本稿で提案する和声解析はまず和音認識を行い、その後に調性認識と和音進行解析を行う。和音進行解析は音楽と自然言語の類似性に着目し、和音を単語、カデンツを句、カデンツのルールを文法規則として捉え、和音列に対して構文解析を行うことでカデンツの構造を解析する。構文解析に必要な文法理論には主辞駆動文法 HPSG (Head-driven Phrase Structure Grammar)[2] を用い、HPSG パーサーは自然言語解析にも利用可能なものを JAVA を用いて実装した。解析に必要な知識には古典西洋音楽の音楽理論である和声学の知識を用いた。解析に際してまず音楽における主辞や補語の概念を定義し、素性構造を用いて和音・和音列・カデンツを表現し、辞書項目に約 600 個の和音の情報を登録した。また、ID-スキーマやプリンスiplは自然言語解析とほぼ同様のものを使用した。

調性認識は調と和音・カデンツとの相関を利用し、本和音進行解析手法を用いてカデンツ構造の解析と同時に進行。またカデンツごとに調を認識することで転調にも対応した。認識する調は長音階、和声的短音階、旋律的短音階の三種類とした。

また本稿では、本和音進行解析手法を用いて和音列をカデンツのルールと照らし合わせることで、和音認識に発生する曖昧性を解消する手法について検討した。これを検証するために簡易的な和音認識システムを二種類用意し、それぞれに発生する曖昧性をどの程度解消することができるか比較実験を行った。本手法で認識する和音は長和音・短和音とそれぞれの七の和音とした。

解析実験は Mozart, Piano Sonata in C, K545 第 1 楽章・第 2 楽章冒頭, Beethoven, Piano Sonata, Op.49, No2 第 1 楽章・第 2 楽章冒頭, Op.49, No1 第 1 楽章冒頭・第 2 楽章冒頭について行った。解析した和音の総数は 203 個、カデンツの総数は 88 個である。解析実験の入力ファイル形式は MIDI ファイルとした。和音認識の正解例の作成は音楽大学の卒業生の方に依頼した。和音進行解析の結果は、システムが出力したカデンツの再現率と、和音認識の精度がどれだけ向上したかで評価した。調性認識の評価は、カデンツの再現率で代用した。これは、本和音進行解析によって認識されたカデンツが正しければその区間で認識された調も正しいことが保証されるためである。また和音認識の結果は、システムが出力した和音の精度で評価した。この評価は本稿で用意した和音認識手法の有効性を検証するためではなく、和音進行解析における曖昧性解消機能を評価するためである。実験によって、和声学に基づく楽曲に対する本和音進行解析・調性認識の有効性が認められた。また、和音認識に発生する曖昧性の解消についてもある程度有効であることが確認された。

目次

第1章	研究背景	1
第2章	和声学	4
2.1	音名	4
2.2	全音と半音	4
2.3	音度差	5
2.4	転回	5
2.5	調	6
2.6	近親調	6
2.7	和音	7
2.8	和音の転回	8
2.9	調と和音の関係と音度表記	9
2.10	カデンツ (cadence) と和音機能	9
第3章	HSPG	11
3.1	主辞の概念と下位範疇化	11
3.1.1	主辞	11
3.1.2	下位範疇化	12
3.2	型階層と単一化	12
3.2.1	型階層	12
3.2.2	型の単一化	13
3.3	素性構造と単一化	13
3.3.1	素性構造	13
3.3.2	素性構造の単一化	14
3.3.3	素性構造による単語・句の表現	16
3.4	文法規則と構文解析	17
3.4.1	ID-スキーマ	17
3.4.2	プリンシプル	18
3.4.3	HPSG による構文解析	18

第4章	和声解析と先行研究	23
4.1	調性認識	23
4.1.1	最も古典的な調性認識手法	23
4.1.2	調性空間を用いた手法	24
4.1.3	適合度を評価する手法	25
4.1.4	確率論的モデル	25
4.1.5	調性認識の問題点	25
4.2	和音認識	26
4.2.1	和音の時間的範囲の特定における問題点	26
4.2.2	和音の構成音の特定における問題点	27
4.2.3	和音名の特定における問題点	27
4.3	和音進行解析	28
4.3.1	基本的なカデンツ解析手法とその問題点	28
4.4	和声解析	29
4.4.1	GTTM	29
4.4.2	和声解析における問題点	30
第5章	提案方式	32
5.1	HPSGを用いた和音進行解析	32
5.1.1	音楽における主辞	32
5.1.2	素性構造による和音の表現	33
5.1.3	素性構造によるカデンツ規則の表現	36
5.1.4	ID-スキーマと Principle	38
5.1.5	カデンツ構造の解析	40
5.1.6	カデンツ規則の拡張	44
5.2	和音進行解析を利用した調性認識	47
5.3	和音進行解析と絡めた和音認識	48
5.3.1	和音認識における曖昧性を低減する手法	48
5.3.2	和音認識のアルゴリズム	49
5.4	和声解析	51
第6章	システム概要	56
6.1	和音進行解析部	56
6.1.1	package : HPSG	58
6.1.2	package : phrase Parser	63
6.1.3	package : Music Parser	63
6.2	和音認識部	65

第7章	評価実験と考察	67
7.1	評価実験	67
7.2	Mozart, Piano Sonata in C, K545	68
7.3	Beethoven, Piano Sonata, Op.49, No2	69
7.4	Beethoven, Piano Sonata, Op.49, No1	71
7.5	考察	72
7.5.1	カデンツ認識の評価と考察	72
7.5.2	調の認識の評価と考察	73
7.5.3	和音の認識精度の考察	74
7.5.4	本和音進行解析手法による和音認識の曖昧性解消機能の評価	74
第8章	まとめ	76

第1章 研究背景

今日、計算機上で音楽を解析または扱うための手法が多数研究されている。これらの研究には作曲・演奏などの音楽活動の補助，音楽知識の習得支援，編曲・伴奏・採譜などの自動化などの幅広い応用が見込まれる。

音楽の解析に関する研究は，生の音源などの音響情報を扱うものと抽象化された音楽情報を扱うものの二つに大別できる。前者は人間の知覚プロセスを紐解く研究であると言える。音響信号をド，レなどの概念情報へ変換する作業や，音源の特定，重なった音の分離，楽音とノイズの識別などのタスクがこれに相当する。これらの研究は，人間と計算機とのインターフェースをより抽象度の低いもの（生の音響情報など）とするために非常に重要である。対して後者はより高次の人間の知的活動の解明を目的とする。抽象化された音楽情報を対象とした楽曲の構造の解析や理解・解釈などのタスクがこれに相当する。これらのタスクは，先に述べたいくつかの応用例を実現するためには必要不可欠である。

後者の研究はさらにリズム解析と和声解析の二種類に分類できる。リズムに関する解析はビートトラッキング，拍節構造解析などと呼ばれ，今日までに多数の研究事例が報告されている。一方，和声解析とは楽曲の和声，すなわち調・和音・カデンツの三者を総合的に解析することを指す。これらの解析はそれぞれ調性認識・和音認識・和音進行解析（カデンツ解析）と呼ばれる。計算機に楽曲の構造を認識させるには，調・和音・カデンツの三者を解析し，リズム解析の結果と組み合わせる必要がある。

上記のうち，和声解析は最も進んでいない分野である。一般的な和声解析ではまず調性認識と和音認識を行い，これらの結果をもとに和音進行解析を行う。ここで問題となるのは各解析の精度である。調性認識については現在までに様々な解析手法が提案されているが，その大半はメロディーに対する解析手法である。各調には近親調が存在し，任意の調とその近親調におけるメロディーは使用される音種が類似することから，このような手法では高い精度で調を認識することはできない。また，和音認識でも和音の範囲の特定や和音構成音と非和音構成音の識別，省略されている和音構成音の特定などの問題の難しさから少なからず曖昧性が生じる。したがって和音進行解析に曖昧性を含む調と和音列が入力され，結果としてカデンツの認識精度も低下することとなる。このため，各解析の精度を上げるための手法を考える必要がある。

調・和音・カデンツの三者は非常に密接な関係にあるため，これらの解析精度を上げるためには解析結果を相互に参照することが最も有効である。しかし，各解析結果を単純に相互参照することはできない。例えば調を決定付ける最大の要因は和音の進行，つまりカデンツであることから，調性認識の曖昧性を解消するにはカデンツの情報を考慮した解

析を行うことが最も有効である．また和音とはカデンツを形成するものであることから，和音認識の曖昧性を解消するためには，カデンツの情報をもとに相応しくない和音を誤認識として排除することが最も有効である．しかし従来の和音進行解析は調と和音の情報が入力として必要なため，事前にカデンツを認識しておくことは不可能である．したがって和音進行解析は曖昧性を含む入力を受け取らざるを得ない．このため，カデンツを解析すると同時に調・和音・カデンツの三者の関係を考慮して曖昧性を解消する和音進行解析手法を考える必要がある．

また，解析に必要な知識の表現方法についても考慮する必要がある．既存の和声解析システムは，制限された環境の下で解析を行うものやある作曲家の楽曲に特化したものが多く，多種多様な音楽に対応することができない．システムは解析対象に応じて知識を選択し入れ替えることが可能で，システム内部では知識と解析を行う部分が完全に切り離されている必要がある．また，解析部と知識とのインターフェースは知識の種類に関係なく単一なものであることが望ましい．

以上の点を踏まえ，本稿では文法理論 HPSG (Head-driven Phrase Structure Grammar)[2] を用いた和音進行解析手法を提案し，これを中心に据えた和声解析手法について論じる．

提案方式ではカデンツ構造の解析に自然言語の解析手法を用いる．音楽研究の場では，以前から音楽と自然言語の類似性・相違性が論じられてきた．音楽と自然言語は必ずしも一対一に対応するものではないが，いくつかの類似性が認められる．この類似性を利用して和音を単語，カデンツを句，カデンツのルールを文法規則に対応付けることができ，自然言語を解析するようにカデンツ構造を解析することが可能である．

カデンツ構造を解析するための文法理論には HPSG を用いる．HPSG は制約ベースの文法理論であり，他の文法理論と比べてルールの数を大幅に削減できるという利点がある．また，HPSG は単語・句などの各カテゴリーの情報や文法規則を素性構造と呼ばれる属性構造によって表現する．この素性構造による知識表現が様々な音楽知識の表現にも非常に有効である．さらに，HPSG は主辞 (Head) の概念を中心に据え下位範疇化と呼ばれる解析プロセスを用いて句の構文を解析する．カデンツを形成する和音列からは自然言語の句と類似した階層構造が見出せるため，この概念と解析プロセスはカデンツ構造の解析にも極めて有効である．

解析に必要な知識には古典西洋音楽の音楽理論である和声学の知識を用いる．これは和声学が最も体系化の進んだ音楽理論であることと，現在でも様々なジャンルの音楽の基本となっているためである．また，HPSG の素性構造を用いた知識表現体系は様々なジャンルの音楽知識を柔軟に表現することが可能であり，同一のルールでそれらを扱うことが可能である．このため，任意の知識に基づいた解析手法を体系化することで，他の知識に基づく解析も同様に行うことが可能となる．

本和声解析手法は和音列から調とカデンツの構造を同時に解析することが可能であり，予め調を認識する必要はない．これは，カデンツを形成する和音列から一意にカデンツ構造を求めることによって，調も一意に求まるためである．また本手法は，曖昧性が含まれる和音列からもカデンツの構造を認識することが可能であり，同時にある程度の曖昧性を

解消することが可能である。これは、カデンツとして相応しい和音列のみからカデンツ構造を見出すことによって、相応しくない和音列を排除することができるためである。

解析対象の音楽ファイル形式には MIDI ファイルを用いる。MIDI ファイルのデータ構造は、ある程度の音楽知識を持つ人間が脳で抽象化した音楽情報に最も近いデータ構造であり、本システムの評価実験に用いる入力情報として理想的である。

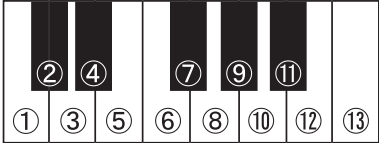
次節ではまず和声学について解説し、続く節では文法理論 HPSG について解説する。続いて和声解析に関連した先行研究とその問題点について述べ、提案手法について述べた後、システムの概要を紹介し、最後に幾つかの実験結果を紹介する。

第2章 和声学

和声学とは、和音の持つ性格や機能に着目しその連結方法について研究した音楽学問であり、現在でも様々なジャンルの音楽の基礎となっている。ここではまず音楽の基礎的な知識について解説し、和声学に基づく調・和音・カデンツの三者とこれらの関係について説明する¹。

2.1 音名

音名とは音の高さを表す固有な名称である。ド、レなどがこれに相当する。音名の英語・イタリア語・日本語による表記とピアノの鍵盤との対応を図 2.1 に示す。本稿では以後、音名は英語で表記する。



	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬
イタリア語	ド	ド♯ レ♭	レ	レ♯ ミ♭	ミ	ファ	ファ♯ ソ♭	ソ	ソ♯ ラ♭	ラ	ラ♯ シ♭	シ	ド
英語	C	C♯ D♭	D	D♯ E♭	E	F	F♯ G♭	G	G♯ A♭	A	A♯ B♭	B	C
日本語	ハ	嬰ハ 変二	ニ	嬰ニ 変ホ	ホ	ヘ	嬰ヘ 変ト	ト	嬰ト 変イ	イ	嬰イ 変ロ	ロ	ハ

図 2.1: 音名と鍵盤の対応

2.2 全音と半音

ピアノの鍵盤上で隣り合うような2音の高さの差を音程差と呼ぶ。図 2.1 中の‘⑤’と‘⑥’のように鍵盤1つ分離れた2音の音程差を半音、‘①’と‘③’のように鍵盤2つ分離れた2音の音程差を全音と呼ぶ。これらの間には、全音 = 半音 × 2 の関係が成り立つ。

¹ここで解説する和声学の知識は、書籍：和声 理論と実習 I [11] に基づくものである

2.3 音度差

和声学では、音程差とは別に音度差と呼ばれるものが定義されている。音度差は、楽譜上の任意の2音が線²または間³いくつ分離れているかを示す。図 2.2 に示すように、同一の線または間上にある2音の音度差を1度、線または間1つ分離れた2音の音度差を2度のように呼ぶ。これらは英語で 1st, 2nd のように表記する。また8度である2音の関係をオクターブと言う。

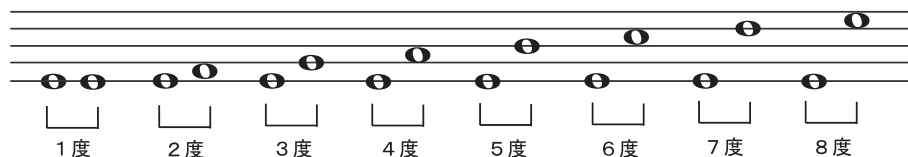


図 2.2: 音度差

また、見た目上同じ音度差でも実際にはその高さの差が異なることがある。例えば、図 2.3 左側の3度の関係にある2音は実際の鍵盤上では半音4つ分離れているが、同図右側の3度の関係にある2音の距離は半音3つ分である。

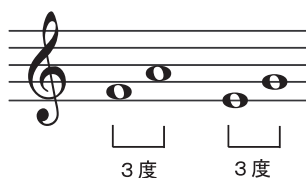


図 2.3: 音度差と実際の音高差

このような違いを音度の前に長 (major), 短 (minor), 完全 (perfect), 増 (augmented), 減 (diminished) などを付加して表現する。上記の例では、左側の3度は長3度、右側の3度は短3度と呼ばれる。表 2.1 に音度表記と実際の音高差の対応を示す。

2.4 転回

任意の2音の上下関係を入れ替えることを転回と呼ぶ。この操作によって2音の音度差は変化する。オクターブの範囲内にある任意の2音に対して転回の操作を行うとき、表 2.2 の法則が成り立つ。

²楽譜に描かれる5本の横線

³線と線の間

完全1度	=	半音 × 0	減5度	=	半音 × 6
短2度	=	半音 × 1	完全5度	=	半音 × 7
長2度	=	半音 × 2	短6度	=	半音 × 8
短3度	=	半音 × 3	長6度	=	半音 × 9
長3度	=	半音 × 4	短7度	=	半音 × 10
完全4度	=	半音 × 5	長7度	=	半音 × 11
増4度	=	半音 × 6	完全8度	=	半音 × 12

表 2.1: 音度差

長 n 度	→	短 $(9 - n)$ 度
短 n 度	→	長 $(9 - n)$ 度
完全 n 度	→	完全 $(9 - n)$ 度
減 n 度	→	増 $(9 - n)$ 度
増 n 度	→	減 $(9 - n)$ 度

表 2.2: 展開に伴う音度の変化

2.5 調

調とは任意の音を基準として規則的に音を集めた音の集合であり、楽曲は任意の調の音を用いて作られる。使用される調の数は楽曲によって様々であるが、楽曲中の任意の時刻における調は基本的に唯一である。

調には大きく分けて長音階 (Major Scale)、短音階 (Minor Scale) の 2 種類が存在する。長音階は長調、短音階は短調とも呼ばれる。前者は明るい響き、後者は暗い響きが得られる。長調と短調では、要素である音を基準音から順に並べその階差をとって得られる音程の並びが異なる。この違いを図 2.4 と図 2.5 に示す。図のように、長調では階差の音程が「全音・全音・半音・全音・全音・全音・半音」のように並ぶのに対して、短調では「全音・半音・全音・全音・半音・全音・全音」のように並ぶ。

C の音を基準に音を並べた長調を八長調 (C Major Scale)、短調を八短調 (C Minor Scale) のように呼ぶ。また基準音を主音と呼ぶ。加えて、7 番目の音を半音上げた短調を旋律的短音階 (Melodic Minor Scale) と呼び、これに対して通常の短調を和声的短音階 (Harmonic Minor Scale) と呼ぶ。

2.6 近親調

任意の調とその構成音が類似する調を近親調と呼ぶ。八長調の近親調を表 2.3 に示す。主音である C の音から長 6 度上の A の音を主音とした短調を並行調、完全 5 度上の G の

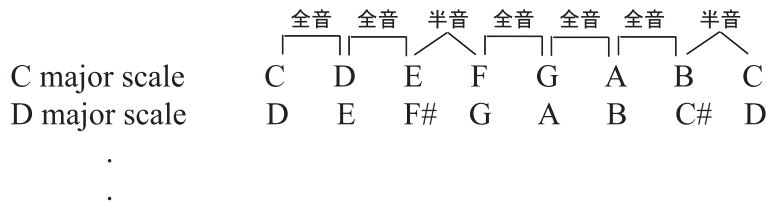


図 2.4: 長調

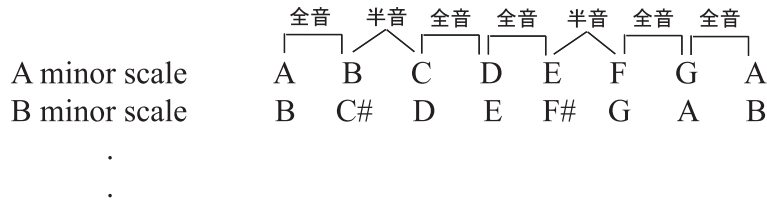


図 2.5: 短調

音を主音とした長調を属調，完全 5 度下の F の音を主音とした長調を下屬調と呼ぶ．主調と並行調はその構成音が完全に一致する．

主調	C Major Scale	.. C D E F G A B C ..
平行調	A Minor Scale	.. A B C D E F G A ..
属調	G Major Scale	.. G A B C D E F# G ..
下屬調	F Major Scale	.. F G A Bb C D E F ..

表 2.3: C Major Scale の近親調

2.7 和音

和音とは幾つかの音が規則的に重なって構成されるものである．主に伴奏のような役割を果たし，楽曲の任意の時刻における響きを決定する．最も基本的な和音は基本三和音と呼ばれる．図 2.6 に示すように，基本三和音は和音の基礎となる根音の上に 3 度上の音と 5 度上の音を重ねたものである．3 度上の音を第 3 音 (3rd)，5 度上の音を第 5 音 (5th) と呼ぶ．基本三和音のうち，第 3 音と根音との音度差が長 3 度であるものは Major Chord (長三和音)，短 3 度であるものは Minor Chord (短三和音) と呼ばれる．Major Chord は明るい響きが得られ，Minor Chord は暗い響きが得られる．また，根音と第 3 音との音度差が短 3 度で第 5 音との音度差が減 5 度である三和音を Diminished Chord (減三和音) と呼ぶ．本稿では，C の音を根音とする長三和音を 'C'，短 3 和音を 'Cm'，減三和音を 'Cdim' のように表記する．

和声学では、基本三和音の他に七の和音が定義されている。これは、基本三和音に根音からの音度差が7度の第7音を付加したものである。七の和音の例を図2.7に示す。同図の左側の和音は長三和音に第7音を付加して得られる七の和音であり、右側の和音は短三和音に第7音を付加して得られる七の和音である。本稿ではC Major Chord に第7音を付加した七の和音を‘C7’、C Minor Chord に第7音を付加した七の和音を‘Cm7’のように表記する。

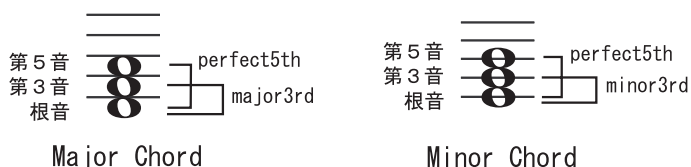


図 2.6: 基本三和音

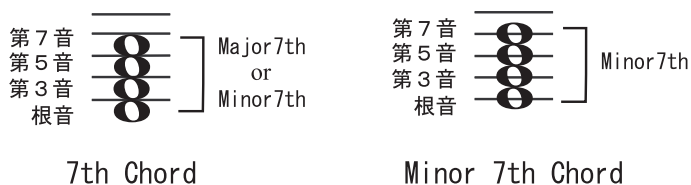


図 2.7: 七の和音

2.8 和音の転回

図2.8に示すように、和音には転回型と呼ばれる根音以外の和音構成音を最低音とした和音形態が存在する。基本三和音の根音を最低音とした形態を基本型、第3音を最低音とした形態を第1転回型、第5音を最低音とした形態を第2転回型と呼ぶ。七の和音の転回型は第3転回型まで存在する。

基本三和音を転回すると、第1展開型は最も低い音の上に3度上と6度上の音を、第2転回型は4度上と6度上の音を重ねた形となり、視覚的に3度・5度の関係を見出すことはできなくなる。

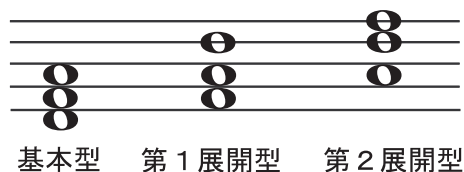


図 2.8: 和音の展開型

2.9 調と和音の関係と音度表記

楽曲は幾つもの和音を連結することで作られる。和音の連結の際は同一の調の構成音によって作成された和音同士を連結することが最も基本的である。任意の調の構成音からは、7つの基本三和音と7つの七の和音を作成することが可能である。八長調の構成音から作成可能な基本三和音を図 2.9 に示す。

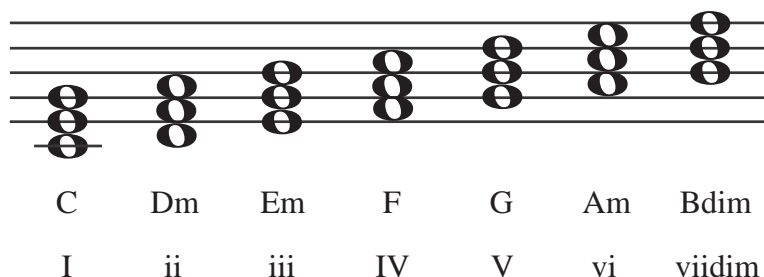


図 2.9: 八長調の構成音から作られる基本 3 和音とその音度表記

同図中に示されるように、任意の調から得られる和音にはローマ数字による名前が与えられる。これを音度表記と呼ぶ。長三和音は大文字、短三和音は小文字で表記し、特に長三和音と短三和音を区別する必要がない場合は全て大文字で表記する。また七の和音は‘V₇’のように表記する。この音度表記を用いることで、具体的な調名や和音名を挙げることなく和音の進行を表現できる。例えば、‘I - IV - V - I’ という音度表記列は、八長調ならば ‘C - F - G - C’、イ短調ならば ‘A - D - E - A’ という和音進行を意味する。また ‘C - G - C’ という和音進行は、八長調であれば ‘I - V - I’、ト長調であれば ‘IV - I - IV’ と表現される。本稿では調と音度表記の組み合わせを ‘C : I - V - I’ のように ‘:’ の左側に調名、右側に音度表記を記述して表現する。

2.10 カデンツ (cadence) と和音機能

カデンツとは和音同士を連結する際の最も重要な規則である。この規則では、任意の調から得られる各和音をその役割ごとに分類し、分類された和音をどのように連結すべきかを定義している。また、カデンツの規則に基づく和音列は人間が聴くと完成した1つのフレーズに聞こえる。

カデンツの規則ではまず任意の調から得られる和音を

Tonic	I or VI
Subdominant	II or IV
Dominant	V

のように Tonic, Subdominant, Dominant の和音機能と呼ばれるものに分類する。III と VII の和音は特殊な働きをするためここでは省略する。Tonic の和音はフレーズ中で中心

的な役割を果たし, Dominant, Subdominant の和音は Tonic の和音へと連結してフレーズを終了させる機能をもつ. また, 任意の和音の機能は基づく調によって決定される. 例えば, C の和音は八長調では I の和音であるから Tonic の役割を果たし, ト長調では IV の和音であるから Subdominant の役割を果たす. 以後本稿では和音機能を 'T', 'S', 'D' と略記する.

T, D, S に対して以下の3つの連結規則が定義されている. これをカデンツの規則と呼ぶ.

cadence	→	T - D - T	K1
cadence	→	T - S - D - T	K2
cadence	→	T - S - T	K3

これらのカデンツは 'K1', 'K2', 'K3' のように表記され, 先頭の Tonic の項は開始 T 項, 最後の Tonic の項は終結 T 項, Dominant, Subdominant の項は D 項, S 項と表記される. また, 'V - I' で終止するカデンツを全終止, 'V - VI' で終止するカデンツを偽終止, 'IV - I' で終止するカデンツを変終止と呼ぶ. 七の和音を用いたカデンツもこのルールに属する. 楽曲はこのカデンツをいくつも連鎖させることで作られる.

第3章 HSPG

HSPG とは主辞 (Head) と呼ばれる概念が中心的役割を果たす文法理論である。この理論では、単語の情報はその役割や制約とともに表現されることから、辞書の情報を充実させることで文法規則の数を大幅に削減することが可能である。

本章では、HPSG の主辞の概念、下位範疇化プロセス、型階層、素性構造による語や句の表現、単一化、文法規則、構文解析プロセスについて解説する。

3.1 主辞の概念と下位範疇化

3.1.1 主辞

HPSG には主辞と呼ばれる概念が中心に据えられていると述べた。この主辞の概念とは、名詞句や動詞句などあらゆる句にはその下位に必ず中心的な役割を果たす句や語が存在し、その中心となる句や語の重要な情報は必ず上位階層に伝えられその性質が受け継がれる、という考え方に基づくものである。これを図 3.1 の名詞句を例に説明する。図中の 'N' は名詞、'Det' は冠詞を意味する。同図に示されるように、名詞句は下位に名詞を必ず保持しており、名詞句は名詞の性質を受け継いでいる。この名詞のように、中心的な役割を果たす語や句のことを主辞と呼び、冠詞のように主辞を補う語や句のことを補語と呼ぶ。また、主辞を中心として形成された句には、' \bar{N} ' のように主辞の名前の上にバーを付加した名前が与えられる。これは、バーを伴わない同名のカテゴリーを抽象化したものであることを意味する。

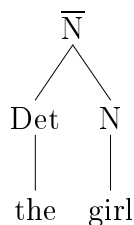


図 3.1: 名詞句

3.1.2 下位範疇化

HPSGでは、生成された上位階層を親、親を形成するいくつかの語や句を子と呼ぶ。また主辞である子が補語である子と結びついて親を形成するプロセスを下位範疇化と呼ぶ。HPSGによる句の構造の解析はこの下位範疇化プロセスを繰り返すことによって行われ、結果として構文木と呼ばれる木構造が得られる。この構文解析プロセスを図3.2を用いて説明する。図中の‘V’は動詞を意味する。まず、名詞句の主辞である‘girl’が補語である‘the’を下位範疇化することで名詞句‘ \bar{N} ’の木構造が得られる。また、先頭の名詞‘He’はこれ単体で名詞句を形成する。次に、動詞句の主辞である‘loves’が補語である右側の‘ \bar{N} ’を下位範疇化して動詞句‘ \bar{V} ’の木構造が得られ、これが最後に先頭の‘ \bar{N} ’を下位範疇化することで、句全体の木構造が得られる。また同図に表れているように、句全体の主辞は動詞‘loves’である。このようにHSPGにおける句全体の主辞は必ず動詞となる。

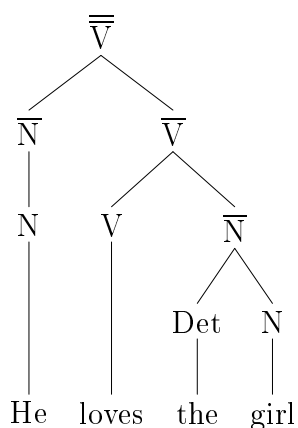


図 3.2: 句の木構造

3.2 型階層と単一化

3.2.1 型階層

HPSGでは解析に必要な様々な値を型 (sort) と呼ぶ。すべての型は型階層と呼ばれる知識階層においてその上位関係が定義される。型階層の例を図3.3に示す。型階層では下位の型ほど一般的な型であり、上位の型はより特殊な型となる。また、上位の型は自身の下位の型の性質を継承している。最も下位にある型はボトム (\perp) とよばれ、全ての型はここから派生する。

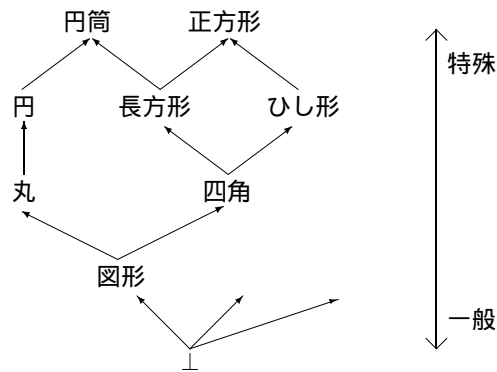


図 3.3: 型階層の例

3.2.2 型の単一化

型階層を用いて、任意の二つの型からより正確な情報を得ることができる。例えば、図 3.3 の型階層より長方形でありひし形である図形は正方形であることがわかる。このとき、正方形は長方形とひし形の最小上界 (Least Upper Bound) と呼ばれ、これを求める操作を型の単一化と呼ぶ。つまり、型の単一化とは任意の二つの型に共通する最も近い祖先を求める操作である。また、円とひし形のように共通の祖先を持たない型同士は単一化することができない。

3.3 素性構造と単一化

HSPG では単語や句などの情報を素性構造と呼ばれるもので表現する。素性構造とは

$$\left[\begin{array}{ll} attribute_1 & value_1 \\ attribute_2 & value_2 \end{array} \right]$$

のように、属性と値のペアを括弧でくくって表現したものである。本節では、この素性構造と素性構造を用いた知識表現、素性構造に関する単一化について解説する。

3.3.1 素性構造

素性 (feature) とは型の属性であり、また任意の型から派生する矢印である。HPSG では、任意の知識に関する情報を素性と型を用いて図 3.4 のように有向グラフによって表現する。この有向グラフを素性構造と呼ぶ。素性構造では素性は大文字、型は小文字で表記される。

上のような素性構造は AVM (Attribute-value matrix) を用いて図 3.5 のように表記される。一つの素性構造は括弧でくくって表記し、括弧内には型の名前が先頭に配置され、

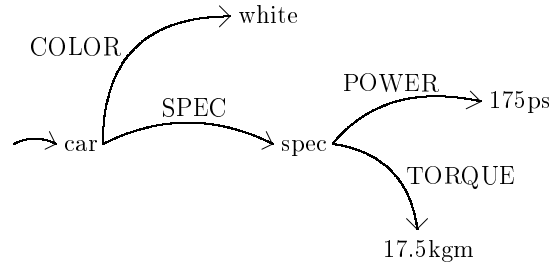


図 3.4: 素性構造

その下にその型から派生する素性名とその先の素性構造のペアが格納される．型名を得に表記する必要のない場合は省略して表記する．

$$\left[\begin{array}{l} \text{car} \\ \text{COLOR} \quad \text{white} \\ \text{SPEC} \quad \left[\begin{array}{l} \text{spec} \\ \text{POWER} \quad 175\text{ps} \\ \text{TORQUE} \quad 17.5\text{kgm} \end{array} \right] \end{array} \right]$$

図 3.5: AVM による素性構造の表記

また，HPSG では \boxed{i} などのインデックスによるポインタによって任意の素性構造全体を表現する．このポインタを用いて，図 3.6 の左側の素性構造は同図右側の素性構造のように表現できる．このように，ポインタを用いて異なる素性が同一の型や素性構造を共有することを構造共有と呼ぶ．

$$\left[\begin{array}{l} F_1 \quad \left[\begin{array}{l} F_3 \quad s_3 \\ F_4 \quad s_4 \end{array} \right] \\ F_2 \quad \left[\begin{array}{l} F_3 \quad s_3 \\ F_4 \quad s_4 \end{array} \right] \end{array} \right] \rightarrow \left[\begin{array}{l} F_1 \quad \boxed{1} \quad \left[\begin{array}{l} F_3 \quad s_3 \\ F_4 \quad s_4 \end{array} \right] \\ F_2 \quad \boxed{1} \end{array} \right]$$

図 3.6: 構造共有

3.3.2 素性構造の単一化

3.3.1 節で述べたインデックスによるポインタを用いて異なる素性構造同士が構造共有される場合，それらは矛盾のない限り単一化される．このような単一化は 2 つの子の素性構造から親の素性構造を導き出す際などに行われる．本稿では単一化の記号に \sqcup を用いる．(3.1) に簡単な単一化の例を示す．この単一化では二つの情報をあわせてより豊富な情報を得ている．

$$\begin{bmatrix} F_1 & s_1 \\ F_2 & s_2 \end{bmatrix} \sqcup \begin{bmatrix} F_3 & s_3 \end{bmatrix} \rightarrow \begin{bmatrix} F_1 & s_1 \\ F_2 & s_2 \\ F_3 & s_3 \end{bmatrix} \quad (3.1)$$

また，単一化に際して同一の素性が異なる型を指している場合は，型階層が参照され最小上界の型が求められる．この時，最小上界が求まらない場合は単一化に失敗する．型階層が図 3.7 のように定義される場合の単一化の例を (3.2) に示す．この単一化では二つの情報からより正確な情報を得ている．

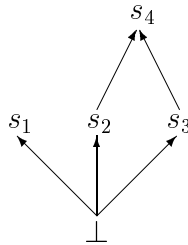


図 3.7: 型階層

$$\begin{bmatrix} F_1 & s_1 \\ F_2 & s_2 \end{bmatrix} \sqcup \begin{bmatrix} F_1 & s_1 \\ F_2 & s_3 \end{bmatrix} \rightarrow \begin{bmatrix} F_1 & s_1 \\ F_2 & s_4 \end{bmatrix} \quad (3.2)$$

以上のような単一化操作は型の継承の際にも行われる．例えば図 3.8 に示すように，型 ‘s1’ が素性 ‘F4’ を持ち，型 ‘s2’ が素性 ‘F5’ を持つ場合，この二つの型を継承する型 ‘s3’ は親の素性 ‘F4’，‘F5’ をそれぞれ受け継ぐ．

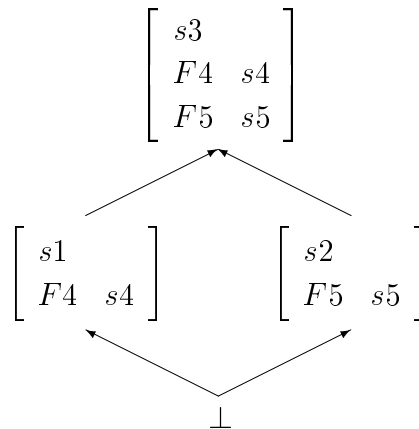


図 3.8: 素性の継承

3.3.3 素性構造による単語・句の表現

素性構造は解析する言語に応じてある程度自由にデザインできるが、伝統的には図 3.9 のような骨格を持つ。‘〈 〉’ はリストを意味する。ここでは、同図の素性構造内の各素性の意味について説明する。

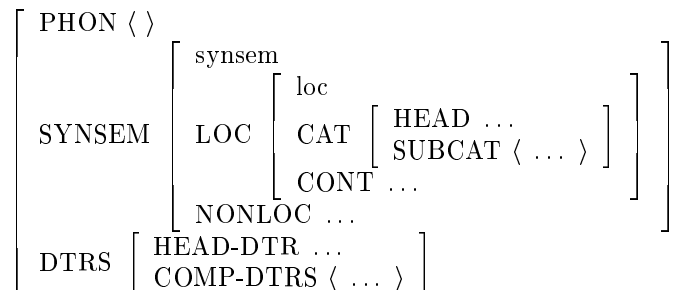


図 3.9: 動詞の素性構造例

- PHON 素性
表現対象の音声情報を格納する。
- SYNSEM 素性
Syntax and Semantics の略。この素性の先に構文や意味の情報を格納する。
- LOC
Local の略。表現対象の局所的な情報を格納する。
- NONLOC
Non-Local の略。表現対象の非局所的な情報を格納する。
- CAT
Category の略。この先の素性構造全体が CFG の非終端記号に相当する。
- HEAD
名詞・動詞・形容詞など、この素性の重要な情報を格納する。またここから派生する素性の先には、性・数・各・人称などの情報が格納される。
- SUBCAT
Subcategorization の略。表現対象が下位範疇化可能な語や句などの素性構造を格納する。

- CONT
Content の略．意味に関する情報を格納する．
- DTRS
表現対象がすでに下位範疇化している語や句などの素性構造を格納する．
- HEAD-DTR
主辞である子の素性構造を格納する．
- COMP-DTRS
補語である子の素性構造を格納する．

上記のように，HPSG では素性構造を用いて単語や句を様々な情報と共に表現する．このため，CFG のように無数のシンボルを定義することなく様々なカテゴリーを表現できる．また単語ごとにそれが何を下位範疇化するかを定義するため，文法規則などのルールの数を大幅に削減することが可能である．

今後本稿では素性構造を見やすく表示するために，直接関係のない素性は省略する．また，連続する素性の表示は ‘|’ を挟んで ‘ $F_1 | F_2 | F_3$ ’ のように表記，もしくは ‘ $F_1 || F_n$ ’ や ‘ $|| F_n$ ’ のように略記する．

3.4 文法規則と構文解析

HPSG の文法規則は ID-スキーマとプリンシプルの 2 つから成り立っている．ここでは ID-スキーマとプリンシプルの代表的な例を紹介し，これらを用いた構文解析の例について説明する．

3.4.1 ID-スキーマ

ID-スキーマ (Schema) とは任意の 2 つの素性構造から親の素性構造を得るための制約である．HPSG ではこれを素性構造によって表現する．ID-スキーマの例を図 3.10 に示す．これは Head-Complement-Schema と呼ばれるもので，主辞の子の SUBCAT 素性の先の先頭の素性構造と補語の子の素性構造は矛盾無く単一化されなければならない，という制約が表現されている．

$$\left[\text{DTRS} \left[\begin{array}{l} \text{HEAD-DTR} \\ \text{COMP-DTRS} \end{array} \left[\begin{array}{l} \text{SYNSEM||SUBCAT} \langle \boxed{1}, \dots \rangle \\ \boxed{1} \end{array} \right] \right] \right]$$

図 3.10: ID-Schema

3.4.2 プリンシプル

プリンシプル (Principle) とは親の素性構造が満たさなければならない制約のことである。プリンシプルもまた素性構造によって表現される。まず、Head-Feature-Principle と呼ばれるものを図 3.11 に示す。このプリンシプルでは、親の HEAD 素性と主辞の子の HEAD 素性は構造共有されなければならない、ということが表現されている。このプリンシプルによって主辞の子の重要な情報が親に伝えられる。

$$\left[\begin{array}{l} \text{SYNSEM||HEAD} \boxed{1} \\ \text{DTRS|HEAD-DTR} \left[\text{SYNSEM||HEAD} \boxed{1} \right] \end{array} \right]$$

図 3.11: Head-Feature Principle

また、図 3.12 は Subcategorization-Principle と呼ばれるものである。このプリンシプルでは、親が下位範疇化すべきカテゴリーは主辞の子が下位範疇化すべきカテゴリーのうち既に下位範疇化されたものを除いたものである、ということが表現されている。

$$\left[\begin{array}{l} \text{SYNSEM||SUBCAT} \langle \boxed{2} \rangle \\ \text{DTRS} \left[\begin{array}{l} \text{HEAD-DTR} \left[\text{SYNSEM||SUBCAT} \langle \boxed{1}, \boxed{2} \rangle \right] \\ \text{COMP-DTRS} \boxed{1} \end{array} \right] \end{array} \right]$$

図 3.12: Subcategorization Principle

この他にもいくつかプリンシプルが存在し、これらの制約をもとに親の素性構造が生成される。

3.4.3 HPSG による構文解析

HPSG の構文解析では、隣り合う 2 つの素性構造に対して ID-スキーマやプリンシプルを適用し、階層的に親の素性構造を求める。この時、親を求める順序や子のペアを指定する必要はない。ID-スキーマによって任意の子のペアから親を求めることが可能か順に調べられるため、解析の手順は自ずと決定される。ここでは 'Mary loves him' という文を例に、構文解析プロセスについて解説する。

まず、名詞 'him' の素性構造を図 3.13 に示す。素性 'CASE' は格を意味し、型 'acc' は対格を意味する。また、素性 'PER' は人称を、型 '3rd' は三人称を、素性 'NUM' は数を、型 'sing' は単数を意味する。この素性構造では、'him' という単語が三人称単数の対格の名詞であることが表現されている。

続いて、名詞 'Mary' の素性構造を図 3.14 に示す。この素性構造では 'Mary' という単語が三人称単数の名詞であることが表現されている。また、格の情報は表現されていない。これは単語 'Mary' は主格や対格となることができるためである。

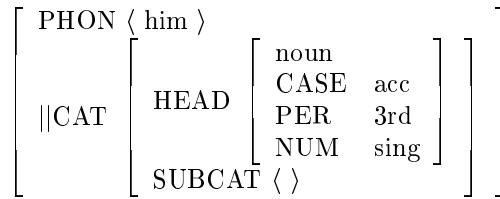


図 3.13: 名詞 ‘him’ の素性構造例

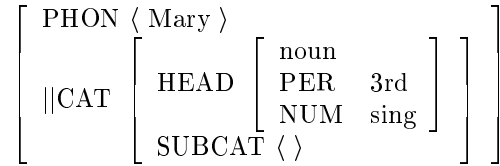


図 3.14: 名詞 ‘Mary’ の素性構造例

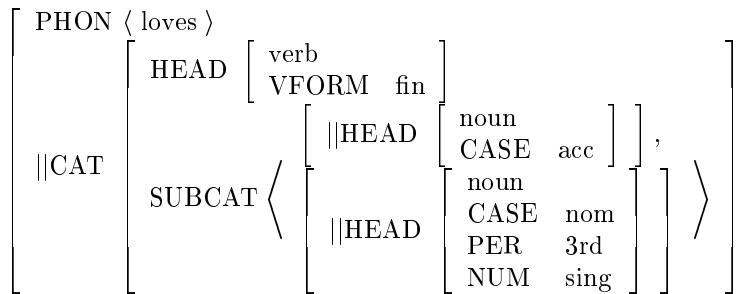


図 3.15: 動詞 ‘loves’ の素性構造例

次に，‘loves’ の素性構造を図 3.15 に示す．素性 ‘VFORM’ は態を意味し，型 ‘fin’ は時制を伴う形を意味する．また，型 ‘nom’ は主格を意味する．この素性構造では，単語 ‘loves’ が時制を伴う形の動詞で，まず対格の名詞を下位範疇化し，後に主格の三人称単数名詞を下位範疇化することで飽和する，ということが表現されている．

最初に ‘loves’ と ‘him’ の素性構造から動詞句の素性構造が導かれる．この時，まず図 3.16 のような空の素性構造を用意し，‘HEAD-DTR’ 素性の先に ‘loves’ の素性構造を，‘COMP-DTRS’ 素性の先に ‘him’ の素性構造を格納する．この結果得られる素性構造を図 3.17 に示す．

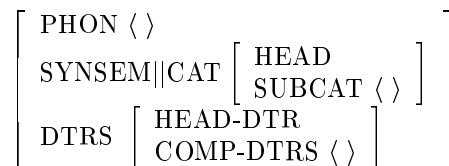


図 3.16: 空の素性構造例

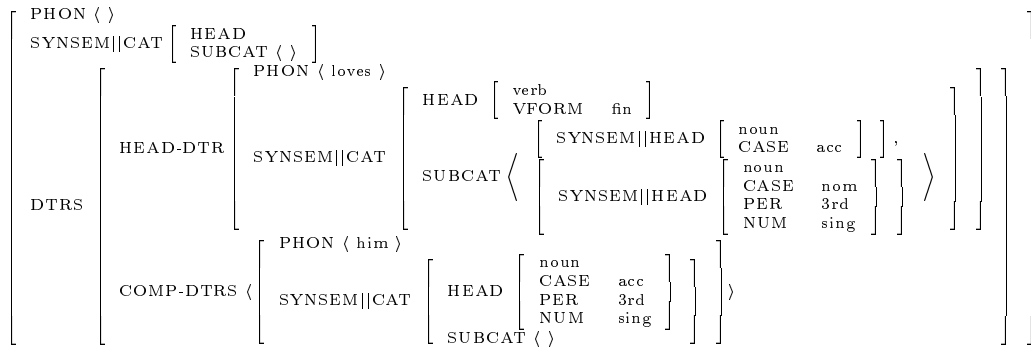


図 3.17: 親の素性構造

続いて、この素性構造と図 3.10 の ID-スキーマとの単一化を行う。この単一化によって、主辞の子である ‘loves’ の SUBCAT 素性の先の素性構造と補語の子の素性構造が構造共有される。この時の素性構造を図 3.18 に示す。また、もし ID-スキーマの適用による構造共有に失敗したならば、主辞の子と補語の子が文として相応しくない組み合わせであることが分かる。

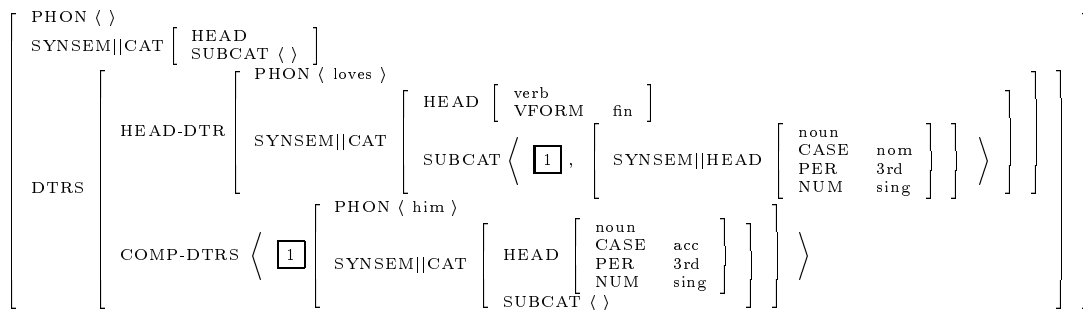


図 3.18: ID-スキーマ 適用後

続いて 3.4.2 節に示した各プリンシプルとの単一化を行う。図 3.18 の素性構造と図 3.11 の Head-Feature Principle とを単一化することで、親と主辞の子との HEAD 素性が構造共有され、主辞の子の重要な情報が親へと伝えられる。さらに、図 3.12 の Subcategorization Principle との単一化を行うことで、親の Subcat 素性に親が下位範疇化すべきカテゴリーの情報が伝えられる。これらのプリンシプルを適用して得られる親の素性構造を図 3.19 に示す。この後、さらに幾つかのプリンシプルを適用することによって親の素性構造が求まる。この操作によって得られる木構造を図 3.21 に示す。

続いて、生成された動詞句と単語 ‘Mary’ から句全体の素性構造が生成され、図 3.21 に示す木構造が得られる。この時、ID-スキーマが適用されたことによって‘3’の ‘loves-him’ の Subcat 素性の先の素性構造と‘4’の ‘Mary’ の素性構造が構造共有され、‘Mary’ の素性構造に素性 ‘CASE’ の情報が追加される。これによって ‘Mary’ が主格の名詞として機能していることが表現される。

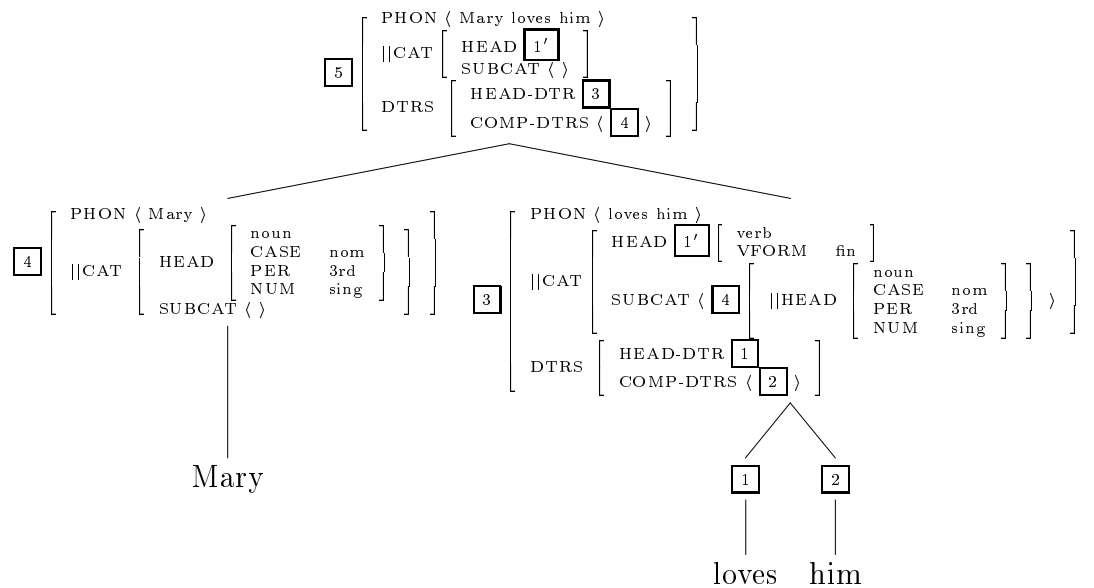


図 3.21: 句の木構造

以上，ここでは HPSG における構文解析のプロセスについて解説した．ここで示されたように，HSPG はごく少数のルールによって文の構造を解析することが可能である．これは，素性構造を用いることで解析に必要な様々な知識が単語の情報と一緒に格納されるためである．また知識とそれを扱う規則が完全に分離されているため，任意の単語の振る舞いに変更されたとしても ID-スキーマなどのルールを変更する必要はない．

第4章 和声解析と先行研究

和声解析にはその目的や用途によって様々な形態が存在する．しかし基本的には図 4.1 のように，楽曲から和音列を認識した後に調の情報を基に和音機能で分類し，カデンツがどのように連鎖しているかを解析するものである．この章では調性認識・和音認識・和音進行解析の三者と，三者を総合的に解析する和声解析について概要と研究事例，問題点などを紹介する．

和音名： C G C F C G C
和音機能： T D T S T D T
cadence K1 cadence K2 cadence K1

図 4.1: 和声解析例 Mozart Piano Sonate in C, K545.

4.1 調性認識

調性認識に関しては幾つかの研究事例が報告されている．調の情報は和音に音度表記を与える際や和音機能で分類する際に用いられる．ここでは調性認識に関する先行研究をいくつか紹介し，これらの問題点について述べる．

4.1.1 最も古典的な調性認識手法

調性認識に関する最も古典的な手法 [10] として以下の三つの手法が挙げられる．

1. メロディーの第 1 音を主音とみなして調を認識する手法
2. メロディーの最終音を主音とみなして調を認識する手法

3. 出現頻度の基も高い音を主音とみなして調を認識する手法

上記の三つの手法は、まず調の主音を特定した後に長調と短調の判別を行う。1 番目の手法は、西洋音楽のメロディーは第 1 音が調の主音であることが多い、という経験、また 2 番目の手法は、西洋音楽のメロディーは主音で終わることが多い、という経験に基づいて調を解析する手法である。しかしこのような経験は必ずしも全てのメロディーに当てはまるわけではい。事実メロディーの第 1 音や最終音が主音ではない楽曲も多いため、これらの手法では正確に調を認識することができない。

3 番目の手法は、メロディー中で最も多く用いられる音が調の主音であることが多い、という経験に基づいている。しかし主音以外の音が最も出現頻度の高い楽曲も多数存在するため、前出の二つの手法と同様にこの手法も正確に調を認識することはできない。

4.1.2 調性空間を用いた手法

Longuet-Higgins と Steedman ら [4] は横向きに完全 5 度ずつ、縦向きに長 3 度ずつ音を並べた調性空間と呼ばれるマトリックス上で調の解析を行う手法を提案している。調性空間を表 4.2 に示す。この調性空間上では、長調と短調はそれぞれ表 4.3 のような枠組みで表現される。この枠組みを調性空間上の任意の位置に移動させることで 12 通りの長調と短調を表現することができる。この手法の解析は、入力された音が調の枠組みの中に全て収まるような位置を探索することで行われる。

入力はメロディーの音高の情報のみで、1 音入力されるたびにその音が枠組みに入らない調を候補から排除していく。全ての音を入力した結果、解が一意に求まった場合はその解を出力し、一つの解も残らなかった場合は 1 前の音が入力された段階まで戻ってその時点での解を出力する。また、いくつかの解が残った場合は主音属音選好規則とよばれるルールで解を決定する。このルールは基本的に 4.1.1 項の 1 番目の手法のように、メロディーの最初の音を主音とするルールである。

この手法では単純なメロディーならばかなり正確に調を解釈することができる。しかし、解を一意に絞り込むことが出来なかった場合は主音属音選好規則によって解が決定されるため、主音以外の音で始まるメロディーの調を正しく解釈できないことがある。

E	B	F \sharp	C \sharp	G \sharp	D \sharp	A \sharp	E \sharp	B \sharp
C	G	D	A	E	B	F \sharp	C \sharp	G \sharp
A \flat	E \flat	B \flat	F	C	G	D	A	E
F \flat	C \flat	G \flat	D \flat	A \flat	E \flat	B \flat	F	C
D $\flat\flat$	A $\flat\flat$	E $\flat\flat$	B $\flat\flat$	F \flat	C \flat	G \flat	D \flat	A \flat

図 4.2: 調性空間

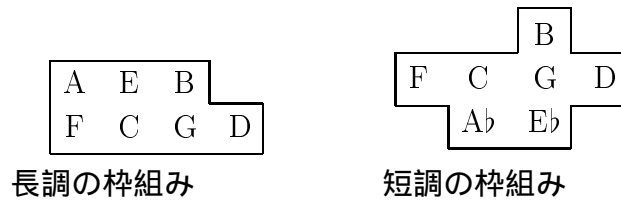


図 4.3: 調性空間上での長調と短調の枠組み

4.1.3 適合度を評価する手法

吉野・阿部ら [5] は，入力されたメロディーが 24 種の調それぞれに対してどの程度適合するかを評価する手法を提案した．モデルにはメロディーが 1 音ずつ入力され，その度に各調の評価値が更新される．

評価値は入力された音が何調のどの音であるかを調べることによって決定される．例えば C の音が入力された場合，これは八長調の主音であるから八長調の評価値を加算する．同時に，C の音はヘ長調の第 5 音であることからヘ長調の評価値も加算する．加算される値は，主音，第 5 音，第 3 音，第 4 音，第 6 音，第 2 音，第 7 音の順に低くなっていく．また，入力された音を含まない調に対してはある程度大きな値を減算する．

この手法はこれまでのものとは大きく異なり，人間の心理的な特徴を反映した手法と言われる．しかし調の構成音以外の音が多く含まれるようなメロディーを解析した場合，減算の値によっては正解の調の評価値が下がってしまうことが予想される．

4.1.4 確率論的モデル

Krumhansl[6] は，人間は長年の音楽経験から楽曲で用いられる各音の出現パターンを無意識的に習得し，実際のメロディーの音高分布パターンを習得したパターンに当てはめることによって調を認識している，と主張した．この主張に基づき Krumhansl は確率論的な手法を提案している．

この手法では予め知覚実験を行い，任意の調のメロディーにおける各音の出現頻度パターン（習得性出現頻度パターン 図 4.4）を用意する．このパターンと，入力されたメロディーの音高分布パターンとを比較することでメロディーの調を解釈する．音高分布パターンは各音の持続時間の総和によって求められる．このため，この手法は音の長さの情報に依存する割合が大きく，主音以外の音が長い音として入力されるとその音を主音として選んでしまうという欠点がある．

4.1.5 調性認識の問題点

この節では調性認識の概要といくつかの調性認識手法について解説した．しかしここで紹介した方式は全てメロディーに対する調性認識手法である．現時点では楽曲中からメロ

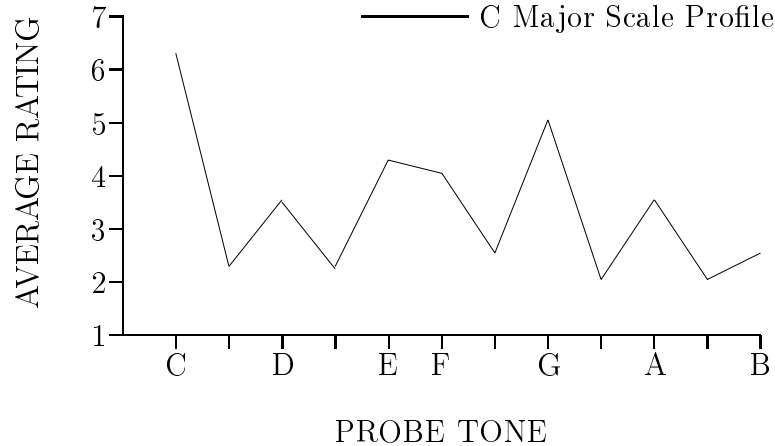


図 4.4: 習得性出現頻度パターン

ディーだけを抽出することは困難である。また、同一のメロディーであっても和音の付け方によって様々な調のメロディーになり得る。このことから、メロディーの情報から楽曲の調を特定することは必ずしも可能ではない。加えて、既存の調性認識が抱える最大の問題は近親調¹の問題である。近親調の関係にある調同士はその構成音が類似するため、メロディーの音高情報のみで調を判断することは不可能である。調ごとの最大の相違点は和音の進行であることから、調の認識においては和音の進行を考慮することが最も有効であると考えられる。以上のことから、伴奏を含む楽曲全体に適用できる調性認識手法が待たれるところである。

また楽曲の調とは必ずしも一つとは限らず、楽曲の任意の部分から調が変化することがある。これを転調と呼ぶ。上記の方式はどれもこの転調については対応していない。この点についても考慮する必要がある。

4.2 和音認識

調性認識・和音認識・和音進行解析の三者のうち、和音認識は最も難しく最も進んでいない分野であると言える。このため、和音認識に関する研究事例はあまり報告されていない。本節では和音認識におけるいくつかの問題点について述べる。

4.2.1 和音の時間的範囲の特定における問題点

和音認識が困難なのは、これがリズム解析とも密接に関わる問題なためである。和音の認識においては、まず一つの和音がどれだけの時間的範囲を支配するかを特定しなければ

¹2.6 参照

ならない．しかし楽曲中にいくつかの和音が含まれ，それぞれがどれだけの時間的範囲を支配するかは，楽曲によって千差万別である．また，和音を構成する音がすべて時間的に重なっているとも限らない．このため，なんらかの情報を基に和音の境界を推定する必要がある．有効な手段としては，GTTM[7]の拍節構造解析やグルーピング解析，また後藤らによるリズムトラッキング手法[8]などの結果を参照し，リズム的な情報を基に和音範囲を推定する手法などが挙げられる．しかし，必ずしもリズム的な切れ目が和音の境界と一致するわけではないため，このような手法でも限界があると考えられる．例えば，図4.5のフレーズのリズム的な切れ目は小節線の上であると考えられる．しかし和音の境界は非常に曖昧であり，1小節目は全体で1つの和音と捉えCもしくはAmと判断することも，2つの和音の進行であると捉えAm - CもしくはC - Amという和音進行と判断することも可能である．



図 4.5: 曖昧性の例

4.2.2 和音の構成音の特定における問題点

仮に一つの和音の時間的範囲が特定できたとしても，そこから一意に和音名が求まるとは限らない．一つの和音の範囲には和音を構成する音（和声音）とそれ以外の音（非和声音）が含まれる．和声音と非和声音の判別も非常に困難である．例えば，図4.5の1小節目全体が一つの和音範囲であると仮定したとき，この和音はGの音以外を和声音と捉えてAmの和音と判断することも，Aの音以外を和声音と捉えてCの和音と判断することも可能である．このような場合和音認識システム単体で解を求めるとすれば，解を複数出力するかいくつかのヒューリスティックスを用いて候補を絞る必要がある．

また，一つの和音範囲には全ての和音構成音が含まれるとは限らない．図4.5の2小節目の和音は，基本三和音を構成する三つの音のうち一つの音が省略されている．この場合，Eの音が省略されていると捉えてEmの和音と判断することも，Gの和音が省略されていると捉えてGの和音であると判断することも可能である．

このように，和音認識では様々な曖昧性が発生する．これを解消するためにはリズムやフレーズ，カデンツなどの情報を参照する必要があると考えられる．

4.2.3 和音名の特定における問題点

実際に和音名を特定する際には，和声音と思われる任意の3音ないし4音の間に3度・5度（・7度）の関係を見出さなければならない．しかし，和音には転回型（2.8節参照）

が存在するため、必ずしも視覚的に3度・5度の関係を見出せるとは限らない。また1つの和音が1オクターブの範囲内に収まっているとも限らないため、何らかの前処理が必要となる。さらに各和音の雛形とのパターンマッチングを行う際、全ての和音形態のパターンを用意すると、それぞれ24通り存在する基本三和音と七の和音に対して基本型・転回型・省略型...と言ったように無数の和音形態を列挙することとなる。これはあまり現実的ではないため、何らかの対策を講じる必要がある。

4.3 和音進行解析解析

和音進行解析では、和音認識によって得られた各和音を調の情報を用いて和音機能で分類し、任意の和音列がどのようなカデンツを形成しているかを解析する。このため和音進行解析はこれ単体で行うことはできず、あらかじめ調と和音列が認識されている必要がある。本節では基本的な和音進行解析の手法とその問題点について述べる。

4.3.1 基本的なカデンツ解析手法とその問題点

仮に調と和音が認識されている場合、任意の長さの和音列を実際のカデンツの形態とパターンマッチングすることになる。しかしすべてのカデンツを列挙すると、基本三和音のみによるIIIとVIIの和音を除いたカデンツだけでも2456通り存在する(旋律的短音階を除く)。これら全てのパターンを列挙することはやはり現実的ではため、調の違いや同じ機能を持つ異なる和音の違いなどを吸収する表現方法が必要となる。

表2.10に示した3通りのカデンツのルールによる和音の進行は、図4.6のような状態遷移図を用いて表現することができる。このためカデンツのルールをContext Free Gram-

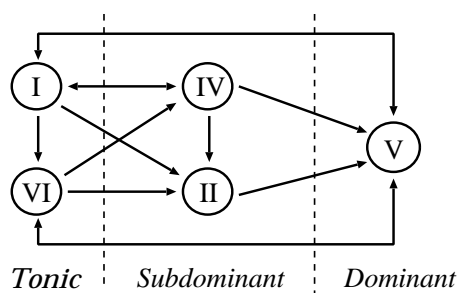


図 4.6: 和音の状態遷移図

mar (CFG) などを用いて記述し和音進行解析を行うという手法が考えられる。和音の進行はCFGを用いて表4.1のように表記できる。

しかしこのような手法においても若干の問題が発生する。例えば、和声学の初歩の知識に厳密に準拠することを考えたとき、変終止の終結 T 項に I の和音の使用は許されるが、

cadence → T, D, T
 cadence → T, S, T
 cadence → T, S, D, T

T → I S → IV D → V
 T → VI S → II
 T → I, VI S → IV, II

表 4.1: Context Free Grammar によるカデンツのルール

VI の和音の使用は許されない。また IV の和音の S 項は I の和音の終結 T 項へ連結可能だが、II の和音や IV - II の和音進行の S 項は I の和音の終結 T 項へ連結不可能となる。このような厳密なルールを CFG で定義していくと、シンボル数の増加やルールの煩雑化を招くこととなり、人間にとっては理解し難いルールとなる可能性がある。

4.4 和声解析

和声解析では、楽曲の前節までに述べた調性認識・和音認識・カデンツ解析を総合的に行う。和声解析を行うシステムはいくつか提案されているが、詳細については触れられていないものや編曲システムなど他のシステムの一部となっていることが多いため、システム単体の具体例を挙げるのは難しい。そこで本稿では、和声解析の手法として現在最も注目されている GTTM と呼ばれる理論について述べ、その問題点について考察する。

4.4.1 GTTM

GTTM[7] は 1983 年に Lerdahl と Jackendoff によって発表された音楽理論である。この理論では自然言語のように音楽を構造的に解析することを提案している。この理論によって Bach Chorale 0 Haupt voll Blut und Wunden を解析した結果を図 4.7 に示す。GTTM の解析は図のように楽曲中から木構造を導き出すことが目的である。

GTTM の解析はグルーピング構造解析、拍節構造解析、タイムスパン簡約、延長的簡約の 4 つに分けられ、それぞれについて解析を行うためのルールが箇条書きされている。グルーピング解析は隣り合う音同士の間隔を調べることによって、図 4.7 の ‘ \cup ’ に示されるような楽曲中のまとまりのあるグループを認識する。拍節構造解析は図 4.7 の ‘?’ や ‘□’ で示されるような、強拍や弱拍などの情報を基にしたリズムの階層的な解析を行う。タイムスパン簡約では図 4.7 の楽譜上部のような木構造を生成する。タイムスパンとは時間的に短く音楽を細分化して得られる間隔のことを指す。この解析ではグルーピング構造解析や拍節構造解析の結果を考慮しながら、音高や和声などの情報を基に各タイムスパン中の重要な音 (ヘッド) を選び、隣り合うタイムスパンのヘッドのうちどちらが重要かを階層

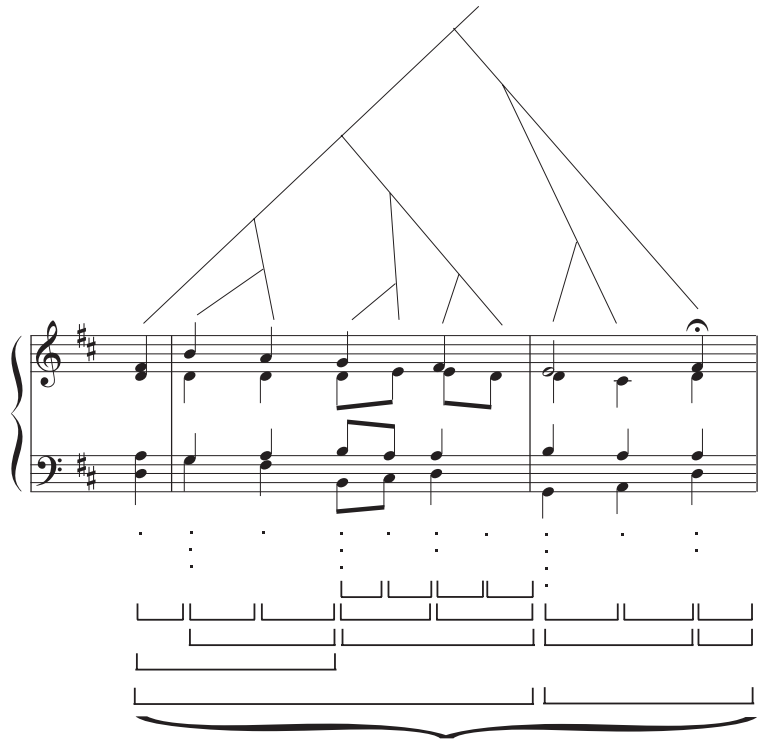


図 4.7: GTTM による和声解析例

的に調べていくことで木構造を導く。また延長的簡約では、タイムスパン簡約で得られた木構造に対して心理学的な立場などから拡張を加える。

GTTM は、音楽を解析するためのルールが箇条書きされていることから計算機上への実装に向いていると言われてきた。しかし、発表されてから 20 年が経過した今でも実装に成功したという例は報告されていない。これは GTTM の理論による音楽の解析が、人間の手によって行われることを前提としているためである。例えば、解析のルールには‘比較的安定した’、‘相対的に長い’、‘局所的な’といった曖昧な表現が用いられており、このような表現が計算機への実装の際に障害となる。またルール同士の競合もしばしば発生するため、多くのヒューリスティクスを必要とする。このため仮に計算機上への実装に成功したとしても、解析する楽曲に応じてヒューリスティック値をチューニングする必要があると考えられる。また、ルールの適用に際してリズムや調・和音・カデンツなどの情報を必要とすることがある。このため事前にこれらを解析しておく必要があるが、その手法については述べられていない。

4.4.2 和声解析における問題点

和声解析では調・和音の認識に発生する曖昧性をどのように解消するかが問題となる。例えば、和音認識から出力された全解候補に対して調性認識から得られた全ての調に基づ

くカデンツを解析した場合，組み合わせの数が膨大なものとなることが予想される．このため，和音やカデンツの前後関係などを考慮し解候補を絞ることが可能な枠組みが要求される．

また，和声解析システムは柔軟な知識表現体系を備えていることも重要である．音楽には様々なジャンルが存在し，それぞれ用いられる調や和音，和音進行が異なることから，単一の音楽知識で全ての音楽を解析することはできない．このため任意の楽曲に特化したルールのみを扱えるような処理体系を構築することは余り好ましくない．様々な音楽知識を柔軟に表現し，知識と処理を分離し，単一の処理体系で様々な知識を処理することが可能な枠組みを用意する必要がある．

第5章 提案方式

ここでは本稿で提案する和声解析手法について述べる．まず，HPSG を用いた和音進行解析手法について述べ，続いて和音進行解析を利用した調性認識手法，和音進行解析と絡めた和音認識手法について述べ，最後に本稿でおこなう和声解析について述べる．

5.1 HPSG を用いた和音進行解析

本節では，文法理論 HPSG を用いてカデンツの解析を行うにあたってまず音楽における主辞を定義し，素性構造を用いた和音・和音列・カデンツの表現と音楽知識に関する型階層を示し，ID-スキーマ・プリンシプルを用いてカデンツをどのように解析するかについて述べる．

5.1.1 音楽における主辞

3.1.2 節で HSPG を用いて自然言語の単語列から木構造を見出す，構文解析プロセスについて述べた．音楽の和音列からも同様の木構造を見出すことが出来る．この例を図 5.1 に示す．図中の ‘T’ は Tonic ，‘D’ は Dominant ，‘S’ は Subdominant を意味する．

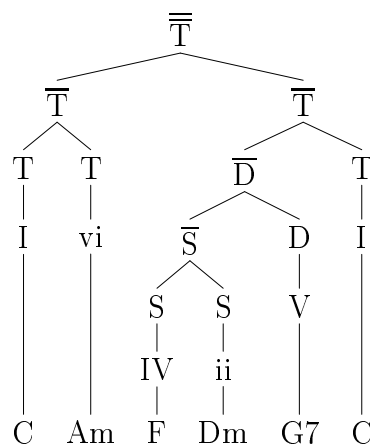


図 5.1: カデンツの木構造

このような木構造を生成するために，本稿では Tonic の和音を和音列中の主辞である

と定義する．これは Tonic がカデンツの中心的な和音であることと，Tonic の和音には必ず調の主音が含まれていること，また Dominant や Subdominant が Tonic を誘発して解決するための和音であること，などの理由によるものである．またカデンツには開始 T 項と終結 T 項の二つの T 項が現れるが，カデンツの主辞となるのは終結 T 項であると定義する．これは終結 T 項の種類によってカデンツが完全終止か偽終止かが決定されるためである．

T 項に I, VI の二つの和音が現れる場合，T 項の主辞は I の和音であると定義する．これは I の和音が調の主音を根音とした和音であり，最も重要な Tonic 和音であるためである．同様に，S 項の主辞は IV の和音であると定義する．また S 項と D 項が並ぶ場合，その局所的な主辞は D 項であると定義する．これは Dominant の和音がより Tonic へと進行する機能が高いためである．

以上の定義から，図 5.1 のような木構造を以下の手順で生成することができる．まず，先頭の 'I' の和音が続く 'vi' の和音を下位範疇化し \bar{T} の木構造が求まる．続いて，'IV' の和音が 'ii' の和音を下位範疇化し \bar{S} の木構造が求まる．この \bar{S} を D 項の 'V' の和音が下位範疇化し \bar{D} の木構造が求まる．さらに，この \bar{D} を最後の 'I' の和音が下位範疇化して \bar{T} が求まり，これが最初に求めた \bar{T} を下位範疇化して $\bar{\bar{T}}$ が求まる．この $\bar{\bar{T}}$ が一つのカデンツに相当する．

5.1.2 素性構造による和音の表現

3.3 章において HPSG の素性構造による単語や句の表現について述べた．楽曲の和音や和音進行も素性構造を用いて効果的に表現することが可能である．まず，型 'head' に関する素性構造を図 5.2 のように定義する．素性 'DEG' は音度表記 (DEGREE) を意味し，この素性の先に和音の音度表記を格納する．素性 'TONAL' は調性 (TONALITY) を意味し，この素性の先に調名を格納する．また，素性 'TYPE' では和音が形成するカデンツのタイプを表現する．この他にも和音の発音時刻や時間的範囲を表現する素性を導入することが考えられるが，本稿では和音の機能にのみ着目するためこれらの情報は省く．

$$\left[\begin{array}{l} \text{head} \\ \text{DEG} \quad \dots \\ \text{TONAL} \quad \dots \\ \text{TYPE} \quad \dots \end{array} \right]$$

図 5.2: head の素性構造

続いて，型 'head' に関する型階層を図 5.3 に定義する．同図に示されるように，型 'tonic'，'dominant'，'subdominant' に型 'head' を継承させる．これによってこの三つの型は 'head' が持つ素性構造を継承し，音度表記，調，カデンツタイプの情報を表現できるようになる．

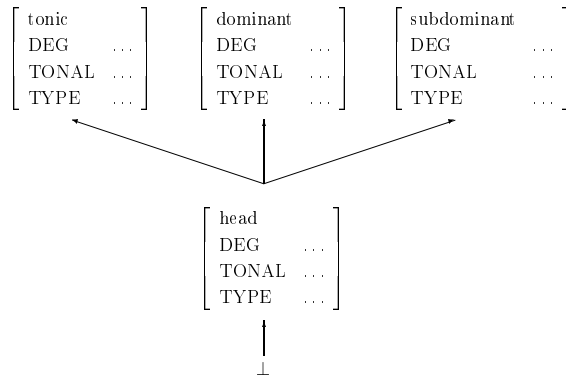


図 5.3: head の型階層

次に，素性 ‘DEG’ に格納される型に関する型階層を図 5.4 のように定義する．型 ‘degree’ の先に型 ‘I’, ‘II’ などの音度表記を配置し，各音度表記の先にそれぞれの七の和音を配置する．これによって，任意の基本三和音を下位範疇化する規則を記述すれば七の和音を下位範疇化する規則を記述する必要がなくなる．また，短三和音やその七の和音に関する型階層も同様に定義する．

続いて，素性 ‘TONAL’ に格納される型に関する型階層を図 5.5 のように定義する．型 ‘major-scale’ の先に ‘C’, ‘D’ などの長調の調名を配置する．短調の型階層も同様に定義する．

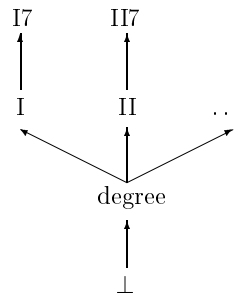


図 5.4: degree の型階層

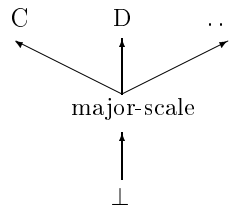


図 5.5: tonality の型階層

加えて，素性 ‘TYPE’ に格納される型に関する型階層を図 5.6 のように定義する．型 ‘type’ の先に ‘K1’, ‘K2’, ‘K3’ の各カデンツタイプを表す型を配置する．

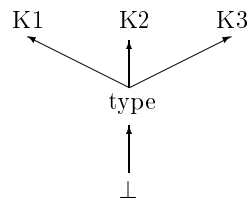


図 5.6: type の型階層

以上の定義より，和音は図 5.7 のように表現できる．素性 ‘PHON’ には和音名を格納する．この素性構造は Tonic である C の和音を表し，またこの和音が八長調の I の和音で，かつ K 1 の和音であるということ表現している．

$$\left[\begin{array}{l} \text{PHON} \langle C \rangle \\ \text{SYNSEM|LOC|CAT|HEAD} \left[\begin{array}{ll} \text{tonic} & \\ \text{DEG} & I \\ \text{TONAL} & C \\ \text{TYPE} & K1 \end{array} \right] \end{array} \right]$$

図 5.7: 素性構造による和音の表現例

同様にして和音列は図 5.8 のように表現できる．この素性構造では八長調における D 項の和音列 ‘IV - ii’ が表現されている．素性 ‘DSEQ (DEGREE-SEQUENCE)’ は本来 unnecessary 素性であるが，和音の進行を見やすく表示するために導入した．図のように，HEAD-DTR 素性に主辞の子である ‘IV’ の和音の素性構造を，COMP-DTR 素性に補語の子である ‘ii’ の和音の素性構造を格納することで和音列を表現する．

$$\left[\begin{array}{l} \text{PHON} \langle F\text{-Dm} \rangle \\ \text{DSEQ} \langle IV\text{-ii} \rangle \\ \text{DTRS} \left[\begin{array}{l} \text{HEAD-DTR} \left[\begin{array}{l} \text{PHON} \langle F \rangle \\ \text{DSEQ} \langle IV \rangle \\ \text{SYNSEM||HEAD} \left[\begin{array}{ll} \text{subdominant} & \\ \text{DEG} & IV \\ \text{TONAL} & C \\ \text{TYPE} & K2 \end{array} \right] \end{array} \right] \\ \text{COMP-DTR} \left[\begin{array}{l} \text{PHON} \langle Dm \rangle \\ \text{DSEQ} \langle ii \rangle \\ \text{SYNSEM||HEAD} \left[\begin{array}{ll} \text{subdominant} & \\ \text{DEG} & ii \\ \text{TONAL} & C \\ \text{TYPE} & K2 \end{array} \right] \end{array} \right] \end{array} \right] \end{array} \right]$$

図 5.8: 素性構造による和音列の表現例

5.1.3 素性構造によるカデンツ規則の表現

2.10 項で和声学における基本的なカデンツ規則について述べた．ここでは和音列から図 5.1 のような木構造を見出すために，Tonic，Dominant，Subdominant の各項の和音進行と，三つのカデンツ規則を素性構造を用いて定義する．

T 項は I と VI の和音の集合であり，可能な和音の進行は I, VI, I-VI の 3 種類である．ここで I の和音の HEAD の素性構造を (5.1)，VI の和音の HEAD の素性構造を (5.2) のように定義する．これらを用いて T 項の和音進行は (5.3)，(5.4)，(5.5) のように定義できる．SUBCAT 素性の先の素性 'R' は自身の右側の和音を下位範疇化することを意味する．また，ここでは '~' を用いて素性構造全体のタイプを表現している．

(5.3) は I の和音のみによる T 項を表現し，(5.5) は VI の和音のみによる T 項を表現している．(5.4) は I - VI の和音進行による T 項を表現しており，主辞の子である I の和音が補語の子である vi の和音を下位範疇化して T 項を形成している．このような操作を行うとき，全素性構造中の素性 'TONAL' と 'TYPE' の型は統一されている必要がある．

$$\boxed{1} \left[\begin{array}{l} \text{tonic} \\ \text{DEG} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \begin{array}{l} \\ \\ \boxed{3} \\ \boxed{4} \end{array} \quad (5.1)$$

$$\boxed{2} \left[\begin{array}{l} \text{tonic} \\ \text{DEG} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \begin{array}{l} \\ \\ \boxed{3} \\ \boxed{4} \end{array} \quad (5.2)$$

$$\left[\begin{array}{l} \sim T \\ ||\text{HEAD} \boxed{1} \end{array} \right] \rightarrow \left[\begin{array}{l} \sim T \\ ||\text{HEAD} \boxed{1} \end{array} \right] \quad (5.3)$$

$$\left[\begin{array}{l} \sim T \\ ||\text{HEAD} \boxed{1} \end{array} \right] \rightarrow \left[\begin{array}{l} \sim T \\ ||\text{HEAD} \boxed{1} \\ ||\text{SUBCAT|R} \langle \boxed{5} \rangle \end{array} \right], \boxed{5} \left[\begin{array}{l} \sim T \\ ||\text{HEAD} \boxed{2} \end{array} \right] \quad (5.4)$$

$$\left[\begin{array}{l} \sim T \\ ||\text{HEAD} \boxed{2} \end{array} \right] \rightarrow \left[\begin{array}{l} \sim T \\ ||\text{HEAD} \boxed{2} \end{array} \right] \quad (5.5)$$

S 項は IV と II の和音の集合であり，可能な和音の進行は IV, II, IV-II の 3 種類である．IV の和音の HEAD の素性構造を (5.6)，II の和音の HEAD の素性構造を (5.7) のように定義すると，S 項の和音進行は (5.8)，(5.9)，(5.10)，のように定義できる．(5.8) は IV の和音による S 項を，(5.10) は II の和音による S 項を表現している．また (5.9) は IV - II の和音進行による S 項を表現している．Subdominant は K1 のカデンツを構成することはないため，TYPE 素性には K2, K3 のいずれかが格納される．

$$\boxed{1} \begin{bmatrix} \text{subdominant} \\ \text{DEG} \\ \text{TONAL} \\ \text{TYPE} \end{bmatrix} \begin{bmatrix} \text{IV} \\ \boxed{3} \\ \boxed{4} \end{bmatrix} \quad (5.6)$$

$$\boxed{2} \begin{bmatrix} \text{subdominant} \\ \text{DEG} \\ \text{TONAL} \\ \text{TYPE} \end{bmatrix} \begin{bmatrix} \text{II} \\ \boxed{3} \\ \boxed{4} \end{bmatrix} \quad (5.7)$$

$$\begin{bmatrix} \sim\bar{S} \\ ||\text{HEAD} \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \sim S \\ ||\text{HEAD} \boxed{1} \end{bmatrix} \quad (5.8)$$

$$\begin{bmatrix} \sim\bar{S} \\ ||\text{HEAD} \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \sim S \\ ||\text{HEAD} \boxed{1} \\ ||\text{SUBCAT|R} \langle \boxed{5} \rangle \end{bmatrix}, \boxed{5} \begin{bmatrix} \sim S \\ ||\text{HEAD} \boxed{2} \end{bmatrix} \quad (5.9)$$

$$\begin{bmatrix} \sim\bar{S} \\ ||\text{HEAD} \boxed{2} \end{bmatrix} \rightarrow \begin{bmatrix} \sim S \\ ||\text{HEAD} \boxed{2} \end{bmatrix} \quad (5.10)$$

D 項は一つの和音から構成される．V の和音の HEAD の素性構造を (5.11) のように定義すると，D 項は (5.12) のように定義できる．Dominant の TYPE 素性には K1, K2 のいずれかが格納される．

$$\boxed{1} \begin{bmatrix} \text{dominant} \\ \text{DEG} \\ \text{TONAL} \\ \text{TYPE} \end{bmatrix} \begin{bmatrix} \text{V} \\ \boxed{2} \\ \boxed{3} \end{bmatrix} \quad (5.11)$$

$$\begin{bmatrix} \sim\bar{D} \\ ||\text{HEAD} \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \sim D \\ ||\text{HEAD} \boxed{1} \end{bmatrix} \quad (5.12)$$

本稿では ‘III’ と ‘VII’ の和音は扱わない．これらの和音は ‘I₇’ の和音もしくは ‘V₇’ の和音の根音省略形態として捉える．以上の各項の定義を用いてカデンツ規則は (5.13), (5.14), (5.15) のように定義できる．SUBCAT 素性の先の素性 ‘L’ は自身の左側の和音を下位範疇化することを意味する．(5.13) は K3 に相当し，終結 T 項が S 項，開始 T 項を順に下位範疇化する．(5.14) は K1 に相当し，終結 T 項が D 項，開始 T 項を順に下位範疇化する．(5.15) は K2 に相当し，D 項が S 項を下位範疇化した後，終結 T 項が D 項，開始 T 項を下位範疇化する．

$$\begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{4} \end{bmatrix} \rightarrow \boxed{5} \begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{1} \end{bmatrix}, \boxed{6} \begin{bmatrix} \sim\bar{S} \\ ||\text{HEAD} \boxed{2} \end{bmatrix}, \begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{4} \\ ||\text{SUBCAT|L} \langle \boxed{6}, \boxed{5} \rangle \end{bmatrix} \quad (5.13)$$

$$\begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{4} \end{bmatrix} \rightarrow \boxed{5} \begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{1} \end{bmatrix}, \boxed{7} \begin{bmatrix} \sim\bar{D} \\ ||\text{HEAD} \boxed{3} \end{bmatrix}, \begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{4} \\ ||\text{SUBCAT|L} \langle \boxed{7}, \boxed{5} \rangle \end{bmatrix} \quad (5.14)$$

$$\begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{4} \end{bmatrix} \rightarrow \boxed{5} \begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{1} \end{bmatrix}, \boxed{6} \begin{bmatrix} \sim\bar{S} \\ ||\text{HEAD} \boxed{2} \end{bmatrix}, \boxed{7} \begin{bmatrix} \sim\bar{D} \\ ||\text{HEAD} \boxed{3} \\ ||\text{SUBCAT|L} \langle \boxed{6} \rangle \end{bmatrix}, \begin{bmatrix} \sim\bar{T} \\ ||\text{HEAD} \boxed{4} \\ ||\text{SUBCAT|L} \langle \boxed{7}, \boxed{5} \rangle \end{bmatrix} \quad (5.15)$$

本稿では，SUBCAT 素性の先に ‘L’，‘R’ 素性を用意し下位範疇化する向きを指定できるようにしている．実際はこのように方向を指定するようなことはしなくても，カデンツ構造の解析は可能である．しかし方向の指定は解析の精度を高める上で非常に重要である．例えば和音認識によって ‘I - V - ii - I’ のような和音列が得られ，この和音列に対して和音進行解析を行う場合，カデンツのルールでは ‘V - ii’ の進行は許されていないことから，この和音列がカデンツとして相応しくないということを認識しなければならない．しかし，下位範疇化の方向を指定せずにカデンツの解析を行うと，図 5.9 のように和音列 ‘V-ii’ から誤った木構造が解析されてしまう．この結果，誤ったカデンツ構造が認識され，システムの解析精度が低下する．このような誤認識を避けるために ‘L’，‘R’ 素性は重要な役割を果たしている．

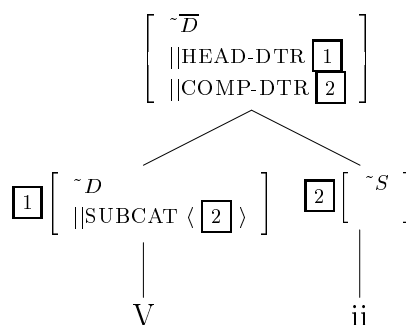


図 5.9: 誤った解析例

5.1.4 ID-スキーマと Principle

前節で定義したカデンツの規則に基いてカデンツの構造を解析するために 2 つの ID-スキーマと 3 つのプリンシプルを定義する．まず，ID-スキーマを (5.16), (5.17) に示す．

$$\left[\text{DTRS} \left[\begin{array}{l} \text{HEAD-DTR} \\ \text{COMP-DTRS} \end{array} \left[\begin{array}{l} \text{SYNSEM||SUBCAT|R} \\ \boxed{1} \end{array} \right] \right] \right] \quad (5.16)$$

$$\left[\text{DTRS} \left[\begin{array}{l} \text{HEAD-DTR} \\ \text{COMP-DTRS} \end{array} \left[\begin{array}{l} \text{SYNSEM||SUBCAT|L} \\ \boxed{1} \end{array} \left[\langle \boxed{1}, \dots \rangle \right] \right] \right] \right] \quad (5.17)$$

(5.16) は主辞の子が左側で補語の子が右側に位置する場合に適用されるスキーマである．主辞の子の SUBCAT|R 素性の要素と補語の子が矛盾なく単一化されなければならないことが表現されている．このスキーマは同一の項の素性構造同士から親の素性構造を導く際に用いられる．(5.17) は主辞の子が右側で補語の子が左側に位置する場合に適用されるスキーマである．主辞の子の SUBCAT|L 素性の先頭の要素と補語の子が，矛盾なく単

一化されなければならないことが表現されている．このスキーマは異なる項の素性構造同士から親の素性構造を導く際に用いられる．

Head Feature Principle は (5.18) のように，3.4.2 節で紹介したものと同一のものを用いる．

$$\left[\begin{array}{l} \text{SYNSEM||HEAD} \boxed{1} \\ \text{DTRS|HEAD-DTR} \left[\text{SYNSEM||HEAD} \boxed{1} \right] \end{array} \right] \quad (5.18)$$

Subcategorization Principle は (5.19), (5.20) の2種類を定義する．まず，(5.19) の Subcategorization Principle は (5.16) の ID-スキーマが適用された際に用いる．主辞の子の SUBCAT|L 素性はすべて親に伝え，下位範疇化済みの SUBCAT|R 素性の内容は親に伝えない．続いて，(5.20) の Subcategorization Principle は (5.17) の ID-スキーマが適用された際に用いる．主辞の子の SUBCAT|L 素性にある要素のうち，先頭の要素を除いたものを親の SUBCAT|R 素性に伝える．この時，SUBCAT|R 素性の要素は親に伝える必要はない．これは (5.17) の ID-スキーマが適用された時点で解析が異なる項同士を統合するステップに移っているためである．また，SUBCAT|R 素性の要素を親に伝えないことによって誤った木構造の出力を避けることができる．この例を図 5.10 に示す．同図の ‘I - V - I - VI’ というカデンツからは，まず ‘I - VI’ の終結 T 項の木構造が生成され，続いて ‘V - I - VI’ の木構造が生成されるべきである．しかしこの例では先に ‘V - I’ の木構造が生成されてしまっている．この時， $\boxed{1}$ の SUBCAT|R 素性の要素を親の $\boxed{4}$ に伝えなければそれ以上解析を進めることが不可能となるため，解析の順序に誤りがあったことが認識される．しかし，この例では SUBCAT|R 素性の要素を親に伝えたために， $\boxed{4}$ が $\boxed{3}$ の ‘VI’ の和音を下位範疇化してしまい，全体として誤った木構造が認識されている．このような誤認識を避けるために，(5.17) の ID-スキーマが適用された後，生成された親に主辞の子の SUBCAT|R 素性の要素を伝えないことは非常に重要である．

$$\left[\begin{array}{l} \text{SYNSEM||SUBCAT|L} \langle \boxed{1}, \boxed{2}, \dots \rangle \\ \text{DTRS} \left[\begin{array}{l} \text{HEAD-DTR} \left[\text{||SUBCAT} \left[\begin{array}{l} \text{L} \langle \boxed{1}, \boxed{2}, \dots \rangle \\ \text{R} \boxed{3} \end{array} \right] \right] \\ \text{COMP-DTRS} \boxed{3} \end{array} \right] \end{array} \right] \quad (5.19)$$

$$\left[\begin{array}{l} \text{SYNSEM||SUBCAT|L} \langle \boxed{2}, \dots \rangle \\ \text{DTRS} \left[\begin{array}{l} \text{HEAD-DTR} \left[\text{||SUBCAT} \left[\begin{array}{l} \text{L} \langle \boxed{1}, \boxed{2}, \dots \rangle \\ \text{R} \boxed{3} \end{array} \right] \right] \\ \text{COMP-DTRS} \boxed{1} \end{array} \right] \end{array} \right] \quad (5.20)$$

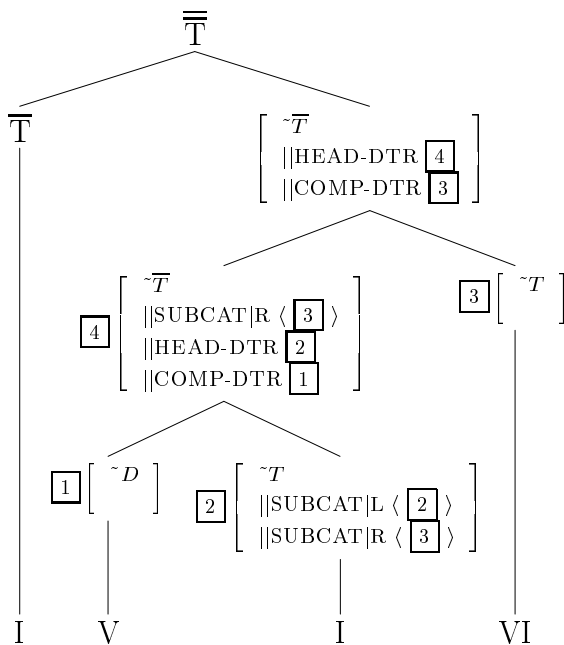


図 5.10: 誤った木構造

5.1.5 カデンツ構造の解析

5.1.3 節で定義したカデンツ規則と前節で定義した ID-スキーマ・プリンシプルを用いて、和音列からカデンツ構造を解析するプロセスを図 5.1 の和音列 ‘C - Am - F - Dm - G - C’ を例に解説する。この和音列は八長調の K2 のカデンツであり、音度表記を用いて ‘I - vi - IV - ii - V - I’ と表記できる。

まず、先頭の ‘I - vi’ の和音列から \bar{T} の木構造を生成する。前節で定義したカデンツの規則より、‘I’ と ‘vi’ の和音の素性構造を以下のように定義する。この二つの素性構造では、この二つの和音が共に左向きに Dominant の和音と Subdominant の和音を下位範疇化することが表現されている。また (5.21) の素性構造では ‘I’ の和音は右向きに ‘vi’ の Tonic 和音を下位範疇化することが表現されている。

$$\left[\begin{array}{l} \text{PHON} \langle C \rangle \\ \text{DSEQ} \langle I \rangle \end{array} \right] \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic} \\ \text{DEG} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \left[\begin{array}{l} I \\ C \\ K2 \end{array} \right] \\ \text{SUBCAT} \left[\begin{array}{l} L \left\langle \left[\begin{array}{l} ||\text{HEAD} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \left[\begin{array}{l} C \\ K2 \end{array} \right] \right\rangle, \\ \left[\begin{array}{l} \text{tonic} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \left[\begin{array}{l} C \\ K2 \end{array} \right] \right\rangle \\ R \left[\begin{array}{l} ||\text{HEAD} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \left[\begin{array}{l} \text{tonic} \\ \text{DEG} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \left[\begin{array}{l} vi \\ C \\ K2 \end{array} \right] \end{array} \right] \end{array} \right] \quad (5.21)$$

$$\left[\begin{array}{l} \text{PHON} \langle \text{Am} \rangle \\ \text{DSEQ} \langle \text{vi} \rangle \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic} \\ \text{DEG} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \left[\begin{array}{l} \text{vi} \\ \text{C} \\ \text{K2} \end{array} \right] \\ \text{SUBCAT} \left[\text{L} \left\langle \begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{dominant} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \left[\begin{array}{l} \text{C} \\ \text{K2} \end{array} \right] \\ \text{||HEAD} \left[\begin{array}{l} \text{tonic} \\ \text{TONAL} \\ \text{TYPE} \end{array} \right] \left[\begin{array}{l} \text{C} \\ \text{K2} \end{array} \right] \end{array} \right\rangle \right] \end{array} \right] \end{array} \right] \end{array} \right] \quad (5.22)$$

この二つの素性構造から先頭の \bar{T} の木構造を図 5.11 のように導く．この木構造を生成する際は，まず (5.16) の ID-スキーマの適用に成功する．その後，(5.18) の Head Feature Principle の適用によって ‘I’ の和音の HEAD 素性の情報が親に伝えられ，(5.19) の Subcategorization Principle の適用によって，親が下位範疇化すべき和音の情報が伝えられる．

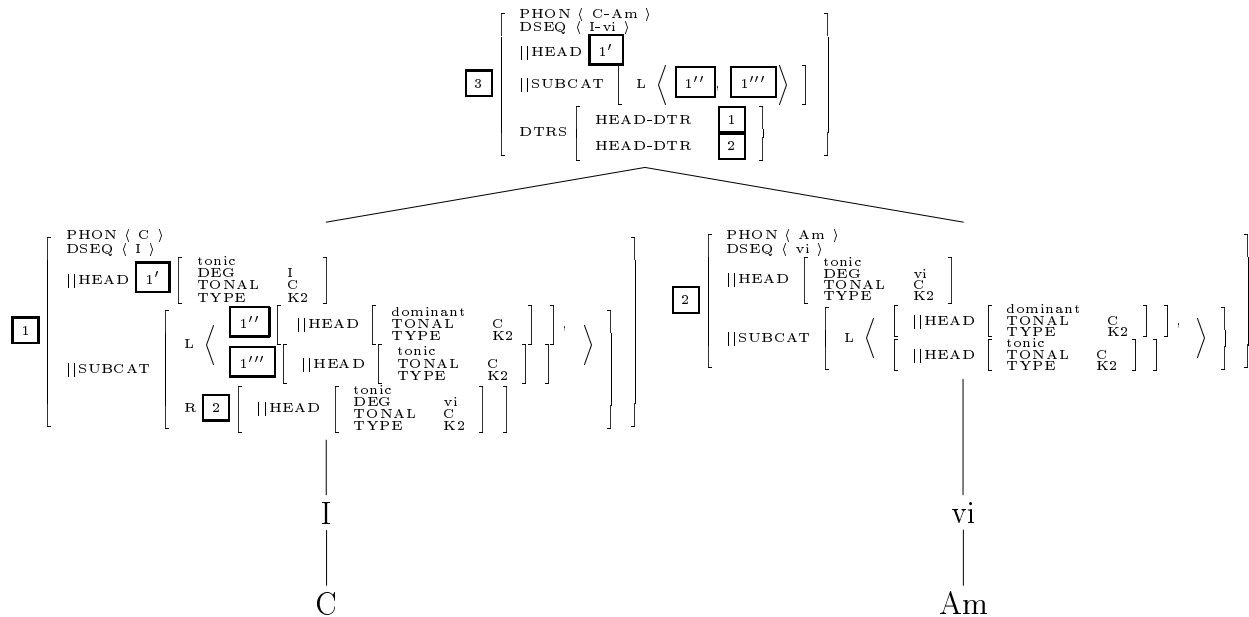


図 5.11: \bar{T} の木構造

次に \bar{S} の木構造を生成する．この木構造を図 5.12 に示す．‘IV’ と ‘ii’ の和音は [4]，[5] のように定義する．これらに対し先ほど同様の操作を行って [6] の素性構造を導き S 項の木構造を求めることができる．

続いて異なる項同士から木構造を見出すプロセスに移る．まず，先ほどの \bar{S} と Dominant の ‘V’ の和音から図 5.13 のような木構造が得られる．‘V’ の和音の素性構造は [7] のように定義する．ここでは右側の ‘V’ の和音が \bar{S} を下位範疇化し \bar{D} を形成している．このとき適用された ID-スキーマは (5.17) であり，プリンシプルは (5.18) の Head Feature Principle と (5.20) の Subcategorization Principle である．

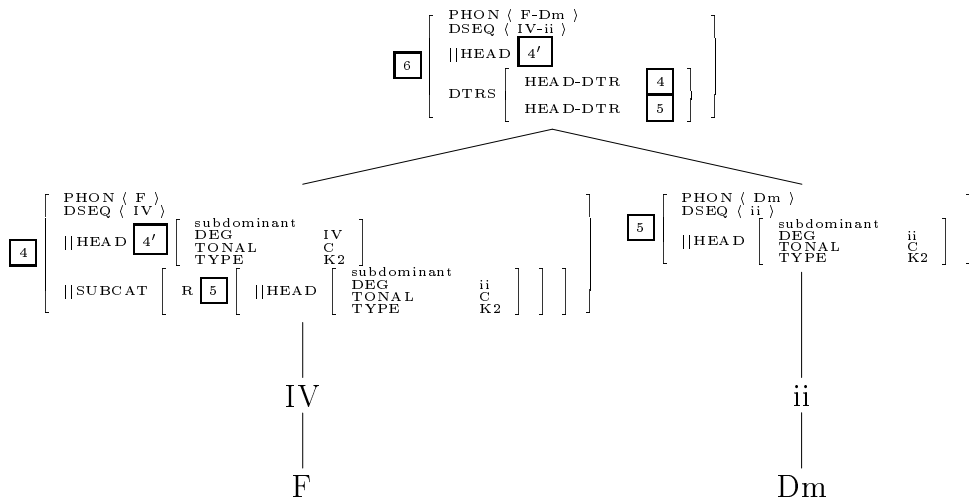


図 5.12: \bar{S} の木構造

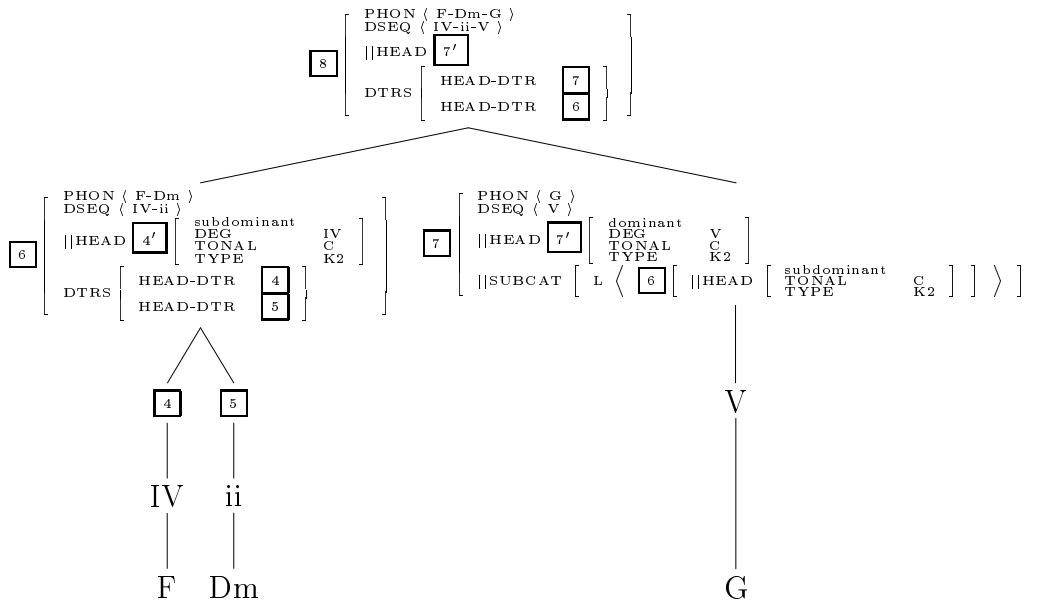


図 5.13: \bar{D} の木構造

次に、先ほど生成した \bar{D} と最後の 'I' の和音から 図 5.14 のように \bar{T} の木構造が得られる。この木構造は Tonic である '9' の 'I' の和音が '8' の \bar{D} を下位範疇化することで得られる。この時用いられる ID-スキーマやプリンスプルは先ほどと同様である。そして、このとき得られた '10' の \bar{T} が最初に得られた '3' の \bar{T} を下位範疇化し、一つの木構造が得られる。これを図 5.15 に示す

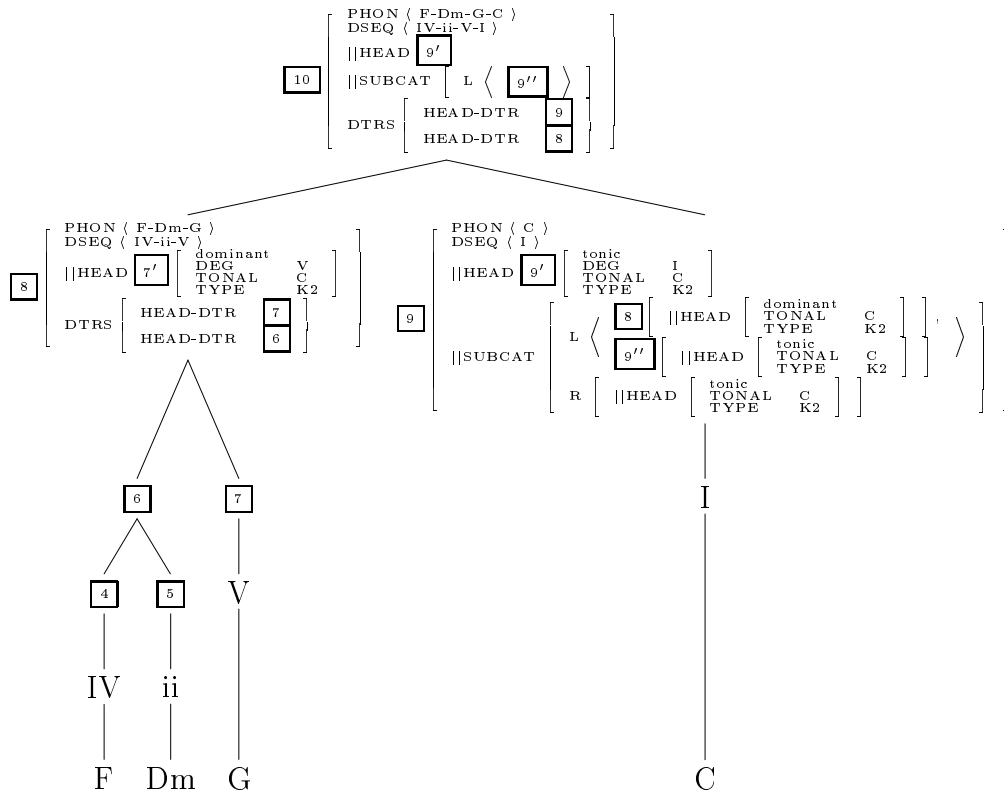


図 5.14: \bar{T} の木構造

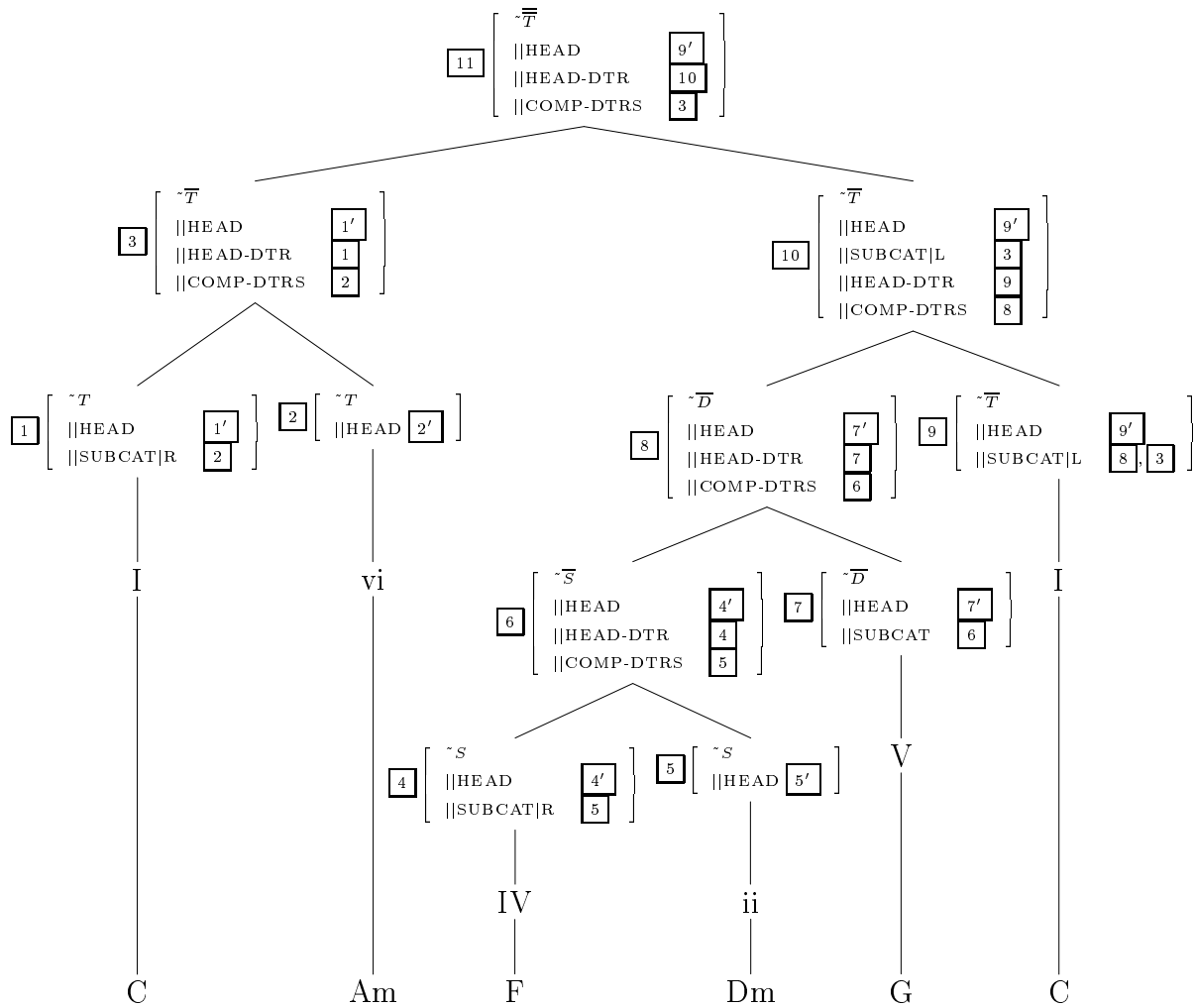


図 5.15: cadence の木構造

5.1.6 カデンツ規則の拡張

5.1.3 節では基本的なカデンツ規則を素性構造を用いて定義した．この定義は辞書の情報を充実あるいは削除することによって簡単に拡張することができる．ここでは，書籍：和声 理論と実習 I [11] のカデンツ規則に基く拡張を行う．このカデンツ規則では，K3 の T-S-T カデンツにおける S 項は必ず IV の和音であり 終結 T 項は必ず I の和音でなければならない，と定義されている．先に定義したカデンツルールでは K3 の S 項には IV, II 両方の和音とこれらの連用が可能であり，終結 T 項は I, VI 両方の和音とこの連用が可能であった．そこで，辞書から TYPE 素性に K3 の型を持つ II の和音の辞書記述を削除し，K3 の型を持つ IV の和音の SUBCAT|R 素性を空にする．これによって II の和音が含まれる S 項は必ず K2 となり，K3 の S 項が IV の和音のみであることが保証される．続いて，Subdominant を直接下位範疇化する VI の和音の辞書記述を削除し，K3 の I の和音の SUBCAT|R 素性を空にする．これによって VI の和音が Subdominant を下

位範疇化できなくなり，K3 の I の和音が VI の和音を下位範疇化できなくなったことから，K3 の終結 T 項は必ず I の和音となる．

次に，旋律的短音階に対応するための拡張を行う．旋律的短音階のカデンツの D 項では短三和音に加えて長三和音も用いることができる．これに対応するには，辞書項目に (5.23) のような HEAD 素性を持つ和音の素性構造を追加すればよい．

$$\left[\begin{array}{l} \text{dominant} \\ \text{DEG} \quad \text{V} \\ \text{TONAL} \quad \text{Cm} \\ \text{TYPE} \end{array} \right] \quad (5.23)$$

続いて，解析の精度を上げるための拡張を行う．現在の規則では‘V-I-V-I’といった和音進行から誤ったカデンツ構造が認識され，カデンツの範囲を正しく認識することができない．これを図 5.16 に示す．この和音列はカデンツとして妥当な形ではない．しかし現在の素性構造定義ではすべての‘I’の和音が左向きに Dominant もしくは Subdominant を下位範疇化できるために，先頭の‘V’の和音が続く‘I’の和音に下位範疇化され，全体として誤った木構造が認識されてしまう．この誤認識は二つの T 項が共に終結 T 項の機能を持つために発生する．これを避けるには開始 T 項と終結 T 項に対して異なった素性構造を定義すればよい．そこで型 tonic に対して図 5.17 のような階層を与える．‘tonic_h’は主辞となる Tonic 和音を，‘tonic_c’は補語となる Tonic 和音を意味する．この内，‘tonic_c’の型を持つ Tonic 和音の SUBCAT|L 素性を空にすることでこのような誤認識を回避することが可能となる．

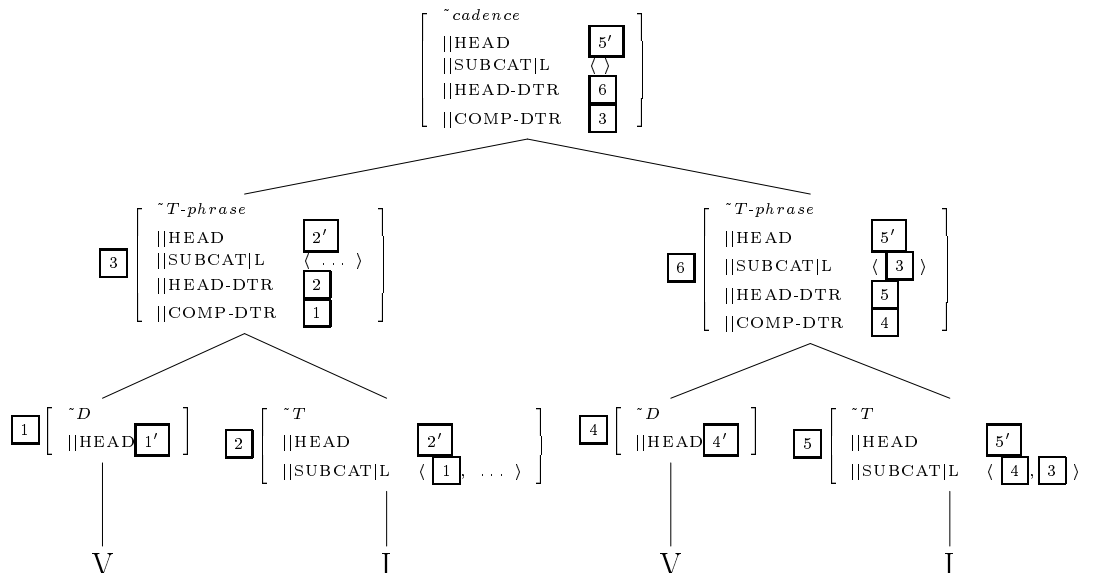


図 5.16: 誤ったカデンツ構造の認識

加えて，辞書記述に構造共有を用いることによって辞書項目の記述量を削減することができる．カデンツのタイプは終結 T 項が何を下位範疇化するかによって決定すると

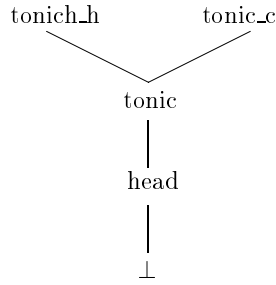


図 5.17: tonic に関する型階層

える．単体の D 項を下位範疇化すれば K1，S 項を下位範疇化すれば K3，S 項 を下位範疇化している D 項を下位範疇化すれば K2 となる．このため，Suddominant，Dominant に関する型階層を図 5.18 のように定義し，それぞれの素性構造の TYPE 素性に形成するカデンツのタイプを格納する．続いて，構造共有を用いて Tonic の和音を図 5.19 のように定義することで，下位範疇化した Dominant もしくは Subdominant から TYPE 素性の型が伝わるので，各カデンツタイプの Tonic を定義する必要がなくなる．しかし，本稿では次章で解説するシステムが辞書記述中での構造共有に対応できないためにこの拡張は用いない．今後導入を検討する．

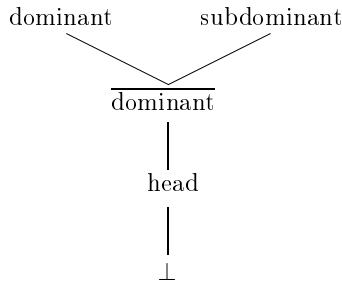


図 5.18: dominant に関する型階層

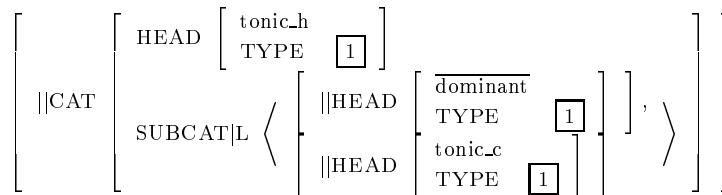


図 5.19: Tonic の素性構造

以上の拡張に準じた各和音の素性構造を C Major Scale を例として章末に定義する．こ

れらを辞書項目に登録する際は，‘調名：和音名’の形の検索キーを与えて登録する．また，七の和音についても同様の素性構造を与える．

5.2 和音進行解析を利用した調性認識

本節では先に述べた和音進行解析手法を利用した調性認識手法について述べる．4.1 節において，楽曲の調はメロディーから解析するのではなく和音の進行から解析すべきであると述べた．これは調ごとに和音の進行が異なるためである．本稿で提案する調性認識手法を図 5.20 の Mozart Piano Sonate in C, K545. を例として解説する．

和音名： C G C F C G C

図 5.20: Mozart Piano Sonate in C, K545.

まず 1 つのカデンツを形成する和音列の調の認識について述べる．図 5.20 中の和音列 ‘C-G-C’ には (5.24) のように 4 種類の調に基く音度表記を与えることができる．

$$\begin{array}{r}
 C : I - V - I \\
 G : IV - I - IV \\
 C - G - C \rightarrow \begin{array}{l} \text{Em} : VI - III - VI \\ \text{Am} : III - VII - III \end{array} \\
 \end{array} \quad (5.24)$$

これらに対し，Tonic，Subdominant，Dominant の各和音機能を与えると (5.25) のようになる．このうち，前節で述べた和音進行解析によってカデンツ構造が見出せるのは ‘C : T-D-T’ のフレーズだけである．このため，C-G-C という和音列の調は八長調であるということが分かる．

$$\begin{array}{r}
 C : T - D - T \\
 G : S - T - S \\
 \text{Em} : T - ? - T \\
 \text{Am} : ? - ? - ? \\
 \end{array} \quad (5.25)$$

また，カデンツが幾つか連鎖する和音列からも調を解析することができる．例えば，図 5.20 の和音列 ‘C-G-C-F-C’ からは (5.26) のように 3 種類の調に対応する音度表記列が得られるが，そこから過不足なくカデンツを見出せるのは八長調の音度表記列のみである．

$$\begin{array}{r}
\text{sequence : C - G - C - F - C} \\
\text{C : T - D - T - S - T} \\
\text{G : S - T - S - - - S} \\
\text{F : D - - - D - T - D}
\end{array} \tag{5.26}$$

この手法ではカデンツごとに調を認識するため、転調も認識することができる。例えば、(5.27) の和音列は先頭の C-G-C が C Major Scale のカデンツ、その後の F-B \flat -C-F が F Major Scale のカデンツであることが認識できることから、F の和音を境いに C Major Scale から F Major Scale に転調していることが分かる。

$$\begin{array}{r}
\text{sequence : C - G - C - F - B}\flat - C - F \\
\text{C : T - D - T - S - - - T - S} \\
\text{G : S - T - S - - - S -} \\
\text{F : D - - - D - T - S - D - T}
\end{array} \tag{5.27}$$

このように、和音の進行を考慮して調を解析することで精度の高い調性認識を行うことができる。また、従来は楽曲の調が認識されていないと和音進行解析を行うことは出来ないと考えられていた。これは調の情報を基に和音に対して音度表記を与えなければならないためである。しかしカデンツを形成する任意の和音列は必ず1つの調に基いていることから、高い精度でカデンツ構造を認識することができれば、同時に調も解析することが可能である。

5.3 和音進行解析と絡めた和音認識

5.3.1 和音認識における曖昧性を低減する手法

本和音進行解析手法は、和音列からカデンツと調性を認識する際に和音列に含まれる曖昧性をある程度解消することが可能である。例えば、和音認識によって (5.28) のような和音列が認識されたとする。この和音列は1番目の和音が C または E m の和音、2番目の和音が E m または G の和音という曖昧性を含んでいる。

$$\left\{ \begin{array}{c} \text{C} \\ \text{E}m \end{array} \right\} - \left\{ \begin{array}{c} \text{E}m \\ \text{G} \end{array} \right\} - \text{A}m \tag{5.28}$$

この和音列を展開すると (5.29) のように4通りの和音列が得られ、これらに対し可能な調名と音度表記の組み合わせを与えると (5.30) のように8通りの進行を得られる。この時、本和音進行解析手法によってカデンツの構造が見出せるのは和音列 'C-G-A m ' を八長調として捉えた 'C : T-D-T' のみである。このことから、1番目の和音は C であり2

番目の和音は G であるということが分かり，同時に調は八長調で K1 のカデンツであるということが認識される．

$$\begin{array}{l}
 C - Em - Am \\
 C - G - Am \\
 Em - Em - Am \\
 Em - G - Am
 \end{array} \tag{5.29}$$

$$\begin{array}{l}
 C - Em - Am \rightarrow \left\{ \begin{array}{l} G: IV - vi - ii \\ Em: VI - i - iv \end{array} \right\} \rightarrow \left\{ \begin{array}{l} G: S - T - S \\ Em: T - T - S \end{array} \right\} \\
 C - G - Am \rightarrow \left\{ \begin{array}{l} C: I - V - vi \\ G: IV - I - ii \end{array} \right\} \rightarrow \left\{ \begin{array}{l} C: T - D - T \\ G: S - T - D \end{array} \right\} \\
 Em - Em - Am \rightarrow \left\{ \begin{array}{l} G: vi - vi - ii \\ Em: i - i - iv \\ Am: v - v - i \end{array} \right\} \rightarrow \left\{ \begin{array}{l} G: T - T - S \\ Em: T - T - S \\ Am: D - D - T \end{array} \right\} \\
 Em - G - Am \rightarrow \left\{ G: vi - I - ii \right\} \rightarrow \left\{ G: T - T - S \right\}
 \end{array} \tag{5.30}$$

5.3.2 和音認識のアルゴリズム

本稿は提案する和音進行解析手法の有効性を示すことを目的としているため，和音認識は簡易的な手法を2つ用意した．1つ目の手法では，図5.21のように手入力で与えた各和音範囲から和音を認識する．このため，この和音認識では和音名の特定に関する曖昧性のみが発生する．認識する和音は基本三和音，七の和音，七の和音の根音省略形態，基本三和音の5度音省略形態，基本三和音の3度音と5度音の省略形態の5種類とした．和音を認識する際のフローチャートを図5.22に示す．同図に示されるように，この手法ではまず基本三和音の認識を行い，基本三和音が認識された場合はそれが七の和音を形成していないかを調べる．基本三和音が認識されなければ七の和音の根音省略形態や基本三和音の5度音省略形態の認識を行う．和声学では基本三和音の5度音の省略と V_7 の和音の根音

The image shows a musical score for Mozart's Piano Sonata in C, K545. It consists of two staves: a treble clef staff and a bass clef staff. The treble staff contains a sequence of notes: C4, E4, G4, A4, B4, C5, B4, A4, G4, F4, E4, D4, C4. The bass staff contains a sequence of notes: C3, E3, G3, A3, B3, C4, B3, A3, G3, F3, E3, D3, C3. Brackets are placed under the notes in the bass staff to indicate the chords: C (C3-E3-G3), G7 (G3-B3-D4-F4), C (C3-E3-G3), F (F3-A3-C4), C (C3-E3-G3), G7 (G3-B3-D4-F4), C (C3-E3-G3), and Em7 (C3-E3-G3-A3). The chords are labeled C, G7, C, F, C, G7, C, and Em7 from left to right.

図 5.21: 和音認識 Mozart Piano Sonate in C, K545.

の省略が許されている．このためこれらの認識を優先的に行うことにした．また， V_7 の根音省略形態は省略された根音の長3度上の音を根音とした減三和音，つまり VII の和音である．このため，減三和音を認識することで七の和音の根音省略形態を認識することができる．

以上の和音を認識できなかった場合，基本三和音の3度音省略形態や3度音と5度音の省略形態などの認識を行う．これらの形態は和声学においては許されてはいないが，実際の楽曲においてはしばしば見られる和音形態である．

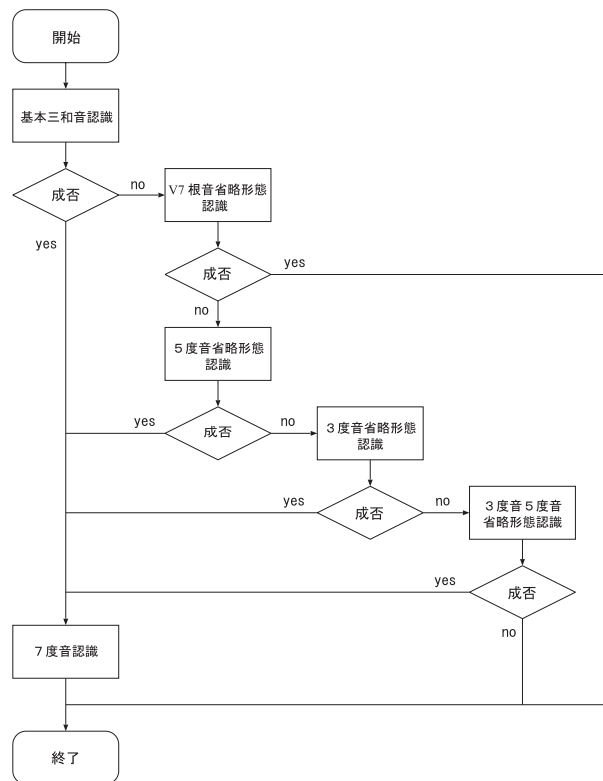


図 5.22: 和音認識 フローチャート

続いて，各認識ステップでのパターンマッチング処理について説明する．基本三和音の認識ではまず機械的に12通りの長三和音と短三和音の雛形を生成し，指定された範囲範囲に含まれる音と各和音の雛形との一致度を以下の計算式で求め，上位2位以内のものを解として出力する．

$$\text{評価値} = \left(\frac{\text{雛形の構成音と一致する音の長さの総和}}{\text{雛形の構成音と一致しない音の長さの総和 (7度音を除く)}} \right)$$

この計算式は指定された範囲に含まれる音が各和音の雛形に対してどれだけ一致し，且つどれだけ一致しないかを求めるものである．このように不一致度も考慮することによって和音認識に発生する曖昧性を低減することが可能である．また，音と音の一致をオクターブの差を無視して判断することで転回型の和音も認識することができる． V_7 の根音

省略形態など任意の音が省略されている和音も同様の雛形を用いて認識することが可能である。

もう1つの和音認識手法では、図 5.23 のように音符列を4分音符もしくは8分音符ごとに区切り、各範囲から和音列を認識し、その後認識された和音列中の連続した同名の和音を1つに統合する。この手法では和音の範囲の認識にも曖昧性が発生することとなる。この和音認識では楽曲を非常に細かく細分化することから、各範囲に必ず根音などが含まれているとは限らなくなるため、和音の根音を省略した和音形態などの認識も行う。

5.4 和声解析

本稿で行う和声解析を図 5.24 に示す。本手法ではまず和音認識を行って和音列を生成し、和音列に対し和音進行解析を行うことでカデンツ列と調名の認識を行う。また、これと同時に和音列中に含まれる曖昧性の解消も行う。

本手法では連鎖するカデンツ列を認識するために、図 5.25 に示すスライディングウィンドウ方式を用いて和音列を適当な長さに区切り、切り出した和音列に対して順に和音進行解析を行っていく。この手法では和音列中に含まれる全カデンツを認識することができるが、(5.31) の K3 ようなカデンツも認識される。このカデンツは局所的には認識されること自体誤りではないが、カデンツの前後関係を考慮すると相応しくないカデンツである。本稿ではこのようなカデンツを排除するような試みは行っていないが、カデンツの認識精度が高ければ正しいカデンツが必ず連続して認識されるため、このようなカデンツを排除することが可能となる。

$$\begin{aligned}
 C - G - C - G - C &\rightarrow \left\{ \begin{array}{l} C: I - V - I - V - I \\ G: IV - I - IV - I - IV \end{array} \right\} \\
 &\rightarrow \left\{ \begin{array}{l} C: \leftarrow K1 \rightarrow \leftarrow K1 \rightarrow \\ G: ? \rightarrow \leftarrow K3 \rightarrow \leftarrow ? \end{array} \right\} \quad (5.31)
 \end{aligned}$$

カデンツ列は図 5.24 のように Tonic の和音を共有して連鎖する形で出力し、連鎖するいくつかのカデンツを含むさらに大きな木構造の生成は行わない。この作業はメロディーやフレーズ、リズムなどの情報を基に行うべきであり、和声の情報のみで行うことはできない。また、2つのカデンツに共有される Tonic 項が 'I - VI' のような和音進行の場合、厳密なカデンツの切れ目がどこであるのかについても言及はしない。この作業もリズムなどの情報を基に行う必要がある。また、実際の切れ目が本質的に曖昧なこともあるため、これを特定することは非常に難しい問題である。

C C C C G G7 C C
 Cm Em Cm Em
 C G7 C

図 5.23: 4分音符区切りの和音認識

K1 K3 K1
 T D T S T D T
 C:I C:V C:I C:IV C:I C:V C:I
 C G C F C G C

図 5.24: 和声解析例 Mozart Piano Sonate in C, K545.

和音列 : C - Am - F - Dm - G - C - Dm - G - C ...

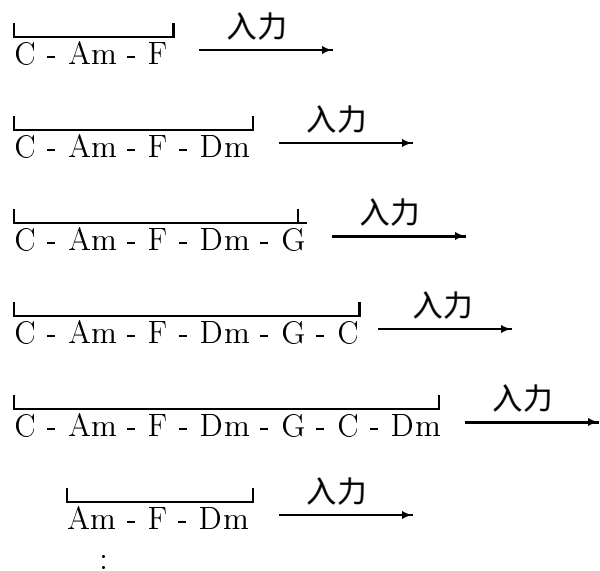


図 5.25: スライディングウィンドウによる和音の区切り方

素性構造

$$\left[\begin{array}{l} \text{PHON} \langle C \rangle \\ \text{DSEQ} \langle I \rangle \\ \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic_c} \\ \text{DEG} \quad I \\ \text{TONAL} \quad C \\ \text{TYPE} \end{array} \right] \\ \\ \text{SUBCAT} \left[\text{R} \left[\text{||HEAD} \left[\begin{array}{l} \text{tonic_c} \\ \text{TONAL} \quad C \\ \text{TYPE} \end{array} \right] \right] \right] \end{array} \right] \end{array} \right] \quad (5.32)$$

開始 T 項, I

$$\left[\begin{array}{l} \text{PHON} \langle C \rangle \\ \text{DSEQ} \langle I \rangle \\ \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic_h} \\ \text{DEG} \quad I \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K1 \end{array} \right] \\ \\ \text{SUBCAT} \left[\text{L} \left\langle \begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{dominant} \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K1 \end{array} \right] \right] , \\ \text{||HEAD} \left[\begin{array}{l} \text{tonic_c} \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K1 \end{array} \right] \end{array} \right\rangle \\ \\ \text{R} \left[\text{||HEAD} \left[\begin{array}{l} \text{tonic_c} \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K1 \end{array} \right] \right] \end{array} \right] \end{array} \right] \quad (5.33)$$

K1, 終結 T 項, I

$$\left[\begin{array}{l} \text{PHON} \langle C \rangle \\ \text{DSEQ} \langle I \rangle \end{array} \right] \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic}_h \\ \text{DEG} \quad I \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K2 \end{array} \right] \\ \text{SUBCAT} \left[\begin{array}{l} L \left\langle \begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{dominant} \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K2 \end{array} \right] \\ \text{||HEAD} \left[\begin{array}{l} \text{tonic}_c \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K2 \end{array} \right] \end{array} \right\rangle, \\ R \left[\text{||HEAD} \left[\begin{array}{l} \text{tonic}_c \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K2 \end{array} \right] \right] \end{array} \right] \end{array} \right] \right] \quad (5.34)$$

K2, 終結 T 項, I

$$\left[\begin{array}{l} \text{PHON} \langle C \rangle \\ \text{DSEQ} \langle I \rangle \end{array} \right] \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic}_h \\ \text{DEG} \quad I \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K3 \end{array} \right] \\ \text{SUBCAT} \left[\begin{array}{l} L \left\langle \begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{subdominant} \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K3 \end{array} \right] \\ \text{||HEAD} \left[\begin{array}{l} \text{tonic}_c \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K3 \end{array} \right] \end{array} \right\rangle, \\ \end{array} \right] \end{array} \right] \right] \quad (5.35)$$

K3, 終結 T 項, I

$$\left[\begin{array}{l} \text{PHON} \langle C \rangle \\ \text{DSEQ} \langle vi \rangle \end{array} \right] \left[\text{||CAT} \left[\text{HEAD} \left[\begin{array}{l} \text{tonic}_c \\ \text{DEG} \quad vi \\ \text{TONAL} \quad C \\ \text{TYPE} \end{array} \right] \right] \right] \quad (5.36)$$

開始 T 項, vi

$$\left[\begin{array}{l} \text{PHON} \langle Am \rangle \\ \text{DSEQ} \langle vi \rangle \end{array} \right] \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic}_h \\ \text{DEG} \quad vi \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K1 \end{array} \right] \\ \text{SUBCAT} \left[\begin{array}{l} L \left\langle \begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{dominant} \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K1 \end{array} \right] \\ \text{||HEAD} \left[\begin{array}{l} \text{tonic}_c \\ \text{TONAL} \quad C \\ \text{TYPE} \quad K1 \end{array} \right] \end{array} \right\rangle, \\ \end{array} \right] \end{array} \right] \right] \quad (5.37)$$

K1, 終結 T 項, vi

$$\left[\begin{array}{l} \text{PHON} \langle \text{Am} \rangle \\ \text{DSEQ} \langle \text{vi} \rangle \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic}_h \\ \text{DEG} \quad \text{vi} \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K2} \end{array} \right] \\ \text{SUBCAT} \left[\text{L} \left\langle \left[\begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{dominant} \quad \text{C} \\ \text{TONAL} \quad \text{K2} \end{array} \right] \right] , \left[\begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{tonic}_c \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K2} \end{array} \right] \right] \right\rangle \right] \end{array} \right] \end{array} \right] \end{array} \right] \quad (5.38)$$

K2, 終結 T 項, vi

$$\left[\begin{array}{l} \text{PHON} \langle \text{F} \rangle \\ \text{DSEQ} \langle \text{IV} \rangle \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{subdominant} \\ \text{DEG} \quad \text{IV} \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K2} \end{array} \right] \\ \text{SUBCAT} \left[\text{R} \left[\begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{subdominant} \\ \text{DEG} \quad \text{ii} \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K2} \end{array} \right] \right] \right] \right] \end{array} \right] \end{array} \right] \end{array} \right] \quad (5.39)$$

K2, S 項, IV

$$\left[\begin{array}{l} \text{PHON} \langle \text{F} \rangle \\ \text{DSEQ} \langle \text{IV} \rangle \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{tonic}_h \\ \text{DEG} \quad \text{IV} \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K3} \end{array} \right] \right] \end{array} \right] \end{array} \right] \quad (5.40)$$

K3, S 項, IV

$$\left[\begin{array}{l} \text{PHON} \langle \text{Dm} \rangle \\ \text{DSEQ} \langle \text{ii} \rangle \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{subdominant} \\ \text{DEG} \quad \text{ii} \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K2} \end{array} \right] \right] \end{array} \right] \end{array} \right] \quad (5.41)$$

K2, S 項, ii

$$\left[\begin{array}{l} \text{PHON} \langle \text{G} \rangle \\ \text{DSEQ} \langle \text{V} \rangle \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{dominant} \\ \text{DEG} \quad \text{V} \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K1} \end{array} \right] \right] \end{array} \right] \end{array} \right] \quad (5.42)$$

K1, D 項, V

$$\left[\begin{array}{l} \text{PHON} \langle \text{G} \rangle \\ \text{DSEQ} \langle \text{V} \rangle \\ \text{||CAT} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{dominant} \\ \text{DEG} \quad \text{V} \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K2} \end{array} \right] \\ \text{SUBCAT} \left[\text{R} \left[\begin{array}{l} \text{||HEAD} \left[\begin{array}{l} \text{subdominant} \\ \text{TONAL} \quad \text{C} \\ \text{TYPE} \quad \text{K2} \end{array} \right] \right] \right] \right] \end{array} \right] \end{array} \right] \end{array} \right] \quad (5.43)$$

K2, D 項, V

第6章 システム概要

本稿で実装したシステムの全体図を図 6.1 に示す。システムは大きく分けて和音認識部と和音進行解析部に別れ、これらを System Facade が統括する。システム全体の入力には MIDI ファイルを XML 化したファイル¹を用い、出力形式は HTML ファイルとした。システムはまず和音認識部で楽曲から和音列を認識し、その結果を和音進行解析部に入力して調とカデンツ列を解析する。また、システムは和音認識単体で使うことも、和音列を入力して和音進行解析単体で使うことも可能である。和音進行解析を単体で行う場合は和音列を記述したテキストファイルを入力とする。本章ではまず和音進行解析部について解説し、続いて和音認識部について解説する。

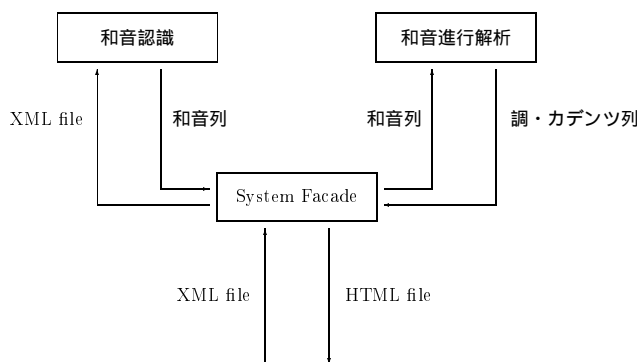


図 6.1: システム概要図

6.1 和音進行解析部

和音進行解析部は図 6.2 のような構成となっている。パッケージ ‘HPSG’ は入力として単語列を受け取り、HSPG に基く構文解析を行う。解析に必要な知識は ‘Definition’ に格納されている。パッケージ ‘Phrase Parser’ は入力された和音列をパッケージ ‘HPSG’ を用いて構文解析し、カデンツと調を認識する。パッケージ ‘Music Parser’ は外部から和音列を受け取り、パッケージ ‘Phrase Parser’ を用いて任意の数のカデンツと調を認識して出力する。

¹ファイル変換には青柳 龍也助教授（津田塾大学）による midi2xml パッケージを仕様した。

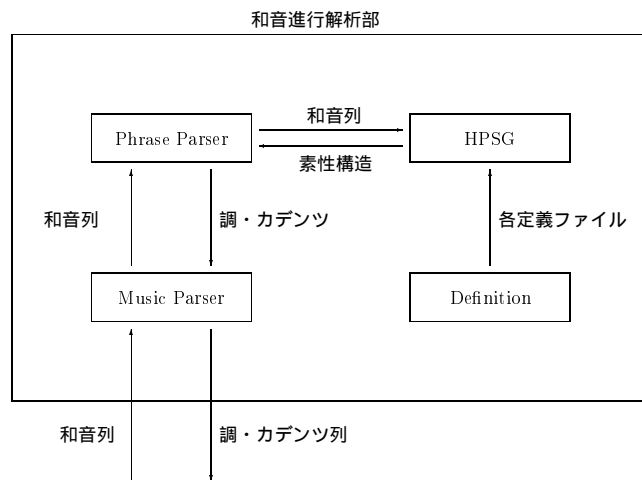


図 6.2: 和音進行解析部のシステム図

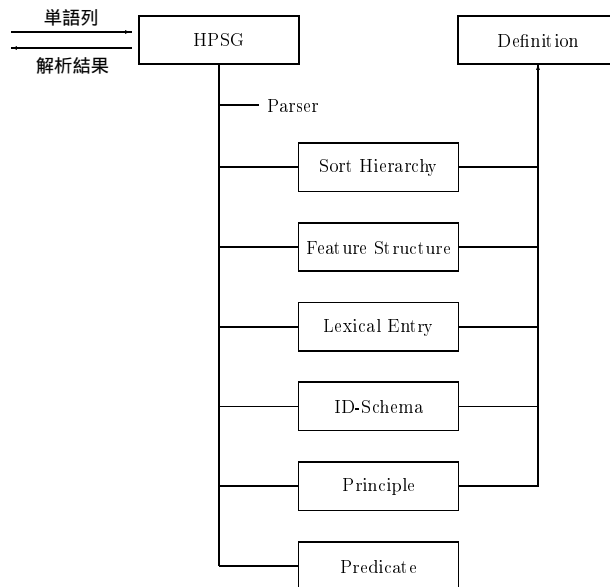


図 6.3: package HPSG

6.1.1 package : HPSG

このパッケージは HPSG による構文解析を行うインタプリタである。入力として単語列を受け取り、出力として解析結果の素性構造を返す。このパッケージは内部にいくつかのサブパッケージを含む。これを図 6.3 に示す。サブパッケージのいくつかは実行時に定義ファルを格納した 'Definition' を参照する。ここでは各サブパッケージの処理内容と各定義ファイルのフォーマットについて述べ、実行例を紹介する。

package : Sort Hierarchy

このパッケージは型階層定義ファイルを読み込んで型階層を生成し、型の管理、単一化などのサービスを提供する。型階層定義ファイルのフォーマットを図 6.5 に示す。宣言する型を '<-' の左側に、継承する型を右側に '[']' で囲って記述する。継承する型は ',' で区切っていくつでも宣言することができる。

```
head <- [bot].  
  
noun <- [head].  
verb <- [head].  
.  
.
```

図 6.4: 型階層定義ファイルのフォーマット

package : Feature Structure

このパッケージは素性構造に関するサービスを提供する。主なサービスは素性構造の生成・複製・表示、素性構造同士の単一化、任意のパスの型の代入などである。単一化や型を操作する際は型階層が参照される。このパッケージが読み込む定義ファイルのフォーマットを図 6.6 に示す。 '<-' の左側に型名を、右側にその型から派生する素性名とその先の型名を記述する。また、ここで用いられる型名は全て型階層において宣言されていない。

package : Lexical Entry

このパッケージは辞書項目の生成・管理・検索などの処理を担当する。辞書項目の定義ファイルのフォーマットを以下に示す。定義ファル冒頭では各型へのショートカットを定

```

per <- [].
num <- [].

head <- [PER\per, NUM\num].
.
.

```

図 6.5: 素性構造定義ファイルのフォーマット

```

.
.
HEAD\ = SYNSEM\LOC\CAT\HEAD\;
VAL\ = SYNSEM\LOC\CAT\VAL\;
COMP\ = SYNSEM\LOC\CAT\COMP\;

VAL\HEAD\ = VAL\ + HEAD\;
COMP\HEAD\ = COMP\ + HEAD\;

.
.
he <- (
    PHON\ = [he];
    HEAD\ = [verb];
    VAL\HEAD\ = [noun];
    .
    .
).

```

図 6.6: 辞書項目定義ファイルのフォーマット

義する．このショートカットを用いることで辞書項目の記述を容易にし，記述量を削減することができる．また素性構造のデザインの変更が行われてもショートカットを修正するだけで済むため，ソースの再利用性が高まる．項目の定義は，‘<-’の左側に検索キーを記述し，右側にショートカットを用いて各型の値を列挙して‘()’で囲む．ここで記述される型名も型階層において定義されている必要があり，厳密な型チェックが行われる．例外として PHON 素性の型には何でも記述できる．

package : ID-Schema, Principle

ID-Schema と Principle の両パッケージは構文解析を行う際に ‘Paraer’ によって呼び出される．まず，ID-Schema 定義ファイルのフォーマットを示す．left, right, head, comp, mothor はシステムの予約語で，left は解析対象の右側の，right は左側の素性構造へのポインタであり，head は主辞の子，comp は補語の子，mothor は親の素性構造へのポインタである．

```
.
.
head-complement-schema <- {

    set( head, left );
    set( comp, right );
    set( mothor\DTRS\HEAD-DTR\, head );
    set( mothor\DTRS\COMP-DTR\, comp );
    unify( mothor\DTRS\HEAD-DTR\SYNSEM\LOC\CAT\SUBCAT\,
           mothor\DTRS\COMP-DTR\,
           true );

}.
.
.
```

図 6.7: ID-Schema 定義ファイルのフォーマット

ここに記述されている述語はシステムに関数として組み込まれている．一つの ID-Schema の適用に際しては，スキーマ内に記述されている全述語の実行に成功しなければならない．以下にシステムに組み込まれている述語について解説する．

- `set(arg1, arg2)`
ポインタ `arg1` にポインタ `arg2` をセットする .
- `unify(arg1, arg2, arg3)`
パス `arg1` の先の素性構造とパス `arg2` の先の素性構造の単一化を行う . また述語の処理結果は , 単一化の成否と `arg3` の真偽値が等しい場合に真となる .
- `append(arg1, arg2, arg3)`
パス `arg1`, `arg2` で指定された型名の文字列を ‘-’ を挟んで結合し `arg3` へ代入する . PHON 素性の型名の結合に用いられる .
- `clear(arg1)`
パス `arg1` で指定された型を初期値に戻し , そこから派生する素性を削除する .
- `sizeIs(arg1, arg2)`
`arg1` で指定されたパスの型から派生する素性の数が `arg2` と等しい場合真 .

任意の ID-Schema の適用に成功した場合 , 定義されている全プリンシプルが親の素性構造に対して適用される . プリンシプルの定義ファイルは図 6.8 のように記述される .

package : Parser, Predicate

パッケージ ‘Parser’ は各サブクラスを用いて単語列の解析を行う . まず , 起動時に各サブパッケージを初期化し待ち状態に入る . 入力として単語列を受け取ると , 各単語の素性構造を辞書項目より取得し素性構造列を生成する . その後生成した素性構造列の $i, i+1$ 番目の素性構造に対して ID-Schema, Principle を適用し親の素性構造を得て , 新たな素性構造列を生成する . この操作を繰り返し , 素性構造列の要素数が 1 となった場合にそれを結果として出力する . ID-Schema や Principle の適用に必要な述語は パッケージ ‘Predicate’ に実装されており , ここに左側の子 , 右側の子 , 主辞 , 補語 , 親の素性構造のポインタと述語名 , 引数を渡して処理を行う .

HPSG パッケージの実行結果

このパッケージを用いて自然言語を解析した際の出力結果を図 6.9 紹介する . これは 3.4.3 項で紹介した素性構造を用いて英文 ‘he loves Mary’ を解析した結果の素性構造である . SUBCAT 素性は省略して表示している .

まず , `DTRS|HEAD-DTR|DTRS|HEAD-DTR` の先の素性構造が文全体の主辞 ‘loves’ の素性構造であり , これが `DTRS|HEAD-DTR|DTRS|COMP-DTR` の先の ‘Mary’ の素性構造を下位範疇化し , 親である `DTRS|HEAD-DTR` の素性構造 ‘loves-Mary’ を形成し

```

.
.
haed-feature-principle <- {

    set( mothor\SYNSEM\LOC\CAT\HEAD\,
          mothor\DTRS\HEAD-DTR\SYNSEM\LOC\CAT\HEAD\ );

}.

phon-principle <- {

    append( left\PHON\, right\PHON\, mothor\PHON\ );

}.

.
.

```

図 6.8: Principle 定義ファイルのフォーマット

```

|~PHON: <he-loves-Mary>~|
|SYNSEM|~LOC|~CAT|~HEAD|~verb~|~|~|~|~| | |
|_|_|_|_|_VF|FORM: <fin>_|_|_|_|_|
|_|_|_|_|_|~PHON: <loves-Mary>~|~|~|~|~|
|SYNSEM|~LOC|~CAT|~HEAD|~verb~|~|~|~|~|
|_|_|_|_|_VF|FORM: <fin>_|_|_|_|_|
|_|_|_|_|_|~PHON: <loves>~|~|~|~|~|
|HEAD-DTR:|SYNSEM|~LOC|~CAT|~HEAD|~verb~|~|~|~|~|
|DTRS:|DTRS:|~PHON: <Mary>~|~|~|~|~|
|COMP-DTR:|SYNSEM:|LOC:|CAT:|HEAD:|CASE: <acc>|~|~|~|~|~|
|PER: <3rd>|~|~|~|~|~|
|_NUM: <sing>_|_|_|_|_|
|~PHON: <he>~|~|~|~|~|
|COMP-DTR:|SYNSEM:|LOC:|CAT:|HEAD:|CASE: <nom>|~|~|~|~|~|
|PER: <3rd>|~|~|~|~|~|
|_NUM: <sing>_|_|_|_|_|

```

図 6.9: HPSG パッケージの実行結果

ている．また，これが DTRS|COMP-DTRS の ‘he’ の素性構造を下位範疇化し，飽和した文となっている．

6.1.2 package : phrase Parser

このパッケージは和音列を入力として受け取り，パッケージ ‘HPSG’ を用いて構文解析を行って和音列から1つのカデンツ構造を見出す．まず入力された和音列に (6.1) のように ‘:’ を挟んで調名を付加する．これは各和音が辞書項目に ‘調名 : 和音名’ の形の検索キーとともに格納されているためである．また，このパッケージで用られる調名は長調，短調あわせて24通りである． \flat を用いて表される調は $D\flat = C\sharp$ のように \sharp を用いて表現している．

$$\begin{array}{ccccccc}
 & & & & C:C & - & C:F & - & C:G & - & C:C \\
 & & & & C\sharp:C & - & C\sharp:F & - & C\sharp:G & - & C\sharp:C \\
 C - F - G - C & \rightarrow & D:C & - & D:F & - & D:G & - & D:C & & \\
 & & D\sharp:C & - & D\sharp:F & - & D\sharp:G & - & D\sharp:C & & (6.1)
 \end{array}$$

このようにして得られた和音列を順に HPSG パッケージに入力してカデンツ構造を解析する．このパッケージによって得られる出力例を図 6.10 に示す．これは5章で述べた和音進行解析手法に基づいて，和音列 ‘G-D7-G’ を解析した結果である．

6.1.3 package : Music Parser

パッケージ ‘Music Parser’ はパッケージ ‘Phrase Parser’ を用いて入力された和音列から任意の数のカデンツと調を認識する．まず，(6.2) のように入力された和音列中に含まれる同名の和音の連鎖を統合する．その後，図 5.25 スライディングウィンドウ方式を用いて和音を区切る．ウィンドウの閾値は3以上7以下に設定してある．これは入力された和音列に対して (6.2) のような操作を行うため，これ以外の長さの和音列から1つのカデンツが見出されることは無いためである．このようにして得られた各和音列を順にパッケージ HPSG へ入力し，カデンツが認識された複数の和音列を解として出力する．

また，このパッケージには曖昧性を含む和音列も入力することができる．この例を (6.3) に示す．この和音列は一番目の和音が C または E_m ，二番目の和音が G または E_m という曖昧性を含んでいる．これを展開して四種類の和音列を生成し，それぞれをパッケージ HPSG に入力することで正しい和音列と調名，カデンツ構造を得る．

$$\dots C - C - G - G - C \dots \quad \rightarrow \quad \dots C - G - G \dots \quad (6.2)$$

```

|~PHON: <G-D7-G>
|DSEQ: <I-V7-I>
|SYNSEM: | LOC: | CAT: | HEAD: | DEG: <I> | | | | | |
|         |     |     |     |     | TONAL: <G> | | | | |
|         |     |     |     |     | _TYPE: <K1>_|_|_|_|
|         |~      |~PHON: <D7-G>
|         |DSEQ: <V7-I>
|         |SYNSEM: | LOC: | CAT: | HEAD: | DEG: <I> | | | | | |
|         |         |     |     |     |     | TONAL: <G> | | | | |
|         |         |     |     |     |     | _TYPE: <K1>_|_|_|_|
|         |         |~      |~PHON: <G>
|         |         |DSEQ: <I>
|HEAD-DTR: |         |HEAD-DTR: |~      |~PHON: <G>
|         |         |         |DSEQ: <I>
|         |         |         |SYNSEM: | LOC: | CAT: | HEAD: | DEG: <I> | | | | | |
|DTRS:     |         |         |         |     |     |     |     | TONAL: <G> | | | | |
|         |         |         |         |     |     |     |     | _TYPE: <K1>_|_|_|_|
|         |         |         |~PHON: <D7>
|         |         |         |DSEQ: <V7>
|         |         |         |COMP-DTR: |~      |~PHON: <G>
|         |         |         |         |DSEQ: <I>
|         |         |         |         |SYNSEM: | LOC: | CAT: | HEAD: | DEG: <V7> | | | | | |
|         |         |         |         |         |     |     |     |     | TONAL: <G> | | | | |
|         |         |         |         |         |     |     |     |     | _TYPE: <K1>_|_|_|_|
|         |         |         |         |~PHON: <G>
|         |         |         |         |DSEQ: <I>
|         |         |         |         |COMP-DTR: |~      |~PHON: <G>
|         |         |         |         |         |DSEQ: <I>
|         |         |         |         |         |SYNSEM: | LOC: | CAT: | HEAD: | DEG: <I> | | | | |
|         |         |         |         |         |         |     |     |     |     | TONAL: <G>_|_|_|_|

```

図 6.10: Phrase Parser パッケージの実行結果

$$\begin{array}{lcl}
 (C, Em) - (G, Em) - Am & \rightarrow & \begin{array}{l} C - G - Am \\ C - Em - Am \\ Em - G - Am \\ Em - Em - Am \end{array}
 \end{array} \tag{6.3}$$

6.2 和音認識部

和音認識部は全体で一つのパッケージとなっている。このパッケージは入力として MIDI ファイルを XML 化したファイルを受け取る。この XML ファイルの例を図 6.11 に示す。タグ ‘noteOn’ は発音を、タグ ‘noteOff’ は音の終了を表現し、属性 ‘noteNumber’ は音高を、‘velocity’ は音強を表現している。noteNumber の ‘60’ はピアノの鍵盤の中心の C の音にあたり、半音上の C♯ は ‘61’、さらに半音上の D は ‘62’ のように表現される。また、タグ ‘deltaTime’ は時間の経過を表し、属性 ‘value’ によって経過時間を示す。時間の単位には ‘tick’ が用いられる。この例では noteNumber 72 の C の音が発音されると同時に noteNumber 60 の C の音が発音され、512 tick 後に noteNumber 60 の C の音が終了している。

和音認識パッケージは XML ファイルの他に和音範囲情報を記述したファイルを入力として受け取る。このファイルのフォーマット図 6.12 を示す。まず、‘QuarterNote’ によって四分音符 1 つあたり何 tick かを指定し、‘per’ によって何拍子かを指定する。また、1 節目の 1 拍目から始まらないような楽曲²の場合、最初の音が発音されるまでの空白の tick を ‘alignment’ によって指定する。続いて、‘1,3’ や ‘2,1’ のように ‘小節番号, 拍’ のペアで何小節目の何拍目に和音の切れ目が生じるかを記述する。また、2 拍と 1/2 拍目などは ‘2.5’ のようにして記述する。これらの情報から和音の切れ目となる tick を求める際は、小節番号を ‘bar’、拍を ‘note’ とすると以下の計算式で求めることができる。

$$\text{tick} = \{(\text{bar} - 1)\text{par} + (\text{note} - 1)\}\text{QuarterNote} - \text{alignment}$$

以上のファイルの情報を基に音符列を指定された範囲ごとに区切る。このとき、任意の和音範囲の要素となる音は、その範囲中で発音している音または終了する音とする。また、各和音の雛形と任意の和音範囲の音とのパターンマッチングを行う際は、音と音が一致するかを調べるために予め以下のような計算式を用いてオクターブの差を吸収しておく。

$$\text{noteNumber} \bmod 12$$

また、4 分音符、8 分音符区切りなどの和音認識を行う場合は、各範囲から和音を認識した後に図 5.23 のような和音の統合を行う。

²弱起の曲と呼ばれる


```
<noteOn channel="1" noteNumber="72(C4)" velocity="64"/>
<deltaTime value="0"/>
<noteOn channel="1" noteNumber="60(C3)" velocity="64"/>
<deltaTime value="512"/>
<noteOff channel="1" noteNumber="60(C3)" velocity="0"/>
<deltaTime value="0"/>
<noteOn channel="1" noteNumber="67(G3)" velocity="64"/>
<deltaTime value="512"/>
<noteOff channel="1" noteNumber="67(G3)" velocity="0"/>
<deltaTime value="0"/>
```

図 6.11: XML File

```
QuarterNote= 1024
alignment= 1024
par= 3
```

```
1,3
2,1
2,2
2,3
3,1
3,2
3,3
```

図 6.12: 和音範囲情報 File

第7章 評価実験と考察

本稿では提案方式を評価するために4種類の実験を行った。実験1では和音列を入力として和音進行解析を行う。この和音列の生成は音楽大学を卒業された方に依頼した。実験2では正しい和音範囲の情報を与え各和音範囲から和音を認識し、認識した和音列に対して和音進行解析を行う。実験3では音符列を4分音符ごとに区切り、各範囲から和音を認識し、和音列中の連続した同名の和音を統合した後に和音進行解析を行う。実験4では音符列を8分音符ごとに区切り、実験3と同様の手順で和音進行解析を行う。

実験1では本和音進行解析による調とカデンツの認識の有効性を検証する。カデンツ認識の有効性は再現率で評価する。再現率の計算式を(7.1)に示す。また、現在はカデンツ列の認識精度を上げるような試みは行っていないため、認識されたカデンツの精度については検討しない。加えて、調の認識の評価はカデンツの再現率で代用する。これは本和音進行解析システムによって認識されたカデンツが正しければ、その区間で認識された調も正しいことが保証されているためである。

実験2から実験4ではカデンツと調の認識の評価に加えて、各和音認識によって認識された和音列中に含まれる曖昧性が和音進行解析システムによってどれだけ解消されたかを検証する。評価は和音進行解析を行う前と行った後の和音列に対しそれぞれ(7.2)の計算式によって精度を求め、これらと比較することで行う。また、各和音認識では発生する曖昧性の性質が異なる。実験2の和音認識では和音範囲による曖昧性は発生せず、和音名の特定における曖昧性のみが生じる。実験3、実験4ではこれに加えて和音範囲の特定における曖昧性が生じる。この曖昧性は実験4の方が大きくなることが予想される。これらの各曖昧性に対し、和音進行解析による曖昧性解消機能がどれだけ有効かを検討する。

$$\text{再現率} = \text{システムが出力した正解の数} / \text{システムが出力すべき正解の数} \quad (7.1)$$

$$\text{精度} = \text{システムが出力した正解の数} / \text{システムの出力総数} \quad (7.2)$$

7.1 評価実験

実験は以下の楽曲について行った。これらの楽曲は和声学の知識に基づいて作曲されており、本システムが現在どれだけ忠実に和声学の知識を表現し、また扱うことができるか評価することを目的として選択した。ここではそれぞれの楽曲に対する実験の結果を順に紹介する。

- Mozart, Piano Sonata in C, K545 第1楽章・第2楽章冒頭
- Beethoven, Piano Sonata, Op.49, No2 第1楽章・第2楽章冒頭
- Beethoven, Piano Sonata, Op.49, No1 第1楽章・第2楽章冒頭

7.2 Mozart, Piano Sonata in C, K545

実験は第1章冒頭の28小節と、第2楽章冒頭16小節について行った。第1楽章にはカデンツは計17、和音は計41個含まれる。調は20小節目の後半を境にハ長調からト長調へと転調している。第2楽章にはカデンツは計12個、和音は計27個含まれており、調はト長調である。

第1楽章の実験結果を表7.1に示す。各実験を通じて、曖昧性を含まない一つの和音列から複数のカデンツ構造が見い出されることはなかった。

実験1の和音進行解析では17個中15個のカデンツを認識し、再現率は0.882となった。次に、実験2の和音認識結果には54個の回答中に36個の正解が含まれ、精度は0.666となった。この和音列に対し和音進行解析を行った結果、12個のカデンツが認識され、6カ所の曖昧性が解消されて精度が0.800へと向上した。実験3の和音認識結果には50個の回答中に32個の正解が含まれ、精度は0.640となった。この和音列に対し和音進行解析を行った結果7個のカデンツが認識されたが、和音列中の曖昧性を解消することはできなかった。実験4の和音認識結果には69個回答中に37個の正解が含まれ、精度は0.536となった。この和音列に対し和音進行解析を行った結果、1個のカデンツが認識され、1カ所の曖昧性が解消されて精度が0.552へと向上した。

		実験1	実験2	実験3	実験4
和音進行解析結果	カデンツ正解出力数	15/17	12/17	7/17	6/17
	カデンツ再現率	0.882	0.705	0.441	0.352
和音認識結果 (和音進行解析前)	和音正解出力数		36/41	32/41	37/41
	和音再現率		0.878	0.780	0.902
	正解数 / 出力数		36/54	32/50	37/69
	和音認識精度		0.666	0.640	0.536
和音認識結果 (和音進行解析後)	曖昧性特定箇所		6/9	0/3	1/8
	正解数 / 出力数		36/45	32/50	37/67
	和音認識精度2		0.800	0.640	0.552

表 7.1: Mozart, Poano Sonate in C, K545 第1楽章実験結果

続いて第2楽章の実験結果を表7.2に示す。ここでも曖昧性のない和音列からのカデンツ構造の認識に曖昧性は発生しなかった。実験1の和音進行解析では12個中7個のカデン

ツを認識し，再現率は 0.583 となった．実験 2 の和音認識結果には 34 個の回答中に 25 個の正解が含まれ，精度は 0.735 となった．この和音列に対し和音進行解析を行った結果，5 個のカデンツが認識され，4 カ所の曖昧性が解消されて精度が 0.883 へと向上した．実験 3 の和音認識結果には 36 個の回答中に 26 個の正解が含まれ，精度は 0.722 となった．この和音列に対し和音進行解析を行った結果 5 個のカデンツが認識され，1 カ所の曖昧性を解消し，精度が 0.742 へと向上した．実験 4 の和音認識結果には 66 個の回答中に 27 個の正解が含まれ，精度は 0.409 となった．この和音列に対し和音進行解析を行った結果，1 個のカデンツが認識され，1 カ所の曖昧性が解消されて精度が 0.415 へと向上した．

		実験 1	実験 2	実験 3	実験 4
和音進行解析結果	カデンツ正解出力数	7/12	5/12	5/12	1/12
	カデンツ再現率	0.583	0.416	0.416	0.083
和音認識結果 (和音進行解析前)	和音正解出力数		25/27	26/27	27/27
	和音再現率		0.925	0.962	1.000
	正解数 / 出力数		25/34	26/36	27/66
	和音認識精度		0.735	0.722	0.409
和音認識結果 (和音進行解析後)	曖昧性特定箇所		4/7	1/4	1/7
	正解数 / 出力数		25/30	26/35	27/65
	和音認識精度 2		0.833	0.742	0.415

表 7.2: Mozart, Poano Sonate in C, K545 第 2 楽章実験結果

7.3 Beethoven, Piano Sonata, Op.49, No2

第 1 楽章は冒頭 20 小節，第 2 楽章は冒頭 21 小節について実験を行った．第 1 楽章には 37 個の和音，16 個のカデンツが含まれ，調は 13 小節目を境にト長調からニ長調へと転調している．第 2 楽章には 24 個の和音，11 個のカデンツが含まれ，調はト長調である．

第 1 楽章の実験結果を表 7.3 に示す．各実験を通じて，任意の和音列に対するカデンツ構造の認識に曖昧性は発生しなかった．

実験 1 の和音進行解析では 16 個中 15 個のカデンツを認識し，再現率は 0.937 となった．実験 2 の和音認識結果には 42 個の回答中に 28 個の正解が含まれ，精度は 0.666 となった．この和音列に対し和音進行解析を行った結果，9 個のカデンツが認識され，2 カ所の曖昧性が解消されて精度が 0.700 へと向上した．実験 3 の和音認識結果には 70 個の回答中に 33 個の正解が含まれ，精度は 0.471 となった．この和音列に対し和音進行解析を行った結果 9 個のカデンツが認識され，5 カ所の曖昧性が解消されて精度が 0.559 へと向上した．実験 4 の和音認識結果には 142 個の回答中に 33 個の正解が含まれ，精度は 0.232 となった．この和音列に対し和音進行解析を行った結果，6 個のカデンツが認識されたが，誤っ

た和音列から多くのカデンツ構造が誤認識されたため、和音列の曖昧性を解消することはできなかった。

		実験 1	実験 2	実験 3	実験 4
和音進行解析結果	カデンツ正解出力数	15/16	9/16	9/16	6/16
	カデンツ再現率	0.937	0.562	0.562	0.375
和音認識結果 (和音進行解析前)	和音正解出力数		28/37	33/37	33/37
	和音再現率		0.756	0.891	0.891
	正解数 / 出力数		28/42	33/70	33/142
	和音認識精度		0.666	0.471	0.232
和音解析結果 (和音進行解析後)	曖昧性特定箇所		2/4	5/13	0/17
	正解数 / 出力数		28/40	33/59	33/142
	和音認識精度 2		0.700	0.559	0.232

表 7.3: Bethooven, Piano Sonate, Op49, No2 第 1 楽章 実験結果

続いて第 2 楽章の実験結果を表 7.2 に示す。実験 1 の和音進行解析では 11 個中 9 個のカデンツを認識し、再現率は 0.818 となった。実験 2 の和音認識結果には 29 個の回答中に 22 個の正解が含まれ、精度は 0.758 となった。この和音列に対し和音進行解析を行った結果、8 個のカデンツが認識され、すべての曖昧性が解消されて精度が 0.880 へと向上した。実験 3 の和音認識結果には 32 個の回答中に 23 個の正解が含まれ、精度は 0.718 となった。この和音列に対し和音進行解析を行った結果 8 個のカデンツが認識され、5 カ所の曖昧性を解消し、精度が 0.851 へと向上した。実験 4 の和音認識結果には 45 個の回答中に 23 個の正解が含まれ、精度は 0.511 となった。この和音列に対し和音進行解析を行った結果、6 個のカデンツが認識され 3 カ所の曖昧性が解消されて精度が 0.547 へと向上した。

		実験 1	実験 2	実験 3	実験 4
和音進行解析結果	カデンツ正解出力数	9/11	8/11	8/11	6/11
	カデンツ再現率	0.818	0.727	0.727	0.545
和音認識結果 (和音進行解析前)	和音正解出力数		22/24	23/24	23/24
	和音再現率		0.916	0.920	0.920
	正解数 / 出力数		22/29	23/32	23/45
	和音認識精度		0.758	0.718	0.511
和音認識結果 (和音進行解析後)	曖昧性特定箇所		4/4	5/7	3/7
	正解数 / 出力数		22/25	23/27	23/42
	和音認識精度 2		0.880	0.851	0.547

表 7.4: Bethooven, Piano Sonate, Op49, No2 第 2 楽章 実験結果

7.4 Beethoven, Piano Sonata, Op.49, No1

第1楽章は冒頭33小節，第2楽章は冒頭13小節について実験を行った．第1楽章には和音は46個，カデンツは21個含まれ，調は16小節目を境にト短調から変ロ長調へと転調している．第2楽章には和音は28個，カデンツは11個含まれ，調はト長調である．

第1楽章の実験結果を表7.5に示す．ここでも，各実験を通じて任意の和音列に対するカデンツ構造の認識に曖昧性は発生しなかった．

実験1の和音進行解析では21個中18個のカデンツを認識し，再現率は0.857となった．実験2の和音認識結果には63個の回答中に40個の正解が含まれ，精度は0.634となった．この和音列に対し和音進行解析を行った結果，14個のカデンツが認識され，7カ所の曖昧性が解消されて精度が0.714へと向上した．実験3の和音認識結果には56個の回答中に24個の正解が含まれ，精度は0.428となった．この和音列に対し和音進行解析を行った結果13個のカデンツが認識され，6カ所の曖昧性が解消されて精度が0.510へと向上した．実験4の和音認識結果には96個の回答中に45個の正解が含まれ，精度は0.468となった．この和音列に対し和音進行解析を行った結果，12個のカデンツが認識され5カ所の曖昧性が解消されて精度が0.542へと向上した．

		実験1	実験2	実験3	実験4
和音進行解析結果	カデンツ正解出力数	18/21	14/21	13/21	12/21
	カデンツ再現率	0.857	0.666	0.619	0.571
和音認識結果 (和音進行解析前)	和音正解出力数		40/46	24/46	45/46
	和音再現率		0.869	0.521	0.978
	正解数 / 出力数		40/63	24/56	45/96
	和音認識精度		0.634	0.428	0.468
和音認識結果 (和音進行解析後)	曖昧性特定箇所		7/11	6/8	5/13
	正解数 / 出力数		40/56	24/47	45/83
	和音認識精度2		0.714	0.510	0.542

表 7.5: Beethoven, Piano Sonata, Op49, No1 第1楽章 実験結果

続いて第2楽章の実験結果を表7.6に示す．実験1の和音進行解析では11個中8個のカデンツを認識し，再現率は0.727となった．実験2の和音認識結果には24個の回答中に22個の正解が含まれ，精度は0.916となった．この和音列に対し和音進行解析を行った結果，5個のカデンツが認識されたが，和音列中の曖昧性を解消することはできなかった．実験3の和音認識結果には32個の回答中に17個の正解が含まれ，精度は0.531となった．この和音列に対し和音進行解析を行った結果4個のカデンツが認識され，2カ所の曖昧性を解消し，精度が0.607へと向上した．実験4の和音認識結果には45個の回答中に25個の正解が含まれ，精度は0.555となった．この和音列に対し和音進行解析を行った結果，6個のカデンツが認識され2カ所の曖昧性が解消されて精度が0.581へと向上した．

		実験 1	実験 2	実験 3	実験 4
和音進行解析結果	カデンツ正解出力数	8/11	5/11	4/11	6/11
	カデンツ再現率	0.727	0.454	0.363	0.545
和音認識結果 (和音進行解析前)	和音正解出力数		22/28	17/28	25/28
	和音再現率		0.785	0.607	0.892
	出力数 / 正解数		22/24	17/32	25/45
	和音認識精度		0.916	0.531	0.555
和音認識結果 (和音進行解析後)	曖昧性特定箇所		0/0	2/6	2/6
	出力数 / 正解数		22/24	17/28	25/43
	和音認識精度 2		0.916	0.607	0.581

表 7.6: Bethooven, Piano Sonate, Op49, No1 第 2 楽章 実験結果

7.5 考察

実験の総合結果を表 7.7 に示す．ここではまず実験 1 の結果からカデンツと調の認識に関する考察を行う．続いて実験 2, 3, 4 の結果から各和音認識の精度について考察した後、和音認識に発生した曖昧性の解消機能について考察する．

		実験 1	実験 2	実験 3	実験 4
和音進行解析結果	カデンツ正解出力数	72/88	53/88	46/88	37/88
	カデンツ再現率	0.818	0.602	0.522	0.420
和音認識結果 (和音進行解析前)	和音正解出力数		173/203	155/203	190/203
	和音再現率		0.852	0.763	0.935
	正解数 / 出力数		173/246	155/276	190/463
	和音認識精度		0.703	0.561	0.410
和音認識結果 (和音進行解析後)	曖昧性特定箇所		23/35	19/41	12/58
	正解数 / 出力数		173/220	155/246	190/442
	和音認識精度 2		0.786	0.630	0.429
	精度比		1.118	1.122	1.046

表 7.7: 総合結果

7.5.1 カデンツ認識の評価と考察

実験 1 の結果から、本和音進行解析手法は和声学のカデンツ規則に基づくカデンツは十分認識できることが分かり、カデンツを形成する曖昧性を含まない和音列からは必ず一通り

のカデンツ構造を出力できることが分かった．また，和音列が曖昧性を含んでいなければ約 8 割のカデンツを認識できることが分かった．

しかし，基本的なカデンツ規則に含まれないカデンツが全体で 15 個存在し，これらを認識することができなかつた．このうち最も多かったのは 'I - ii - I' 進行のカデンツであった．この進行は 89 個のカデンツに 5 個含まれていた．和声学ではこの進行は特殊な進行として扱われており，K1 のカデンツと結びつき 'I - ii - I - V - I' という形で 1 つのカデンツを形成すると定義されている．この時，中間の I の和音は Tonic ではなく Dominant として働き，全体としては 'T - S - D - D - T' という K2 のカデンツとなる．また Dominant として機能する Tonic は 第 2 転回型でなければならないという制約がある．このカデンツに対応するには，HEAD 素性の中に転回型に関する素性を導入し，Dominant として機能する I の和音の素性構造を辞書項目に追加すればよいと考えられる．しかし単純に辞書項目への追加を行うと，'VI - I - VI' などのような和音進行から誤ったカデンツ構造を認識してしまうことが予想されるため，なんらかの対策を講じる必要がある．

続いて多かったのは転調を内部に含むカデンツであった．このカデンツは 3 個含まれていた．和声学における転調は，任意の和音を 2 つの調の和音に見立てることによって転調を行う．これの例を以下に示す．例では G の和音を境に八長調から二長調への転調が行われている．

$$\begin{array}{cccc} \text{chord sequence :} & C & - & F & - & G & - & D \\ & C : & I & - & IV & - & V & \\ & & & & & & \parallel & \\ & D : & & & IV & - & I & \end{array}$$

このような転調に対応するためには，各和音がどの調の何の和音となりうるかを十分に検討し，辞書項目を充実させ，カデンツが認識されなかつた区間に対して転調の可能性を考慮に入れたカデンツ構造の解析を行う必要がある．

その他の認識できなかったカデンツには，例外的な規則のカデンツや一時的に転調を行うカデンツなどが見られる．これらについても，素性構造のデザインを変更したり和音の前後関係を十分に調べることによって対応することが可能であると考えられるが，再現率の向上にはあまり結びつかないことが予想される．

7.5.2 調の認識の評価と考察

調性認識の評価はカデンツの再現率で代用した．カデンツの再現率は 0.802 であったことから，全実験曲を通じて約 8 割の区間の調を正しく認識できたと評価できる．またカデンツごとに調を認識することで，転調が行われている楽曲にも十分対応できた．このため，和音進行の情報をもとに調を認識する手法は有効であると考えられる．しかしカデンツと同時に調を解析しているため，カデンツが認識されない区間からは調を認識することはできない．このため，例外的なカデンツや転調を内部に含んでいるカデンツなどの調は認識できなかった．

7.5.3 和音の認識精度の考察

実験の結果，三種類の和音認識のうち実験4の8分音符区切りの和音認識が最も再現率が高く，且つ最も精度が低かった．最も精度が高かったのは実験2の和音範囲の情報を与えた和音認識となり，続いて実験3の四分音符区切りの和音認識の精度が高くなった．

8分音符区切りの和音認識が再現率最もが高くなったのは，楽曲を細かく区切ることによって，図7.1のような和音構成音以外の音が多く含まれるような区間からも正解の和音名を出力できるようになったためである．本和音認識手法では和音構成音の音長の総和から非和音構成音の音長の総和を減算したものを和音の評価値としている．このため，同図のような和音範囲に対して実験2の手法で和音認識を行うと，正解の和音の評価値が下がってしまうことがある．対して実験4のように8分音符ごとに区切って和音を認識した場合，非和音構成音による評価値の減算が低減されるため，正しい和音名を出力できるようになる．しかし同図の2，3拍目のように，休符などによって和音構成音がとぎれメロディーのみが演奏される区間からは，誤った和音名が認識されることとなり，結果和音の認識精度が低下している．



図 7.1: GTTM による和声解析例

7.5.4 本和音進行解析手法による和音認識の曖昧性解消機能の評価

和音進行解析を利用することで，曖昧性が発生した箇所のうち実験2では約65%，実験3では約46%，実験4では約20%の曖昧性を解消することに成功した．実験4の和音認識における曖昧性をあまり解消できなかったのは，和音認識に発生する誤認識が原因である．前節で考察したようにこの和音認識では，例えば 'F - C' という和音列が認識されるべき区間から，'F - ... - C' のように間に正しくない和音が混入した和音列が認識されてしまう．つまり，この手法では和音の範囲が正しく定まらないために，曖昧性に加えて誤認識が多く発生する．このため，実験4の和音の再現率は最も高かったのにもかかわらずカデンツの再現率は最も低くなっている．このような誤認識が含まれることによってカデンツを認識することができなかったために，曖昧性もあまり解消することができなかった．このことから，本和音進行解析による和音列中の曖昧性解消機能は，和音の範囲が正しく求まらない和音認識手法に対してはあまり効果を発揮することができないことが分かった．

しかし実験2の結果から，このような曖昧性や誤認識以外にはある程度有効であること

が確認された．最もその効果が発揮されたのは，(7.3) のような曖昧性に対してであった．これは，和音の3度音が省略され長三和音と短三和音の判断を行うことができなかつたために発生した曖昧性である．

$$\left\{ \begin{array}{c} G \\ Gm \end{array} \right\} - D - G7 \quad (7.3)$$

また和音の構成音が類似するために発生する曖昧性については，その前後関係によって解消可能なものと不可能なものが存在することが分かった．例えば (7.4) のように，曖昧な和音が異なる機能で分類されればかなりの確率で曖昧性を解消することが可能であるが，(7.5) のように同一の和音機能で分類されるとあまり曖昧性を解消できないことが確認された．

$$D - G - \left\{ \begin{array}{c} D \\ G \end{array} \right\} \rightarrow I - V - \left\{ \begin{array}{c} I \\ V \end{array} \right\} \rightarrow I - V - I \quad (7.4)$$

$$D - G - \left\{ \begin{array}{c} D \\ Bm \end{array} \right\} \rightarrow I - V - \left\{ \begin{array}{c} I \\ VI \end{array} \right\} \rightarrow \begin{array}{l} I - V - I \\ I - V - VI \end{array} \quad (7.5)$$

また (7.5) のような曖昧性は，これが D 項に現れた場合は必ず曖昧性は解消され，S 項に現れた場合は，カデンツが K3 であれば解消され T 項に現れた場合は解消することができないことがわかっている．これは，D 項には V の和音しか用いることができず，T 項や K2 の S 項には複数の和音を用いることができるためである．

第8章 まとめ

本稿で提案する和音進行解析手法では楽曲中に含まれるカデンツの内約8割を認識することができ、調の認識や転調の認識にも対応できることが確認された。また、和音認識における曖昧性の解消機能についてはある程度有効であることが確認された。

カデンツの認識に関しては、転調を含むものや例外的なものを認識することができなかったことから、今後、カデンツの認識精度を上げるためにこのようなカデンツにも対応していく必要があると考えられる。

和音認識における曖昧性の解消については、和音範囲の情報がないために発生する曖昧性や誤認識にはあまり対応することができなかった。このため、他の情報をもとに和音の曖昧性を解消する手法や、和音の誤認識箇所を特定し再度和音を認識する手法などについて検討する必要がある。

和声解析全体としてはまだまだ初歩的な段階にあり、今後拡張を重ねていく必要がある。例えばより大きな木構造を出力するために、メロディーやリズムなどの情報などを参照したり、GTTM 拍節構造解析やグルーピング構造解析などと組み合わせる、といった拡張が考えられる。またこれとは別に、本手法を用いて GTTM のタイムスパン簡約を実現する可能性についても検討していきたい。

謝辞

本研究を進めるにあたって、本学 東条 敏 教授より熱心な御指導、有益な御助言をいただきました。東条 敏 教授に深謝いたします。

本研究に際し、数多くのご指導、有益な御助言をいただきました本学 鳥澤 健太郎 助教授、永田 裕一 助手に感謝いたします。

最後になりましたが、御協力いただきました 東条研究室、鳥澤研究室の皆様にご心よりお礼申し上げます。

関連図書

- [1] Tojo, S. Harmony Analysis in Head-driven Phrase Structure Grammar. In Proceeding of the workshop on Musical Constraints, 2001.
- [2] Pollard, C. and Sag, I. A Head-driven Phrase Structure Grammar. University of Chicago Press, 1994.
- [3] Pollard, C. and Sag, I. Information-based Syntax and Semantics. 産業図書出版, 1995.
- [4] H, C, Longuet-Higgins, and M, J, Steedman. On interpreting Bach, In B, Meltzer and D, Michie (Eds.). Machine intelligence, Vol. 6, Edinburgh University Press, 1971.
- [5] I, Yoshio and J, Abe. Cognitive modeling of the process of tonal organization in melody perception. International Journal of Psychology, Vol.31, p.51, 1996.
- [6] C, L, Krumhansl. Cognitive foundations of musical pitch. Oxford University Press, 1990.
- [7] Lerdahl, F. and Jackendoff, R. A Generative Theory of Tonal Music. MIT Press, 1993.
- [8] Goto, M. and Muraoka, Y. Real-time Rhythm Tracking for Musical Decisions. IJCAI-97 Workshop on Computational Auditory Scene Analysis.
- [9] T, Winograd. Linguistics and the computer analysis of tonal harmony. Journal of Music Theory, Vol. 12, No. 1, pp.2-49, 1968. Also in 21), pp.112-153.
- [10] 長嶋 洋一, 橋本 周司, 平賀 譲, 平田 圭二. bit 別冊 コンピュータと音楽の世界 基礎からフロンティアまで. 共立出版, 1998.
- [11] 島岡 譲, 丸田 昭三, 小林 秀雄. 和声 理論と実習 I. 音楽之友社, 1964.
- [12] 山縣 茂太郎. 音楽通論. 音楽之友社, 1958.
- [13] 大蔵 康義. 音と音楽の基礎知識. 国書刊行会, 1999.