

修士論文

ジャズ和声を含む楽曲の組み合わせ範疇文法に基づく解析システムの実装

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

福成 豪

2015年3月

修 士 論 文

ジャズ和声を含む楽曲の組み合わせ範疇文法に基づく解析システムの実装

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 白井 清昭 准教授

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

1410038 福成 毅

提出年月: 2015 年 2 月

概要

近年、音楽情報科学の産業的な応用が盛んに行われている。自動作編曲や作編曲支援などといった音楽家向けの技術だけでなく、音楽情報検索や音楽レコメンドなどエンドユーザーが恩恵を受ける技術にも注目が集まっている。

これらの技術の実現のために、音楽理論は大きな役割を果たすが、音楽理論を計算機上へ実装させるために、音楽理論を計算機科学の視点から解き直そうとする計算論的音楽理論に関する研究が進められている。これにより音楽理論を上記の技術に応用することができるのではないかと考えられる。

本研究では和声に着目する。和声は調性音楽における最も基本的な要素の一つであり、和声を機能的に分析する和声解析を行うことによって、楽曲中の表現技法や音楽的特徴を構造的・機能的に把握することができる。そのため、和声解析を計算機によって行うことで、作編曲・演奏等を工学的な側面からサポートすることができ、自動採譜などの産業的な応用が可能になるのではないかと考えられている。

また、本研究はジャズに着目する。ここでのジャズとは1950年代におけるビバップの時代まで演奏されてきたものを指すが、そこで用いられた和声や理論が現在流通している商業音楽に大きな影響を与えている。このジャズで用いられている和声理論を和声解析システムに取り入れることで、より多くの楽曲の機能的解析が可能になるのではないかと考えられる。

以上を踏まえてジャズ和声を含む商業音楽に向けた和声解析システムの実装を本研究の目的とする。

先行研究としてジャズ和声の解析に自然言語処理の手法のひとつである組み合わせ範疇文法を用いた研究がある。この研究は入力をコードネーム表記による和音進行を入力とし、和音の役割を表す機能というものを出力としている。しかしながら、複数の解釈が可能な和音に対して一つの機能しか出力してない、和音に対して不適切な機能の付与が行われる、などの問題がある。

本研究ではこれらの問題に対して、コードネームではなく、予め調の情報を含んだディグリーネームを入力とし、先行研究で扱っていなかったサブドミナントマイナーやセカンドリー・ドミナントなどの機能を追加しつつ辞書項目の改訂を行った。また、複数の解釈が可能な一例としてピボットコードを用いた転調があるが、このピボットコードを認識できるようにアルゴリズムの改良を行った。

また、上記により入力をディグリーネームとしたため、調推定が必要になるが、本研究では組み合わせ範疇文法による和声解析の前段階として Tonal Pitch Space を用いた和声解析の手法を取り入れた。Tonal Pitch Space とは、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。TPS を

ジャズでよく用いられるノンダイアトニックコードに対応させた研究があり、若干の改訂を行いながらもこの研究で用いられた手法をシステムの調推定に用いた。

これらを用いることでシステムの実装を行い、妥当性を評価するためにいくつかのコードネーム表記による和音進行を入力として実験を行った。実験の結果、ピボットコードによる転調を含む長調に関しては正しい解析を行うことができた。また、提案手法により新たに加えた機能に関しても適切な出力を得ることができた。一方で、ピボットコード以外による転調や、複数の短音階をもつ短調に対しては、つねに適切な出力を得ることが難しく、さらなる改良が必要である。

今後の課題としては、メロディ依存の和音進行や、モード・ジャズへの対応が挙げられる。また、多くの楽曲に対しての解析による定量的な評価が必要となってくる。

目次

第1章 はじめに	1
1.1 研究の背景	1
1.1.1 音楽情報科学	1
1.1.2 計算論的生成音楽理論	1
1.1.3 音楽理論	2
1.1.4 和声とその応用	2
1.1.5 ジャズ	2
1.1.6 先行研究	4
1.2 研究の目的	5
1.3 本稿の構成	5
第2章 Tonal Pitch Space	6
2.1 Tonal Pitch Space (TPS)	6
2.1.1 ピッチクラス	6
2.1.2 和音の表記方法について	7
2.1.3 近親調における和音間距離	7
2.1.4 坂本らによる TPS を用いた和声解析	9
2.2 TPS のジャズへの適用 [7]	11
2.2.1 文献 [7] の指摘による現行の TPS の問題点	11
2.2.2 文献 [7] による TPS の提案手法	11
2.2.3 文献 [7] による和声進行を入力とした実験	14
第3章 組み合わせ範疇文法 (Combinatory Categorial Grammar)	16
3.1 Combinatory Categorial Grammar (CCG)	16
3.1.1 カテゴリ	18
3.1.2 意味表示	19
3.1.3 組合せ規則	19
3.2 和声解析に向けた組み合わせ範疇文法の応用 [11]	20
3.2.1 組合せ規則	20
3.2.2 意味表示	22
3.2.3 辞書項目	23

第4章 組み合わせ範疇文法を用いたジャズ和声解析手法の改良	31
4.1 文献 [11] における諸問題	31
4.1.1 サブドミナントからドミナントへの進行	31
4.1.2 三和音に対するカテゴリの不適切な付与	32
4.1.3 ピボットコードを含む転調	33
4.2 本研究における文献 [11] の諸問題の解決	34
4.2.1 ディグリーネーム表記への変更	34
4.2.2 ジャズ特有の和音および機能への対応	34
4.2.3 辞書項目の改良	35
4.2.4 転調間におけるピボットコードの発見	38
4.3 ジャズ和声に対応したTPS[7]の導入	41
4.3.1 文献 [7] の手法を用いた転調を含む和音進行の解析実験	41
4.3.2 四和音への対応	42
第5章 ジャズ和声解析システムの実装	44
5.1 TPS 和声解析部	44
5.1.1 ノード生成部	44
5.1.2 和音間距離計算部	45
5.1.3 最短距離計算部	47
5.2 CCG 和声解析部	47
5.2.1 analyzeCCG メソッド	47
5.2.2 innerCCG メソッド	48
5.2.3 degreeNameConversion メソッド	49
5.2.4 ruleCheck メソッド	50
第6章 実験	55
6.1 実験結果	56
6.2 評価・考察	64
6.2.1 長調を想定した和音進行に関して	64
6.2.2 短調を想定した和音進行に関して	64
6.2.3 短い転調を繰り返した和音進行に関して	64
第7章 おわりに	67

図 目 次

2.1	調の五度圏	8
2.2	和音の五度圏	9
2.3	ベーシックスペースの例: (a)I/C; (b)iv/b	10
2.4	坂本らによるTPSを用いた和声解析	10
2.5	II ₇ /Cのベーシックスペース	12
2.6	II ₇ /Cのベーシックスペース	12
2.7	拡張された和音の五度圏	13
2.8	拡張された和音の五度圏距離	14
2.9	文献[7]によるノンダイアトニックコードを含む和音進行を解析した例	14
3.1	CFGに基づく構文解析	17
3.2	CCGに基づく構文解析	18
3.3	CCGに基づく構文解析（証明システムに準ずる記述）	19
3.4	関数適用規則の例	21
3.5	関数合成規則の例	21
3.6	等位接続規則の例	22
3.7	意味構造とオイラー格子への対応	23
3.8	文献[11]で用いられた辞書項目	24
3.9	文献[11]における辞書項目からのカテゴリの割り当て1	26
3.10	文献[11]における辞書項目からのカテゴリの割り当て2	26
3.11	文献[11]における辞書項目からのカテゴリの割り当て3	27
3.12	文献[11]における辞書項目からのカテゴリの割り当て4	27
3.13	文献[11]における辞書項目からのカテゴリの割り当て5	28
3.14	文献[11]における辞書項目からのカテゴリの割り当て6	28
3.15	文献[11]における辞書項目からのカテゴリの割り当て7	29
3.16	文献[11]における辞書項目からのカテゴリの割り当て8	29
3.17	文献[11]における辞書項目からのカテゴリの割り当て9	30
4.1	Dm7 → G7 → Cに対する解析	32
4.2	C → Am → Dm → G → Cに対する解析	32
4.3	提案手法による転調を含む和音進行の解析1	38
4.4	提案手法による転調を含む和音進行の解析2	39

4.5 提案手法による転調を含む和音進行の解析 3	39
4.6 提案手法による転調を含む和音進行の解析 4	40
4.7 提案手法による転調を含む和音進行の解析 5	40
4.8 文献 [7] の提案手法を用いた転調を含む和音進行に対する実験結果の一部	42
4.9 四和音に対応した長調における和音の五度圏	43
4.10 四和音に対応した短調における和音の五度圏	43
 6.1 実行の様子	55
6.2 実験結果 1	57
6.3 実験結果 2	58
6.4 実験結果 3	59
6.5 実験結果 4	60
6.6 実験結果 5	61
6.7 実験結果 6	62
6.8 実験結果 7	63
6.9 実験結果 5 の TPS 和声解析部による調推定	65
6.10 実験結果 6 の TPS 和声解析部による調推定	65
6.11 実験結果 7 の TPS 和声解析部による調推定	66
6.12 三種類の短音階に対応した和音の五度圏	66

表 目 次

2.1	ピッチクラスの値と音名	6
2.2	ノンダイアトニックコードとハ長調上の各ダイアトニックコードとのベーシックスペース距離	13
3.1	CFG の生成規則の例	16
4.1	ピボットコードを含む和音進行の解釈	33
4.2	提案手法による辞書項目（トニック・サブドミナント・ドミナント・サブドミナントマイナー）	36
4.3	提案手法による辞書項目（リレイテッドセカンドマイナー・セカンドダイードミナント・パッシングディミニッシュ）	37
5.1	調の主音と調ピッチクラスの値	44
5.5	長調における辞書項目の配列の一部と要素	51
5.6	短調における辞書項目の配列と要素	52
5.2	$CN[2] = \{9, m7\}$ に対するノードの生成例	53
5.3	ダイアトニックコードとダイアトニックピッチクラス値	53
5.4	ディグリーネーム変換テーブル	54

第1章 はじめに

1.1 研究の背景

1.1.1 音楽情報科学

近年、音楽情報科学が大きな注目を集めている[1]。シンセサイザーによる電子音楽やデスクトップミュージック(DTM)などの作曲・編曲・演奏支援の研究が行われてきた。このような技術は現代の音楽制作に欠かせないものとなっているが、その多くは音楽家向けであり、音楽の初学者にとっては敷居が高いのが現状である。そのため、音楽知識に乏しいユーザでも手軽に作編曲ができる、自動作編曲、作編曲支援、などといった技術への需要が高まっている[2]。

また、近年では音楽家だけでなくエンドユーザが恩恵を受ける技術にも注目が集まっている。2000年代に入ると、計算機やインターネットの性能向上・普及・低価格化を背景に、iPodやWalkmanなどの小型音楽再生機が普及した。これにより誰もが膨大な数の楽曲を持ち歩くようになった。さらに2015年にはApple社やGoogle社などの大手IT企業が揃って定額制のストリーミング配信を開始した。世界中で「聴き放題」の聴取スタイルが定着しつつあるなか、数千万曲の膨大な楽曲を所有し配信するこのようなサービスは、ユーザ自身がキーワードやジャンル検索をせずとも、聴きたい音楽や新しい楽曲との出会えるということをコンセプトの一つとしているため、楽曲間の類似度推定に基づいた音楽情報検索や音楽レコメンドに関する研究が注目され始めている。

1.1.2 計算論的生成音楽理論

上記のような音楽情報科学の研究は隠れマルコフモデルやクロマベクトルを用いているものが多いが[3]、音楽理論に基づく研究は少ない。従来の音楽理論は経験則によって成り立っており、その曖昧さから計算機への実装を難しくしていることが原因ではないかと考えられる。そこで、音楽理論を計算機への実装させるために、音楽理論を数学的な手法を用いて定量的に捉え直した、計算論的生成音楽理論の研究が盛んに行われている。

1.1.3 音楽理論

音の要素は、基本周波数、その中に含まれる周波数、大きさ、周期性、音源の方向などが主なものとして挙げられる。音楽は複数の音が時間軸方向と周波数軸方向に分布することによって成り立つ。音を無秩序に並べただけでは、音楽とは知覚されないが、文法や語彙、構造などの一定の秩序に基づいて音が配置されることで音の集合は音楽として知覚される。この秩序を知識体系によって説明したものが音楽理論である。作曲者や演奏者は音楽理論の知識に基づいて作曲や楽曲分析活動を行っている。

音楽理論は文化圏や時代によって様々であるが、現在の音楽大学で中心的に教えられているのは19世紀後半あたりまでのドイツなどで体系づけられたものが中心となっており、「楽典」「和声学」「対位法」「楽式論」などに細分化することができる。

この19世紀後半に構築された理論を基本とし、その後の西洋音楽においては近代・現代音楽の時代になるにつれ、シェーンベルクによる十二音技法などの新しい理論を生み出していく。

1.1.4 和声とその応用

本研究では和声(harmony)に着目する。和声は、調性をもつ音楽においては旋律(melody)、律動(rhythm)と並び音楽の最重要要素の一つである。

和声を解析することは、一般的には調推定と和音推定を行い、それらをもとに和音がどのような機能や役割をもっているのかを分析することである。

人間が採譜を行う際、和音を認識してから個々の音を認識するというアプローチがあるが、鳴っているすべての音がどの音高なのかを特定するのは難しい。しかし、和声に基づいた曲は和音構成音を含むことが多いことから、和声進行を手がかりに全構成音を推定している。これと同様のアプローチが計算機によって行うことができれば、自動採譜の性能を向上させることができると考えられる。つまり、計算機による和声解析が自動採譜の要素技術として重要であると考えられる[2]。

また、ある和音と同じ機能をもつ和音に置き換えることが可能である。既存の和音進行を変化させるリハーモナイズと呼ばれる編曲技法がある。人間による編曲では和声解析を行った後にリハーモナイズを適用するが、計算機に対しても和声解析を基にリハーモナイズの実現が可能になるのではないかと考えられる。

その他、和声進行を用いることで、音楽情報検索や曲間の類似度によるカバー曲の識別などへの応用できるのではないかと考えられる。

1.1.5 ジャズ

本研究ではジャズに着目する。ここでのジャズとは1950年代におけるビバップの時代まで演奏されてきたものを指すが、そこで用いられた和声や理論が現在流通している商業

音楽に大きな影響を与えており、このジャズで用いられている和声理論を和声解析システムに取り入れることで、より多くの楽曲の機能的解析が可能になるのではないかと考えられる。

以下は本研究と深く関係がある和音進行や調性に重点を置き、ジャズの歴史について示した[15]。

ラグ・タイム

諸説あるが、20世紀初頭のラグ・タイムがその起源だと言われている。和音進行には調性があり、セカンダリー・ドミナントも存在する。

ビバップ

1940年～1950年はビバップの時代である。この時代よりアドリブが行われるようになった。Charlie Parker(1920～1955)を中心として、原曲のコード進行を様々なサブコード（裏コードを含む）を用いてリハーモナイズすることでアドリブを行うスタイルが確立された。頻繁な転調やセカンダリードミナントやテンション等の使用によって調性は崩壊寸前まで陥った。西洋音楽における後期ロマン派の音楽と本質的に同じ内容を持っている。また、アドリブはコードの構成音に基づいて行われるため、誰が行っても同じようなアドリブになったため、1950年の終わりごろには行き詰まりを呈した。

モード・ジャズ

Miles Davis(1926～)によって旋法（Mode）の手法が導入された。西洋音楽において後期ロマン派音楽の行き詰まりを開拓するためのDebussyが果たした役割と等しい。あるモードで構成された曲または曲の一部のアドリブはモードの構成音を自由に用いて行われる。あるモードに基づくダイアトニックコードを用いて、曲または曲の一部を構成するモーダル・ハーモニーという手法が用いられた。John Coltrane(1926～1970)によって、モード・ジャズ以前のスタンダード曲におけるアドリブも、各和音のアベイラブル・ノート・スケールをモードとみなして行われた。

フリー・ジャズ

ビバップの行き詰まりを開拓する方法としてモードの手法とは別に、既成の概念を全て否定するフリー・ジャズが1960年代にOrnette Coleman(1930～)やJohn Coltrane(1926～1970)によって行われた。ここで既成の概念とは調性、メロディ、コード進行、リズムを含む。

ポスト・フリー

1970 年代はポスト・フリーの時代と呼ばれる。フリー・ジャズの時代が、既成の概念をただ闇雲に否定するのに対して、ポスト・フリーの時代は、既成の概念を否定しつつも新しい秩序を生み出そうとする模索の時代である。現代のジャズもポスト・フリーに含まれ、西洋音楽における現代音楽と同じ精神をもっている。ポスト・フリーでは、フリー・ジャズで否定されたコードやモードが再び用いられるようになるが、以下のような新しい秩序の中で用いられるように工夫される。これらは現代音楽で用いられる手法を独自の発想で導入している。

1. ドミナント・モーションをもたないコード進行

- (a) 頻繁な自由な転調
- (b) 12 音技法をコード進行に取り入れる方法
- (c) 分数コード（スーパー・インポーズド・コード）

2. モード手法の発展

- (a) 新しいモードの創造（コンポジット・スケール）
- (b) モード的フレージングにおける新しい技法（アッパー・ストラクチャー・トライアドの応用、ペンタトニックスケールの応用、その他）
- (c) ポリ・モード

このようにジャズの定義は時代によって様々であるが、本研究の対象は、ラグ・タイムからビバップまでとする。

1.1.6 先行研究

和声に着目した計算論的音楽理論で有名なものとしては、クラシック音楽の和声解析を前提とした F.Lerdahl による Tonal Pitch Space (TPS) が挙げられるが [5] , TPS は取り扱える和音の種類が限定されているため、現在流通している音楽における和声を機能的に把握することができない。ジャズをはじめとした商業音楽は非調構成音を含む和音を用いていることに着目し、ジャズに対応した TPS の拡張の研究が行われている [7]。

また、ジャズ理論を用いた和声の解析手法として、Granroth らによる組合せ範疇文法 (Combinatory Categorial Grammar (以下 CCG とする)) [10] を利用した解析手法がある [11]。これは音楽における言語起源説 [4] の立場から自然言語処理の手法を和声解析に応用したものである。

1.2 研究の目的

本研究の目的は、ジャズ和声を含む商業音楽に向けた和声解析システムの実装である。和声解析の入力は調の情報を含まないコードネームを前提とする。先行研究にてCCGを用いたジャズ和声の解析手法[11]が提案されているが、本研究では[11]に対する諸問題を調査し、かつ解決させたものを手法に取り入れる。また、その過程で調推定が必要になるため、先行研究[7]にて提案されたTPSによる和声解析手法をCCGによる解析の前段階として取り入れる。

1.3 本稿の構成

本稿の構成は次の通りである。

第2章でTPSと先行研究[7]で提案されたTPSについて述べる。

第3章でCCGと先行研究[11]で提案されたCCGのジャズ和声解析の手法について述べる。

第4章で本研究で扱うジャズ和声解析の提案手法について述べる。

第5章で具体的な実装方法について述べる。

第6章で提案手法の妥当性を示すための実験とその結果・考察について述べる。

第7章で本稿のまとめと今後の展望について述べる。

第2章 Tonal Pitch Space

2.1 Tonal Pitch Space (TPS)

TPS[5] は、ピッチや和音、調のような音楽的要素が変化することで聴き手がどのように感じるかを定量的に表現する、既知の音楽理論を基礎とした認知科学的モデルである。2001年にF. Lerdahlによって提案された認知科学的な音楽理論であり、LerdahlとJackendoffの提案した認知的な観点から楽曲の木構造を構築するルールベースの理論である GTTM[?] を基礎として、その補完を行うために TPS は提案されたものだが、GTTM とは独立な理論である。

本研究では和声解析を定量的に行うことを目的として TPS を用いる。TPS は五度圏などの既知の音楽理論だけでなく後述するベーシックスペースという新しい概念を用いることにより、それぞれある調に基づいた二つの和音間にに関して、音程間、和音間、調間に定量的な距離を定め、この距離の和が近いほどそれらは安定して心地良い進行とし、逆に距離が長いほど不自然で違和感ある進行としている。

本研究では TPS の根幹部分である和音間距離の計算部分について [5] より述べることにする。

2.1.1 ピッチクラス

TPS では、ピッチクラスという音名に関する概念を取り入れている。表 2.1 のように 1 オクターブの中に含まれている 12 個の C～B までの半音階にラベルとして数字を割り当てる。F3 と F4 のようなオクターブの異なる音は、オクターブを考慮せずに同一のピッチクラス値を与えていている。TPS を用いる目的は和声解析であり、ここでは和音構成音に重点を置いているためである。また、十二平均律で考えているため、純正律における C[#]4 と D^b4 のような異名同音は同一のピッチクラスに属している。以降は、本研究におけるピッチクラス値は、p0, p1, のように頭に p を付けて表現する。

表 2.1 ピッチクラスの値と音名

音名	C	C [#] /D ^b	D	D [#] /E ^b	E	F	F [#] /G ^b	G	G [#] /A ^b	A	A [#] /B ^b	B
ピッチクラス値	0	1	2	3	4	5	6	7	8	9	10	11

2.1.2 和音の表記方法について

便宜上、本章における和音の表記は以下の方法による表記とする。

- 調はアルファベットで表す。長調は大文字、短調は小文字での表記とする。
- コードはギリシャ数字で表す。これらの数字はそのコードが属する調の音階に対して何度の音にあたるのかを示す。
- 長調や長和音は大文字、短調や短和音は小文字の表記とする。
- ”和音/調”の形式で表記する。例えばハ長調の4度長和音はIV/Cと表す。
- 四和音およびディミニッシュなどの三和音以外の情報はギリシャ数字の右肩に表記する。
- 半音低い、または高いことを示す場合はギリシャ数字の左側に♭や♯をつけて表記する。
- コードネームの表記はバークリーメソッド[13]に従う。

2.1.3 近親調における和音間距離

和音間距離は近親調間の場合と遠隔調間の場合で異なり、遠隔調における和音間距離 Δ は近親調における和音間距離 δ を基礎として導出される。近親調における和音間距離は以下の式で表される。

$$\delta(x, y) = i(x, y) + j(x, y) + k(x, y) \quad (2.1)$$

ここで、引数である x と y はそれぞれ進行前の和音、進行後の和音である。 i は調間距離、 j は和音の五度圏距離、 k はベーシックスペース距離を表しており、これらの3つの距離の和が近親調における和音間距離となる。 i, j, k についての詳細を以降で述べる。

- 調間距離 i

和音 x, y が属している調が図2.1に示す調の五度圏[?]上でどれだけ離れているかを示したのが調間距離である。調の五度圏とはCからBまでの各調の主音を、隣同士が完全五度の関係になるように円環状に並べたものである。調号で考えると、図2.1は右隣に1ステップ進むと♯が1つ増え、♭が一個減る。また左隣に1ステップ進むと♭が1つ増え、♯が1つ減っている。そのため、平行調の関係にある和音間の調間距離は0となる。

例として、I/Cからiv/bへ進行する際の調間距離を求める。I/Cはハ長調でiv/bはロ短調であり、図においてハ長調から右に2ステップ進むとロ短調になる。従ってこのときの調間距離は2となる。

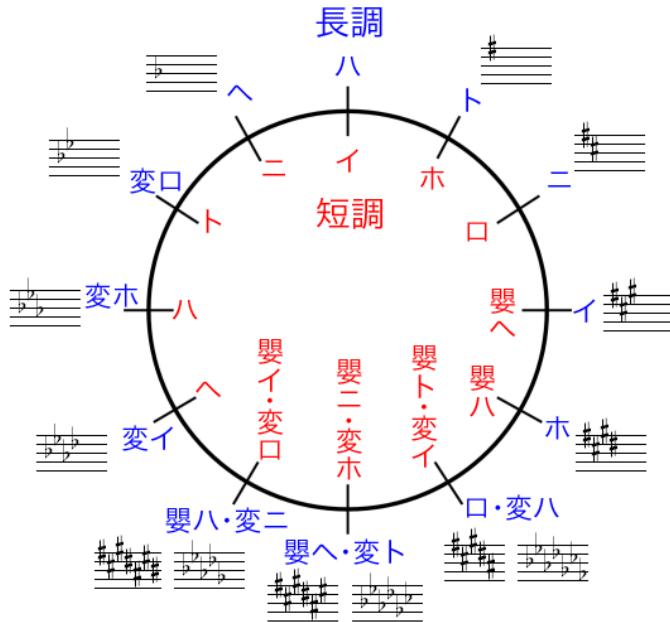


図 2.1 調の五度圏
([13] より)

- 和音の五度圏距離 j

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。

つまり、図 2.2 に示す和音の五度圏上において、和音 x のルート音から和音 y のルート音へ到達するために必要なステップ数の最短距離が和音の五度圏距離である。

例として、I/C から iv/b へ進行する際の和音の五度圏距離を求める。I/C のルート音 p0 から iv/b のルート音 p4 へ到達するためのステップ数は、図 2.2 の和音の五度圏距離において 3 である。従ってこのときの和音の五度圏距離は 3 である。

- ベーシックスペース距離 k

ベーシックスペースとは、各音がその和音のなかでどれだけ重要であるのか重み付けを行い、一つの和音に対して、横軸をピッチクラス、縦軸をその影響度の level とした二次元空間である。例として I/C のベーシックスペースを図 2.3 の (a) に示す。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

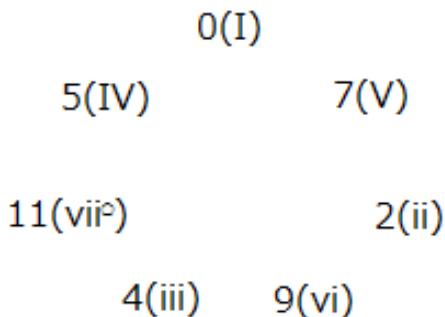


図 2.2 和音の五度圏
([6] より)

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。つまり、ある音がベーシックスペースで上のレベルにくる音ほど、その和音の中で影響度の高い音であることがいえる。また、和音の構成音だけでなく、その和音が置かれている調もベーシックスペース上で表現される。

続いて、ベーシックスペース距離について述べる。進行前の和音 x のベーシックスペースではなく、進行後の和音 y のベーシックスペースで新たに増えたピッチクラスの数がベーシックスペース距離である。

例として I/C から iv/b へ進行する際のベーシックスペース距離を求める。図 2.3 の (a) と (b) を比較すると、(a) のベーシックスペースから (b) のベーシックスペースで増えたピッチクラスは、p1 が 1 つ、p4 が 2 つ、p6 が 1 つ、p11 が 2 つであり（下線部），これらの数の合計は 6 となる。これがベーシックスペース距離である。

2.1.4 坂本らによる TPS を用いた和声解析

TPS で計算される和音間距離は人間の音楽認知との近似がみられ、一般的な和声学の感覚に近い。また、如何なる和音に対しても計算ができるため全域的なアルゴリズムを可能とし、なおかつシンプルな設計になっている。このことから坂本らは文献 [6] で TPS を和声解析に用いている。大まかではあるが、その手順を以下で述べる。

1. コードネームの配列を入力として受け取る
2. 各コードネームの解釈として可能性のある和音をノードとして生成する
3. 各ノードを連結し、図??のようにパスを作成する。
4. すべてのノード間で和音間距離を計算し、楽曲の始まりから終わりまでの最短距離になるようなノードを選択する

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

図 2.3 ベーシックスペースの例: (a)I/C; (b)iv/b

5. そのノードが表す和音の配列を解析結果として出力する

つまり、この計算結果は、人間の音楽認知的に最も自然な解釈になるような和音が各コードネームに割り当てられるということである。

しかしながら、TPS は取り扱える和音の種類が限定されており、そのままではジャズで用いられる和音進行に応用することはできない。そのため次節で述べる提案手法を用いたいと考えている。

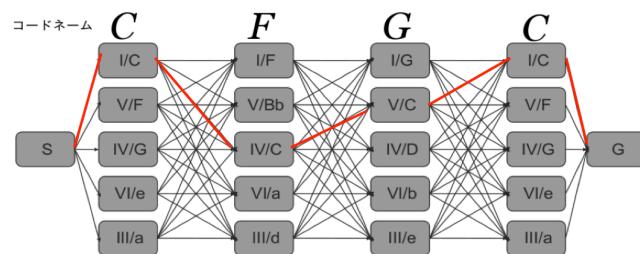


図 2.4 坂本らによる TPS を用いた和声解析
([6] より)

2.2 TPS のジャズへの適用 [7]

本研究では、ジャズで用いられる和音を用いたい。そこで、山口らが提案した TPS[7] を用いたいと考えている。

2.2.1 文献 [7] の指摘による現行の TPS の問題点

山口らは TPS をジャズ和音に適用するにあたり、現行の TPS に対して以下の問題点を指摘している。

問題点 1 ベーシックスペースにおいて、あるレベルで存在しているピッチクラスは、その下位のレベルにも存在していなければならない。

- 言い換えると、和音構成音は調構成音でなくてはならないため、解析の対象はダイアトニックコードのみに限られる。
- 例えば、調構成音がない音を和音構成音にもつ和音（ノンダイアトニックコード）の一つである II_7/C をベーシックスペース上で表現すると、図2.5のように記号 ϕ で示した部分に『抜け』が生じる。
- このように、現行の TPS はジャズでよく用いられるノンダイアトニックコードに対応していない。

問題点 2 ベーシックスペースにおいて、主音と五度音以外の和音構成音はすべて level c とされる。

- つまり、和音内での 3 度音と 7 度音などの付加音の影響度が等しいとされている。
- 付加音は Omit しても和音の機能は変わらないが、それと比較して 3 度音は和音の響きを決める重要な音である。付加音と 3 度音の重みを同じくするべきではない。

問題点 3 和音の五度圏距離においても chordal circle-of-fifth ルールによりダイアトニックコードにしか対応していない

2.2.2 文献 [7] による TPS の提案手法

これらの問題を解決するため、山口ら [7] は TPS を以降で述べるように改訂することを提案している。

- ベーシックスペースの改訂

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

図 2.5 II_7/C のベーシックスペース

提案手法の 1 つめに、ベーシックスペースの改訂が挙げられている。前節の問題点 2 への対応として、Level をもう一段増やし、ベーシックスペースの縦軸を level a～level f の重みをつける。つまり、level a では根音、level b では根音と五度音、level c では根音・五度音・三度音、level d では和音構成音、level e では調構成音、level f ではすべて音がそれぞれ含まれるようにすることである。これにより、根音・五度音・三度音とその他の和音構成音との間に重みの差をつけることが可能となる。

また、問題点 1 で述べられている記号 ϕ のような「抜け」の問題については、 ϕ を上下のピッチクラス値で埋める。すなわち、調構成音でない音が和音構成音になった場合、一時的にその音を調構成音として認めることである。

以上をまとめると、図 2.5 は以下のようになる。

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

図 2.6 II_7/C のベーシックスペース

• 和音の五度圏の拡張

提案手法の 2 つめに、和音の五度圏の拡張が挙げられる。その手順を以下にて述べる。

1. 入力として与えられた二つの遷移前後の和音と、その和音が属する調のダイアトニックコードとのベーシックスペース距離をそれぞれ求める。
2. 得られた距離が最小となったダイアトニックコードと、与えられた和音とをそれぞれ連結し、和音の五度圏を拡張する。
3. 拡張によって新たに得られた和音の五度圏上で、与えられた二つの和音の和音の五度圏距離を求める。

例として, ii_7/C から V_7/C に進行するときの和音の五度圏距離を求める。

まず, 二つの和音はどちらもハ長調に属しているため, 表 2.2 のように与えられた二つの和音とハ長調上の各ダイアトニックコードのベーシックスペース距離を計算する。ここでの $\text{basic}(x, y)$ は和音 x から和音 y へのベーシックスペース距離を表す。

表 2.2 ノンダイアトニックコードとハ長調上の各ダイアトニックコードとのベーシックスペース距離

x	I	ii	iii	IV	V	vi	vii ^(b5)
$\text{basic}(\text{ii}_7/\text{C}, x)$	9	1	10	4	8	6	8
$\text{basic}(\text{V}_7/\text{C}, x)$	8	7	5	9	1	10	5

表 2.2 の下線部で示した部分が計算によって得られた距離が最小となったダイアトニックコードである。これらと入力時に与えられた二つの和音をそれぞれ連結し、図 2.7 のように和音の五度圏を拡張する。

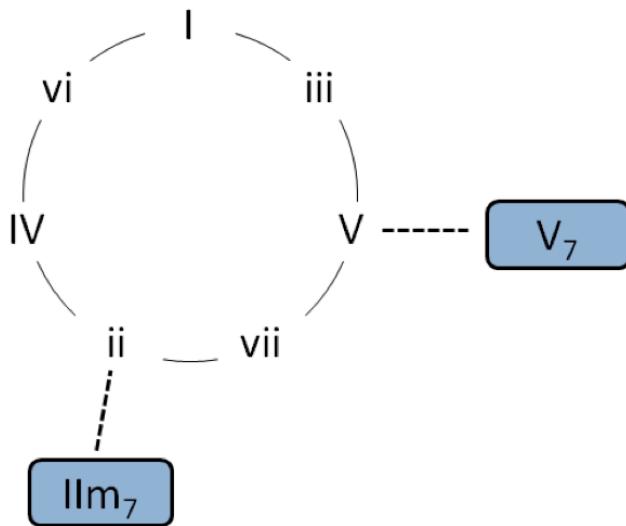


図 2.7 拡張された和音の五度圏

([7] より)

図 2.8 のように拡張された和音の五度圏上で、入力時に与えられた二つの和音間の距離を求める。この例では ii_7/C から V_7/C に進行するときの和音の五度圏距離は 4 となる。

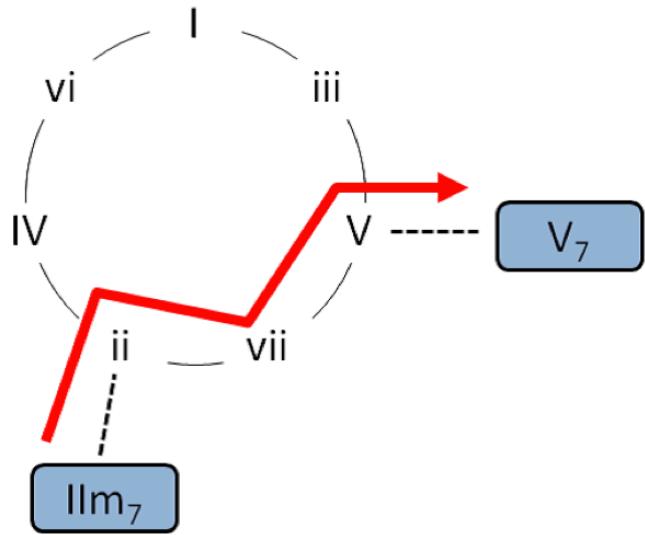


図 2.8 拡張された和音の五度圏距離
([7] より)

2.2.3 文献 [7] による和声進行を入力とした実験

文献 [7] では、和音進行 $C \rightarrow A7 \rightarrow Dm \rightarrow G7 \rightarrow C$ を入力とした和声解析の実験を行っている。図 2.9 はその結果の一部であるが、全ての和音がハ長調上にあり、転調は起きないという結果になっている。A7 以外の和音はハ長調上のダイアトニックコードであるため、この和音進行全体をハ長調上の進行と仮定すると、A7 はノンダイアトニックコードであるが、このように前節で述べた提案手法を TPS に取り入れることによって、従来の TPS と異なり、ノンダイアトニックコードが含まれる和音進行でも和声解析を行うことができる。

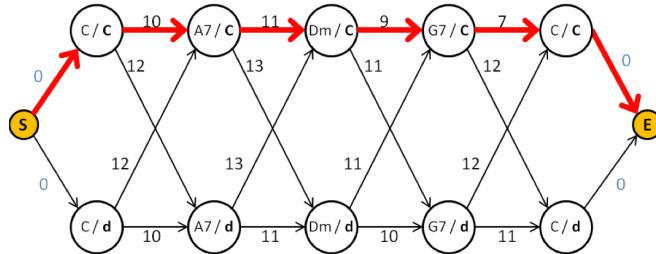


図 2.9 文献 [7] によるノンダイアトニックコードを含む和音進行を解析した例
([7] より)

A7はセカンダリー・ドミナントとよばれるものであるが、セカンダリー・ドミナントを自身の前後の和音の調に属するものとみなして進行全体で転調していないと解釈するか、セカンダリー・ドミナント進行の次の和音がトニックとなるように部分的な転調が起こっていると解釈するかで見解が分かれている。文献[15]によると、前者の解釈が正しいとしているため、本研究においても前者の解釈が正しいものとして議論を進める。

第3章 組み合わせ範疇文法 (Combinatory Categorial Grammar)

3.1 Combinatory Categorial Grammar (CCG)

形式文法には様々な種類があるが、最も基本的なものの一つとして文脈自由文法 (Context Free Grammar (以下 CFG とする)) [9] が挙げられる。CFG は非終端記号、終端記号、開始記号、生成規則から成る。開始記号からはじまり、生成規則を適用することで文字列を生成し、文字列が終端記号のみになれば終了するというルールで生成を行う。例えば、以下の表 3.1 にあるような CFG が与えられたとき、図 3.1 のような結果が導出される。

表 3.1 CFG の生成規則の例

開始記号	:	文
終端記号	:	単語文字列
非終端記号	:	品詞・句
生成規則	:	文 → 格助詞句 動詞句
		格助詞句 → 名詞句 格助詞
		名詞句 → 名詞
		動詞句 → 動詞 格助詞句 動詞句
		格助詞 → が を に
		名詞 → きつね 油揚げ
		動詞 → 買う 食べる

CFG は単純なアルゴリズムのため、数学的な研究手法には使いやすいが、完全な文法を書くためにはルール数が増大してしまうという問題点もある。例えば、先程の文法例に「～買わない」という句の解析をするためには、

- 動詞句 → 動詞 助動詞
- 動詞 → 買わ

- 助動詞 → ない

という生成規則の追加が必要である。しかし、この規則を追加するだけでは「～買うない」や「～買わ」などの非文も正しい文だとみなされてしまう。この場合は動詞の活用と助動詞の対応がとれていないため、これを修正するためには以下の生成規則が新たに必要となる。

- 動詞-終止形 → 買う
- 動詞-未然形 → 買わ
- 動詞句-終止形 → 格助詞 動詞句-終止形 | 動詞-未然形 助動詞-未然形-終止形

「～買わない」の解析するだけの必要な生成規則を追加したが、このように全ての表現を解析するには生成規則を無限に追加していかなければならない。

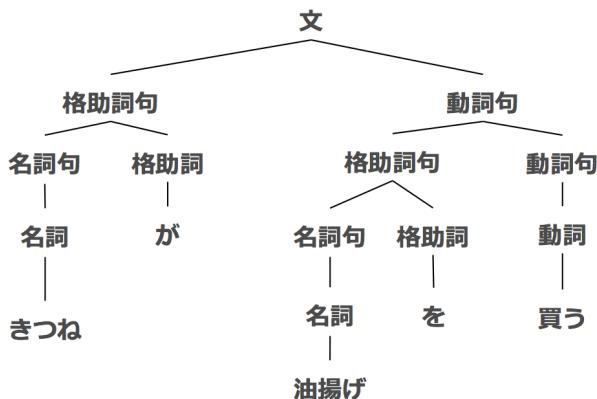


図 3.1 CFG に基づく構文解析

これを解決するために、生成規則を少数に抑え、単語自体に情報を詰め込む語彙化文法という方法が存在する。組合せ範疇文法 (Combinatory Categorial Grammar (以下 CCG とする)) [10] はそうした語彙化文法の一種であり、本研究では形式文法として CCG を採用する。

CCG は辞書といいくつかの組合せ規則によって構成される。辞書は多数の語彙項目を含んでおり、語彙項目は表層形、カテゴリ、意味表示からなる。例えば、

$$\begin{aligned}
 \text{Mary} &:= NP : j \\
 \text{loves} &:= S \setminus NP / NP : \lambda y. \lambda x. \lambda e. \text{love}(e, x, y) \\
 \text{Manny} &:= NP : v
 \end{aligned}$$

のように各語について語彙項目が定められている。ここで S は sentence (文), NP は noun phrase (名詞句) 略であり、どちらもカテゴリである。また”:”の右側は意味表示を表している。

次節以降ではカテゴリ、組合せ規則、意味表示について説明する。

3.1.1 カテゴリ

カテゴリは言語学における品詞、プログラミング言語における型に相当する。カテゴリは基底カテゴリの集合として定義されており、自然言語においては、普通名詞 N , 固有名詞 NP , 文 S , 伝達文, 接続詞 $CONJ$ の 5 種類の基底カテゴリが定義されている。これらから構成されるカテゴリの集合は自動詞 $S \setminus NP$ や他動詞 ($S \setminus NP) / NP$ のように “/” や “\” によって表現される。カテゴリ X / Y や $X \setminus Y$ における演算子 ‘/’ ‘\’ の直感的な意味は以下の通りである。

演算子 ‘/’ の直感的な意味

カテゴリが X / Y である語句は、自身の右側にカテゴリ Y を持つ語句が現れたとき、それと組み合わせると、カテゴリ X をもつ語句となる。

演算子 ‘\’ の直感的な意味

カテゴリが $X \setminus Y$ である語句は、自身の左側にカテゴリ Y を持つ語句が現れたとき、それと組み合わせると、カテゴリ X をもつ語句となる。

ここで、文 Mary loves Manny の統語構造を以下の図 3.2 に示す。

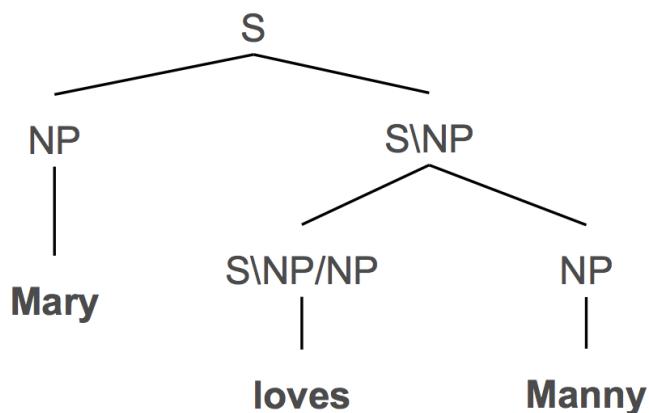


図 3.2 CCG に基づく構文解析

CCG における統語構造の導出は証明システムに準ずる記述で書くのが一般的であるため、以後は以下の図 3.3 のような逆さまの木構造として表記する。

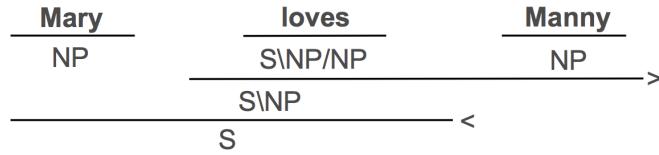


図 3.3 CCG に基づく構文解析（証明システムに準ずる記述）

また、現実的な文法を記述するにあたり、カテゴリについてより詳しい指定が必要である。たとえば、”eats”の主語は三人称単数であり、目的語は対格を持たなければならないが、このような情報は統語素性として、基本カテゴリに添字として記す。例えば、統語素性 nom, acc を持つカテゴリ NP は、それぞれ主格 (nominative case), 対格 (accusative case) の名詞句を表し、統語素性 3S, 3P を持つカテゴリ NP は、それぞれ三人称単数、三人称複数の名詞句を表す。nom | acc のような記法は「nom または acc」という意味に解釈するものとする。

3.1.2 意味表示

各カテゴリに対して意味構造の λ 式が割り当てられる。 λ 計算を行うことで意味構造を計算することができる。

3.1.3 組合せ規則

組合せ規則は、各語彙から文を生成するための規則であり、カテゴリと意味表示のそれぞれの合成方法を定義したものである。主な規則は以下の 4 つである。

関数適用規則

$$X/Y : f \quad Y : a \quad \rightarrow \quad X : fa \quad (>) \quad (3.1)$$

$$Y : a \quad X \setminus Y : f \quad \rightarrow \quad X : fa \quad (<) \quad (3.2)$$

関数合成規則

$$X/Y : f \quad Y/Z : g \quad \rightarrow \quad X/Z : \lambda x. fgx \quad (> B) \quad (3.3)$$

$$Y \setminus Z : g \quad X \setminus Y : f \quad \rightarrow \quad X \setminus Z : \lambda x. fgx \quad (< B) \quad (3.4)$$

等位接続規則

$$X : f_1 \dots CONJ : \circ X : f_m \quad \rightarrow \quad X : \lambda x. (f_1 x) \circ \dots \circ (f_m x) \quad (< \phi >) \quad (3.5)$$

型繰り上げ規則

$$X : a \rightarrow T / (T \setminus X) : \lambda P.Pa \quad (> T) \quad (3.6)$$

$$X : a \rightarrow T \setminus (T / X) : \lambda P.Pa \quad (> T) \quad (3.7)$$

辞書と組合せ規則を用いると”Mary loves Manny”について以下の導出が得られる。

この導出の最終結果 $S : \lambda e.love(e, j, v)$ は”Mary loves Manny”が正しく文 (S) であり、意味が $\lambda e.love(e, j, v)$ と表せるこことを示している。

3.2 和声解析に向けた組み合わせ範疇文法の応用 [11]

ここで、CCG を直感的に捉える。例えば、英語における他動詞の後ろには目的語が来なければならない。つまり他動詞からみると目的語が後に来るこを『期待する』ような状態であり、他動詞のカテゴリを X / Y とすると $/$ の右側の Y が『期待する』対象のカテゴリとなる。従ってこの Y は目的語を表すカテゴリと一致する。これと同様のことを和音進行に当てはめて和声解析を行ったのが文献 [11] である。例えば、ハ長調における $G7$ から C への進行はドミナント・モーションと呼ばれる。ドミナント・モーションとは、ドミナントの機能を持つ和音からトニックの機能をもつ和音へ進行することである。聴き手は、不安定な響きをもつドミナントを聴いたらその次の和音はトニックが来るこを期待する。ここでもドミナントである $G7$ からみたらトニックの C を『期待している』状態である。そのため、 $G7$ のカテゴリは $G^D / C^{D|T}$ と表現することができる。カテゴリの添字は前節と同様に統語素性であり、 D と T はそれぞれドミナントとトニックを表す。また、トニックである C にも同様のカテゴリによる表現が可能だが、トニックはいかなる和音への進行が可能なので $/$ などを用いない基本カテゴリ C^T で表現することができる。さて、 $G7 \rightarrow C$ という進行をカテゴリで並べると $G^D / C^{D|T} C^T$ となるが、この場合、前で述べた規則の一番目を用いることができる。つまりカテゴリ C^T がカテゴリ $G^D / C^{D|T}$ の右側に来たら一つの基本カテゴリとして組み合わされるということである。

3.2.1 組合せ規則

文献 [11] では和音進行を対象としたCCGにおいて次の4つの式を用いている。以下に各式の詳細を記述する。

関数適用規則

$$X / Y : f \quad Y - Z : x \rightarrow X - Z : f(x) \quad (>) \quad (3.8)$$

$$X - Y : x \quad Z \setminus Y : f \rightarrow X - Z : f(x) \quad (<) \quad (3.9)$$

式(3.8)と式(3.9)は関数適用規則である。式(3.8)は前述でも述べたようなドミナント・モーションであったり、サブドミナントの機能を持つ和音からトニックの機能を持つ和音へ進行するサブドミナント・モーションにも用いられる。また、和音は自身の後ろの和音と係り受けの構造をなすので、式(3.9)はあまり用いられない。関数適用規則を用いて和音進行を解析した例が図3.4である。

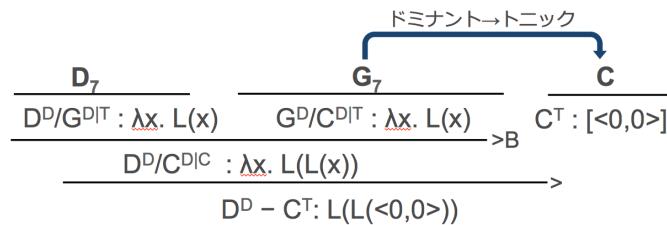


図3.4 関数適用規則の例

規則を用いた後にカテゴリは X - Y のような形のカテゴリとなっているが、これはカテゴリ X に由来する和音からカテゴリ Y に由来する和音へ拡張されたケーデンスを意味している。

関数合成規則

$$X/Y : f \quad Y/Z : g \quad \rightarrow \quad X/Z : \lambda x. f(g(x)) \quad (> B) \quad (3.10)$$

$$X\backslash Y : g \quad Z\backslash X : f \quad \rightarrow \quad Z\backslash Y : \lambda x. f(g(x)) \quad (< B) \quad (3.11)$$

式(3.10)と式(3.11)は関数合成規則である。主に用いられるのは式(3.10)であり、サブドミナントの機能をもつ和音が進行した先の和音がドミナントの機能をもつ場合にこの規則を用いる。また、セカンダリー・ドミナント・モーションの場合にもこの規則が用いられる。また関数適用規則と同様に式(3.11)は文献[11]において重要ではないとされている。関数合成規則を用いて和音進行を解析した例が図3.5である。

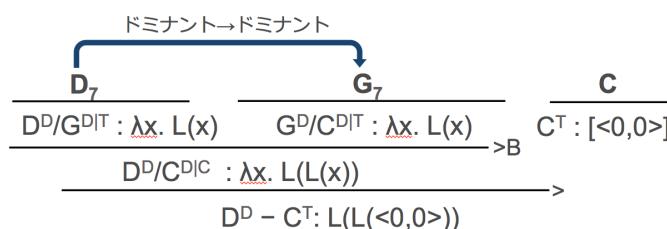


図3.5 関数合成規則の例

図3.5の D7 はセカンダリー・ドミナントである。自身を二次的なドミナントと見なして、根音が4度上である G7 を仮初めのトニックとみなしている。しかし、この G7 自身

もドミナントの機能を持っているため複合カテゴリで表される。このような D7 と G7 に対して関数合成規則を用いることができる。

等位接続規則

$$X^F/Y \quad Z^F/Y \quad \rightarrow \quad X^F/Y \quad F \in D, S \quad (3.12)$$

式(3.12)は等位接続規則である。これはある二つの和音が自身の後ろにあるトニックを期待している場合に用いられる。自然言語の場合と同じく遠くの依存関係が見ることができ。等位接続規則用いて和音進行を解析した例が図 3.6 である。

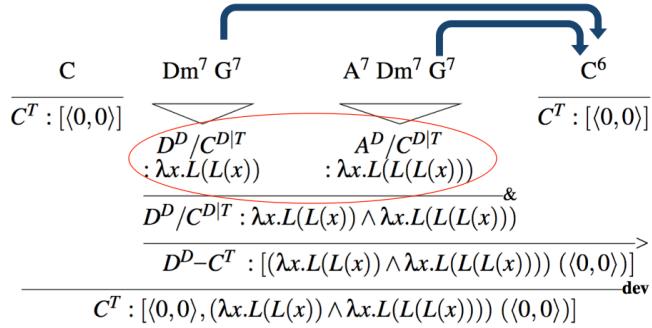


図 3.6 等位接続規則の例

関数合成規則を繰り返し用いると図 3.6 の赤丸のような複合カテゴリが並ぶ。二つの複合カテゴリはともにカテゴリ C がくることを期待している状態である。左の複合カテゴリからみると、ドミナントの直後にトニックが来なくてもドミナントモーションが成立している。このような進行も音楽ではよく用いられ、例えばジャズのスタンダードナンバーの一つである『IF I WERE A BELL』で用いられている [14]。

Development 規則

$$V - W \quad X - Y \quad \rightarrow \quad V - Y \quad (3.13)$$

式(3.13)は development 規則である。これは既に解析が終了したケーデンスを一つのケーデンスに統合することである。CCG の計算は規則を用いて最終的に一つのカテゴリに組み合わされなくてはならないため、二つの解析が終了した和音進行に対して解析の手続き上後処理のようなものを行うのがこの式の役割であると考えられる。

3.2.2 意味表示

文献 [11] で用いられる意味表示は $λx.leftonto(x), λx.rightonto(x), < a, b >$ である。以下にそれぞれについて説明する。

$\lambda x.\text{leftonto}(x)$

オイラー格子[12]上で「左に1マス移動する」という操作を意味する。ケーデンスでよく用いられるドミナント・モーションは、進行前後の二つの和音A,Bの根音に注目すると、和音Bの根音は和音Aの根音の完全4度上である。オイラー格子上では自身の隣の音は完全4度上の関係にある。従ってこの意味表示は二つの和音の根音が完全4度上の関係となる和音進行を表している。

$\lambda x.\text{rightonto}(x)$

オイラー格子上で「右に1マス移動する」という操作を意味する。ケーデンスでよく用いられるサブドミナント・モーションは、進行前後の二つの和音A,Bの根音に注目すると、和音Bの根音は和音Aの根音の完全5度上である。オイラー格子上では自身の隣の音は完全5度上の関係にある。従ってこの意味表示は二つの和音の根音が完全5度上の関係となる和音進行を表している。

$\langle a, b \rangle$

a, b は整数が入る。オイラー格子上で基準となる原点を決め、そこから右に a , 上に b の位置にある音を表している。

例として、CCGによる和音進行の解析とそれがオイラー格子にどのように対応しているかを図に3.7示す。

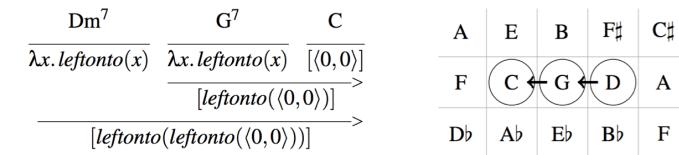


図 3.7 意味構造とオイラー格子への対応

このように文献[11]ではケーデンスをオイラー格子によって表現している。オイラー格子は和音進行を表現することができ、それぞれの和音の根音の変化をオイラー格子上で経路としている。

3.2.3 辞書項目

文献[11]では図3.8のような辞書項目を用いて入力されたコードネームにカテゴリを割り当てている。

Mnemonic label	Category schema	chord	Example syntactic type
Ton.	X(m) := $I^T : [(0,0)]$	CM7	C^T
Ton-III.	Xm := $\flat V I^T : [(0,2)]$	Em	C^T
Ton-bVI.	X := $III^T : [(0,1)]$	A♭M7	C^T
Dom.	X(m) ⁷ := $I^D / IV^{D T} : \lambda x.\text{leftonto}(x)$	G ⁷	$G^D / C^{D T}$
Dom-backdoor.	X(m) ⁷ := $VI^D / II^{D T} : \lambda x.\text{leftonto}(x)$	B♭ ⁷	$G^D / C^{D T}$
Dom-tritone.	X(m) ⁷ := $\flat V^D / VII^{D T} : \lambda x.\text{leftonto}(x)$	D♭ ⁷	$G^D / C^{D T}$
Dom-bartok.	X(m) ⁷ := $\flat III^D / \flat VI^{D T} : \lambda x.\text{leftonto}(x)$	E ⁷	$G^D / C^{D T}$
Subdom.	X(m) := $I^S / V^{S T} : \lambda x.\text{rightonto}(x)$	F	$F^S / C^{S T}$
Subdom-bIII.	X := $VI^S / III^{S T} : \lambda x.\text{rightonto}(x)$	A♭	$F^S / C^{S T}$
Dim-bVII.	Xo := $IV^D / \flat VII^{D T} : \lambda x.\text{leftonto}(x)$	Ddim ⁷	$G^D / C^{D T}$
Dim-V.	Xo := $II^D / V^{D T} : \lambda x.\text{leftonto}(x)$	Fdim ⁷	$G^D / C^{D T}$
Dim-III.	Xo := $VII^D / III^{D T} : \lambda x.\text{leftonto}(x)$	A♭dim ⁷	$G^D / C^{D T}$
Dim-bII.	Xo := $\flat VI^D / \flat II^{D T} : \lambda x.\text{leftonto}(x)$	Bdim ⁷	$G^D / C^{D T}$
Pass-I.	Xo := $I^T / I^T : \lambda x.x$	Cdim ⁷	C^T / C^T
	Xo := $I^D / I^D : \lambda x.x$	Gdim ⁷	G^D / G^D
Pass-VI.	Xo := $VI^T / VT : \lambda x.x$	Adim ⁷	C^T / C^T
	Xo := $VI^D / VI^D : \lambda x.x$	Edim ⁷	G^D / G^D
Pass-bV.	Xo := $\flat V^T / VT : \lambda x.x$	G♭dim ⁷	C^T / C^T
	Xo := $\flat V^D / V^D : \lambda x.x$	D♭dim ⁷	G^D / G^D
Pass-bIII.	Xo := $\flat III^T / \flat III^T : \lambda x.x$	E♭dim ⁷	C^T / C^T
	Xo := $\flat III^D / \flat III^D : \lambda x.x$	B♭dim ⁷	G^D / G^D
Aug-bII.	X ⁷ := $\flat VI^D / \flat II^{D T} : \lambda x.\text{leftonto}(x)$	Baug	$G^D / C^{D T}$
Aug-VI.	X ⁷ := $III^D / VI^{D T} : \lambda x.\text{leftonto}(x)$	Ebaug	$G^D / C^{D T}$
Colour-IVf.	X(m) := $V^T / VT : \lambda x.x$	F	C^T / C^T
Colour-IVb.	X(m) := $VT / V^T : \lambda x.x$	F	C^T / C^T
Colour-IIf.	X(m) := $\flat VII^T / \flat VII^T : \lambda x.x$	Dm	C^T / C^T
Colour-IIb.	X(m) := $\flat VII^T / VT : \lambda x.x$	Dm	C^T / C^T
Dom-IVm.	Xm := $II^D / VT : \lambda x.\text{leftonto}(x)$	Fm ⁶	G^D / C^T

図 3.8 文献 [11] で用いられた辞書項目
([11] より)

例として $D7 \rightarrow Db7 \rightarrow C$ というコードネームによる和音進行が入力されたときの辞書項目を用いた解析の手順を述べる。

1. まず、 $D7$ はその表層形から $X7$ の一種であるため、図 3.8 の上から 4 番目から 7 番目のカテゴリを割り当てることができる。割り当てが可能なカテゴリが複数ある場合は、辞書項目の上のほうから優先的に割り当てるので、はじめに 4 番目のカテゴリを割り当てる。(図 3.9)
2. 次に $Db7$ も同様に 4 番目から 7 番目のカテゴリを割り当てることができる。4 番目のカテゴリを割り当てる。(図 3.10)
3. 割り当てた二つのカテゴリに対して前述した 4 種類の組合せ規則が適用できるか試みる。この場合は規則が適用できないので何もせずに次の操作に移る。
4. 次の C に対しては 1 番目から 3 番目のカテゴリを割り当てることが可能である。1 番目のカテゴリを割り当てる。(図 3.11)

5. 割り当てている 3 つのカテゴリのなかで隣接しているカテゴリに対して組合せ規則が適用できるかそれぞれ試みる。この場合はいかなるカテゴリ間に対してもいかなる規則が適用できず、C が最後のコードネームのため C に割り当てられているカテゴリを消去する。
6. C に対して同様に 2 番目、3 番目のカテゴリを割り当てる。この場合も適用できる規則が存在しない。(図 3.12)
7. 一つ前の Db7 に対しての割り当てが適切ではないと疑われる所以、Db7 に割り当てられた 4 番目のカテゴリ消去し、次の候補である 5 番目のカテゴリを割り当てる。(図 3.13)
8. 3.~7. の操作を繰り返す。このように解析が失敗した時点で一つ前の操作に戻り、パズルを組み合わせるようにカテゴリの割り当てを試みていく。
9. Db7 に対して 6 番目のカテゴリを割り当てる、組合せ規則が適用できるか確かめる。関数合成規則が適用できるので、解析を進める。(図 3.14)
10. C に対して 1 番目のカテゴリを割り当てる。(図 3.15)
11. 存在している二つのカテゴリに対して組合せ規則が適用できるか確かめる。関数適用規則が適用できるので、解析を進める。(図 3.16)
12. カテゴリが一つに統合されたら、解析を終了する。入力されたコードネームに対して、それぞれカテゴリが割り当てられているが、図 3.8 よりそれに対応する label を割り当てる。この label が和音の機能を表しているので、出力はコードネームと label のセットとなる。解析の結果、D7 → Db7 → C という進行はそれぞれ Dom (ドミナント) → Dom-tritone (ドミナントの裏コード) → Ton (トニック) という解釈が与えられる。(図 3.17)

このように解析が失敗した時点で一つ前の操作に戻り、パズルを組み合わせるようにカテゴリの割り当てを試みていくので、縦型探索のようなアルゴリズムになっていると考えられる。

Mnemonic label	Category schema	Example chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$Xm := \flat VI^T : [(0,2)]$	Em	C^T
Ton-bVI.	$X := III^T : [(0,1)]$	A♭M7	C^T
Dom.	$X(m)^7 := I^D / IV^D T : \lambda x. leftonto(x)$	G ⁷	$G^D / C^D T$
Dom-backdoor.	$X(m)^7 := VI^D / II^D T : \lambda x. leftonto(x)$	B♭ ⁷	$G^D / C^D T$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^D T : \lambda x. leftonto(x)$	D♭ ⁷	$G^D / C^D T$
Dom-bartok.	$X(m)^7 := \flat III^D / \flat VI^D T : \lambda x. leftonto(x)$	E ⁷	$G^D / C^D T$
Subdom.	$X(m) := I^S / V^S T : \lambda x. rightonto(x)$	F	$F^S / C^S T$
Subdom-bIII.	$X := VI^S / III^S T : \lambda x. rightonto(x)$	A♭	$F^S / C^S T$
Dim-bVII.	$Xo := IV^D / \flat VII^D T : \lambda x. leftonto(x)$	Ddim ⁷	$G^D / C^D T$
Dim-V.	$Xo := II^D / V^D T : \lambda x. leftonto(x)$	Fdim ⁷	$G^D / C^D T$
Dim-III.	$Xo := VII^D / III^D T : \lambda x. leftonto(x)$	A♭dim ⁷	$G^D / C^D T$
Dim-bII.	$Xo := \flat VI^D / \flat II^D T : \lambda x. leftonto(x)$	Bdim ⁷	$G^D / C^D T$

D₇ **D b₇** **C**

D^{D/G^{D|T}} : λx. L(x)

図 3.9 文献 [11] における辞書項目からのカテゴリの割り当て 1

Mnemonic label	Category schema	Example chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$Xm := \flat VI^T : [(0,2)]$	Em	C^T
Ton-bVI.	$X := III^T : [(0,1)]$	A♭M7	C^T
Dom.	$X(m)^7 := I^D / IV^D T : \lambda x. leftonto(x)$	G ⁷	$G^D / C^D T$
Dom-backdoor.	$X(m)^7 := VI^D / II^D T : \lambda x. leftonto(x)$	B♭ ⁷	$G^D / C^D T$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^D T : \lambda x. leftonto(x)$	D♭ ⁷	$G^D / C^D T$
Dom-bartok.	$X(m)^7 := \flat III^D / \flat VI^D T : \lambda x. leftonto(x)$	E ⁷	$G^D / C^D T$
Subdom.	$X(m) := I^S / V^S T : \lambda x. rightonto(x)$	F	$F^S / C^S T$
Subdom-bIII.	$X := VI^S / III^S T : \lambda x. rightonto(x)$	A♭	$F^S / C^S T$
Dim-bVII.	$Xo := IV^D / \flat VII^D T : \lambda x. leftonto(x)$	Ddim ⁷	$G^D / C^D T$
Dim-V.	$Xo := II^D / V^D T : \lambda x. leftonto(x)$	Fdim ⁷	$G^D / C^D T$
Dim-III.	$Xo := VII^D / III^D T : \lambda x. leftonto(x)$	A♭dim ⁷	$G^D / C^D T$
Dim-bII.	$Xo := \flat VI^D / \flat II^D T : \lambda x. leftonto(x)$	Bdim ⁷	$G^D / C^D T$

D₇ **D b₇** **C**

D^{D/G^{D|T}} : λx. L(x)

図 3.10 文献 [11] における辞書項目からのカテゴリの割り当て 2

Mnemonic label	Category schema	chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$\flat V I^T : [(0,2)]$	Em	C^T
Ton-bVI.	$III^T : [(0,1)]$	A♭M7	C^T
Dom.	$X(m)^7 := I^D / IV^{D T} : \lambda x. leftonto(x)$	G ⁷	$G^D / C^{D T}$
Dom-backdoor.	$X(m)^7 := VI^D / II^{D T} : \lambda x. leftonto(x)$	B♭ ⁷	$G^D / C^{D T}$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^{D T} : \lambda x. leftonto(x)$	D♭ ⁷	$G^D / C^{D T}$
Dom-bartok.	$X(m)^7 := \flat III^D / \flat VI^{D T} : \lambda x. leftonto(x)$	E ⁷	$G^D / C^{D T}$
Subdom.	$X(m) := I^S / V^{S T} : \lambda x. rightonto(x)$	F	$F^S / C^{S T}$
Subdom-bIII.	$X := VI^S / III^{S T} : \lambda x. rightonto(x)$	A♭	$F^S / C^{S T}$
Dim-bVII.	$Xo := IV^D / \flat VII^{D T} : \lambda x. leftonto(x)$	Ddim ⁷	$G^D / C^{D T}$
Dim-V.	$Xo := II^D / V^{D T} : \lambda x. leftonto(x)$	Fdim ⁷	$G^D / C^{D T}$
Dim-III.	$Xo := VII^D / III^{D T} : \lambda x. leftonto(x)$	A♭dim ⁷	$G^D / C^{D T}$
Dim-bII.	$Xo := \flat VI^D / \flat II^{D T} : \lambda x. leftonto(x)$	Bdim ⁷	$G^D / C^{D T}$

$\overline{D_7} \quad \overline{D \flat_7} \quad \overline{C}$

$D^D / G^{D|T} : \lambda x. L(x) \quad D \flat^D / G \flat^{D|T} : \lambda x. L(x) \quad C^T : [0,0]$

図 3.11 文献 [11] における辞書項目からのカテゴリの割り当て 3

Mnemonic label	Category schema	chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$\flat V I^T : [(0,2)]$	Em	C^T
Ton-bVI.	$III^T : [(0,1)]$	A♭M7	C^T
Dom.	$X(m)^7 := I^D / \flat V^{D T} : \lambda x. leftonto(x)$	G ⁷	$G^D / C^{D T}$
Dom-backdoor.	$X(m)^7 := VI^D / II^{D T} : \lambda x. leftonto(x)$	B♭ ⁷	$G^D / C^{D T}$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^{D T} : \lambda x. leftonto(x)$	D♭ ⁷	$G^D / C^{D T}$
Dom-bartok.	$X(m)^7 := \flat III^D / \flat VI^{D T} : \lambda x. leftonto(x)$	E ⁷	$G^D / C^{D T}$
Subdom.	$X(m) := I^S / V^{S T} : \lambda x. rightonto(x)$	F	$F^S / C^{S T}$
Subdom-bIII.	$X := VI^S / III^{S T} : \lambda x. rightonto(x)$	A♭	$F^S / C^{S T}$
Dim-bVII.	$Xo := IV^D / \flat VII^{D T} : \lambda x. leftonto(x)$	Ddim ⁷	$G^D / C^{D T}$
Dim-V.	$Xo := II^D / V^{D T} : \lambda x. leftonto(x)$	Fdim ⁷	$G^D / C^{D T}$
Dim-III.	$Xo := VII^D / III^{D T} : \lambda x. leftonto(x)$	A♭dim ⁷	$G^D / C^{D T}$
Dim-bII.	$Xo := \flat VI^D / \flat II^{D T} : \lambda x. leftonto(x)$	Bdim ⁷	$G^D / C^{D T}$

$\overline{D_7} \quad \overline{D \flat_7} \quad \overline{C}$

$D^D / G^{D|T} : \lambda x. L(x) \quad D \flat^D / G \flat^{D|T} : \lambda x. L(x) \quad E^T : [0,1]$

図 3.12 文献 [11] における辞書項目からのカテゴリの割り当て 4

Mnemonic label	Category schema	chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$Xm := \flat VI^T : [(0,2)]$	Em	C^T
Ton-bVI.	$X := III^T : [(0,1)]$	A♭M7	C^T
Dom.	$X(m)^7 := I^D / IV^{D T} : \lambda x.\text{leftonto}(x)$	G ⁷	$G^D / C^{D T}$
Dom-backdoor.	$X(m)^7 := VI^D / II^{D T} : \lambda x.\text{leftonto}(x)$	B♭ ⁷	$G^D / C^{D T}$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^{D T} : \lambda x.\text{leftonto}(x)$	D♭ ⁷	$G^D / C^{D T}$
Dom-bartok.	$X(m)^7 := III^D / \flat VI^{D T} : \lambda x.\text{leftonto}(x)$	E ⁷	$G^D / C^{D T}$
Subdom.	$X(m) := I^S / V^{S T} : \lambda x.\text{rightonto}(x)$	F	$F^S / C^{S T}$
Subdom-bIII.	$X := VI^S / IV^{S T} : \lambda x.\text{rightonto}(x)$	A♭	$F^S / C^{S T}$
Dim-bVII.	$Xo := IV^D / \flat VII^{D T} : \lambda x.\text{leftonto}(x)$	Ddim ⁷	$G^D / C^{D T}$
Dim-V.	$Xo := II^D / V^{D T} : \lambda x.\text{leftonto}(x)$	Fdim ⁷	$G^D / C^{D T}$
Dim-III.	$Xo := VII^D / III^{D T} : \lambda x.\text{leftonto}(x)$	A♭dim ⁷	$G^D / C^{D T}$
Dim-bII.	$Xo := \flat VI^D / II^{D T} : \lambda x.\text{leftonto}(x)$	Bdim ⁷	$G^D / C^{D T}$

D_7
 $D \flat 7$
 C

$D^D / G^{D|T} : \lambda x. L(x)$
 $B \flat D / E \flat D|T : \lambda x. L(x)$

図 3.13 文献 [11] における辞書項目からのカテゴリの割り当て 5

Mnemonic label	Category schema	chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$Xm := \flat VI^T : [(0,2)]$	Em	C^T
Ton-bVI.	$X := III^T : [(0,1)]$	A♭M7	C^T
Dom.	$X(m)^7 := I^D / IV^{D T} : \lambda x.\text{leftonto}(x)$	G ⁷	$G^D / C^{D T}$
Dom-backdoor.	$X(m)^7 := VI^D / II^{D T} : \lambda x.\text{leftonto}(x)$	B♭ ⁷	$G^D / C^{D T}$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^{D T} : \lambda x.\text{leftonto}(x)$	D♭ ⁷	$G^D / C^{D T}$
Dom-bartok.	$X(m)^7 := III^D / \flat VI^{D T} : \lambda x.\text{leftonto}(x)$	E ⁷	$G^D / C^{D T}$
Subdom.	$X(m) := I^S / V^{S T} : \lambda x.\text{rightonto}(x)$	F	$F^S / C^{S T}$
Subdom-bIII.	$X := VI^S / IV^{S T} : \lambda x.\text{rightonto}(x)$	A♭	$F^S / C^{S T}$
Dim-bVII.	$Xo := IV^D / \flat VII^{D T} : \lambda x.\text{leftonto}(x)$	Ddim ⁷	$G^D / C^{D T}$
Dim-V.	$Xo := II^D / V^{D T} : \lambda x.\text{leftonto}(x)$	Fdim ⁷	$G^D / C^{D T}$
Dim-III.	$Xo := VII^D / III^{D T} : \lambda x.\text{leftonto}(x)$	A♭dim ⁷	$G^D / C^{D T}$
Dim-bII.	$Xo := \flat VI^D / II^{D T} : \lambda x.\text{leftonto}(x)$	Bdim ⁷	$G^D / C^{D T}$

D_7
 $D \flat 7$
 C

$D^D / G^{D|T} : \lambda x. L(x)$
 $G^D / C^{D|T} : \lambda x. L(x)$

$D^D / C^{D|C} : \lambda x. L(L(x))$
>B

図 3.14 文献 [11] における辞書項目からのカテゴリの割り当て 6

Mnemonic label	Category schema	chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$Xm := \flat V I^T : [(0,2)]$	Em	C^T
Ton-bVI.	$X := III^T : [(0,1)]$	A♭M7	C^T
Dom.	$X(m)^7 := I^D / IV^D T : \lambda x. leftonto(x)$	G ⁷	$G^D / C^{D T}$
Dom-backdoor.	$X(m)^7 := VI^D / II^D T : \lambda x. leftonto(x)$	B♭ ⁷	$G^D / C^{D T}$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^D T : \lambda x. leftonto(x)$	D♭ ⁷	$G^D / C^{D T}$
Dom-bartok.	$X(m)^7 := \flat III^D / \flat VI^D T : \lambda x. leftonto(x)$	E ⁷	$G^D / C^{D T}$
Subdom.	$X(m) := I^S / V^S T : \lambda x. rightonto(x)$	F	$F^S / C^S T$
Subdom-bIII.	$X := VI^S / III^S T : \lambda x. rightonto(x)$	A♭	$F^S / C^S T$
Dim-bVII.	$Xo := IV^D / \flat VII^D T : \lambda x. leftonto(x)$	Ddim ⁷	$G^D / C^{D T}$
Dim-V.	$Xo := II^D / V^D T : \lambda x. leftonto(x)$	Fdim ⁷	$G^D / C^{D T}$
Dim-III.	$Xo := VII^D / III^D T : \lambda x. leftonto(x)$	A♭dim ⁷	$G^D / C^{D T}$
Dim-bII.	$Xo := \flat VI^D / \flat II^D T : \lambda x. leftonto(x)$	Bdim ⁷	$G^D / C^{D T}$

\mathbf{D}_7
 $\mathbf{D} \flat_7$
 \mathbf{C}

$D^D / G^D | T : \lambda x. L(x)$
 $G^D / C^D | T : \lambda x. L(x)$
 $C^T : [<0,0>]$

$>B$

$D^D / C^D | C : \lambda x. L(L(x))$

図 3.15 文献 [11] における辞書項目からのカテゴリの割り当て 7

Mnemonic label	Category schema	chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$Xm := \flat V I^T : [(0,2)]$	Em	C^T
Ton-bVI.	$X := III^T : [(0,1)]$	A♭M7	C^T
Dom.	$X(m)^7 := I^D / IV^D T : \lambda x. leftonto(x)$	G ⁷	$G^D / C^{D T}$
Dom-backdoor.	$X(m)^7 := VI^D / II^D T : \lambda x. leftonto(x)$	B♭ ⁷	$G^D / C^{D T}$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^D T : \lambda x. leftonto(x)$	D♭ ⁷	$G^D / C^{D T}$
Dom-bartok.	$X(m)^7 := \flat III^D / \flat VI^D T : \lambda x. leftonto(x)$	E ⁷	$G^D / C^{D T}$
Subdom.	$X(m) := I^S / V^S T : \lambda x. rightonto(x)$	F	$F^S / C^S T$
Subdom-bIII.	$X := VI^S / III^S T : \lambda x. rightonto(x)$	A♭	$F^S / C^S T$
Dim-bVII.	$Xo := IV^D / \flat VII^D T : \lambda x. leftonto(x)$	Ddim ⁷	$G^D / C^{D T}$
Dim-V.	$Xo := II^D / V^D T : \lambda x. leftonto(x)$	Fdim ⁷	$G^D / C^{D T}$
Dim-III.	$Xo := VII^D / III^D T : \lambda x. leftonto(x)$	A♭dim ⁷	$G^D / C^{D T}$
Dim-bII.	$Xo := \flat VI^D / \flat II^D T : \lambda x. leftonto(x)$	Bdim ⁷	$G^D / C^{D T}$

\mathbf{D}_7
 $\mathbf{D} \flat_7$
 \mathbf{C}

$D^D / G^D | T : \lambda x. L(x)$
 $G^D / C^D | T : \lambda x. L(x)$
 $C^T : [<0,0>]$

$>B$

$D^D / C^D | C : \lambda x. L(L(x))$

$>$

$D^D - C^T : L(L(<0,0>))$

図 3.16 文献 [11] における辞書項目からのカテゴリの割り当て 8

Mnemonic label	Category schema	chord	Example syntactic type
Ton.	$X(m) := I^T : [(0,0)]$	CM7	C^T
Ton-III.	$Xm := \flat VI^T : [(0,2)]$	Em	C^T
Ton-bVI.	$X := III^T : [(0,1)]$	AbM7	C^T
Dom.	$X(m)^7 := I^D / IV^D T : \lambda x. leftonto(x)$	G ⁷	$G^D / CD T$
Dom-backdoor.	$X(m)^7 := VI^D / II^D T : \lambda x. leftonto(x)$	B \flat ⁷	$G^D / CD T$
Dom-tritone.	$X(m)^7 := \flat V^D / VII^D T : \lambda x. leftonto(x)$	D \flat ⁷	$G^D / CD T$
Dom-bartok.	$X(m)^7 := \flat III^D / \flat VI^D T : \lambda x. leftonto(x)$	E ⁷	$G^D / CD T$
Subdom.	$X(m) := I^S / V^S T : \lambda x. rightonto(x)$	F	$F^S / CS T$
Subdom-bIII.	$X := VI^S / III^S T : \lambda x. rightonto(x)$	Ab	$F^S / CS T$
Dim-bVII.	$Xo := IV^D / \flat VII^D T : \lambda x. leftonto(x)$	Ddim ⁷	$G^D / CD T$
Dim-V.	$Xo := II^D / V^D T : \lambda x. leftonto(x)$	Fdim ⁷	$G^D / CD T$
Dim-III.	$Xo := VII^D / III^D T : \lambda x. leftonto(x)$	Abdim ⁷	$G^D / CD T$
Dim-bII.	$Xo := \flat VI^D / \flat II^D T : \lambda x. leftonto(x)$	Bdim ⁷	$G^D / CD T$

Dom. Dom-tritone. Ton.

D_7 $D \flat_7$ C

$D^D / G^D|T : \lambda x. L(x)$ $G^D / C^D|T : \lambda x. L(x)$ $C^T : [<0,0>]$

$D^D / C^D|C : \lambda x. L(L(x))$ >B >

$D^D - C^T : L(L(<0,0>))$

図 3.17 文献 [11] における辞書項目からのカテゴリの割り当て 9

第4章 組み合わせ範疇文法を用いたジャズ和声解析手法の改良

本研究ではジャズ和声を含む和声解析の手法として3.2で述べた[11]の手法を主に取り入れる。しかしながら、次節で述べるように、[11]の手法にはいくつか問題点があり、本研究ではこれらに対して解決させたものを手法として取り入れる。

4.1 文献[11]における諸問題

文献[11]の手法による和声解析にはいくつか問題点がある。

4.1.1 サブドミナントからドミナントへの進行

一つ目の問題点は、サブドミナントからドミナントへの進行を想定していないという点である。

文献[15]によると、サブドミナントの機能をもつ和音からドミナントの機能を持つ和音への進行は正しいものとしている。しかしながら文献[11]においては、オイラー格子の性質より、解析できる和音進行はトニックの機能をもつ和音から任意の和音、根音が4度上の関係にある和音どうし、根音が5度上の関係にある和音どうしに限られる。つまり、トニックから任意の和音、サブドミナントからトニックへの進行、ドミナントからトニックへの進行の3種類のみを想定しているということである。ここで、文献[11]の解析手法を用いて現代の商業音楽で頻繁に用いられる $Dm7 \rightarrow G7 \rightarrow C$ となるような進行を解析した例を図4.1に示す。

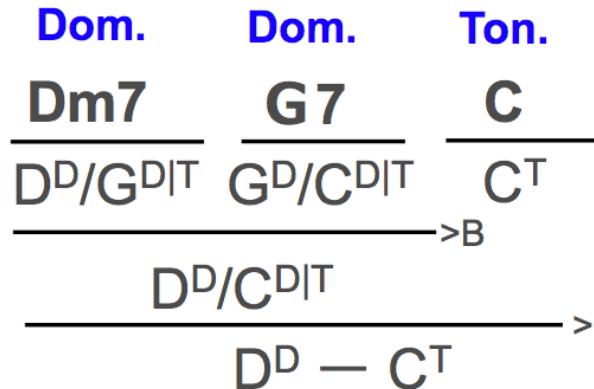


図 4.1 $Dm7 \rightarrow G7 \rightarrow C$ に対する解析

文献 [15] によると、この和音進行はツーファイブ・ドミナントとよばれ、この $Dm7$ は F の代理和音である。したがって $Dm7$ はサブドミナントと解釈されるのが妥当であるが、用いている辞書項目により $Dm7$ はドミナントであると解釈される。

4.1.2 三和音に対するカテゴリの不適切な付与

二つ目の問題点は、トライアドコードに対して容易にトニックであると解釈される点である。例として、文献 [11] による $C \rightarrow Am \rightarrow Dm \rightarrow G \rightarrow C$ となるような進行の和声解析を図 4.2 に示す。

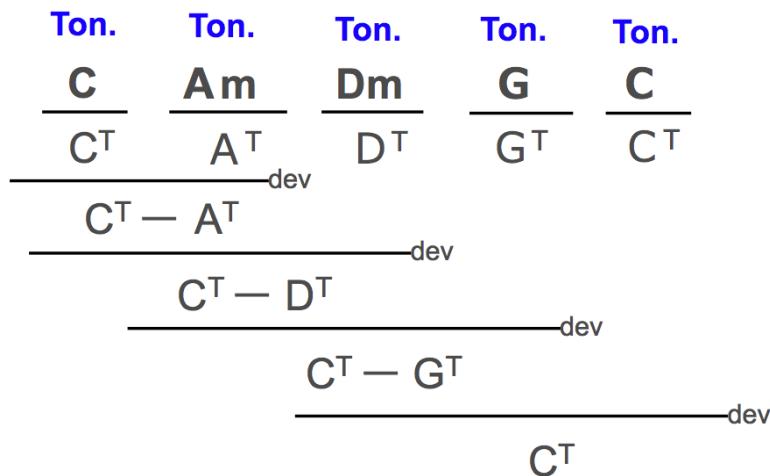


図 4.2 $C \rightarrow Am \rightarrow Dm \rightarrow G \rightarrow C$ に対する解析

文献[15]によると $C \rightarrow Am \rightarrow Dm \rightarrow G \rightarrow C$ はハ長調におけるダイアトニック・コードによる進行である。したがって左からそれぞれトニック、トニック、サブドミナント、ドミナント、トニックと解釈されるのが自然であるとしている。しかしながら文献[11]による和声解析では、すべてのコードネームに対して辞書項目の1番目のカテゴリが割り当てられ、すべて式(3.13)によりトニック単体のケーデンスの集合であると解釈される。

この問題を文献[11]では隠れマルコフ過程などを用いて機械学習的に解決しているが、本研究ではルールベースで和声解析を行っていきたいと考えているため、このような辞書項目の作り方は望ましくないと判断した。

4.1.3 ピボットコードを含む転調

三つ目の問題点は複数の解釈がなされる和音進行に対してただ一つの解釈のみを出力としている点である。ピボットコードがその例である。 $C \rightarrow D7 \rightarrow G \rightarrow A7 \rightarrow D$ と進行するようなコードネームの和音進行が入力として与えられたとき、妥当な解釈は表4.1のように二通り存在する。

表4.1 ピボットコードを含む和音進行の解釈

コードネーム	C	D7	G	A7	D
解釈 1	IV/G	V/G	I/G	V/D	I/D
解釈 2	IV/G	V/G	IV/D	V/D	I/D

解釈1ではGとA7との間で転調が生じ、Gはト長調のトニックであるという解釈になる。解釈2ではD7とGとの間で転調が生じ、Gはニ長調のサブドミナントであるという解釈になる。どちらの解釈も正しい。ここでGがピボットコードであり、Gは二つの意味をもつことになる。このように聴き手に明示的な転調を与えずにスムーズな転調を与える作曲手法が様々なジャンルの音楽で用いられている。しかし、文献[11]のような解析ではそれぞれのコードネームに対して一対一の出力しか与えないでの、ピボットコードを認識することができない。

4.2 本研究における文献[11]の諸問題の解決

4.1で示した問題点に対処するため、CCGを用いた和声解析の手法を次のように提案する。

4.2.1 ディグリーネーム表記への変更

文献[11]の手法では、入力された和音進行がどのような調に属しているのかといった情報が前もって与えられていないことが原因で、トライアドコードが無条件にトニックであると解釈されるようなカテゴリが与えられると考えられる。そのため、4.1の問題点2の対策として、CCGで扱う和音をコードネームではなく、調の情報を含んだディグリーネームとする。

以降のディグリーネームの表記に関しては以下の通りとする。

- コードはギリシャ数字で表す。ただしすべて大文字の表記とする。
- 短和音の場合はギリシャ数字の左側に『-』をつけて表す。

これにより、例えばハ長調上におけるDmはディグリーネームでⅡ-と表され、少なくともこれがトニックであるという解釈を避けることができる。

4.2.2 ジャズ特有の和音および機能への対応

4.1の問題点として挙げていないが、より多くのジャズ和声を含んだ楽曲に対して和声解析を可能にするためには、文献[11]では扱われなかった和音の機能にも対応する必要がある。

ジャズ特有の和音には様々なものがあるが、本研究においてはサブドミナントマイナーやセカンドドミナント、リレイテッド・セカンドマイナー・セブンス、裏コードやその他のノンダイアトニックコードを扱う。以下ではこれらの和音について述べる。

サブドミナントマイナー

サブドミナントマイナーとは自然的短音階上のダイアトニックコードであり、構成音に短六度の音を含む。長音階は短六度の音を持たないため、長調ではノンダイアトニックコードであるが、モーダル・インターチェンジとよばれる技法により同主短調からの借用和音として長調でも用いることができるため、長調の主要和音の一つと考えられる。ジャズに限らず現代の商業音楽においてもサブドミナントマイナーは頻繁に用いられ、楽曲をより豊かにする効果がある。

セカンダリードミナント, リレイテッド・セカンドマイナー・セブンス

セカンダリー・ドミナントとは, 進行先のダイアトニックコードを仮のトニックと見立ててそれに向かってドミナント・モーションするような和音である。そして, そのセカンダリー・ドミナントに四度の強進行をするような和音をリレイテッド・セカンドマイナー・セブンスという。ジャズでは既存の曲の和音進行をそのまま演奏することは少なく, 和音進行を各々がリハーモナイズと呼ばれる編曲を施して演奏する。その際にこれらの和音を用いてリハーモナイズすることから重要であるとし, 本研究に取り入れる。

4.2.3 辞書項目の改良

4.1 の一つ目の問題点の対応として, 新たにサブドミナントからドミナントへの進行を認識できるようにする必要があり, すなわちここで和音進行のルールを見直す。

機能をもった和音間における進行はある一定の法則が存在し, これをケーデンスの法則という。ケーデンスの法則は時代によって形を変えているが, 本研究におけるケーデンスの法則は [15] に基づき以下の通りとする。

- トニックはドミナント, サブドミナント, サブドミナントマイナーのいずれへも進行することができる。
- ドミナントは必ずトニックへ解決しサブドミナント, サブドミナントマイナーへ進行することはない。
- サブドミナントはトニック, ドミナント, サブドミナントマイナーのいずれへも進行することができる。
- サブドミナントマイナーはトニック, ドミナントへ進行することができるが, サブドミナントへ進行することはない。

4.2.1 と 4.2.2 も踏まえて, 前述のケーデンスの法則を満たすように図 3.8 の辞書項目を表 4.2 と表 4.3 のように改訂する。

表 4.2 と表 4.3 に示すカテゴリは T (トニック), S (サブドミナント), D (ドミナント), SM (サブドミナントマイナー), R (リレイテッド・セカンドマイナーセブンス), SD (セカンダリー・ドミナント), PD (パッシングディミニッシュ) である。カテゴリの右肩に添字として表しているのは統語素性である。統語素性を持たないカテゴリに関しては, 任意の統語素性をもつカテゴリとしてみなすものとする。リレイテッド・セカンドマイナーセブンスはそれぞれ直後に必ず特定のセカンダリードミナントないしドミナントへ進行する。これはリレイテッド・セカンドマイナーセブンスが, 同じセカンダリー・ドミナントであっても不適切なセカンダリードミナントへ進行するのを防ぐ役割をもつ。

表 4.2 提案手法による辞書項目（トニック・サブドミナント・ドミナント・サブドミナントマイナー）

	Tonic	Sub Dominant	Dominant	Sub Dominant Minor
Major Key	$I := T^I$ $I_6 := T^{I_6}$ $I_{M7} := T^{I_{M7}}$ $III_- := T^{III_-}$ $III_{-7} := T^{III_{-7}}$ $VI_- := T^{VI_-}$ $VI_{-7} := T^{VI_{-7}}$ $I_7 := T^{I_7}$ $\#IV_{-7}^{(b5)} := T^{\#IV_{-7}^{(b5)}}$	$IV := S^{IV}/(T D SM)$ $IV_6 := S^{IV_6}/(T D SM)$ $IV_{M7} := S^{IV_{M7}}/(T D SM)$ $II_- := S^{II_-}/(T D SM)$ $II_{-7} := S^{II_{-7}}/(T D SM)$ $IV_7 := S^{IV_7}/(T D SM)$ $VII_7 := S^{VII_7}/(T D SM)$ $\#IV_{-7}^{(b5)} := S^{\#IV_{-7}^{(b5)}}/(T D SM)$	$V := D^V/T$ $V_7 := D^{V_7}/T$ $VII_{-7}^{(b5)} := D^{VII_{-7}^{(b5)}}/T$ $VII_{-7}^{(b5)} := D^{VII_{-7}^{(b5)}}/T$ $\flat II_7 := D^{\flat II_7}/T$	$\flat VII_7 := SM^{\flat VII_7}/(T D)$ $\flat II_{M7} := SM^{\flat II_{M7}}/(T D)$ $\flat VI_7 := SM^{\flat VI_7}/(T D)$
Minor Key	$I_- := T^{I_-}$ $I_{-6} := T^{I_{-6}}$ $I_{-7} := T^{I_{-7}}$ $I_{-M7} := T^{I_{-M7}}$ $\flat III := T^{\flat III}$ $\flat III_{M7} := T^{\flat III_{M7}}$ $\flat III^+ := T^{\flat III^+}$ $\flat III_{M7}^+ := T^{\flat III_{M7}^+}$ $VI_-^{(b5)} := T^{VI_-^{(b5)}}$ $VI_{-7}^{(b5)} := T^{VI_{-7}^{(b5)}}$	$IV := S^{IV}/(T D SM)$ $IV_6 := S^{IV_6}/(T D SM)$ $IV_7 := S^{IV_7}/(T D SM)$ $II_- := S^{II_-}/(T D SM)$ $II_{-7} := S^{II_{-7}}/(T D SM)$ $VII_7 := S^{VII_7}/(T D SM)$	$V := D^V/T$ $V_7 := D^{V_7}/T$ $V_{-7} := D^{V_{-7}}/T$ $VII_0 := D^{VII_0}/T$ $VII_{-7}^{(b5)} := D^{VII_{-7}^{(b5)}}/T$ $\flat II_7 := D^{\flat II_7}/T$	$IV_- := SM^{IV_-}/(T D)$ $IV_{-6} := SM^{IV_{-6}}/(T D)$ $IV_{-7} := SM^{IV_{-7}}/(T D)$ $IV_{-M7} := SM^{IV_{-M7}}/(T D)$ $\flat II_{-7}^{(b5)} := SM^{\flat II_{-7}^{(b5)}}/(T D)$ $\flat II_{-7}^{(b5)} := SM^{\flat II_{-7}^{(b5)}}/(T D)$ $\flat VI := SM^{\flat VI}/(T D)$ $\flat VI_{M7} := SM^{\flat VI_{M7}}/(T D)$ $\flat VII := SM^{\flat VII}/(T D)$ $\flat VII_7 := SM^{\flat VII_7}/(T D)$ $\flat II_{M7} := SM^{\flat II_{M7}}/(T D)$ $\flat VI_7 := SM^{\flat VI_7}/(T D)$

表4.3 提案手法による辞書項目（リレイテッドセカンドマイナー・セカンダリードミナント・パッシングディミニッシュ）

	Related II ₋₇	Secondary Dominant	Passing Diminish	
Major Key 37	$\text{II}_{-7} := R^{\text{II}_{-7}} / D^{[V V_7 \flat\text{II}_7]}$ $\text{III}_{-7} := R^{\text{II}_{-7}} / SD^{[VI_7 \flat\text{III}_7]}$ $\text{III}_{-7}^{(\flat 5)} := R^{\text{II}_{-7}} / SD^{[VI_7 \flat\text{III}_7]}$ $\#IV_{-7} := R^{\#IV_{-7}} / SD^{[VII_7 IV_7]}$ $\#IV_{-7}^{(\flat 5)} := R^{\#IV_{-7}} / SD^{[VII_7 IV_7]}$ $V_{-7} := R^{V_{-7}} / SD^{[I_7 \flat V_7]}$ $VI_{-7} := R^{VI_{-7}} / SD^{[I_7 \flat VI_7]}$ $VII_{-7} := R^{VII_{-7}} / SD^{[III_7 \flat VII_7]}$ $VII_{-7}^{(\flat 5)} := R^{VII_{-7}} / SD^{[III_7 \flat VII_7]}$	$\flat\text{VI}_{-7} := R^{\flat VI_{-7}} / D^{[V V_7 \flat\text{II}_7]}$ $\flat\text{VII}_{-7} := R^{\flat VI_{-7}} / SD^{[VI_7 \flat\text{III}_7]}$ $\flat\text{VII}_{-7}^{(\flat 5)} := R^{\flat VI_{-7}} / SD^{[VI_7 \flat\text{III}_7]}$ $I_{-7} := R^{I_{-7}} / SD^{[VII_7 IV_7]}$ $I_{-7}^{(\flat 5)} := R^{I_{-7}} / SD^{[VII_7 IV_7]}$ $\flat\text{II}_{-7} := R^{\flat II_{-7}} / SD^{[I_7 \flat V_7]}$ $\flat\text{III}_{-7} := R^{\flat III_{-7}} / SD^{[I_7 \flat VI_7]}$ $IV_{-7} := R^{IV_{-7}} / SD^{[III_7 \flat VII_7]}$ $IV_{-7}^{(\flat 5)} := R^{IV_{-7}} / SD^{[III_7 \flat VII_7]}$	$I_7 := SD^{I_7} / S^{[IV IV_6 IV_{M7}]}$ $\text{II}_7 := SD^{II_7} / D^{[V V_7]}$ $\text{III}_7 := SD^{III_7} / T^{[VI_{-} VI_{-7}]}$ $\text{VI}_7 := SD^{VI_7} / S^{[II_{-} II_{-7}]}$ $VII_7 := SD^{VII_7} / T^{[III_{-} III_{-7}]}$ $\flat\text{V}_7 := SD^{\flat V_7} / S^{[IV IV_6 IV_{M7}]}$ $\flat\text{VI}_7 := SD^{\flat VI_7} / D^{[V V_7]}$ $\flat\text{VII}_7 := SD^{\flat VII_7} / T^{[VI_{-} VI_{-7}]}$ $\flat\text{III}_7 := SD^{\flat III_7} / S^{[II_{-} II_{-7}]}$ $IV_7 := SD^{IV_7} / T^{[III_{-} III_{-7}]}$	$\#I_{\circ} := PD^{\#I_{\circ}}$ $\#II_{\circ} := PD^{\#II_{\circ}}$ $\#IV_{\circ} := PD^{\#IV_{\circ}}$ $\#V_{\circ} := PD^{\#V_{\circ}}$ $\#VI_{\circ} := PD^{\#VI_{\circ}}$
Minor Key	$\text{II}_{-7} := R^{\text{II}_{-7}} / D^{[V V_7 \flat\text{II}_7]}$ $\text{II}_{-7}^{(\flat 5)} := R^{\text{II}_{-7}} / D^{[V V_7 \flat\text{II}_7]}$ $IV_{-7} := R^{IV_{-7}} / SD^{[\flat VII_7 \flat III_7]}$ $V_{-7} := R^{V_{-7}} / SD^{[I_7 \flat V_7]}$ $V_{-7}^{(\flat 5)} := R^{V_{-7}} / SD^{[I_7 \flat V_7]}$ $VI_{-7} := R^{VI_{-7}} / SD^{[I_7 \flat VI_7]}$ $VI_{-7}^{(\flat 5)} := R^{VI_{-7}} / SD^{[I_7 \flat VI_7]}$ $\flat\text{VII}_{-7} := R^{\flat VII_{-7}} / SD^{[\flat III_7 VI_7]}$ $I_{-7} := R^{I_{-7}} / SD^{[IV_7 VII_7]}$	$\flat\text{VI}_{-7} := R^{\flat VI_{-7}} / D^{[V V_7 \flat\text{II}_7]}$ $\flat\text{VI}_{-7}^{(\flat 5)} := R^{\flat VI_{-7}} / D^{[V V_7 \flat\text{II}_7]}$ $VII_{-7} := R^{VII_{-7}} / SD^{[\flat VII_7 \flat III_7]}$ $\flat\text{II}_{-7} := R^{\flat II_{-7}} / SD^{[I_7 \flat V_7]}$ $\flat\text{II}_{-7}^{(\flat 5)} := R^{\flat II_{-7}} / SD^{[I_7 \flat V_7]}$ $\flat\text{III}_{-7} := R^{\flat III_{-7}} / SD^{[I_7 \flat VI_7]}$ $\flat\text{III}_{-7}^{(\flat 5)} := R^{\flat III_{-7}} / SD^{[I_7 \flat VI_7]}$ $\flat\text{III}_{-7} := R^{\flat III_{-7}} / SD^{[I_7 \flat VI_7]}$ $\flat\text{V}_7 := R^{\flat V_7} / SD^{[IV_6 IV_7]}$	$I_7 := SD^{I_7} / (S^{[IV IV_6 IV_7]} SM^{[IV_{-} IV_{-6} IV_{-7} IV_{-M7}]})$ $\text{II}_7 := SD^{II_7} / D^{[V V_7 V_{-} V_{-7}]}$ $\flat\text{III}_7 := SD^{\flat III_7} / SM^{[\flat VII_{-} \flat VI_{M7}]}$ $\text{IV}_7 := SD^{IV_7} / SM^{[\flat VII_{-} \flat VII_{-}]}$ $\text{VI}_7 := SD^{VI_7} / S^{[II_{-} II_{-7}]}$ $\flat\text{VII}_7 := SD^{\flat VII_7} / T^{[\flat III_{-} \flat III_{M7}]}$ $\flat\text{V}_7 := SD^{\flat V_7} / (S^{[IV IV_6 IV_7]} SM^{[IV_{-} IV_{-6} IV_{-7} IV_{-M7}]})$ $\flat\text{VI}_7 := SD^{\flat VI_7} / D^{[V V_7 V_{-} V_{-7}]}$ $\text{VI}_7 := SD^{VI_7} / SM^{[\flat VII_{-} \flat VI_{M7}]}$ $VII_7 := SD^{VII_7} / SM^{[\flat VII_{-} \flat VII_{-}]}$ $\flat\text{III}_7 := SD^{\flat III_7} / S^{[II_{-} II_{-7}]}$ $III_7 := SD^{III_7} / T^{[\flat III_{-} \flat III_{M7}]}$	$\#I_{\circ} := PD^{\#I_{\circ}}$ $\#III_{\circ} := PD^{\#III_{\circ}}$ $\#IV_{\circ} := PD^{\#IV_{\circ}}$ $\#VI_{\circ} := PD^{VI_{\circ}}$ $\#VII_{\circ} := PD^{VII_{\circ}}$

また、青字で示したディグリーネームは、ジャズで頻繁に用いられるノンダイアトニックコードを表す。本来はダイアトニックコードのみを用いても音楽は成立するが、ノンダイアトニックコードを用いることでより豊かな和音進行が実現する。ノンダイアトニックコードをダイアトニックコードの代理和音とみなし、[15]を基に辞書項目に取り入れた。

4.2.4 転調間におけるピボットコードの発見

4.1 の問題点 3 への対応として、 $C \rightarrow D7 \rightarrow G \rightarrow A7 \rightarrow D$ と進行するようなコードネームを例として、以下の手法で解析を進める。なお、前節より入力の対象をディグリーネームにするため、和音進行がどの調に基づいているのかという調推定の手順が必要となるが、これに関しては次節で述べる。

1. 入力された和音進行に対して TPS による和声解析の手法を用いる。解析の結果、 $C, D7, G$ がト長調で、 $A7, D$ がニ長調となった。(図 4.3)
2. 左端の和音から転調後の中で左端に位置する和音に対して CCG による和声解析を行う。つまりこの例では C から $A7$ に対して解析を行う。ここで、ニ長調だとされている $A7$ は転調前のト長調の和音であると一時的に見なす。(図 4.4)
3. 解析が失敗したので、解析の範囲を C から G に狭め、再び CCG による和声解析を行う。ここでは解析が成功したので解析を行った和音に対してそれぞれ機能を与える。(図 4.5)
4. 転調前の和音の中で右端に位置する和音から右端の和音に対して CCG による和声解析を行う。つまりこの例では G から D に対して解析を行う。ここで、ト長調とされている G は転調後のニ長調の和音であると一時的に見なす。(図 4.6)
5. 今度は解析が成功したので、解析を行った和音に対してそれぞれ機能を与える。(図 4.7)

	C	D7	G	A7	D
TPSによる 出力	IV in G	V7 in G	I in G	V7 in D	I in D

図 4.3 提案手法による転調を含む和音進行の解析 1

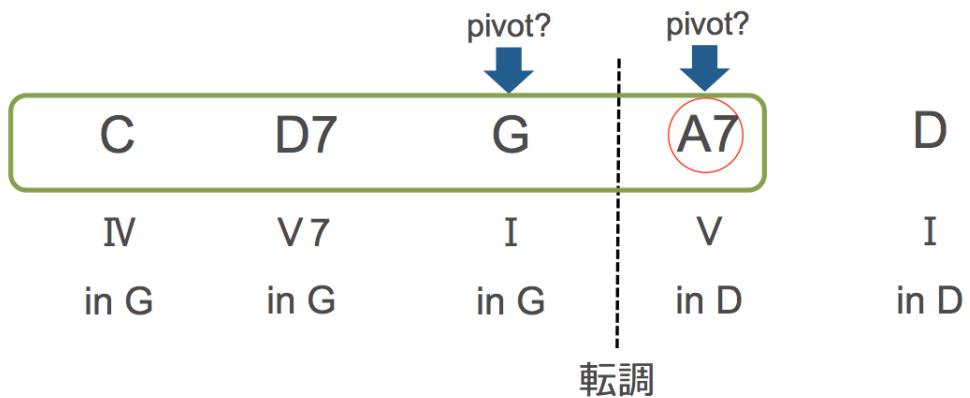


図 4.4 提案手法による転調を含む和音進行の解析 2

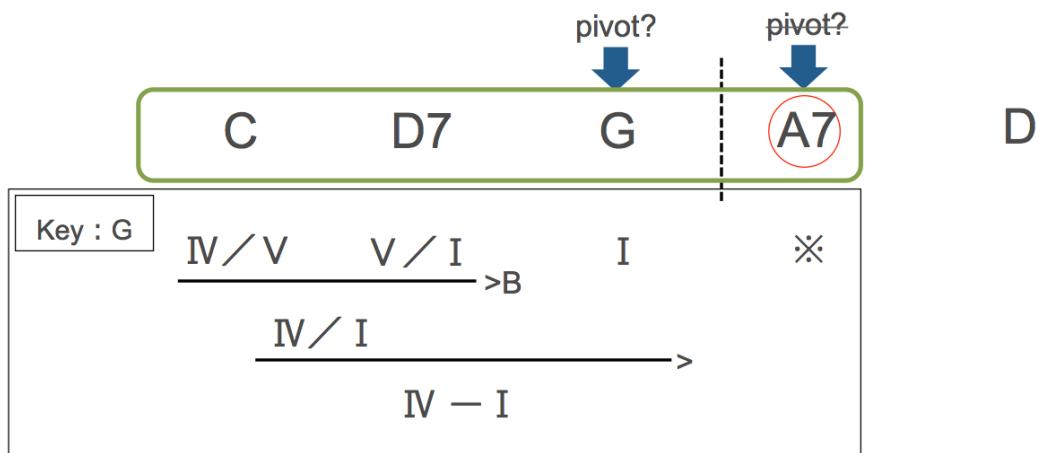


図 4.5 提案手法による転調を含む和音進行の解析 3

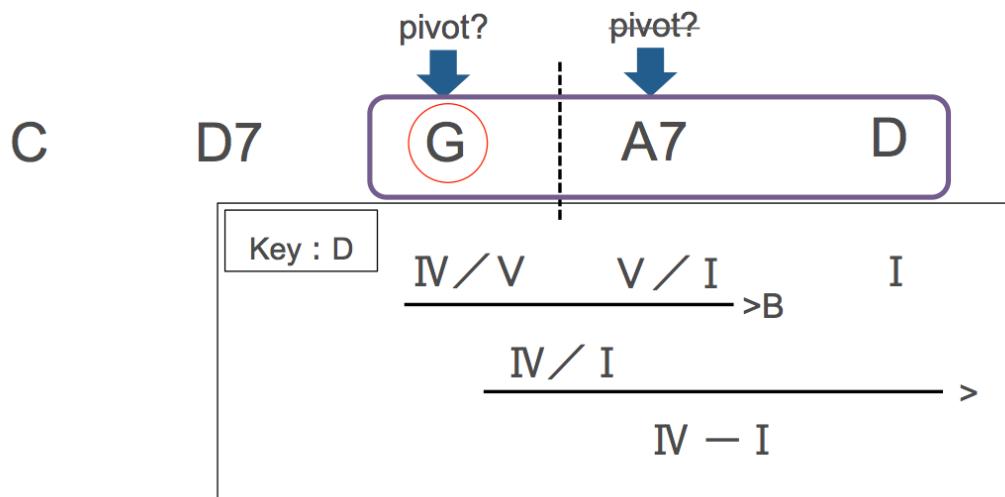


図 4.6 提案手法による転調を含む和音進行の解析 4

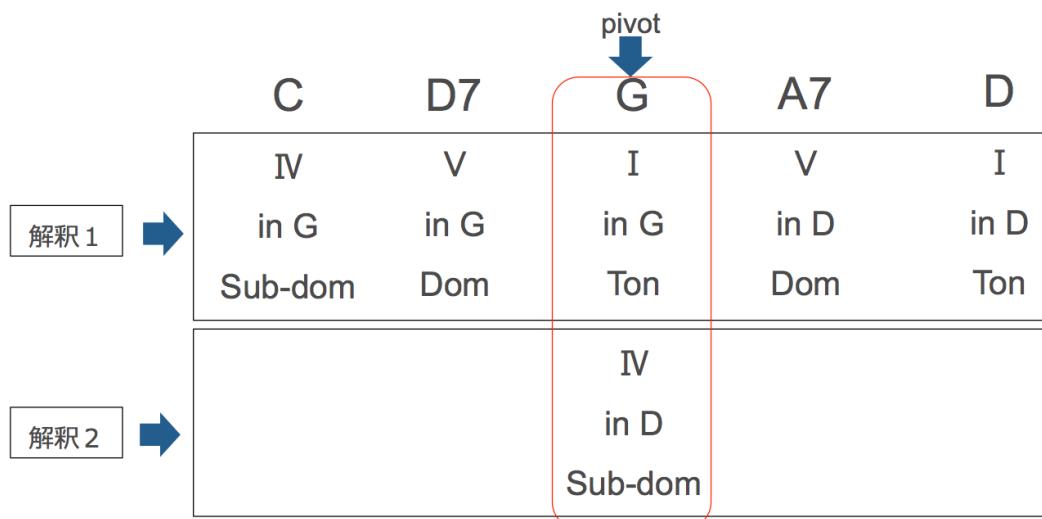


図 4.7 提案手法による転調を含む和音進行の解析 5

この手法の結果、G は転調前のト長調におけるトニックであるという出力と、転調後の二長調におけるサブドミナントであるという出力を得る。つまり G はピボットコードである。このように TPS による和声解析で明らかになった転調付近の二つの和音に対して、転調前の調であるという仮説と転調後の調であるという仮説を立て、CCG による和声解析でそれらの仮説が正しいかどうか検証することで、転調間で隠れているピボットコードを明らかにすることが可能になると考えられる。

4.3 ジャズ和声に対応したTPS[7]の導入

先行研究 [11]への優位性を示すため、本研究でも和声解析の入力を調の情報をもたないコードネームとする。前述よりCCGによる和声解析の対象をディグリーネームとしたため、入力されたコードネームからディグリーネームへの変換をしなければならない。つまり調推定の手法が必要になる。そこで本研究では、CCGによる和声解析の前段階としてTonal Pitch Space (TPS)を用いた和声解析の手法を取り入れる。特に、本研究ではジャズ和声を扱う事を前提としているため、2.2でも述べたジャズ和声に対応したTPS[7]の手法を採用する。

4.3.1 文献[7]の手法を用いた転調を含む和音進行の解析実験

本研究でTPSを扱うにあたって、文献[7]による提案手法にはいくつか問題点がある。

問題点1 式(2.1)における x と y のどちらの調を用いて和音の五度圏を扱っているのか不明である

問題点2 文献[7]の文章中では和音の五度圏を用いると述べているが、実際は和音の三度圏を用いている

問題点3 転調を含んだ和音進行に対しての実験が行われていない

このうち、問題点1・2に対しては以下のように対処する。

- 問題点1への対処

文献[7]で行われた実験の数値(図2.9)と完全に一致したため、2.1における x の調を用いることとする。

- 問題点2への対処

おそらく文献[7]の著者のミスだと思われるため、原著に則り和音の五度圏を用いる。

問題点3に関しては、転調を含む和音進行の例としてDm → G7 → C → Cm → F7 → B♭を用いて、文献[7]の提案手法を用いて和声解析を行うことで文献[7]の手法の妥当性を論じる。

文献[15]によると上記の進行は前半の3つの和音がハ長調上のツーファイブ・ドミナント進行であり、後半3つの和音が変ロ長調上のツーファイブ・ドミナント進行であると解釈される。つまり前半部と後半部で転調が起こっているということである。しかし、和声解析を行うと図4.8のように全ての和音が変ロ長調であると解釈された。つまり文献[7]の手法では本来あるべき転調が認識しなくなったということである。

図4.8の前半3つの和音は、ハ長調におけるダイアトニックコードであり、変ロ長調で考えるとすべてノンダイアトニック・コードである。ノンダイアトニック・コードに対する

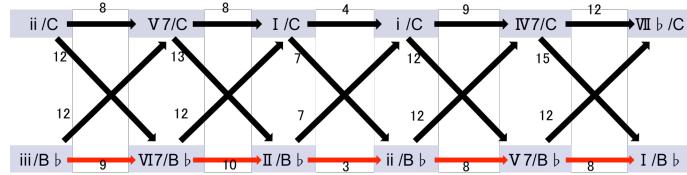


図 4.8 文献 [7] の提案手法を用いた転調を含む和音進行に対する実験結果の一部

る和音の五度圏距離はダイアトニックコードと比べて 1だけ大きいので、前半 3つの和音における和音間距離はハ長調上で考えたときが最小になる。従って [7] の手法は、なるべく多くの和音がダイアトニックコードとなるように調を選択することで、より和音進行全体の和音間距離が小さくなるものであると考えられる。

しかしながら、[7] では四和音のダイアトニックコードに関してノンダイアトニック・コードと見なして、和音の五度圏を拡張させている。そのため図 4.8 の G7 はハ長調上でノンダイアトニック・コードであるとみなされ、ハ長調上の和音間距離が大きくなつたと考えられる。そして、前半 3つの和音はノンダイアトニック・コードを一つ含んだハ長調であり、4 番目の和音から変ロ長調に転調すると捉えるよりも、前半 3つの和音はノンダイアトニック・コードを二つ含んだ変ロ長調であり、そのまま転調しないと捉えたことで和音間距離が小さくなつていると考えられる。

4.3.2 四和音への対応

4.3.1 で明らかになった問題に対処するために、四和音に対してもダイアトニックコードであるとみなし、和音の五度圏の拡張を行わない。つまり、本研究で用いる和音の五度圏は図 4.9 と図 4.10 のようになるとを考えた。

図 4.10 は自然的短音階のダイアトニックコードを和音の五度圏上に並べたものであるが、そこに V と V₇ を含めている。これらは本来自然的短音階に基づくダイアトニックコードではないが、V や V₇ に含まれる p11 の音が導音の役割を果たし、p0 を含む和音、つまりトニックへの進行を強く促すため、本研究ではダイアトニックコードとして扱う。

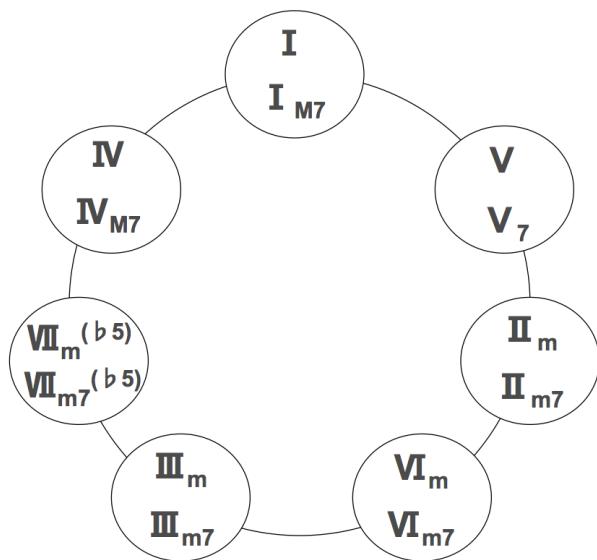


図 4.9 四和音に対応した長調における和音の五度圏

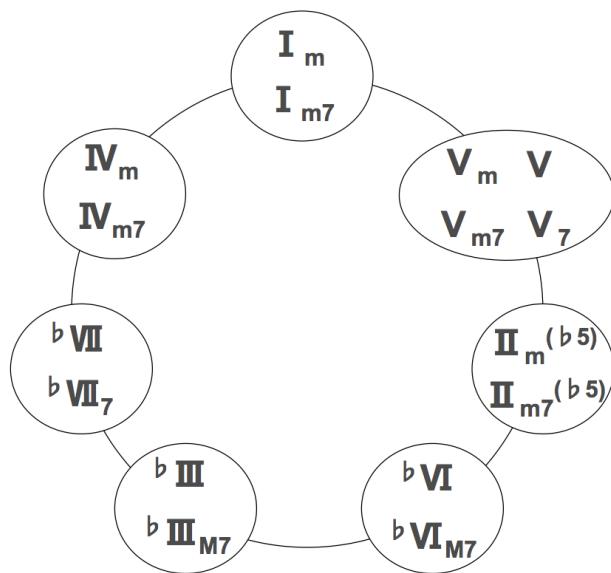


図 4.10 四和音に対応した短調における和音の五度圏

第5章 ジャズ和声解析システムの実装

提案手法により和声解析システムを Java 言語を用いて構築した。システムは TPS 和声解析部と CCG 和声解析部に分けられる。コードネームを入力すると、TPS 和声解析部によって各コードネームはどのような調を持っているのかといった調解釈を与える。これとともに CCG 和声解析部によって各和音の詳細な機能の解釈を与え、和声解析システムの結果として出力する。以降のプログラムに関する記述は Java 言語を想定したものとする。

5.1 TPS 和声解析部

TPS 和声解析部はノード生成部と和音間距離計算部と最短距離計算部に分けられる。計算の便宜上、以降では調ピッチクラスという概念を導入する。調ピッチクラスを以下の表 5.1 のように定義する。

表 5.1 調の主音と調ピッチクラスの値

調の主音	C	C#/D ^b	D	D#/E ^b	E	F	F#/G ^b	G	G#/A ^b	A	A#/B ^b	B
調ピッチクラス値	0	1	2	3	4	5	6	7	8	9	10	11

5.1.1 ノード生成部

コードネームの配列を入力として受け取る。N 個のコードネームの配列はコードネームの根音とコードタイプの2つの要素をもつ。根音は音名のピッチクラス値、コードタイプは M,m,M7 などの文字列で表す。例えば、G,E7,Am7,D7,G というコードネームの進行が与えられた場合、コードネームの配列は以下のように格納される。

$$\begin{aligned}
CN[0] &= \{7, M\} \\
CN[1] &= \{4, Mm7\} \\
CN[2] &= \{9, m7\} \\
CN[3] &= \{2, Mm7\} \\
CN[4] &= \{7, M\}
\end{aligned} \tag{5.1}$$

入力されたコードネームの配列は Chord クラスによってノードを生成する。ノードは各コードネームの解釈として可能性のある和音を意味しており、与えられたコードネームを長調・短調含めた全ての調に当てはめて、有向グラフのノードとする。また生成されたノードはプロパティとしてコードネームの根音 (root)、コードタイプ (chordType)、調の主音 (keynote)、調タイプ (modeType)、距離の最小値 (minDistance) の 5 つをもつ。調の主音は調ピッチクラス、調タイプは長調であれば M、短調であれば m の文字列で表す。距離の最小値は初期値 0 である。例えば、前述の CN[2] のノードは表 5.2 のようになる。

隣り合うコードネーム同士の全てのノードはエッジで接続され、トレリス状のグラフが生成される。各エッジについて提案手法に則した和音間距離を求め、それをエッジのコストとする。和音間距離の計算については 5.1.2 で述べる。また、最初のコードネームと最後のコードは、便宜的にそれぞれグラフの開始ノード START、グラフの終端ノード END とコスト 0 のエッジで接続される。

5.1.2 和音間距離計算部

和音間距離の計算は TonalPitchSpace クラスの calcTonalPitchSpace メソッドによって行われる。calcTonalPitchSpace メソッドは calcRegion メソッドと calcDiatonicCircleDistance メソッドと calcBasicSpace メソッドのそれぞれの戻り値の和によって表される。以後ここで渡される 2 つの和音は x と y と表す。

- calcRegion メソッド

calcTonalPitchSpace メソッドから calcRegion メソッドに 2 つの和音が与えられると、それぞれの調ピッチクラスを fromkey、tokey に格納する。fromkey と tokey について

$$\delta_r = \begin{cases} 0 & (|fromkey - tokey| = 0) \\ 1 & (|fromkey - tokey| = 7 \text{ または } |fromkey - tokey| = 5) \\ 2 & (|fromkey - tokey| = 2 \text{ または } |fromkey - tokey| = 10) \\ 3 & (|fromkey - tokey| = 9 \text{ または } |fromkey - tokey| = 3) \\ 4 & (|fromkey - tokey| = 4 \text{ または } |fromkey - tokey| = 8) \\ 5 & (|fromkey - tokey| = 11 \text{ または } |fromkey - tokey| = 1) \\ 6 & (|fromkey - tokey| = 6) \end{cases} \tag{5.2}$$

を計算し、 δ_r の値を calcRegion メソッドの結果として返す。

- calcBasicSpace メソッド

calcTonalPitchSpace メソッドから calcBasicSpace メソッドに 2 つの和音が与えられると、それぞれのベーシックスペースを生成する。 x, y から生成されたベーシックスペースをそれぞれ長さ 12 の配列 fromVec, toVec とする。配列の各要素はピッチクラスを表し、格納されている各値は各ピッチクラスのレベルを表す。

ベーシックスペースの生成後、両者を比較してその変化分を求める。ベーシックスペースの全ての要素について

$$\delta_s[i] = \begin{cases} fromVec[i] - toVec[i] & (fromVec[i] \geq toVec[i]) \\ 0 & (\text{otherwise}) \end{cases} \quad (5.3)$$

を計算する。その後 $\delta[i]$ の各要素の合計を求め、その値を calcBasicSpace メソッドの結果として返す。

- calcDiatonicCircleDistance メソッド

calcTonalPitchSpace メソッドから calcDiatonicCircleDistance メソッドに 2 つの和音が与えられると、 x が置かれている調のダイアトニックコードを生成する。計算の便宜上、ダイアトニックピッチクラスを表 5.3 のように定義する。

x と各ダイアトニックコードを calcBasicSpace メソッドにより計算し、値が最小になったダイアトニックコードに対応するダイアトニックピッチクラス値を fromDiatonic に格納する。最小になった値がもし 0 でなければ、 x に対して和音の五度圏を拡張する必要があることを示すフラグを true にする。

y と各ダイアトニックコードに対しても同様に calcBasicSpace メソッドにより計算し、値が最小になったダイアトニックコードに対応するダイアトニックピッチクラス値を toDiatonic に格納する。最小になった値がもし 0 でなければ、 y に対して和音の五度圏を拡張する必要があることを示すフラグを true にする。

fromDiatonic と toDiatonic について

$$\delta_d = \begin{cases} 0 & (|fromDiatonic - toDiatonic| = 0) \\ 1 & (|fromDiatonic - toDiatonic| = 4 \text{ または } |fromDiatonic - toDiatonic| = 3) \\ 2 & (|fromDiatonic - toDiatonic| = 1 \text{ または } |fromDiatonic - toDiatonic| = 6) \\ 3 & (|fromDiatonic - toDiatonic| = 2 \text{ または } |fromDiatonic - toDiatonic| = 5) \end{cases} \quad (5.4)$$

を計算する。どちらか一方が true であれば 3 を、両方とも true であれば 6 を δ_d に加算し、 δ_d の値を calcDiatonicCircleDistance メソッドの結果として返す。

5.1.3 最短距離計算部

和音間距離計算部により得られたコストを用い、Vitervi アルゴリズム [?] を用いて最短経路を求める。

これにより求められたグラフの経路は和音間距離の累計が最も小さい経路であり、すなわち全体的に見て最も和音進行が自然な調解釈であると推定できる。

最短経路上にあるノードは N 個の和音の配列に格納される。和音の配列はコードネームの根音、コードタイプ、調の主音、調タイプの 4 つの要素をもつ。例えば、式(5.1)のようにコードネームの配列が与えられている場合、TPS 和声解析部の計算によって和音の配列は式(5.5)のように格納される。

$$\begin{aligned} CO[0] &= \{7, M, 7, M\} \\ CO[1] &= \{4, Mm7, 7, M\} \\ CO[2] &= \{9, m7, 7, M\} \\ CO[3] &= \{2, Mm7, 7, M\} \\ CO[4] &= \{7, M, 7, M\} \end{aligned} \tag{5.5}$$

5.2 CCG 和声解析部

CCG 和声解析部は CCG クラスの `analyzeCCG` メソッドによって解析が行われる。

長調における辞書項目の配列 L_M は表層形のディグリーネームの根音、表層形のコードタイプ、カテゴリ、期待するカテゴリ、和音の機能の 6 つの要素を持つ。表 5.5 にその一部を示す。

短調における辞書項目の配列 L_m も同様の要素を持つ。表 5.6 にその一部を示す。

5.2.1 `analyzeCCG` メソッド

TPS 和声解析部より渡された $CO[0]$ から $CO[n - 1]$ の n 個の和音の配列は `analyzeCCG` メソッドによって解析が行われる。以下にそのアルゴリズム示す。

ここでの $O[0]$ から $O[n - 1]$ は最終的な和声解析の出力を表しており、それぞれコードネームの根音、コードタイプ、調の主音、調タイプ、和音の機能を要素にもつ。

また、`analyzeCCG` メソッドは内部に `innerCCG` メソッドを含む。`innerCCG` メソッドについては 5.2.2 で述べる。

1. はじめに i , k_1 , k_2 を 0 で初期化する。
2. i の値を調べる。

- (a) i の値が $n - 1$ であれば $CO[k_1 - 1]$ から $CO[n - 1]$ までの和音の配列を innerCCG 関数に与える.
- innerCCG 関数より解析成功のフラグが返されたら, 解析を終了する.
 - innerCCG 関数より解析失敗のフラグが返されたら, $CO[k_1]$ から $CO[n - 1]$ までの和音の配列を innerCCG 関数に与え, 解析を終了する.
- (b) i の値が $n - 1$ でなければ, $CO[i]$ と $CO[i + 1]$ の調の主音と調タイプを比較する.
- $CO[i]$ と $CO[i + 1]$ の調の主音と調タイプのいずれかが異なる場合, k_2 に $i + 1$ を代入する. $CO[k_1 - 1]$ から $CO[k_2]$ までの和音の配列を innerCCG 関数に与える. ただし, k_1 が 0 のときは $CO[k_1]$ から $CO[k_2]$ までとする.
 - innerCCG 関数より解析成功のフラグが返されたら, $CO[k_1 - 1]$ から $CO[k_2]$ までの和音の要素を $O[k_1 - 1]$ から $O[k_2]$ に格納する. k_1 に k_2 を代入し, i に 1 を加え, 2 に戻る.
 - innerCCG 関数より解析失敗のフラグが返されたら, $CO[k_1 - 1]$ (k_1 が 0 の場合は $CO[k_1]$) から $CO[k_2 - 1]$ までの和音の配列を innerCCG 関数に与える.
 - innerCCG 関数より解析成功のフラグが返されたら, $CO[k_1 - 1]$ から $CO[k_2 - 1]$ までの和音の要素を $O[k_1 - 1]$ から $O[k_2 - 1]$ に格納する. k_1 に k_2 を代入し, i に 1 を加え, 2 に戻る.
 - innerCCG 関数より解析失敗のフラグが返されたら, $CO[k_1]$ から $CO[k_2 - 1]$ までの和音の配列を innerCCG 関数に与える. その後に $CO[k_1]$ から $CO[k_2 - 1]$ までの和音の要素を $O[k_1]$ から $O[k_2 - 1]$ に格納する. k_1 に k_2 を代入し, i に 1 を加え, 2 に戻る. ただし k_1 が 0 の場合はこの処理を行わない.
 - $CO[i]$ と $CO[i + 1]$ の調の主音と調タイプのいずれも等しい場合, k_1 に k_2 を代入し, i に 1 を加え, 2 に戻る.

5.2.2 innerCCG メソッド

analyzeCCG メソッドより渡された m 個の和音の配列は $c[0]$ から $c[m - 1]$ に格納し, innerCCG メソッドによって解析が行われる. 以下にそのアルゴリズムを示す.

また, innerCCG メソッドは内部に degreeNameConversion メソッドと ruleCheck メソッドを含む. degreeNameConversion メソッドについては 5.2.3, ruleCheck メソッドについては 5.2.4 で述べる.

- 和音の配列を degreeNameConversion メソッドに渡し, degreeNameConversion メソッドを実行する.
- i を 0 で初期化する. また一次元配列 $T[0]$ から $T[m - 1]$ を 0 で初期化する.

3. j を 0 で初期化する.
4. $c[i]$ と $L[j]$ のディグリーネームの根音とコードタイプを比較する.
 - (a) $c[i]$ と $L[j]$ のディグリーネームの根音とコードタイプのいずれかが異なる場合, j に 1 を加える.
 - i. この時点で j が k となった場合, i の値を 1 減らす.
 - A. この時点で i が -1 となった場合, 解析失敗のフラグを analyzeCCG メソッドに渡し, 終了する.
 - B. それ以外の場合, j に $T[i]$ を代入して 4 に戻る.
 - ii. それ以外の場合は 4 に戻る.
 - (b) $c[i]$ と $L[j]$ のディグリーネームの根音とコードタイプのいずれも等しい場合, $T[i]$ に j の値を代入し, $c[i]$ のカテゴリと機能に $L[j]$ のそれらを代入する. また, 配列 S の要素にカテゴリを追加する.
 - i. S の要素が 2 つ以上であるなら, ruleCheck メソッドを実行する.
 - ii. i の値を調べる.
 - A. i が $m - 1$ でない場合, i に 1 を加えて 3 に戻る.
 - B. i が $m - 1$ である場合, 以下を実行する.
 - S の要素が一つでない場合, $c[i]$ のカテゴリと機能を消去する. j に $T[i] + 1$ を代入して 4 に戻る.
 - S の要素が一つの場合, 解析成功のフラグを analyzeCCG メソッドに渡し, 終了する.

5.2.3 degreeNameConversion メソッド

innerCCG メソッドで解析される和音は同じ調でなくてはならない. そのため, 以下の処理を行う.

1. $c[0]$ と $c[1]$ の調の主音と調タイプを比べる. $c[0]$ と $c[1]$ の調の主音と調タイプのいずれかが異なる場合, $c[0]$ の調の主音と調タイプに $c[1]$ の調の主音と調タイプを代入する.
2. $c[m-2]$ と $c[m-1]$ の調の主音と調タイプを比べる. $c[0]$ と $c[1]$ の調の主音と調タイプのいずれかが異なる場合, $c[0]$ の調の主音と調タイプに $c[1]$ の調の主音と調タイプを代入する.

また, CCG による和声解析はディグリーネームで表された和音を対象にしている. そのため innerCCG メソッドによって扱われる和音はディグリーネームでなくてはならない.

そこで, $c[i]$ の要素のひとつであるコードネームの根音を表5.4のようなルックアップテーブルを参照し, ディグリーネームの根音に変換する.

これにより, 例えば式(5.6)は表5.4により式(5.7)のように変換される.

$$\begin{aligned}
 c[0] &= \{7, M, 7, M\} \\
 c[1] &= \{4, Mm7, 7, M\} \\
 c[2] &= \{9, m7, 7, M\} \\
 c[3] &= \{2, Mm7, 7, M\} \\
 c[4] &= \{7, M, 7, M\}
 \end{aligned} \tag{5.6}$$

$$\begin{aligned}
 c[0] &= \{0, M, 7, M\} \\
 c[1] &= \{9, Mm7, 7, M\} \\
 c[2] &= \{2, m7, 7, M\} \\
 c[3] &= \{7, Mm7, 7, M\} \\
 c[4] &= \{0, M, 7, M\}
 \end{aligned} \tag{5.7}$$

変換された和音の配列は innerCCG メソッドに渡される.

5.2.4 ruleCheck メソッド

ruleCheck メソッドは $S[0]$ から $S[m-1]$ までのカテゴリについて規則の適用を行っている. ここでの m は ruleCheck メソッドが実行された時点での要素の数とする.

ruleCheck メソッドのアルゴリズムを以下に示す.

1. i を 0 で初期化する.
2. $S[i]$ と $S[i+1]$ のカテゴリが規則を用いる状態になっているかどうか調べる.
 - (a) 規則を用いることができるなら, $S[i]$ に規則を用いた後のカテゴリを代入し, $S[i+1]$ を消去する.
 - i. もしこの時点で S の要素が 1 つになつたら, 終了する.
 - ii. S の要素が 2 つ以上あれば, $S[i+2]$ 以降を繰り上げ, m の値を 1 減らす. その後 1 に戻る.
 - (b) 規則を用いることができないなら, i に 1 を加える.
 - i. もしこの時点で i の値が m と等しいならば, 終了する.

ii. そうでないならば、2に戻る。

表 5.5 長調における辞書項目の配列の一部と要素

	和音の根音	コードタイプ	カテゴリ 1	カテゴリ 2	和音の機能
$L_M[1]$	0	M	T		Tonic
$L_M[2]$	0	M6	T		Tonic
$L_M[3]$	0	M7	T		Tonic
$L_M[4]$	4	m	T		Tonic
$L_M[5]$	4	m7	T		Tonic
$L_M[6]$	9	m	T		Tonic
$L_M[7]$	9	m7	T		Tonic
$L_M[8]$	1	Mm7	T		Tonic
$L_M[9]$	6	mFlat57	T		Tonic
$L_M[10]$	4	m	4m		Tonic
$L_M[11]$	4	m7	4m7		Tonic
$L_M[12]$	9	m	9m		Tonic
$L_M[13]$	9	m7	9m7		Tonic
$L_M[14]$	5	M	S	T	Sub Dominant
$L_M[15]$	5	M6	S	T	Sub Dominant
$L_M[16]$	5	M7	S	T	Sub Dominant
$L_M[17]$	2	m	S	T	Sub Dominant
$L_M[18]$	2	m7	S	T	Sub Dominant
$L_M[19]$	5	m7	S	T	Sub Dominant
$L_M[20]$	11	m7	S	T	Sub Dominant
$L_M[21]$	6	mFlat57	S	T	Sub Dominant
$L_M[22]$	5	M	5M	T	Sub Dominant
$L_M[23]$	5	M6	5M6	T	Sub Dominant
$L_M[24]$	5	M7	5M7	T	Sub Dominant
$L_M[25]$	2	m	2m	T	Sub Dominant
$L_M[26]$	2	m7	2m7	T	Sub Dominant

表 5.6 短調における辞書項目の配列と要素

	和音の根音	コードタイプ	カテゴリ 1	カテゴリ 2	和音の機能
$L_M[1]$	0	m	T		Tonic
$L_M[2]$	0	m6	T		Tonic
$L_M[3]$	0	m7	T		Tonic
$L_M[4]$	0	mm7	T		Tonic
$L_M[5]$	3	M	T		Tonic
$L_M[6]$	3	M7	T		Tonic
$L_M[7]$	3	aug	T		Tonic
$L_M[8]$	3	augM7	T		Tonic
$L_M[9]$	9	mFlat5	T		Tonic
$L_M[10]$	9	mFlat57	T		Tonic
$L_M[11]$	3	M	3M		Tonic
$L_M[12]$	3	M7	3M7		Tonic
$L_M[13]$	5	M	S	T	Sub Dominant
$L_M[14]$	5	M6	S	T	Sub Dominant
$L_M[15]$	5	Mm7	S	T	Sub Dominant
$L_M[16]$	2	m	S	T	Sub Dominant
$L_M[17]$	2	m7	S	T	Sub Dominant
$L_M[18]$	11	Mm7	S	T	Sub Dominant
$L_M[19]$	5	M	5M	T	Sub Dominant
$L_M[20]$	5	M6	5M6	T	Sub Dominant
$L_M[21]$	5	Mm7	5M7	T	Sub Dominant
$L_M[22]$	2	m	2m	T	Sub Dominant
$L_M[23]$	2	m7	2m7	T	Sub Dominant

表 5.2 $CN[2] = \{9, m7\}$ に対するノードの生成例

	root	chordType	keynote	modeType	minDistance
$CH[2][0]$	9	m7	0	M	0
$CH[2][1]$	9	m7	1	M	0
$CH[2][2]$	9	m7	2	M	0
$CH[2][3]$	9	m7	3	M	0
$CH[2][4]$	9	m7	4	M	0
$CH[2][5]$	9	m7	5	M	0
$CH[2][6]$	9	m7	6	M	0
$CH[2][7]$	9	m7	7	M	0
$CH[2][8]$	9	m7	8	M	0
$CH[2][9]$	9	m7	9	M	0
$CH[2][10]$	9	m7	10	M	0
$CH[2][11]$	9	m7	11	M	0
$CH[2][12]$	9	m7	0	m	0
$CH[2][13]$	9	m7	1	m	0
$CH[2][14]$	9	m7	2	m	0
$CH[2][15]$	9	m7	3	m	0
$CH[2][16]$	9	m7	4	m	0
$CH[2][17]$	9	m7	5	m	0
$CH[2][18]$	9	m7	6	m	0
$CH[2][19]$	9	m7	7	m	0
$CH[2][20]$	9	m7	8	m	0
$CH[2][21]$	9	m7	9	m	0
$CH[2][22]$	9	m7	10	m	0
$CH[2][23]$	9	m7	11	m	0

表 5.3 ダイアトニックコードとダイアトニックピッチクラス値

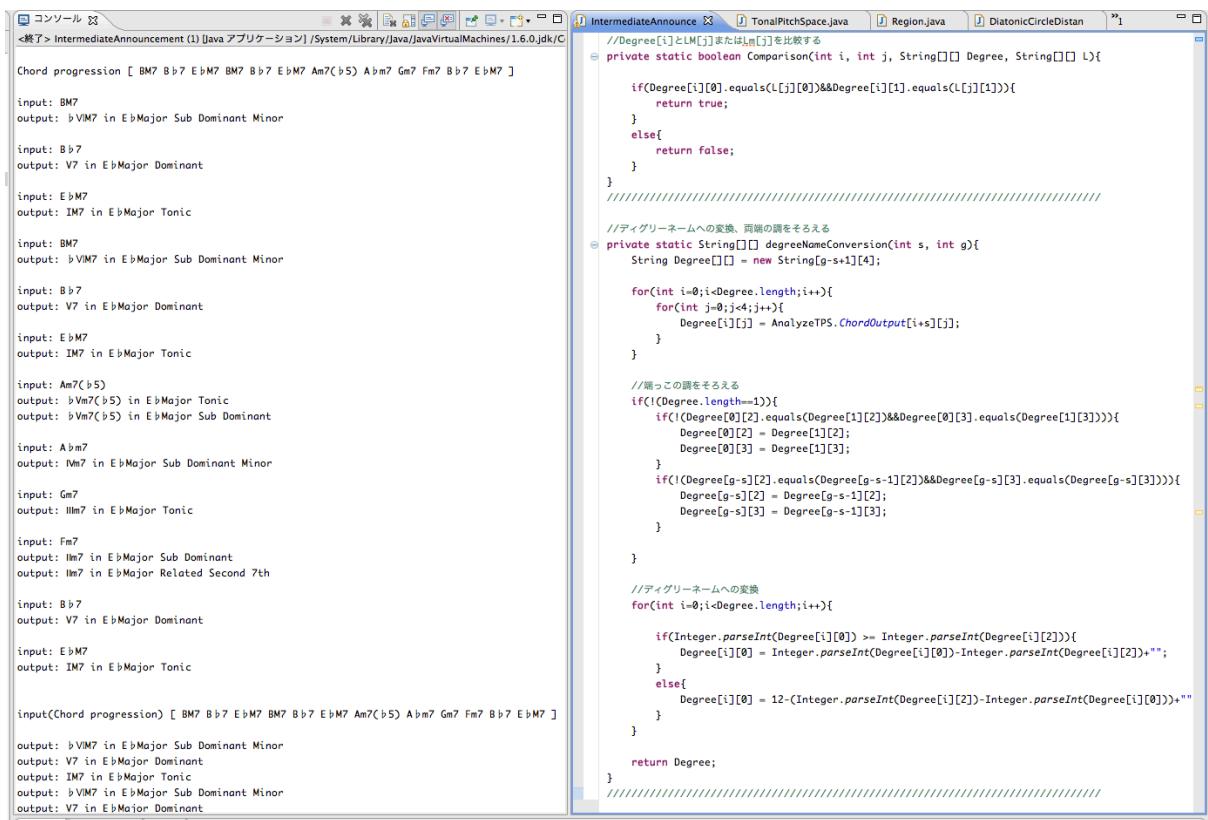
ダイアトニックコード（長調）	I	II ₋	III ₋	IV	V	VI ₋	VII ^(b5)
ダイアトニックコード（短調）	I ₋	II ₋ ^(b5)	^b III	IV ₋	V	^b VI	^b VII
ダイアトニックピッチクラス値	0	1	2	3	4	5	6

表 5.4 ディグリーネーム変換テーブル

		調の主音											
		0	1	2	3	4	5	6	7	8	9	10	11
	0	0	11	10	9	8	7	6	5	4	3	2	1
	1	1	0	11	10	9	8	7	6	5	4	3	2
コ	2	2	1	0	11	10	9	8	7	6	5	4	3
ード	3	3	2	1	0	11	10	9	8	7	6	5	4
ネ	4	4	3	2	1	0	11	10	9	8	7	6	5
ーム	5	5	4	3	2	1	0	11	10	9	8	7	6
の	6	6	5	4	3	2	1	0	11	10	9	8	7
根	7	7	6	5	4	3	2	1	0	11	10	9	8
音	8	8	7	6	5	4	3	2	1	0	11	10	9
	9	9	8	7	6	5	4	3	2	1	0	11	10
	10	10	9	8	7	6	5	4	3	2	1	0	11
	11	11	10	9	8	7	6	5	4	3	2	1	0

第6章 実験

提案手法の妥当性を調査するために、前章で述べたシステムを用いて和音進行に対して和声解析を行った。システム実行の様子を図6.1に示す。



The screenshot shows a Java development environment with two windows. On the left is the 'Console' window, which displays the output of a program named 'IntermediateAnnouncement'. The console output shows various chord progressions and their analysis results. On the right is the 'TonalPitchSpace.java' editor window, containing Java code for comparing chords and performing degree name conversion.

```
Chord progression [ BM7 B♭7 E♭M7 BM7 B♭7 E♭M7 Am7(♭5) Abm7 Gm7 Fm7 B♭7 EbM7 ]  
input: BM7  
output: bVIM7 in EbMajor Sub Dominant Minor  
  
input: B♭7  
output: V7 in EbMajor Dominant  
  
input: EbM7  
output: IM7 in EbMajor Tonic  
  
input: BM7  
output: bVIM7 in EbMajor Sub Dominant Minor  
  
input: B♭7  
output: V7 in EbMajor Dominant  
  
input: EbM7  
output: IM7 in EbMajor Tonic  
  
input: Am7(♭5)  
output: bVm7(♭5) in EbMajor Tonic  
output: bVm7(♭5) in EbMajor Sub Dominant  
  
input: Abm7  
output: Vm7 in EbMajor Sub Dominant Minor  
  
input: Gm7  
output: Ilm7 in EbMajor Tonic  
  
input: Fm7  
output: Ilm7 in EbMajor Sub Dominant  
output: Ilm7 in EbMajor Related Second 7th  
  
input: B♭7  
output: V7 in EbMajor Dominant  
  
input: EbM7  
output: IM7 in EbMajor Tonic  
  
input<(Chord progression> [ BM7 B♭7 E♭M7 BM7 B♭7 E♭M7 Am7(♭5) Abm7 Gm7 Fm7 B♭7 EbM7 ]  
output: bVIM7 in EbMajor Sub Dominant Minor  
output: V7 in EbMajor Dominant  
output: IM7 in EbMajor Tonic  
output: bVIM7 in EbMajor Sub Dominant Minor  
output: V7 in EbMajor Dominant
```

```
//Degree[]とLM[]またはLm[]を比較する  
private static boolean Comparison(int i, int j, String[] Degree, String[] L){  
    if(Degree[i][0].equals(L[j][0])&&Degree[i][1].equals(L[j][1])){  
        return true;  
    }  
    else{  
        return false;  
    }  
}  
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
//ディグリーネームへの変換、両側の調をそろえる  
private static String[] degreeNameConversion(int s, int g){  
    String Degree[] = new String[g-s+1][4];  
  
    for(int i=0;i<Degree.length;i++){  
        for(int j=0;j<4;j++){  
            Degree[i][j] = AnalyzeTPS.ChordOutput[i+s][j];  
        }  
    }  
  
    //黒っこの調をそろえる  
    if(!Degree.length==1){  
        if((Degree[0][2].equals(Degree[1][2])&&Degree[0][3].equals(Degree[1][3])){  
            Degree[0][2] = Degree[1][2];  
            Degree[0][3] = Degree[1][3];  
        }  
        if((Degree[g-s][2].equals(Degree[g-s-1][2])&&Degree[g-s][3].equals(Degree[g-s][3])){  
            Degree[g-s][2] = Degree[g-s-1][2];  
            Degree[g-s][3] = Degree[g-s-1][3];  
        }  
    }  
  
    //ディグリーネームへの変換  
    for(int i=0;i<Degree.length;i++){  
        if(Integer.parseInt(Degree[i][0]) >= Integer.parseInt(Degree[i][2])){  
            Degree[i][0] = Integer.parseInt(Degree[i][0])-Integer.parseInt(Degree[i][2])+";";  
        }  
        else{  
            Degree[i][0] = 12-(Integer.parseInt(Degree[i][2])-Integer.parseInt(Degree[i][0]))+";"  
        }  
    }  
}  
return Degree;
```

図 6.1 実行の様子

入力に用いた和音進行はから抜粋し、調の情報を持たないコードネームを前提とした。また、本研究ではビバップ時代までのジャズ和声を前提としているため、教会旋法などのモードに基づいた和音進行は入力の対象外とした。

6.1 実験結果

実験の結果を図 6.2～図 6.8 に示す。それぞれの結果の一行目は入力した和音進行を表し、一行おきにそれぞれの和音に対して入力と出力をセットで表示している。

すべての入力において実行時間は一つの和音進行あたり約 1 秒程度であり、問題はないと思われる。

図 6.2 の入力は、5 番目と 6 番目の和音の間でハ長調からヘ長調への転調が起こっていることを想定している。システムの出力によると、5 番目と 6 番目だけでなく、4 番目と 5 番目にも転調が起こっていることがわかった。そのため 5 番目の和音はハ長調、ニ長調のどちらの解釈も可能であり、ピボットコードを認識することができた。また、1 番目はトニックであり、3 番目と 6 番目の和音はサブドミナントであるが、後に根音が四度上のセブンスコードがきていることからリレイテッドツーマイナー 7th であると考えられる。システムの出力では、先行研究では考慮されていなかったリレイテッドツーマイナー 7th を含んだ二通りの解釈が与えられ、ピボットコード以外の 2 つの機能をもつ和音を認識することができた。

図 6.3 の入力は図に示したピボットコードを含んだ和音進行と同じものである。図では 4 番目の和音がピボットコードの役割を成し、二通りの解釈をもつていると述べたが、出力により、3 番目の和音に対して三通りの解釈が可能であることがわかった。そのため、2 番目と 3 番目、3 番目と 4 番目、4 番目と 5 番目で転調が起こっていることが認識できた。

図 6.4 の入力はサブドミナントマイナーを含んだ和音進行である。先行研究ではサブドミナントマイナーは考慮されていなかったが、本システムの出力では適切にサブドミナントマイナーを認識することができた。

図 6.5 の四番目の和音は裏コードである。出力でも裏コードを認識することができた。

図 6.6 の入力は短調を想定した和音進行である。しかしながら出力は途中で解析失敗となつた。

図 6.7 の入力は長調から平行短調への転調を想定した和音進行である。しかしながら出力は途中で解析失敗となつた。

図 6.8 の入力は短い転調の繰り返しを想定した和音進行である。しかしながら出力は途中で解析失敗となつた。

```
Chord progression [ Em7 A7 Dm7 G7 C Gm7 C7 F ]  
  
input: Em7  
output: IIIm7 in CMajor Tonic  
output: IIIm7 in CMajor Related Second 7th  
  
input: A7  
output: VI7 in CMajor Secondary Dominant  
  
input: Dm7  
output: IIm7 in CMajor Sub Dominant  
output: IIm7 in CMajor Related Second 7th  
  
input: G7  
output: V7 in CMajor Dominant  
  
input: C  
output: I in CMajor Tonic  
output: V in FMajor Dominant  
  
input: Gm7  
output: IIm7 in FMajor Sub Dominant  
output: IIm7 in FMajor Related Second 7th  
  
input: C7  
output: V7 in FMajor Dominant  
  
input: F  
output: I in FMajor Tonic
```

図 6.2 実験結果 1

```
Chord progression [ G C D7 G A7 D ]  
  
input: G  
output: I in GMajor Tonic  
  
input: C  
output: IV in GMajor Sub Dominant  
  
input: D7  
output: V7 in GMajor Dominant  
output: I7 in DMajor Tonic  
output: I7 in DMajor Secondary Dominant  
  
input: G  
output: I in GMajor Tonic  
output: IV in DMajor Sub Dominant  
  
input: A7  
output: V7 in DMajor Dominant  
  
input: D  
output: I in DMajor Tonic
```

図 6.3 実験結果 2

```
Chord progression [ BM7 B♭7 E♭M7 BM7 B♭7 E♭M7 Am7(♭5) A♭m7 Gm7 Fm7 B♭7 E♭M7 ]  
  
input: BM7  
output: ♫VIM7 in E♭Major Sub Dominant Minor  
  
input: B♭7  
output: V7 in E♭Major Dominant  
  
input: E♭M7  
output: IM7 in E♭Major Tonic  
  
input: BM7  
output: ♫VIM7 in E♭Major Sub Dominant Minor  
  
input: B♭7  
output: V7 in E♭Major Dominant  
  
input: E♭M7  
output: IM7 in E♭Major Tonic  
  
input: Am7(♭5)  
output: ♫Vm7(♭5) in E♭Major Tonic  
output: ♫Vm7(♭5) in E♭Major Sub Dominant  
  
input: A♭m7  
output: IVm7 in E♭Major Sub Dominant Minor  
  
input: Gm7  
output: IIIm7 in E♭Major Tonic  
  
input: Fm7  
output: IIIm7 in E♭Major Sub Dominant  
output: IIIm7 in E♭Major Related Second 7th  
  
input: B♭7  
output: V7 in E♭Major Dominant  
  
input: E♭M7  
output: IM7 in E♭Major Tonic
```

図 6.4 実験結果 3

```
Chord progression [ Em7 A7 Dm7 D♭7 C Dm7 G7 C ]  
  
input: Em7  
output: IIIm7 in CMajor Tonic  
output: IIIm7 in CMajor Related Second 7th  
  
input: A7  
output: VI7 in CMajor Secondary Dominant  
  
input: Dm7  
output: IIIm7 in CMajor Sub Dominant  
  
input: D♭7  
output: ♭II7 in CMajor Dominant  
  
input: C  
output: I in CMajor Tonic  
  
input: Dm7  
output: IIIm7 in CMajor Sub Dominant  
output: IIIm7 in CMajor Related Second 7th  
  
input: G7  
output: V7 in CMajor Dominant  
  
input: C  
output: I in CMajor Tonic
```

図 6.5 実験結果4

```
Chord progression [ Am7(b5) D7 Gm Am7(b5) D7 Gm ]  
  
input: Am7(b5)  
output: IIm7(b5) in GMinor Sub Dominant Minor  
output: IIm7(b5) in GMinor Related Second 7th  
  
input: D7  
output: V7 in GMinor Dominant  
  
input: Gm  
output: Im in GMinor Tonic  
  
input: Am7(b5)  
  
input: D7  
  
input: Gm
```

図 6.6 実験結果 5

```
Chord progression [ Cm7 F7 B♭M7 E♭M7 Am7(♭5) D7 Gm ]  
input: Cm7  
input: F7  
input: B♭M7  
input: E♭M7  
input: Am7(♭5)  
input: D7  
input: Gm
```

図 6.7 実験結果 6

```
Chord progression [ FM7 B♭ C7 FM7 Em7(♭5) A7 Dm7 Bm7(♭5) E7 AM7 Gm7 C7 F ]  
  
input: FM7  
output: Im7 in F Major Tonic  
  
input: B♭  
output: IV in F Major Sub Dominant  
  
input: C7  
output: V7 in F Major Dominant  
  
input: FM7  
output: Im7 in F Major Tonic  
  
input: Em7(♭5)  
output: IIm7(♭5) in D Minor Sub Dominant Minor  
output: IIm7(♭5) in D Minor Related Second 7th  
  
input: A7  
output: V7 in D Minor Dominant  
  
input: Dm7  
output: Im7 in D Minor Tonic  
  
input: Bm7(♭5)  
  
input: E7  
  
input: AM7  
  
input: Gm7  
  
input: C7  
  
input: F
```

図 6.8 実験結果 7

6.2 評価・考察

6.2.1 長調を想定した和音進行に関して

図6.2～図6.5の入力は長調を想定した和音進行である。いずれも[15]による解釈と同じであることから、正しく解析できたのではないかと考えられる。

また、図6.2～図6.5では[11]では考慮していなかったリレイテッド・セカンド・マイナー・セブンス、セカンダリー・ドミナントを和音の機能として認識させることができたことで、ピボットコード以外の和音に対しても複数の解釈を与えることができたのではないかと考えられる。

従って長調に関しては、本研究のシステムにより、[11]等の従来の和声解析システムでは実現できなかった、一つの和音に対して複数の解釈を与えることができたのではないかと考えられる。

6.2.2 短調を想定した和音進行に関して

図6.6と図6.7の入力は短調や長調から短調への転調を想定した和音進行であるが、実験結果よりいずれも解析がうまくいかなかった。以下ではこの原因について考察する。

CCG和声解析部に渡された和音進行は最終的にトニックに解決するような進行を前提としている。しかしながら、図6.7の4番目の和音のようにトニックに解決せずに転調を起こしている曲に関しては解析がうまくいかないのではないかと考えられる。そのため、渡された和音進行の最後の和音がドミナントやサブドミナントの機能をもつような和音進行に関しても解析できるように辞書項目を追加する必要があると考えられる。

図6.6と図6.7の入力におけるTPSによる和声解析の結果をみると、(図6.9、図6.10)ト短調を想定していた和音Gmは、ノンダイアトニック・コードであるのも関わらず変ロ長調であると解釈されている。これは平行調間での調間距離を0としていることが原因ではないかと考えられる。短音階には自然的短音階、和声的短音階、旋律的短音階の三種類が存在し、短調の曲では3つの短音階のうち一種類のみが用いられるることは少なく、三種類の短音階が併用して用いられていることが多い。以上を考えると、長調と平行短調では調性が全く同じであるとは考えにくく、平行調間での調間距離に何らかの値が必要になってくるのではないかと考える。また、和音の五度圏に関しては、図6.12のように三種類の短音階のダイアトニックコードに属する和音を和音の五度圏上で扱う必要があるのではないかと考えられる。

6.2.3 短い転調を繰り返した和音進行に関して

図6.8で扱った和音進行は、特に後半部において3つの和音ごとに転調を繰り返すものを見たが、図6.11よりイ長調をであるはずの部分が転調後のヘ長調として解釈

されている。これにより、8番目から13番目の和音をへ長調だとみなして解析を行ったことでカテゴリーの割り当てに失敗し、解析が上手くいかなかったのではないかと考えられる。また、短い転調を含む和音進行に対してTPSによる和声解析がうまくいかないと思われる。従って、どの程度の長さまでを入力として許すのか明らかにする必要がある。

The screenshot shows a software interface for harmonic analysis. The title bar reads "<終> > Intermediate". The main window displays a list of musical chords in a hierarchical tree structure. The root node is "9 mFlat57 7 m", which has three children: "2 Mm7 7 m", "7 m 10 M", and another "9 mFlat57 7 m" node. This second "9 mFlat57 7 m" node also has three children: "2 Mm7 7 m", "7 m 10 M", and "7 m 10 M".

図 6.9 実験結果 5 の TPS 和声解析部による調推定

The screenshot shows a software interface for harmonic analysis. The title bar reads "<終> > Intermediate". The main window displays a list of musical chords in a hierarchical tree structure. The root node is "0 m7 10 M", which has four children: "5 Mm7 10 M", "10 M7 10 M", "3 M7 10 M", and "9 mFlat57 7 m". The "9 mFlat57 7 m" node has three children: "2 Mm7 7 m", "7 m 10 M", and "7 m 10 M".

図 6.10 実験結果 6 の TPS 和声解析部による調推定

▽ <終了> IntermediateAnnounce

5 M7 5 M
10 M 5 M
0 Mm7 5 M
5 M7 5 M
4 mFlat57 2 m
9 Mm7 2 m
2 m7 2 m
11 mFlat57 5 M
4 Mm7 5 M
9 M7 5 M
7 m7 5 M
0 Mm7 5 M
5 M 5 M

図 6.11 実験結果 7 の TPS 和声解析部による調推定

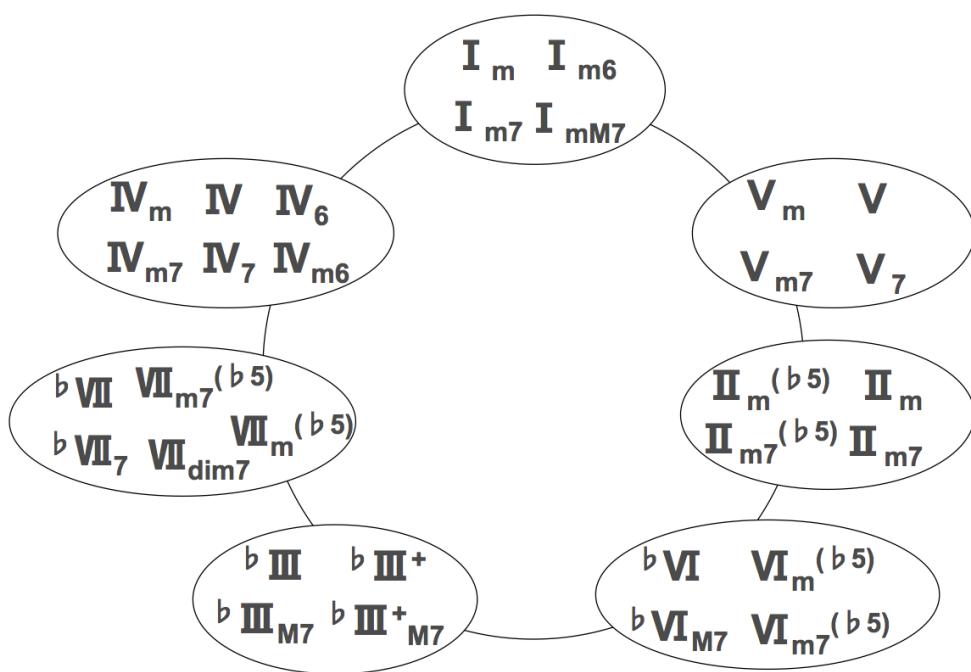


図 6.12 三種類の短音階に対応した和音の五度圏

第7章 おわりに

本研究ではジャズ和声を含む商業音楽に向けた和声解析システムの実装に向けて、二つの先行研究 ([11], [7]) のそれぞれに対して手法の提案を行った。

[11] に関しては入力をコードネームではなく、予め調の情報を含んだディグリーネームを入力とし、先行研究で扱っていなかったサブドミナントマイナーやセカンドリー・ドミナントなどの機能を追加しつつ辞書項目の改訂を行った。また、複数の解釈が可能な一例としてピボットコードを用いた転調があるが、このピボットコードを認識できるようにアルゴリズムの改良を行った。

また、[7] に関しては四和音に対してもダイアトニックコードであるとみなし、四和音に対応した和音の五度圏を長調・短調それぞれに関して捉え直した。

上記の提案手法を取り入れることでシステムの実装を行い、妥当性を評価するためにいくつかのコードネーム表記による和音進行を入力として実験を行った。実験の結果、ピボットコードによる転調を含む長調に関しては正しい解析を行うことができた。また、提案手法により新たに加えた機能に関しても適切な出力を得ることができた。一方で、ピボットコード以外による転調や、複数の短音階をもつ短調に対しては、つねに適切な出力を得ることが難しく、さらなる改良が必要である。

今後の課題としては、メロディ依存の和音進行や、モード・ジャズへの対応が挙げられる。また、多くの楽曲に対しての解析による定量的な評価が必要となってくる。

参考文献

- [1] 後藤真孝 「新しい○○情報学」, 情報処理 51(6), 661-668, 2010-06-15.
- [2] 嵐峨山茂樹: 「和声解析・リズム認識・自動伴奏・運指決定」, 情報処理 50(8), 735-745, 2009-08-15.
- [3] 上田雄, 小野順貴, 嵩峨山茂樹: 「機能和声モデルによる音楽信号からの和声推定」, 情報処理学会研究報告 (IPSJ2010), Vol.2010-MUS-86 No.13, Jul 2010.
- [4] Bernstein, L., "Unanswered Question", Lecture II, Musical Syntax (DVD), Norton Lectures, 1973.
- [5] F. Ledahl, *Tonal Pitch Space*, New York, Oxford University Press, 2001, 411p.
- [6] 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.
- [7] 山口直彦, 菅村昇: 「非調構成音を含む和音への対応を目的とした TPS(Tonal Pitch Space) の拡張 —ジャズ音楽理論への適用を目指して—」, 情報処理学会研究報告 (IPSJ2011), Vol.2011-MUS-89 No.10, Feb 2011.
- [8] Andrew, J.Viterbi, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Transactions on Information Theory 13(2):260-269, April 1967.
- [9] Chomsky, Noam, *Three models for the description of language*, IEEE Transactions on Information Theory 2(3):113-124, Sep 1956.
- [10] Steedman, M., *Surface structure and interpretation*, Vol. 30, The MIT press, Cambridge, MA , 1996.
- [11] Mark Granroth-Wilding, Mark Steedman, *Robust Paser-Interpreter for Jazz Chord Sequence*, Journal of New Music Research, Volume 43, Issue 4, 2014.
- [12] Euler, L., *Tentamen novae theoriae musicae ex certissimis harmoniae principiis dilucide expositae*, Saint Petersberg Academy, 1739.

- [13] 北川祐, 『絶対分かる！コード理論』, Ritter Music, 2014.
- [14] 納浩一, 『ジャズ・スタンダード・セオリー』, Ritter Music, 2014.
- [15] 小山大宣, 『JAZZ THORY WORKSHOP』, 武蔵野音楽学院出版部, 1980.

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 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ます、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東条研究室の皆様に厚く御礼申し上げます。