

修士論文 2020 年度（令和 2 年度）

音楽的期待に基づく
ジャズ和声の漸進的構造解析

東京理科大学 理工学研究科 情報科学専攻

6319508 小椋 裕太

指導教員 桂田浩一 准教授
指導補助教員 大村英史 講師

2021 年 2 月

目 次

第 1 章 序論	1
第 2 章 音楽理論	3
2.1 音楽学の基礎知識	3
2.1.1 音高, 音名, 音程	3
2.1.2 音階 (scale) と調性 (Tonality)	7
2.1.3 コード理論	8
2.1.4 バークリーメソッド	11
2.1.5 ジャズの楽曲ジャンルと楽曲形式	14
2.2 認知的音楽理論	15
2.2.1 生成音楽理論 (Generative Theory of Tonal Music: GTTM)	15
2.2.2 暗意・実現モデル (Implication-Realization Model: IRM)	17
2.2.3 言語理論に基づいた音楽の認知モデル	18
第 3 章 Generative Syntax Model	20
3.1 GSM の概要	20
3.2 ジャズ和声における GSM	20
3.3 Jazz Harmony Treebank	24
第 4 章 漸進的和声構造解析システム ECSA の構築	27
4.1 提案手法と実装システムの概要	27
4.2 確率文脈自由文法に基づく漸進的和声解析	28
4.2.1 漸進的チャート解析のアルゴリズム	28
4.2.2 PCFG への拡張	32
4.3 Ex-based Chord Sequence Analyzer	34
第 5 章 ECSA を用いた楽曲解析実験	36
5.1 実験内容と条件	36
5.2 解析結果と考察	37
5.2.1 コードステップごとの木構造について	37
5.2.2 意外性の検出	43
第 6 章 結論	53

謝辭	55
參考文献	56

図 目 次

2.1 複音程の例	6
2.2 音名と鍵盤、楽譜の関係	6
2.3 全音階の例。左がCメジャースケール、右がCマイナースケール.	7
2.4 五度圏	8
2.5 3度堆積によるコード	9
2.6 4和音の種類	9
2.7 Cメジャースケール上にできるダイアトニックコードとその機能	10
2.8 ジャズスタンダードにおけるトゥー・ファイブ・ワンの例	12
2.9 <i>Stella By Starlight</i> のリードシートの一部	13
2.10 <i>Stella By Starlight</i> のピアノによる演奏例	13
2.11 GTTMによる音楽構造解析の例	16
2.12 ATTAの実行画面	17
2.13 IRMにおける基本パターンと解析例	18
3.1 Birks's Worksの和声構造表現の例	21
3.2 Why Don't You Do Right? の和声構造表現の例	23
3.3 JHTアノテーションの統計情報	25
3.4 JHTアノテーションの統計情報（続き）	26
4.1 漸進的和声解析システム ECSA の入出力関係	28
4.2 完全部分木（左）、不完全部分木（右）	32
4.3 Expectation-based Chord Sequence Analyzerの実行画面	34
4.4 適用確率確認画面	35
5.1 <i>Take the A Train</i> のリードシート（Aパート）.	37
5.2 <i>Take the A Train</i> の木構造解析結果（抜粋）	38
5.3 <i>Autumn Leaves</i> のリードシート（Aパート）.	39
5.4 <i>Autumn Leaves</i> の木構造解析結果（抜粋・前半）	40
5.5 <i>Autumn Leaves</i> の木構造解析結果（抜粋・後半）	41
5.6 <i>Autumn Leaves</i> の木構造解析における根の予測結果数の遷移（上位6カテゴリ） .	42
5.7 <i>Cute</i> のリードシート（後半16小節）	43
5.8 <i>Cute</i> の木構造解析結果（各1位のみ・前半）.	45
5.9 <i>Cute</i> の木構造解析結果（各1位のみ・後半）.	46

5.10	<i>Cute</i> の各コードステップにおける意外性 U の値	47
5.11	<i>lady bird</i> のリードシート	48
5.12	<i>Lady Bird</i> の木構造解析結果 (各 1 位のみ・前半)	49
5.13	<i>Lady Bird</i> の木構造解析結果 (各 1 位のみ・後半)	50
5.14	<i>Lady Bird</i> の木構造解析における根の予測結果数の遷移 (上位 6 カテゴリ)	51
5.15	<i>Lady Bird</i> の各コードステップにおける意外性 U の値	52

表 目 次

2.1 音名	4
2.2 音程の数え方	5
2.3 ジャズスタンダードの楽曲形式による分類	14
4.1 構文解析のための文法と辞書の例	30
4.2 <i>Take the A Train</i> のコード進行に対する漸進的な解析結果	31
4.3 文法規則の確率	33

第1章 序論

音楽とは何か。この問いを巡って、我々人類は古代ギリシアの時代より様々な音楽理論を考案してきた。この音楽理論の発展と共に、多様な作曲表現や音楽ジャンルが生まれてきたことは音楽史の歴史を見れば明らかである。一方で、「我々は音楽をどのように理解しているのか」といった根源的な問い合わせてはいまだに明らかになっていないことが多い。認知的音楽理論は、このような問い合わせてはいまだに明らかになっていないことが多い。認知的音楽理論は、このような問い合わせてはいまだに明らかになっていないことが多い。認知的音楽理論は、このような問い合わせてはいまだに明らかになっていないことが多い。

Rohrmeyerによる *Generative Syntax Model* (GSM) は楽曲の持つ階層構造を分析する認知的音楽理論の一つである。楽曲の持つ階層構造については、Schenker が提唱した、楽曲の聴取者は全てのピッチイベント（音符や和音）を相対的重要度による階層構造として整理しようとするという簡約仮説に端を発し、Lerdahl と Jackendoff による *A Generative Theory of Tonal Music* (GTTM) をはじめとしてこれまで多くの認知的音楽理論研究において着目してきた。その中でも GSM は、チョムスキーの生成文法により忠実に対応した理論であり、和声進行における文脈自由文法を定義することで、和声進行の持つ階層的構造を明らかにしている。

このような認知的音楽理論であるが、音楽認知に関して考慮できていない点がいくつか存在する。その一つが楽曲途中の認知構造の反映である。これまでの認知的音楽理論が対象としてきたのは、楽曲を全て聴き終わった後の認知構造である。またそのような音楽理論を用いたシステムの出力は楽曲全体に対する一意的な解析結果のみであった。一方で、実際に我々が音楽を聴く時は、音楽は時系列に沿って聴かれ、その音の流れに誘発されて様々な情動が我々の中に生じていると考えるのが自然であろう。また、哲学者 Meyer は、音楽の意味は、先行が後続の音を何らかの形で期待 (expectation) させ、具体化された後続によって先行の確認や見直しをはからうとするよう音の関係性から生じるとも述べている。

そこで本研究は、楽曲途中の認知構造について着目する。漸進的チャート解析と呼ばれる構文解析アルゴリズムを GSM に用いることで、これまで議論されて来なかった楽曲途中の階層構造を明らかにする手法を提案する。また、GSM の構文規則を確率文脈自由文法に拡張することで、構文木間の重要度を比較検討する。本研究では、これら提案手法を和声解析システムとして実装し、実際の楽曲を用いて漸進的解析の有用性について議論した。検討の結果、提案手法によって、これまでの理論では扱えなかった解釈の多様性や楽曲における意外性の生じるポイントについて評価することが可能であることがわかった。

本稿の構成は次の通りである。第2章では、まず本研究で必要となる基礎的な音楽的知識について概説し、代表的な認知的音楽理論である GTTM と IR 理論について述べる。第3章では、本研究の文法的な基盤となる Rohrmeyer の GSM について述べる。第4章では、まず本研究の提案

する漸進的構造解析方法について述べ、それを実装した和声解析システムについて述べる。第5章では、第4章のシステムを用いた楽曲解析実験とその結果について述べる。第6章では、本研究のまとめと今後の課題について述べる。

第2章 音楽理論

本章では、まず2.1節で音楽理論に関する用語の曖昧さを排除することを目的とし、本研究の前提となる基本的な音楽の知識について述べる。続く2.2節では、音楽を「聴く側」の理論である認知的音楽理論について解説する。まず言語と音楽の関わりについて2.2.3節で概説したのち、代表的な理論であるGTTMとIR理論についてそれぞれ2.2.1節と2.2.2節で解説する。なお、本節における説明は[1, 2, 3, 4, 5, 6]を参考にしている。

2.1 音楽学の基礎知識

本節では、現在最も普及している12音からなる平均律を前提として、いわゆる「音楽理論」に関する説明を行う。

2.1.1 音高、音名、音程

本節では、単音の呼び方及び2音の関係について述べる。なお、音節では解説中に「音階」という語を用いているが、音階と音名や音程の概念は密接に結びついているため、どちらか一方を独立して解説することは困難である。音階については2.1.2節で解説するが、ここで略説すると音階とは音楽を構成するために用いられる音を、1オクターブの範囲内でその高さの順に並べたものである。

(1) 楽音と噪音、倍音

人の歌声や、ピアノなどの楽器音のように音高が明瞭に感じられる音のことを楽音(musical tone)という。一方で、スネアドラムの音のような音高が明確でない音を噪音(unpitched sound)という。楽音の音高とされる周波数に対し、整数倍の周波数を持つ音を倍音(harmonic overtone)という。弦楽器や管楽器の音を正弦波成分の集合に分解すると、元の音と同じ高さの波の他に、その倍音が多数現れる。この倍音の構成の違いが音色(tone color)として知覚されている。

(2) 音高

音高とは音の高さピッチ(pitch)のことであり、大きさ(loudness)・音色(color)と並んで音の三属性の一つとされている。世間ではマスコミを含めて「音程」という語がしばしば「音高」の

意味で用いられている。例えば、カラオケでうまく歌えない時に「音程をはずした！」ということがある。「音程」(interval)については後述するが、これは本来、2音間の「高さの間隔」ないし「高さの差」を表す単語である。そのため、先ほどの発言は「基準とする音高からの音程をはずした」という表現から「基準とする音高からの」が抜け落ちたものと考えられる。このように「音高」「音程」という語には注意が必要である。

(3) 音名

音の呼び方には絶対的なもの(音名)と相対的なもの(階名)の2種類があり、音名は物理的な音の高さ(音高)に対応する。

音名は表2.1に示すように各国で様々な呼び名が使われるが、いずれにしてもこの表中の7つの音を幹音と呼ぶ。幹音は、主にピアノの白鍵盤に対応する。本稿ではこれ以降、英米式の音名(CDEFGAB)を用いる。そして派生音(主にピアノの黒鍵版に対応)には、幹音の音名に♯や♭を表す言葉を付け加える。

階名は音階中の位置関係によって決定され、ある音階において最も基本となる音(主音)をドとして、音階を構成する7つの音をドから初めて低い順にドレミファソラシ、とイタリア語で呼ぶのが一般的である。改名で歌うことを階名唱法またはドの音高が音名に即して移動するので移動ド唱法とも呼ばれる。これに対して、音名で歌うことを音名唱法または固定ド唱法と呼ぶ。

音名と階名のどちらも、1オクターヴ離れた音には同じ名前が与えられる。オクターヴ(octave)とは2つの音の周波数比が1:2となるような関係のことである。

表2.1: 音名

日本	ハ	ニ	ホ	ヘ	ト	イ	ロ
英・米	C	D	E	F	G	A	B
ドイツ	C	D	E	F	G	A	H
フランス	Ut	R	Mi	Fa	Sol	La	Si
イタリア	Do	Re	Mi	Fa	Sol	La	Si

(4) 音程

2つの音におけるピッチの差を音程(interval)という。12音平均律において、音程の最小単位を半音(semitone)という。

音程は、音階において自身と同じ弦の関係を1度あるいは同度と呼び、その隣の音との関係を2度、2つ離れた音との関係を3度、…のように数え、1オクターヴは8度の音程に相当する。

また、例えば同じ2度であってもCとDの音程は半音2個分(これを全音と呼ぶ)であるのに対し、EとFの音程は半音1個分という違いがあり、これらをそれぞれ長2度、短2度と読ん

で区別する。2度以外の音程についても同様であり、「完全・短・長・減・増」(Perfect · Minor · Major · Diminished · Augmented)などの結合辞を伴って初めて明確な音程になる。1オクターヴ以内の音程の数え方を表2.2にまとめる。なお、各音程について例示している2音は、完全1度を除いて、後者の方が高いものとする。

表2.2: 音程の数え方

音程	半音の数	名称	略称	例
1度	0	完全1度 (perfect unison)	p1	CとC (同じ高さ)
2度	1	短2度 (minor 2nd)	m2	EとF
	2	長2度 (major 2nd)	M2	CとD
3度	3	短3度 (minor 3rd)	m3	DとF
	4	長3度 (major 3rd)	M3	CとE
4度	5	完全4度 (perfect 4th)	p4	CとF
	6	増4度 (augment 4th) ¹	+4	FとB
5度	6	減5度 (diminish 5th) ¹	-5	BとF
	7	完全5度 (perfect 5th)	p5	CとG
6度	8	短6度 (minor 6th)	m6	EとC
	9	長6度 (major 6th)	M6	CとA
7度	10	短7度 (minor 7th)	m7	DとC
	11	長7度 (major 7th)	M7	CとB
8度	12	完全8度 (perfect octave)	p8	CとC (1オクターヴ)

表2.2に示したもの以外にも、1オクターヴを超える9度以上の音程を数えることも可能であり、これを複音程 (compound intervals) と呼ぶ。逆に、1オクターヴ以内の音程のことを単音程 (simple intervals) と呼ぶ。複音程の音程を判断するには、単音程に直して考えればよい。例えば、図2.1左側のC-F♯の音程は11度であるが、これは右側のように1オクターヴ狭くして考えると増4度であることがわかる。そこで、譜例左側の音程を「完全11度」または「1オクターヴと完全4度」のように呼ぶ。後述するジャズのコード理論では、9, 11, 13度がテンションという概念でよく使われるため理解しておく必要がある。

(5) 変位

ある音において半音を単位として上下させることを変位と呼び、変位された音を派生音と呼ぶ。幹音を半音上げた音には嬰記号 (シャープ記号, ♯), 半音下げた音には変記号 (フラット記号, ♭) を付けて、C♯, E♭などのように表記する。これらの派生音に対し、幹音であることを明示する場

¹全音3つ分(半音6つ分)の音程を三全音、トライトーン (tritone) と呼ぶことがある。三全音は、不協和音の中で最も響きの悪い不快なものとされ、「音楽の悪魔」(diabolus in musica) と称されることもあるが、ジャズ和声においては後述するドミナントコードに現れる音程であり、極めて重要な音程であると言える。

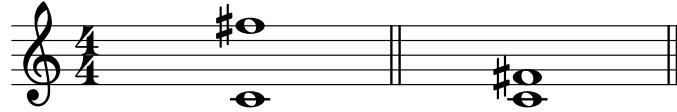


図 2.1: 複音程の例

合にナチュラル記号 (♮) を用いる場合がある。♮は楽譜上で♯や♭の効果を打ち消すために用いられる記号である。

また, C♯とDb, E♯とFなどのように, 音名が異なるが同じ音高を指すものを異名同音 (enharmonic) と呼ぶ。ここで注意しなければならないのは, 音程は幹音における 2 音の関係を基準に数えるということである。例えば, D♯とEb は異名同音であり, C との間の音程はどちらも半音 3 つ分に相当するが, これをそれぞれ増 2 度, 短 3 度と数える。以上の説明で, 12 音平均律におけるすべての音に名前を与えられるようになった。ここで, ピアノの鍵盤および楽譜との対応を図 2.2 に示す。鍵盤の図からわかるように, E と F, および B と C の間には黒鍵盤が無く, これらの音程は短 2 度である。

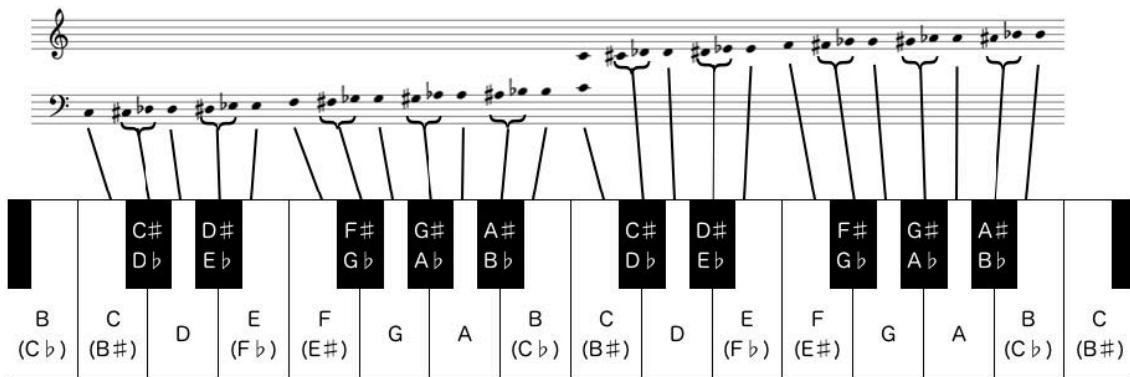


図 2.2: 音名と鍵盤, 楽譜の関係

2.1.2 音階 (scale) と調性 (Tonality)

ある音楽を構成する多くの音を、1オクターヴの範囲内で音高の高低の順に並べたものを音階 (scale) という。音階には様々なものがあり、特にジャズにおいては、中世ヨーロッパにおける教会旋法 (church mode) が即興演奏や作曲のアイデアとして用いられることがあるが、ここでは今後の説明に必要となり、最も基本的である長音階 (major scale) と短音階 (minor scale) について述べる。また、長音階と半音階を合わせて全音階 (diatonic scale) と呼び、音階はある音を基準として、その音から特定の音程だけ上の音の集まりとして構成される。長音階と短音階は図 2.3 に示すように、全音を W, 半音を H で表すようにすると、長音階は出発音から順に WWHWWWH, 短音階は WHWWHWW という音程の並びを持つ。出発音は 12 個の音のうちいずれを選んでもよいため、長音階や短音階は 12 種類ずつ存在する。音楽を構成するメロディやハーモニーがある一つの音 (主音 (tonic)) を中心として構成される時、その音楽は調性 (tonality)¹を持つといい、調性を持った音組織を調 (key, キー) と呼ぶ。長音階は長調 (major key, メジャーキー) を、短音階は短調 (minor key, マイナーキー) を、それぞれその音階の最低音を主音として形成する。

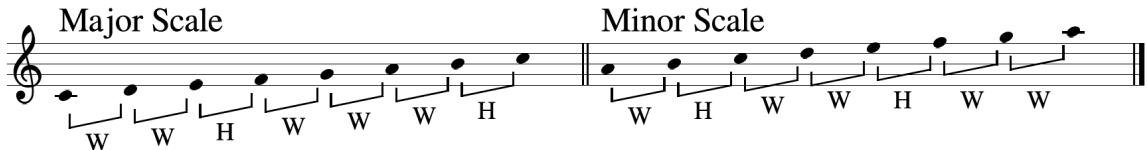


図 2.3: 全音階の例。左が C メジャースケール、右が C マイナースケール。

調号 (key signature) とは、五線譜上の音部記号 (ト音記号やヘ音記号) のすぐ右に置かれて調を指定する嬰記号 (#) や変記号 (b) の一群のことである。平行調 (relative key) とは同じ調号をもつ長調と短調を指し、同名調、同主調 (parallel key) とは主音を同じくする長調と短調を指す。属調 (dominant key) とはある調の完全 5 度上の調を、下属調 (sub-dominant key) とはある調の完全 4 度上の調を指す (いずれも長短に関しては保存される)。近親調 (relative key) とは平行調、同主調、属調、下属調およびそれらの組み合わせの調を指す。

五度圏 (circle of fifths) とは、それぞれの調性を属調 (完全 5 度上)・下属調 (完全 4 度上) を左右に配置し、結果として 12 の長調・12 の短調を円周となるように配置したものである (図 2.4)。

¹調性という言葉は、広義においては支配的な中心音を有する音体系を、狭義においては、長短調いずれかの主和音を持つ和声的な音体系をさす。ここでいう調性とは後者の旋法性を含まない概念のことという [3]。

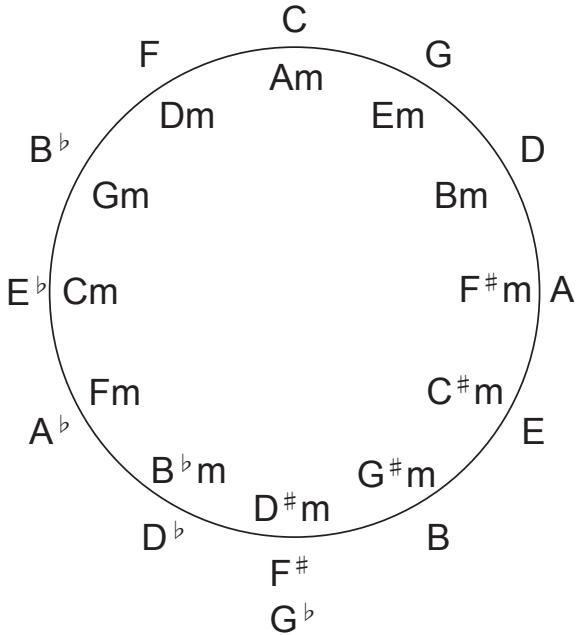


図 2.4: 五度圏

2.1.3 コード理論

和音 (chord) は、ピッチの異なる複数の音が同時に鳴ることで生み出される音響である。このようにして構成された和音を継続的に連結することを和声 (harmony) という。和音や和声に関する音楽理論には流派のようなものがあり、クラシックの和声法と、ジャズやポピュラー音楽で使われるバークリーメソッド (Berklee method) では、和音の表記法など様々な点で差異がみられる。本研究ではジャズを対象とするため、バークリーメソッドにおける和音の表記について述べる。

(1) 3度堆積による和音

ジャズで基本となるコードの構成は、図 2.5 に示すように、ある音を基準として、上に 3 度 (長 3 度でも短 3 度でもよい) ずつ音を積み上げたものであり、各音は下から順に根音 (Root), 第 3 音 (3rd), 第 5 音 (5th), 第 7 音 (7th), 第 9 音 (9th), 第 11 音 (11th), 第 13 音 (13th) という。(8 個目で元の音と同じ音、2 オクターヴ上の C となる)

理論的には根音から第 13 音までの全部の音を構成する和音を考えられうるが、ジャズでは 3 つの音による和音 (Triad) や 4 つの音による和音 (Tetrad, 4 Note Chord) を主に使用するため、Root, 3rd, 5th, 7th をコードの構成音 (Chord Tone) といい、それ以外の音は非構成音 (Non Chord Tone) という。

また、コードの構成音を音符ではなく、アルファベットと、それに付した記号や数字で表したもの、コードネーム (Chord Symbol) という。コードネームとは、そのコードが何の音を根音として作られたかを A, B, C... 等の音名で示し、Chord の性格 (Chord Type) を M7, min7, 7,

dim 等の記号で示したものである。コードネームやその略記の仕方にはいくつかの流派があり、本稿で述べる記法以外の方法も用いられることがある。図 2.6 には、C を根音とした時の 4 和音について記す。

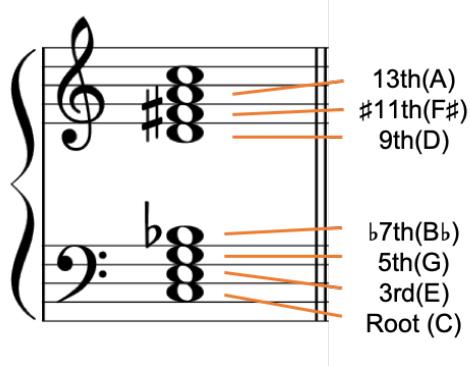


図 2.5: 3 度堆積によるコード

		構成音	名称	Chord Name	備考
M J A O R 系	$\begin{pmatrix} 7 \\ 5 \\ 3 \\ 1 \end{pmatrix}$		(シ-) メジャーセブンス	C7, C△7 CMaj7, CMA7	Major 7th Chordと同様に扱われるChordとしてMajor 6th Chordがある。 C6
	$\begin{pmatrix} b7 \\ 5 \\ 3 \\ 1 \end{pmatrix}$		(シ-) ドミナントセブンス 又は (シ-)セブンス	C7	
M I N O R 系	$\begin{pmatrix} 7 \\ 5 \\ b3 \\ 1 \end{pmatrix}$		(シ-)マイナーメジャーセブンス	C-M7, Cm△7 CminMaj7	minor KeyのTonicにはむしろminor 6th Chordの方がよく使用される。 Cm6
	$\begin{pmatrix} b7 \\ 5 \\ b3 \\ 1 \end{pmatrix}$		(シ-)マイナーセブンス	C-7, Cm7 Cmin7	
	$\begin{pmatrix} b7 \\ b5 \\ b3 \\ 1 \end{pmatrix}$		(シ-)マイナーセブンス フラットファイブ 又は (シ-)ハーフディミニッシュ	C-7(b5), Cm7(b5) Cmin7(b5), Cm7(-5), Cφ	
Major系にもminor系にも属さない	$\begin{pmatrix} b \\ b7 \\ b5 \\ b3 \\ 1 \end{pmatrix}$		(シ-)ディミニッシュセブンス	Cdim7, Co7 Co (本来はDiminished Triadを表す記号だがジャズではDiminished 7thとしても使う)	

図 2.6: 4 和音の種類

またメジャーコードの第 5 音が半音上がった Augment(aug) コードや第 3 音が半音上がって完全 4 度の音程に変化した Suspended 4th (sus4) といった特殊なコードも存在する。

(2) コードの機能 (Function)

今、図 2.7 のように、ある全音階上 (ここでは C Major scale 上) に、その音階上の音のみを使用して 3 度構成による和音を作る (ここでは 4 音による和音を作る). 楽譜の上部には、それぞれの和音のコードネームを、下部にはディグリーネームを表している. ディグリーネーム (Degree Name) とは、そのコードが全音階上の第何番目の音を根音として作られた和音であるかを I, II, III, ... 等のローマ数字で示し、Chord Type を記号で示すものである. これらの和音はその音階を構成する調に対して関連を持つ. その意味から、ダイアトニックコード (Diatonic Chord) と呼ぶ. それに対してダイアトニックコード以外の和音をノンダイアトニックコード (Non Diatonic Chord) と呼ぶ. ダイアトニックコードの各和音は、それぞれ機能を持つとされ、トニック (Tonic, 以下 T), ドミナント (Dominant, 以下 D), サブドミナント (Sub-Dominant, 以下 S) に大別できる. 長音階上の各和音が持つ機能を、図 2.7 のディグリーネームの下に記す.

トニックは主和音とも呼ばれ、その調の主となる和音である. ドミナントは属和音とも呼ばれ、トニックを導く和音である. ドミナントの和音からトニックの和音に進行することを解決 (Resolution) するという. サブドミナントは下属和音とも呼ばれ、トニックとドミナントの中間程度の緊張度を持つ. これらは、後述するコード進行を形作る基本的な考え方である. また同様にして短調に対してもダイアトニックコードを考えることができ、それぞれ機能を持つとされる. 短調の場合は、IV の和音がマイナーコードになることがある. これをサブドミナント・マイナー (Sub-Dominant minor, 以下 SM) であるという. サブドミナント・マイナーの和音には、必ず音階上の \flat vi の音 (ハ短調では \flat ラ) が含まれる.

A musical staff in G clef. Above the staff, seven chords are listed: CM7, Dm7, Em7, FM7, G7, Am7, and Bm7(\flat 5). Below the staff, seven boxes are labeled: IM7, IIIm7, IIIIm7, IVM7, V7, VIIm7, and VIIIm7(\flat 5). Below each box is a letter: T, S, T, S, D, T, D. This indicates that CM7 is the tonic (T), Dm7 is the subdominant (S), Em7 is the tonic (T), FM7 is the subdominant (S), G7 is the dominant (D), Am7 is the tonic (T), and Bm7(\flat 5) is the dominant (D).

図 2.7: C メジャースケール上にできるダイアトニックコードとその機能

(3) コード進行 (Chord Progression)

和音の連なりのことをコード進行 (Chord progression) という. 和音は孤立して存在するものではなく、前後の流れの中で、お互いが関連を持って存在している. 前後関係には一定の法則があり、全体として音楽的意味を持つとされる.

コード進行の基本となる原理としてカデンツ (Cadence) の原理がある. これは前述した和音の機能であるトニック、ドミナント、サブドミナントの定型的な組み合わせを指すもので以下の 3 種類があり、これらを連結することによってより大きなコード進行が形成される.

1. $T \rightarrow D \rightarrow T$
2. $T \rightarrow S \rightarrow T$
3. $T \rightarrow S \rightarrow D \rightarrow T$

しかしながら、実際の楽曲にコード進行はこの原理に基づかないものもあり、その解釈の方法も様々であるため統一された理論は存在しない。

また、本稿ではコード進行とほぼ同義の語としてコード系列 (Chord Sequence) という語も用いる。

(4) トゥー・ファイブ・ワン (two-five-one)

ジャズ楽曲において最も頻出するコード進行としてトゥー・ファイブ・ワン (two-five-one, II-V-I) と呼ばれるものがある。これは文字通り、ディグリーネームにおける II と V の和音からなるコード進行である。C メジャーキーにおけるトゥー・ファイブ・ワンは Dm7 G7 CM7 である。図 2.8 にジャズスタンダード楽曲の一片を示す。図 2.8a は Victor Young 作曲の *Stella By Starlight* (5~7 小節目) であり、キー Eb メジャーの II-V-I である。図 2.8b は Miles Davis 作曲の *Tune Up* (1~4 小節目) であり、D メジャーキーの II-V-I である。図 2.8c は John Coltrane 作曲の *Giant Steps* (10~13 小節目) であり、初めの 2 小節は G メジャーキーの II-V-I、後の 2 小節は B メジャーキーの II-V-I である。このように多くのジャズスタンダード楽曲のコード進行には複数のキーの II-V-I が出現する。

また、マイナーキーにおける II-V も存在する。マイナーキーの場合は II のコードがハーフディミニッシュコードとなって、C マイナーキーの場合は, Ddim7 G7 CM7 となる。こちらも様々なジャズスタンダードにおいて頻繁に見られるコード進行である。

2.1.4 バークリーメソッド

バークリーメソッドは元々バークリー音楽院で教えられていた音楽の理論である。バークリーメソッドは、主として和声進行と旋律の関係を整理・体系化した理論で、記号的に音楽を表現し機械的に操作する方法論を提供している。バークリーメソッド最大の功績は、特に機能和声と称される和声進行の記号化を徹底的に押し進め、その記号レベルでの編曲技法を確立したことにある。それ以前の和声論や旋律論は、音楽家のための実践的なヒューリスティクスの集まりであり、体系化は不十分だった、一方、バークリーメソッドは、和声論をまるでゲームのルールのように記述し、和声進行の創作をパズル遊びのように変えてしまった。²

バークリーメソッドと偶然同時期にニューヨークで誕生したビバップというジャズ音楽は、作曲や演奏における構成性や言語記述性が高く曖昧な部分が少ないと特徴をもち、まるで競技規則に従って闘うスポーツのように即興演奏 (インプロビゼーション) を繰り広げるスタイルを持つ

²和声理論のパズルゲーム化と呼ばれる [7]

Musical score for 'Stella By Starlight' (Measures 5~7). The key signature is A major (no sharps or flats). The melody consists of eighth-note patterns. The chords labeled are Fm7, B♭7, and E♭Δ.

(a) *Stella By Starlight*(5~7 小節目)

Musical score for 'Tune Up' (Measures 1~4). The key signature changes between A major (no sharps or flats), D major (one sharp), and G major (two sharps). The melody consists of quarter notes and eighth-note pairs. The chords labeled are Em7, A7, and DΔ.

(b) *Tune Up*(1~4 小節目)

Musical score for 'Giant Steps' (Measures 10~13). The key signature changes frequently between A minor (no sharps or flats), D major (one sharp), G major (two sharps), C♯m7 (one sharp), F♯7 (two sharps), and BΔ (no sharps or flats). The melody consists of eighth-note patterns. The chords labeled are Am7, D7, GΔ, C♯m7, F♯7, and BΔ.

(c) *Giant Steps*(10~13 小節目)

図 2.8: ジャズスタンダードにおけるトゥー・ファイブ・ワンの例

ていた。ビバップとバークリーメソッドは元から相性がよかつたが、バークリーメソッドはビバップを分析するためにさらに発展し、ビバップはバークリーメソッドの応用によってさらに理論性と表現力を増していったという歴史がある。

バークリーメソッド式の譜面例を図 2.9 に示す。バークリーメソッドの標準的な譜面構成では、1 段の五線譜に単旋律が書かれていて、その五線譜のすぐ上に和音名の列（和声進行）が書かれている。バークリーメソッドはあらゆる楽曲をこの書式で記述し、演奏者の間では一般にリードシートと呼ばれる。リードシートの記述コストは、一般的なクラシック音楽の譜面に見られるような、ピアノ曲であれば 2 段譜に、オーケストラの楽曲であれば、多段組の五線譜みに演奏すべき音が全て記入されているのに比べて、大幅に低減されている。これは、ジャズという音楽がコード進行に基づいた即興演奏を中心に発展してきた音楽であることに起因するといえるであろう。実際にジャズミュージシャンは、先ほどのリードシートを元にしてメロディのリズムを変えたり、バックキング³を付け加えてたりして図 2.10 に示すような演奏を行っている。東条らはこれを、著書 ([5], p101) で表現力と記述コストのトレードオフとよび、バークリーメソッドの発展について論じている。



図 2.9: *Stella By Starlight* のリードシートの一部



図 2.10: *Stella By Starlight* のピアノによる演奏例

³コードによる伴奏のこと。コンピングともいう

2.1.5 ジャズの楽曲ジャンルと楽曲形式

ジャズは20世紀初頭に生まれ発展してきたが、その歴史において、ジャズミュージシャンたちの自由で流動的な音楽性を探求する前衛的な思想が新たな演奏スタイルやサブジャンルを常に生み出してきた。それぞれのサブジャンルの名称に明確な定義や区別があるわけではないが、有名なジャンルとして、Swing, Jazz Blues, Bebop, Cool Jazz, Hard Bop, Bossa Nova と言ったもののがあげられる。また2.1.4でも述べたように、ジャズの歴史は音楽理論の発達の歴史でもあるが、1960年代頃には従来の機能和声理論から脱却した音楽ジャンルとして Modal Jazz や Free Jazz と呼ばれる楽曲もみられる様になった。本研究や3章で述べるGSMでは、Swing, Jazz Blues, Bebop, Cool Jazz, Hard Bop, Bossa Nova といった一般的に機能和声理論において解釈できるジャンルの楽曲を扱っている。

ジャズの楽曲形式には様々なものがあるが、通常12小節の長さのブルースを除いて、多くのジャズ楽曲は8小節単位のフレーズで構成されていることが多い。例えば32小節AABA形式の楽曲といえば、

- 初めの8小節(A)
- 初めの8小節と同じメロディ・コード進行を持つ8小節(A)
- 続いて異なるメロディ・コード進行を持つ8小節(B)
- 冒頭と同様のメロディ・コード進行を持つ8小節(A)

とういう構成を持つ楽曲のことをいう。これらの楽曲形式とコード進行は密接に関係しており、楽曲の和声構造を解析する際も考慮する必要がある。参考として表2.3にジャズスタンダードをその楽曲形式によって分類したものを見よ[6]。

表2.3: ジャズスタンダードの楽曲形式による分類

楽曲形式	楽曲名(作曲者)
AABA	Daahoud(Clifford Brown), Satin Doll(Duke Ellington), Take The A Train(Billy Strayhorn)
ABAC	If I Were A Bell(Franck Loesser), There Will Never Be Another You(Harry Warren), Someday My Prince Will Come(Frank Churchill)
AABC	The Song Is You(Jerome Kern), I Concentrate On You(Cole Porter)
AB	Tune Up(Miles Davis), Lady Bird(Tadd Dameron), Giant Steps(John Coltrane)

2.2 認知的音楽理論

認知的音楽理論とは、伝統的音楽理論（和声学、対位法、楽式論等）が音楽を「作る側」の理論であるのに対し、「聴く側」の認知過程を踏まえた音楽分析の理論をいう。以下では、まず、その代表的な理論とされる GTTM と IR 理論について説明した後、そのほかの音楽の計算論的認知モデルについて説明する。

2.2.1 生成音楽理論 (Generative Theory of Tonal Music: GTTM)

A Generative Theory of Tonal Music (GTTM) はラーダールとジャッケンドフによる音楽構造の理論で、著書のタイトルがそのまま名称および略称となっている [8]。これはオーストリアの音楽学者ハインリッヒ・シェンカーの提唱した、楽曲はより簡単な構造に簡約できるという簡約仮説 (reduction hypothesis) に基づいた理論となっている。簡約とは、段階的に重要な音（ピッチイベント）を残し重要でない音を削除し、楽曲の骨組みに相当する構造⁴を抽出する操作である⁵。なお、簡約の逆のプロセスは精緻化 (elaboration) と呼ばれる。楽曲には、そもそも、それを支える潜在的に重要なピッチイベントが存在し、その重要なピッチイベントを出発点として、アルペジオや線形進行といった音を増やす操作（精緻化）を経ることによって実際の楽曲として現れてくるということである。

GTTM は楽曲の階層的な構造を総合的に分析・抽出することを目的とし、以下の 4 つの部分構造（および対応する部分理論）から構成される。

- グルーピング構造
- 拍節構造
- タイムスパン簡約
- 延長的簡約

グルーピング構造は、動機や楽節など音楽的にまとまった単位（グループ）を、包含関係（小さなグループは大きなグループに含まれる等）による階層的構造として表す。拍節構造は、拍・小節など拍の強弱に基づく時間的な区切りを階層的に表す。タイムスパン簡約は、隣接するグループ間の構造的重要度で主従を決定し、楽曲形式に相当する階層構造を得る。延長的簡約は主として和声に基づく緊張・弛緩関係で主従を決定し、局所的なグループ階層を超えて「持続（延長）」される階層構造を表す。タイムスパン簡約と延長的簡約のいずれにおいても従属する音を消去し、代表となる音（や和音）を残すと簡約した音列が得られる。

図 2.11 に GTTM による楽曲解析の例を示す。中央にある楽譜の直下、黒点は拍節構造を表し、点の数が縦に多いほど強い拍となる。その下にある横カッコはグルーピング構造を表す。カッコが複数段に別れていることからわかるように、ある楽曲はそれ全体を最大レベルのグループとして、階層的により小さいグループに分割されていることがわかる。また (A) と (B) のように、グ

⁴ 音楽を支える潜在的な構造 (underlying structure) とも言う。

⁵ 音楽理論において、簡約は階層 (hierarchy) と同義に使われることが多い [9]。情報学や数理論理学における簡約とは、項をより単純な形に書き換える操作のことである。

ルーピングは何通りか考えられることがある。楽譜上部に記されているものがタイムスパン木と呼ばれるもので、まっすぐ伸びた線は主たる枝を、行き止まりの線は従属する枝を表す。

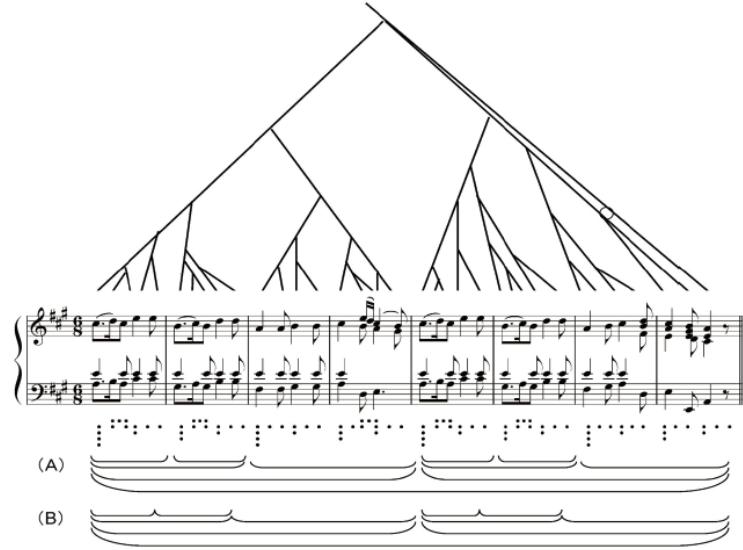


図 2.11: GTTM による音楽構造解析の例 [10]

GTTM はルールシステムとして定式化され、4 つの部分構造のそれぞれが、構成規則 (well-formedness rule) と選好規則 (preference rule) の2種類の規則で構成される。構成規則はその名の通りグループなり拍節なりを構成する上で必ず満たすべき要件を示し、選好規則は一般的な傾向を記述した蓋然性を含む規則である。

選好規則は適用しうる場合を述べただけで、規則どうしが競合する場合も多い。その場合には優先度をつけて適用する規則を取捨選択する必要がある。また低次レベルの規則はゲシュタルト原理⁶などに依拠していて、局所的・形式的に適用しうる一方、高次レベルの規則は「類似したメッセージには類似した構造を割り当てる」「和声的に安定したノードを選ぶ」といった抽象的なものが多い。このような選好規則の存在により一般に、一つの楽曲には一つ以上の木構造（意味や意図）が対応する一方で、この点に起因する曖昧さが GTTM に基づく分析の計算機実装を難しくしている要因でもある。

この GTTM を計算機に実装する試みとして浜中らの exGTTM [11] があげられる。exGTTM は、計算機実装する上で曖昧な部分の残る GTTM を数学的に再定義し、選好規則に対して調節可能な優先度を割り当てることで半自動的な楽曲解析を可能にした理論である。浜中らは [11]において、exGTTM を提案すると共に、これを実装した自動解析システム ATTA(Automatic Time-span Analyzer)についても述べている。ATTa は、楽譜を表すフォーマットである MusicXML が入力として与えられ、解析の結果得られたグルーピング構造、拍節構造、タイムスパン木をそれぞれ GroupingXML, MerticalXML, Time-spanXML という独自の形式で出力する。また、ユー

⁶個々の構成要素（音楽でいうと1つのピッチイベント）には意味を見出せないがそれらを統合したときに初めて何か別の意味が現れ、認識される心理・知覚現象のこと。全体が部分の総和以上の意味を持つような心理・知覚現象とも換言できる。

ザが楽譜を入力し、各選好規則の優先度を設定すると、それに従った解析結果を得ることができる(図 2.12).

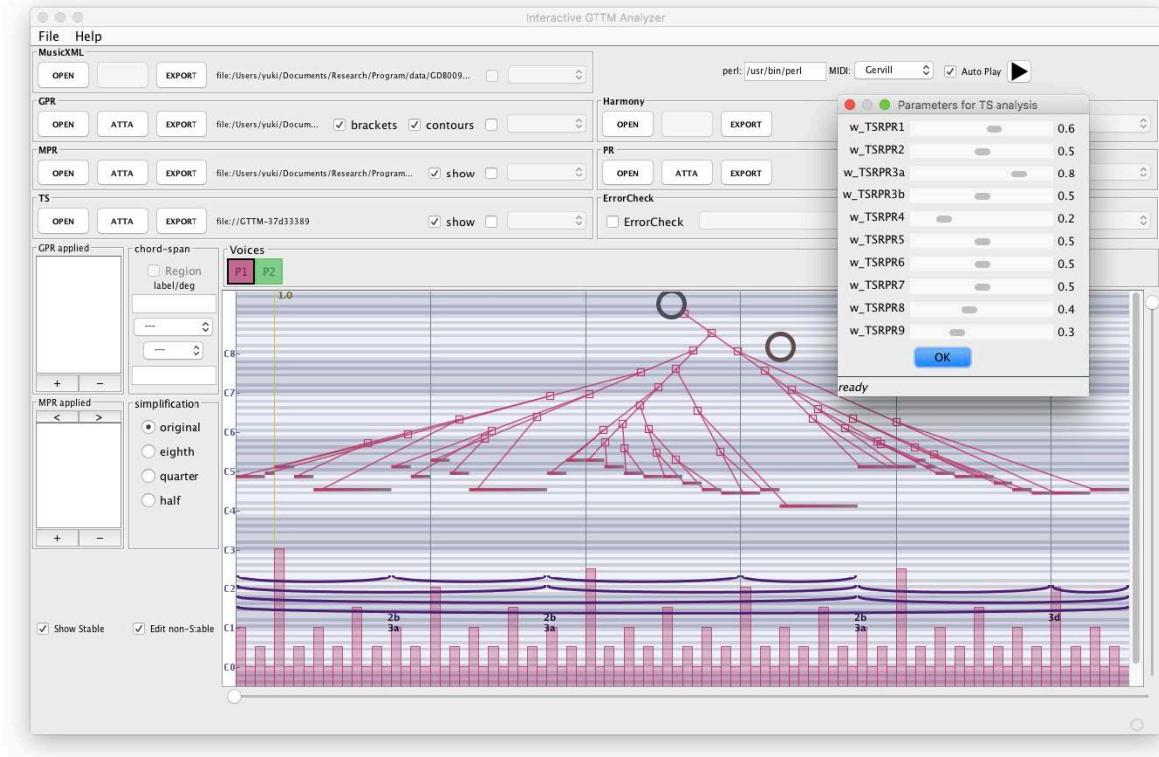


図 2.12: ATTA の実行画面

2.2.2 暗意・実現モデル (Implication-Realization Model: IRM)

音楽は現在の進行から将来何が起きるかを暗意し、人間は音楽を単に受動的に聴くのではなく、その暗意に基づいて(無意識化にせよ)将来の進行を予測ないし期待する。予測通りであれば「解決」として実現される一方、予測が否定されたり、保留されたりすると「未解決」な状態が実現されることになる。いずれの場合にも聞き手の認識や情動が生じて、解決であれば安定感や終始感が、未解決であれば不安定感や緊張、持続感が感じられる。これが「暗意・実現 (implication-realization)」の基本的な考え方である。

暗意・実現の概念はマイヤーの発案によるが、ナームア (Narmour) はそれを大規模に発展・改変し、体系的・形式的な理論の構築を目指した。それを暗意・実現モデル (Implication-Realization Model: IRM) と呼ぶ [12]。IRM ではまず、2 音の音程を質的に分類し(方向: 上がる・下がる・等しい、音程の幅: 大きい・小さい・等しい) 図 2.13 上部に示すような 3 音の組みを基本としたパターンを構成する。基本的なパターンは 16 個あり、他に補助的なパターンやきごうも用いる。これらのパターンを基本として旋律に適用し、得られたパターン列を用いて旋律の構造的特徴を記述・分類し、また聴き手にもたらす効果を分析しようというのが IRM の骨子である。GTTM

が楽曲全体の構造を記述しようという、静的な認知構造の理論であるのに対し、IRMは曲の時間進行に従って生じる動的な認知プロセスの理論であるといえる。

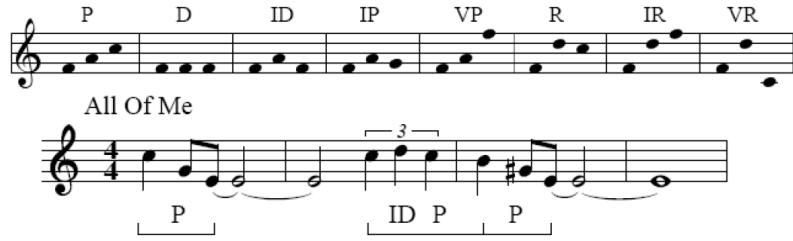


図 2.13: IRM における基本パターンと解析例

2.2.3 言語理論に基づいた音楽の認知モデル

音楽の認知モデルとは、音楽の主として聴取における認知過程を何らかの形でモデル化したものをいう。以下ではそれをコンピュータ上に実働モデル構築しようという研究アプローチについて述べる。音楽の認知モデルと、音楽そのものを解析しようという音楽解析システムとの違いはあまり明確ではない。認知過程を意識し、それを取り込もうとする度合いが高いものが認知モデルといえるが、そうでない解析システムの場合でも、認知過程について様々な知見や示唆が得られる場合も多い。本節でも両者を区別せずに扱う。

また、認知モデルが対象とする音楽の側面は様々である。音楽知覚レベルを対象とするものから、高次の音楽構造を対象とするものまで多岐にわたる。例えば、調性認識、リズム認識、和声解析、メロディ抽出といったような観点から音楽認知モデルや解析システムの研究が行われている。

初期の研究には和声解析システムがある。ウィノグラード [13] は以下のようないくつかの規則についての文脈自由規則

$$\begin{aligned}
 cadence &\rightarrow \text{plagal} \mid \text{authentic} \\
 \text{authentic} &\rightarrow \text{dominant tonic} \\
 \text{dominant} &\rightarrow V \\
 V &\rightarrow V_{\text{seventh}} \mid V_{\text{triad}} \\
 V_{\text{seventh}} &\rightarrow V_7 \mid V_5^6
 \end{aligned}$$

を定め、これを用いてコラール曲に対して転調や借用和音の分析を行うシステムを実装した。

東条らは主辞駆動句構造文法 (Head-driven Phrase Structure Grammar; HPSG) を用いて和声進行解析する手法を提案した [14]。また組み合わせ範疇文法を用いた楽曲解析の手法もいくつか提案されている。グラノロスら [15] はCCGを用いてジャズのコード進行に対し和声解析をするシステムを考案している。また福成らは調性認識に関して Tonal Pitch Space と呼ばれる音楽理論を用いてこれを改良したシステムを提案している [16]。

漸進的に楽曲を解析するというアイデアに近いものとしては、東条による様相論理を用いた Generative Syntax Model への意味づけがある [17]. 様相論理 (modal logic) とは以下に示す様相オペレータまたはモーダルオペレータ (modal operator) と呼ばれる記号を含む命題に関する論理のことをいう [18].

\Box : アクセス可能な全ての世界で

\Diamond : アクセス可能のある世界で

ここでいう世界とは、可能世界意味論における用語である。ピッチイベントが様々に遷移することによって音楽は成り立っているが、ピッチイベントが多種多様に遷移する可能性、その一つ一つをあり得たかもしれない世界であると可能世界意味論では考える。アクセス可能とはある世界を基準として、そこから他の世界の様子が観察可能である状態のことをいう。例えば $\Diamond \text{dom}(x)$ とは近い将来にドミナントの機能を有するコードが期待されることを意味する。東条は GSM の各規則についてこのような様相論理を用いた意味論を構成し、コード（ピッチイベント）の入力に伴って漸進的に楽曲の解釈が行われる様子を示した。

第3章 Generative Syntax Model

生成的統語モデル (Generative Syntax Model; GSM) は Martin Rohrmeier によって提案されている認知的音楽理論である。これは、楽曲の和声進行に関してチョムスキイ的な階層的木構造を生成するモデルまたはその一連の研究の名称である。本節では、本研究における和声解析の土台となる基本的な部分について解説する。

3.1 GSM の概要

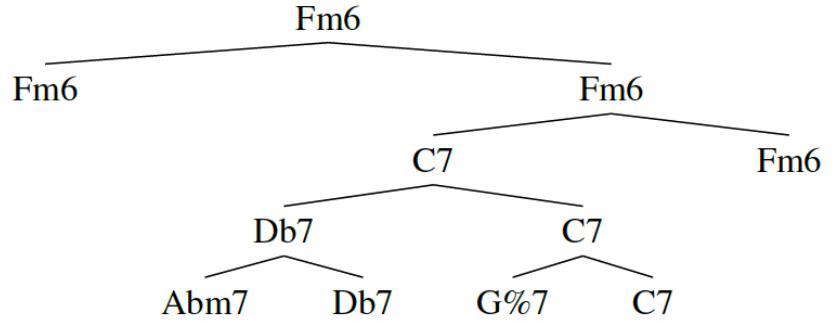
GSM の初出は Rohrmeier による *Towards a Generative Syntax of Tonal Harmony*[19] である。GSM も 2.2.1 節で説明した GTTM と同じく、階層的木構造を生成するモデルである。しかし、GTTM が明示的な文脈自由規則を提示せずに議論を進めているのに対して、GSM はチョムスキイの生成文法理論 [20, 21, 22] に強く基づいており、明示的な和声進行に関する規則を提示しているところに違いがある。

GSM では、一つの和音は前後の和音と依存関係を持ち、特に隣接する和音とはどちらがどちらを支配するかの関係において「機能的ヘッド」（二つの和音のうち、支配的な和音が上位カテゴリーとなる機能）を持つという仮定のもと議論が進められる。GSM で提示されている句構造規則は、その対象とする音楽や考慮する音楽構造の違いによって隨時修正がなされているが、以下の節では、本研究の対象とするジャズ楽曲のため提案された構文規則 [23, 24] を中心に説明を行う。

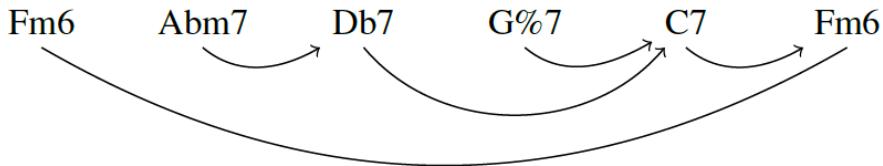
3.2 ジャズ和声における GSM

GSM による楽曲の木構造は楽曲聴取者の認知構造を示しているとされ、楽曲の骨組みがどうであるか、またそこから、どのように精緻化されて実際のコード進行となつたかが表現されている。ジャズ楽曲における精緻化とは、同じ機能を持つコードを使用してコードを延長 (prolongation) したり、あるコードに向かう準備 (preparation) 的な機能を持つドミナントコードを付加したりすることであるといえる。木構造分析はその性質から、1 回目の聴取における認知構造を表すというより、聴取者が意識的または無意識的に認知している楽曲の和声構造についてのより抽象的な表現であるといえる。

GSM で提示される全ての構文規則は、**Prolongation**(延長的) 原則と **Preparation**(準備的) 原則のどちらかに従うとされる。以下では図 3.1a で示されている Birk's Works というジャズス



(a) GSM による和声構文木



(b) 期待-実現の構造

図 3.1: Birks's Works の和声構造表現の例 [24]

タンダードのコード進行 ($Fm6\ Abm7\ Db7\ G\%7\ C7\ Fm6$)¹に対する GSM での解析を例として, Prolongation 原則と Preparation 原則について説明を行う.

まず, Birks's Works のコード進行において, 初めの $Fm6$ はトニックとして認知され, この楽曲は $Fm6$ で終了することが期待される. 続く $Abm7$ と $Db7$ はそれぞれサブドミナントの代理コードと $C7$ のドミナントコード ($\flat II 7$) と考えることができ, 時間的に離れた位置にある $C7$ を期待するといえる. また $G\%7$ は F minor key におけるサブドミナントコードと考えることができ, 続く $C7$, $Fm6$ と共にマイナーのトゥー・ファイブを形成する. 以上のような和声進行における期待 - 実現の構造を図示すると, 図 3.1b のようになる. 図において矢印で表現されているコードのペアは前者のコードが後者のコードの準備的な機能をしているという意味で Preparation 原則に基づくものとされ, 矢印のついていない $Fm6 - Fm6$ のようなペアは Prolongation 原則に基づくものとされる. これは図 3.1a の木構造と一対一対応しているため, 和声進行の木構造は和声進行における期待 - 実現 (expectation - realization) の構造を明らかにしているといえる.

また, Prolongation 原則には強延長 (Strong Prolongation) と弱延長 (Weak Prolongation) の 2 種類がある.² 強延長とは, 同じルートの同じコードタイプを持つ延長のことをいい, 弱延長とは同じ機能を持つコードの延長のことをいう (e.g., C メジャーキーにおける C と Am による延長).

強延長は任意のコードシンボル X に対して $X \rightarrow X\ X$ というふうに表される (e.g., $Fm6 \rightarrow$

¹後述する Jazz Harmony Treebank での表記法にならって, \flat 記号はアルファベット小文字 b を用いて表記する. また, \sharp のつくコードは異名同音の \flat のつくコードに変換して表記する. %はハーフディミニッシュを表す.

²これらは GTTM[8] の延長的簡約というアイデアに基づく規則であるが, ここではジャズ和声に対応するために, より一般化されたものとなっている.

Fm6 Fm6). 弱延長は機能的に等しいコードシンボル X, Y に関して $X \rightarrow Y X$ または $X \rightarrow X Y$ の形で表される (e.g., Fm6 → Ab Fm6). Preparation 原則を満たす規則は、機能的に等しくないコードシンボル X, Y に関して $X \rightarrow Y X$ の形で表される (e.g., Fm6 → C7 Fm6). Prolongation 原則と Preparation 原則についてまとめると以下のようになる.

強延長 $X \rightarrow X X$ (任意のコード X)

弱延長 $X \rightarrow Y X | X Y$ (任意のコード X と機能的に同等なコード Y)

準備 $X \rightarrow Y X$ (任意のコード X と X に解決するコード Y)

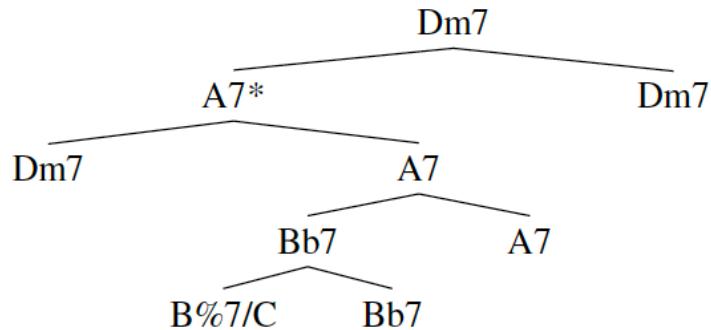
これらの構文規則は、非終端記号に独自のカテゴリを持たず、左辺の記号が必ず右辺に現れるような二分木の形で表現されるという特徴がある。この特徴は依存文法 (dependency grammar)[25] や組み合わせカタゴリー文法 (Combinatorial Category Grammar; CCG)[26] といった文法理論にもみることができる。矢印の両辺に現れる記号はヘッド (head, 主辞) と呼ばれ、Prolongation 原則・Preparation 原則の元では、延長される側 (または、準備される側) のコードがヘッドとなる。つまり、弱延長原則においては子のコードシンボルのうち左右いずれもがヘッドとなりうるが、準備原則においては右側の子のみがヘッドとなりうる。

また、楽曲のリズム構造やラメントバス³を考慮した Open Constituents という木構造の表現方法も提案されている (図 3.2a) が、こちらは図 3.2b で示すようなその楽曲の和声進行における期待 - 実現 (expectation - realization) の構造と一対一の対応をするものではない。しかし、Open Constituents の木構造にはあるノードにアスタリスク (*) が付加されており、このアスタリスクを基準として次のような変換

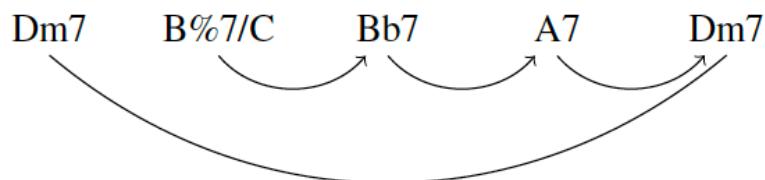


を再帰的に行うことにより、図 3.2c のような期待 - 実現 (expectation - realization) の構造と一対一の対応する木構造を得ることができるとされている。この期待 - 実現 (expectation - realization) の構造と一対一の対応する木構造を Complete Constituents と呼ぶ。後述する Jazz Harmony Treebank では、一部の楽曲において、Complete Constituents の木構造に加えて、Open Constituents を用いた木構造のアノテーションもなされていることがあるが、本研究では、あくまで楽曲の和声進行にのみ着目するため、Complete Constituents の木構造のみを用いることとする。

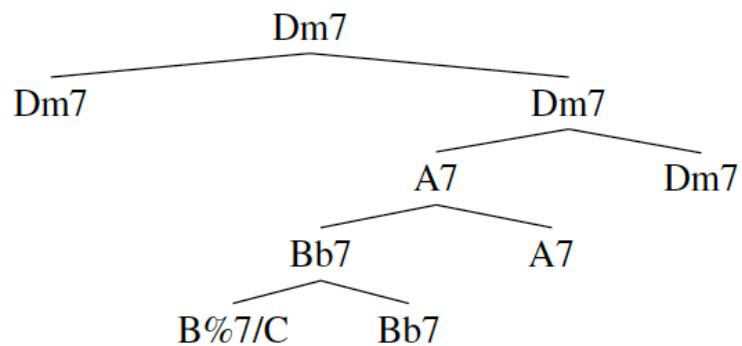
³低音が半音階で下降すること。クリシェともよばれる



(a) GSM による和声構文木 (Open Constituents)



(b) 期待-実現の構造



(c) GSM による和声構文木 (Complete Constituents)

図 3.2: Why Don't You Do Right? の和声構造表現の例 [24]

3.3 Jazz Harmony Treebank

Jazz Harmony Treebank⁴(以下, JHT)[24] は, Harasim らによって作成されたジャズスタンダードの和声進行について専門家による階層的分析の結果がアノテーションされたデータセットである. 階層的分析は 3.2 節で説明した文法規則に基づいて行われている.

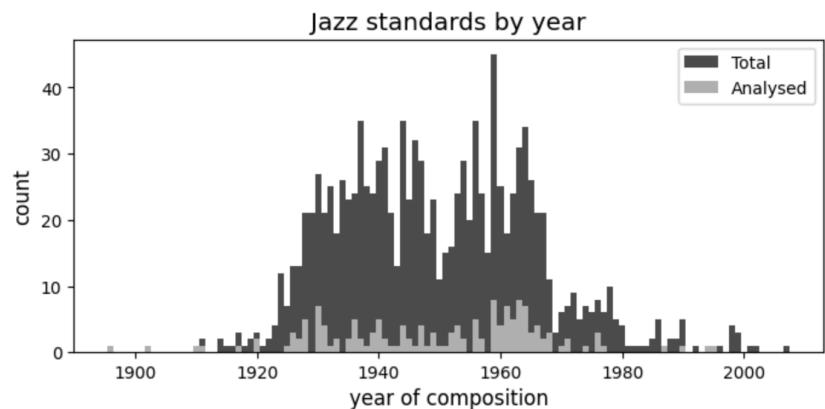
JHT の対象とするジャズのジャンルは, Swing, Bossa Nova, Jazz Blues, Bebop, Cool Jazz, Hard Bop とされており, traditional Blues, Modal Jazz, Free Jazz, Modern Jazz といったジャンルの楽曲や, ジャズスタンダードと言われる楽曲でも木構造分析では記述が難しいとされる Groovin'High や Out of Nowhere といったものは除かれている.

JHT で分析が行われている楽曲は, Shanahan らによって作られた iRealPro データセット⁵[27] に採用されている楽曲である. その中から, JHT で対象にする楽曲ジャンルや和声構造分析にふさわしい楽曲 150 曲が Harasim らの手によって選ばれた. 木構造のアノテーションは Harasim と数名の学生アシスタントによって行われ, 最終的な確認を Harasim が行っている. 1 小節以上続くコードについては, 複数のコードシンボルに分割された. また, 9 th, 11th といったテンションの表記は除去された. アノテーションされた木構造は JSON(JavaScript Object Notation) 形式によって, 楽曲タイトルや作曲者, 作曲年, 調性といったメタ情報と共に提供されている.

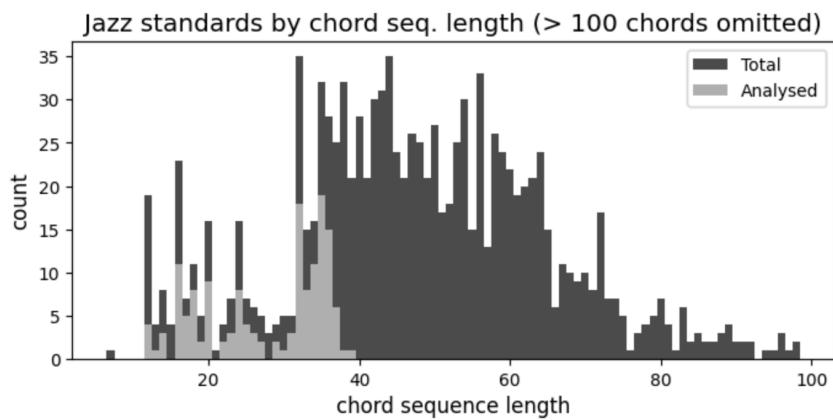
150 曲のコード進行の平均長は 27.75 コードシンボル, 合計 11697 コードシンボルであった(うち重複をのぞくと 92 コードシンボル). 構文木は合計で 3899 の binary 規則によって構成され, 木の高さの平均は 7.57 であった. そのほかの JHT の統計情報を図 3.3, 3.4 に示す. 図 3.3a は iRealPro データセットの中から今回選択された楽曲 150 曲の作曲年の分布を示している. 図 3.3b は選択された楽曲のコード進行の長さを示している. JHT では和声構造へのアノテーションを確実に行うために比較的短い楽曲が選ばれていることがわかる. 図 3.3c はコード進行の長さとアノテーションされた木構造の深さとの関係を示している. 図 3.4a, 3.4b はマイナーキー, メジャーキーそれぞれの木構造アノテーションに出現した上位 20 の構文規則を示す. なお, この構文規則に現れるコードは全て C マイナーキー, C メジャーキーにそれぞれ移調した後のコードである.

⁴<https://github.com/DCMLab/JazzHarmonyTreebank>

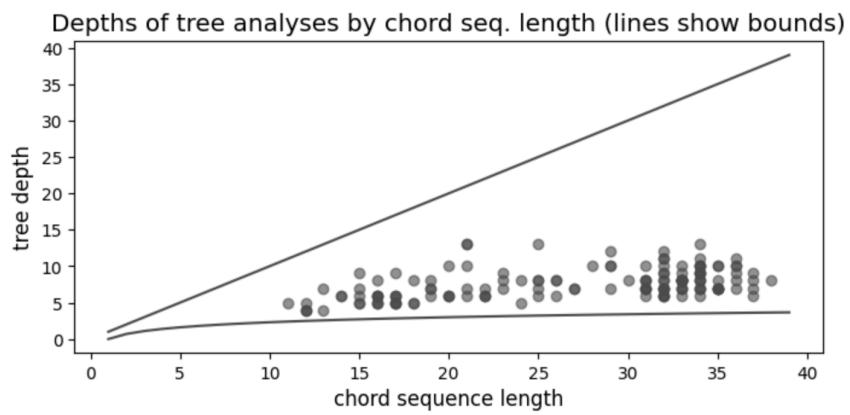
⁵<http://doi.org/10.5281/zenodo.3546040>



(a) 木構造解析を行った楽曲の作曲年

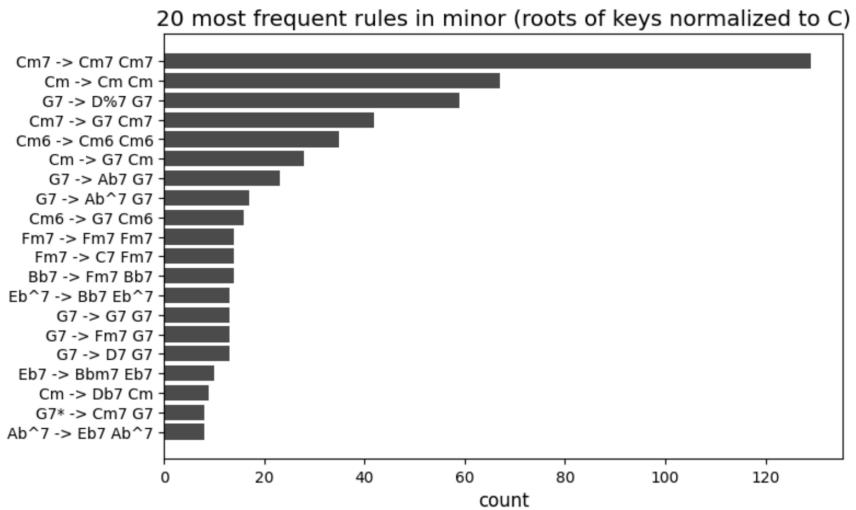


(b) 木構造解析を行った楽曲のコード進行の長さ

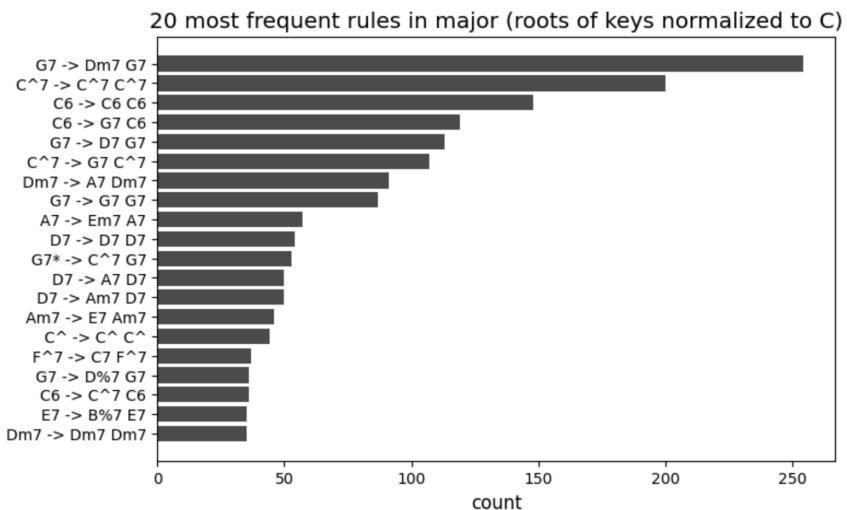


(c) コード進行の長さと木構造の深さとの関係

図 3.3: JHT アノテーションの統計情報 [24]



(a) マイナーキーにおける上位 20 の構文規則 (ルートを C に正規化)



(b) メジャーキーにおける上位 20 の構文規則 (ルートを C に正規化)

図 3.4: JHT アノテーションの統計情報 (続き) [24]

第4章 漸進的和声構造解析システムECSAの構築

本節では、2章と3章で述べた内容を基に、本研究で提案する漸進的和声解析の手法とそれを実装したシステムについて述べる。まず、4.1節で提案手法全体を概説し、ついでその手法の詳細について4.2節で述べる。最後に4.3節において、実装したシステムであるExpectation-based Chord Sequence Analyzerの使用方法を含めて述べる。

4.1 提案手法と実装システムの概要

本研究では、既存の認知的音楽理論では楽曲途中の認知構造の表示ができないという問題意識に基づいて、漸進的に和声進行を解析し、その木構造の途中経過を表示する手法を提案する。木構造の途中経過を表示するというアイデアは2.2.3節で説明した、東条による様相論理に基づく音楽理論[17]から得たものである。和声進行の構文規則は、3章で述べた、ジャズ和声に関するGSMに基づくものを使用する。本研究で提案する漸進的和声構造解析は、主に次の2段階のプロセスによって行われる。

1. PCFGのパラメータ推定：3.3節で述べたJazz Harmony Treebank[24]から和声進行に関する構文規則を抽出する。また、構文規則の出現頻度から構文規則それぞれの適用確率を推定する。(4.2.2節)
2. 漸進的チャート解析：自然言語の構文解析手法の一つである漸進的チャート解析を用いてコード進行を漸進的に解析し、その結果を出力する。(4.2.1節)

また、本研究ではこの手法をデスクトップアプリケーションとして実装したシステム、Expectation-based Chord Sequence Analyzer(以下、ECSA)を開発した。このシステムは、解析結果である木構造を画像として出力するため、ユーザは入力したコード進行に対する構造解析の結果を直感的に理解することができるようになっている。

以上、ECSAの入出力の関係をイメージ図4.1にまとめた。まずシステムへの入力はあるジャズ楽曲のコード進行の一部または全体とする。これを漸進的チャート解析器(Incremental Parser)を用いて構造解析することによって、解析結果である和声構文木を得る。図ではDm7 G7 CM7という3つのコードが入力されている状態を示しており、システムの出力としての画像はDm7 G7まで入力時の画像のみを書いているが、実際には、Dm7までを入力とした時の木構造分析の結果、Dm7 G7までを入力とした時の木構造分析の結果、Dm7 G7 CM7までを入力とした時の

木構造分析の結果のそれがシステムの出力として表示されるようになっている。ECSA の詳細は 4.3 節で述べる。

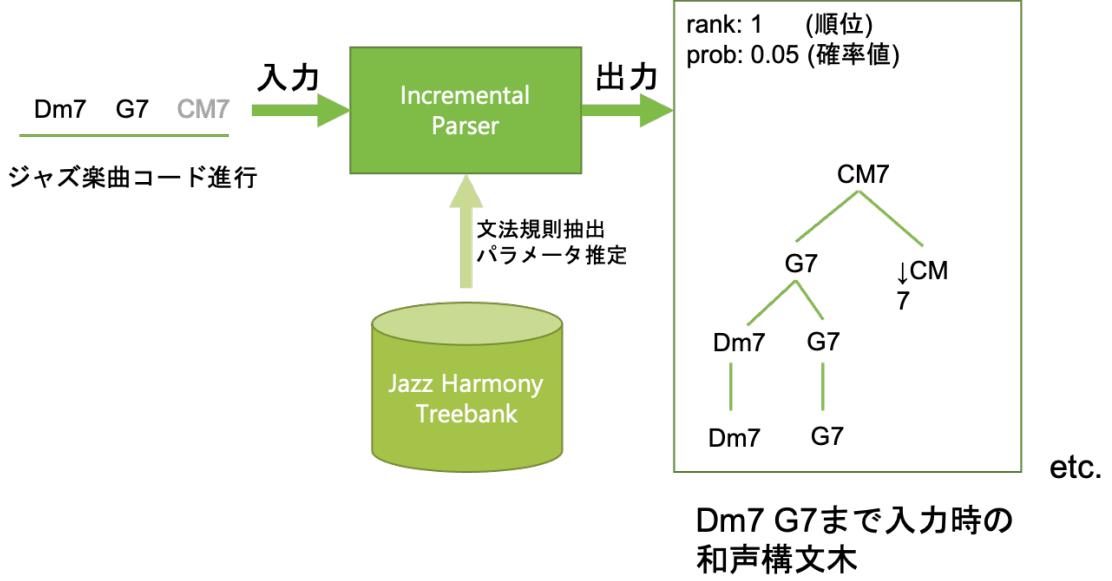


図 4.1: 漸進的和声解析システム ECSA の入出力関係

4.2 確率文脈自由文法に基づく漸進的和声解析

本研究で提案する漸進的和声解析手法は自然言語の構文解析手法である漸進的チャート解析の枠組みに基づいている。そのため、まずこの漸進的チャート解析を用いた和声解析について 4.2.1 節で説明する。なお、この漸進的チャート解析を用いた和声解析法は以前著者が、SMC2020 にて発表した内容 [28] を含んでいる。次に、[28] で提案された内容を確率的文脈自由文法 (PCFG) に拡張したアルゴリズムについて 4.2.2 節で述べる。

4.2.1 漸進的チャート解析のアルゴリズム

漸進的チャート解析とは松原ら [29] によって提案された、自然言語処理における構文解析手法の一つである。本手法はもともと、リアルタイム翻訳と呼ばれる、発話に伴って逐次的に翻訳結果を出力するようなシステムへの応用が期待され提案された手法であるが、本研究ではその入力文に対して逐次的に解析結果を表示することができるという特徴を利用する。以下ではまず漸進的チャート解析について説明する。漸進的チャート解析はチャート (**chart**) と呼ばれるグラフ構造を用いて解析結果を保持する。チャートは節点の集合、及び弧の集合から構成される。節点 (**node**) とは入力文中の単語と単語の間に位置し、 i 番目の単語 w_i と $i+1$ 番目の単語 w_{i+1} の間の節点は、番号 i でラベル付けされる。以下では、番号 i でラベル付された節点を単に節点 i と呼ぶ。弧 (**edge**) は節点と節点を結び、文中でその弧が覆っている部分に対する構文木をラベルとして持つ。

この構文木は項 (**term**) と呼ばれるデータ構造で表され, 記法 $[\alpha]_X$ で表現される. ここで, X は範疇 (category) であり, α は, 単語 (コード), 記号 $[?]$, 項のリストのいずれかである. $[\alpha]_X$ という項 σ に対して, X を σ の範疇と呼ぶ. $[?]_X$ のような項は未決定項 (**undecided term**) と呼ばれ, 構造が決定されていないことを表す. 項の中に出現する未決定項のうち, 最も左に位置するものを最左未決定項と呼ぶ. 弧にラベル付された項の中に未決定項が存在する時, この弧を活性弧 (**active edge**) と呼び, そうでない時, 不活性弧 (**inactive edge**) と呼ぶ.

Algorithm 1 漸進的チャート解析のアルゴリズム

```

function CHART_PARSING( $G\_chart, w$ )
     $L\_chart \leftarrow \{\}$       /*Local charts*/
     $temp \leftarrow \{\}$ 

    /*step1 辞書引き*/
    for  $\alpha \in \text{Lexicon}$  do
        if  $w = \alpha$  then
             $L\_chart \leftarrow L\_chart \cup \{[w]_\alpha\}$ 

    /*step2 文法規則の適用*/
    for  $\sigma \in L\_chart$  and  $\beta \rightarrow \beta_1\beta_2\dots\beta_n \in \text{Rules}$  do
        if  $\sigma = \beta_1$  then
             $L\_chart \leftarrow$ 
             $L\_chart \cup \{[\sigma[?]_{\beta_2}]_\beta\}$ 

    /*step3 項の置き換え*/
    for  $\phi \in G\_chart$  and  $\psi \in L\_chart$  do
        if  $\gamma = lut(\phi) \wedge \gamma = \psi$  then
            replace  $lut(\phi)$  with  $\psi$ 
             $temp \leftarrow temp \cup \{\phi\}$ 

     $G\_chart \leftarrow temp$       /*Global charts*/
    return  $G\_chart$ 

/*main*/
 $G\_chart \leftarrow [?]_S$ 
for i=1,...,last do
     $w_i \leftarrow \text{input\_chord}$ 
     $G\_chart \leftarrow \text{CHART\_PARSING}(G\_chart, w_i)$ 

```

漸進的チャート解析の Algorithm 1 に記す。アルゴリズム中では項 ϕ の最左未決定項を $lut(\phi)$ と表している。 i 番目の単語(コード) W_i が入力された時、次の操作を順次実行する。

辞書引き 単語 w_i の範疇が X ならば、項 $[w_i]_X$ をラベルとして持つ不活性弧をローカルチャートの節点 $i - 1$ と節点 i に追加する。

文法規則の適用 ローカルチャートの節点 $i - 1$ と節点 i を結ぶ弧に、項 $[\dots]_X$ がラベルづけされている時、文法規則 $A \rightarrow XY\dots Z$ が存在するならば、項 $[[\dots]_X?]_Y\dots[?]_Z]_A$ をラベルとして持つ弧をローカルチャートの節点 $i - 1$ と節点 i の間に追加する。

項の置き換え グローバルチャートの節点 0 と節点 $i - 1$ を結ぶ活性弧にラベルづけされた項 ϕ の最左未決定項を $[u]_X$ とする。この時、ローカルチャートの節点 $i - 1$ と節点 i を結ぶ弧にラベルづけされた項 ψ の範疇が X であるならば、項 ψ の最左未決定項を項 ϕ で置き換えた項をラベルとして持つ弧をグローバルチャートの節点 0 と節点 i の間に追加する。

一般的なチャートパーサはボトムアップなものとトップダウンのものとがある。本アルゴリズムにおいて local chart はボトムアップなものに対応し、global chart はトップダウンなものに対応している。本手法では従来のボトムアップ型チャート解析に、活性弧に対して文法規則を適用する操作、及び活性弧にラベル付された項の最左未決定項を別の活性弧にラベル付された項で置き換える操作を導入している。これによって単語が入力されるごとに、それまでの入力に対する項を生成することができるようになっている。

例えば表 4.1 に示す文法と辞書を用いて、*Take the A Train* というジャズスタンダードの冒頭のコード進行

Cmaj7 D7 Dmin7 G7 Cmaj7

を漸進的に解析することを考える。

表 4.1: 構文解析のための文法と辞書の例

文法	辞書
(r1) $S \rightarrow I_C$	(l1) $I_C \rightarrow Cmaj7$
(r2) $I_C \rightarrow I_C I_C$	(l2) $I_C \rightarrow Dmin7$
(r3) $I_C \rightarrow V_C I_C$	(l3) $V_C \rightarrow G7$
(r4) $V_C \rightarrow II_C V_C$	(l4) $V_G \rightarrow D7$
(r5) $V_C \rightarrow V_G V_C$	

表 4.1 の左側、文法において、 S は開始記号を示している。また、ローマ数字と下付き文字で示される記号はディグリーネームであり、各々のコードの属する範疇を示す。例えば、 V_C であれば C メジャーキーの V の和音(と同等の機能を持つ和音)を意味している。一方、表の右側である辞書は、コードの属する範疇から実際に譜面上に存在する(認知される)コード記号へ接続する規則を示している。

表 4.2: *Take the A Train* のコード進行に対する漸進的な解析結果

(a) ローカルチャート

input chord	local_chart			
	ID	edge	term	rules
Cmaj7	(2)	0-1	$[Cmaj7]_{I_C}$	(l1)
	(3)	0-1	$[[Cmaj7]_{I_C}]_S$	(r1)(2)
	(4)	0-1	$[[Cmaj7]_{I_C} [?]_{I_C}]_{I_C}$	(r2)(2)
	(5)	0-1	$[[[Cmaj7]_{I_C} [?]_{I_C}]_{I_C}]_S$	(r1)(4)
D7	(8)	1-2	$[D7]_{V_G}$	(l4)
	(9)	1-2	$[[D7]_{V_G} [?]_{V_C}]_{V_C}$	(r5)(8)
	(10)	1-2	$[[[D7]_{V_G} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C}$	(r3)(9)
	(11)	1-2	$[[[[D7]_{V_G} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C}]_S$	(r1)(10)
	(12)	1-2	$[[[[D7]_{V_G} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}$	(r2)(10)
	(13)	1-2	$[[[[[D7]_{V_G} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}]_S$	(r1)(12)
Dmin7	(16)	2-3	$[Dmin7]_{II_C}$	(l2)
	(17)	2-3	$[[Dmin7]_{II_C} [?]_{V_C}]_{V_C}$	(r4)(16)
	(18)	2-3	$[[[Dmin7]_{II_C} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C}$	(r3)(17)
	(19)	2-3	$[[[[Dmin7]_{II_C} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C}]_S$	(r1)(18)
	(20)	2-3	$[[[[Dmin7]_{II_C} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}$	(r2)(18)
	(21)	2-3	$[[[[[Dmin7]_{II_C} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}]_S$	(r1)(20)
G7	(24)	3-4	$[G7]_{V_C}$	(l3)
	(25)	3-4	$[[G7]_{V_C} [?]_{I_C}]_{I_C}$	(r3)(24)
	(26)	3-4	$[[[G7]_{V_C} [?]_{I_C}]_{I_C}]_S$	(r1)(25)
	(27)	3-4	$[[[[G7]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}]$	(r2)(25)
	(28)	3-4	$[[[[G7]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}]_S$	(r1)(27)
Cmaj7	(31)	3-4	$[Cmaj7]_{I_C}$	(l1)
	(32)	3-4	$[[Cmaj7]_{I_C}]_S$	(r1)(31)
	(33)	4-5	$[[Cmaj7]_{I_C} [?]_{I_C}]_{I_C}$	(r2)(31)
	(34)	4-5	$[[[Cmaj7]_{I_C} [?]_{I_C}]_{I_C}]_S$	(r1)(33)

(b) グローバルチャート

input chord	global_chart			
	ID	edge	term	rules
	(1)	0-0	$[?]_S$	
Cmaj7	(6)	0-1	$[[Cmaj7]_{I_C}]_S$	(1)(3)
	(7)	0-1	$[[[Cmaj7]_{I_C} [?]_{I_C}]_{I_C}]_S$	(1)(5)
D7	(14)	0-2	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C}]_{I_C}]_S$	(7)(10)
	(15)	0-2	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}]_S$	(7)(12)
Dmin7	(22)	0-3	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [[Dmin7]_{II_C} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C}]_{I_C}]_S$	(14)(17)
	(23)	0-3	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [[Dmin7]_{II_C} [?]_{V_C}]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}]_S$	(15)(17)
G7	(29)	0-4	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [[Dmin7]_{II_C} [G7]_{V_C}]_{V_C} [?]_{I_C}]_{I_C}]_{I_C}]_S$	(22)(24)
	(30)	0-4	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [[Dmin7]_{II_C} [G7]_{V_C}]_{V_C} [?]_{I_C}]_{I_C} [?]_{I_C}]_{I_C}]_S$	(23)(24)
Cmaj7	(35)	0-5	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [[Dmin7]_{II_C} [G7]_{V_C}]_{V_C} [Cmaj7]_{I_C}]_{I_C}]_{I_C}]_S$	(29)(31)
	(36)	0-5	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [[Dmin7]_{II_C} [G7]_{V_C}]_{V_C} [Cmaj7]_{I_C} [?]_{I_C}]_{I_C}]_{I_C}]_S$	(30)(31)
	(37)	0-5	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [[Dmin7]_{II_C} [G7]_{V_C}]_{V_C} [Cmaj7]_{I_C} [?]_{I_C}]_{I_C}]_{I_C}]_S$	(29)(33)
	(38)	0-5	$[[[Cmaj7]_{I_C} [[[D7]_{V_G} [[Dmin7]_{II_C} [G7]_{V_C}]_{V_C} [Cmaj7]_{I_C} [?]_{I_C}]_{I_C}]_{I_C}]_S$	(30)(33)

漸進的チャート解析における解析処理の過程を表 4.2 に示す。表 4.3a はローカルチャート、表 4.3b はグローバルチャートをそれぞれ示している。表の各行はチャートの弧に対応しており、ID の列は生成順に付けられた弧の番号、edge の列は弧の始点と終点の対を表す。term の行は項を表す。rules の行は、該当する項を導き出すために使用された、構文規則の番号またはチャートに保存されている ID を示す。例えば、“D7” が入力された時点において、まず辞書引きが行われる。辞書 (l2) とマッチし、項 (8) が生成されローカルチャートに保存される。続いて、(8) の項に対して各ルールが適用され、項 (9) から (13) までが生成されローカルチャートに保存される。次に、項の置き換えが行われる。前段階である “Cmaj7” の入力後までにグローバルチャートで残っている項は (7) だけである。(7) の範疇は S であるから、ローカルチャートの項 (10) と (12) はグローバルチャートの項 (14) と (15) にそれぞれ置き換えられる。

通常のボトムアップ型チャート解析では、ローカルチャートに相当する項しか生成しないが、漸進的チャート解析では、これらに加えて、項の置き換えによって、入力文の先頭からの関係をグローバルチャートに保存する。これにより、コード間の依存関係を捉えることができる。

上述したアルゴリズムでは、文法規則の適用のステップにおいて、全てのルールを平等に扱おうとするが、文法規則中に左再帰規則¹が存在した場合、そこで無限ループに陥ってしまう可能性がある。これを避けるため、本稿では本ステップにおいて左再帰規則の適応回数を 1 回までに制限することにする。また解析途中の文（楽曲）において、未決定項が存在しない不活性弧、つまり完全に閉じた木が現れることがある。そのような木を完全部分木であるといい、文（楽曲）中で安定した構造を持っていることを表す。例えば、表 4.3b における (6) や (35) の状態を完全部分木であるという。

木構造を視覚的に表すために、本研究では図 4.2 のような図示を行う。図 4.2において、上矢印 (\uparrow) その時点で入力されたコードを表し、下矢印 (\downarrow) は未決定項を表す。木が完全部分木 (complete tree) であるならば、図 4.2 左側に示すように下矢印 (\downarrow) が一つも現れない木となる。

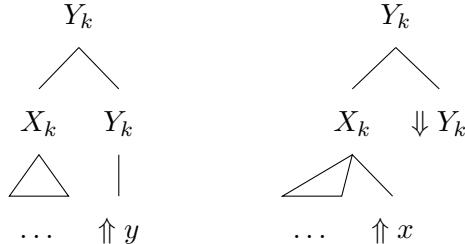


図 4.2: 完全部分木 (左), 不完全部分木 (右)

4.2.2 PCFG への拡張

確率文脈自由文法 (Probabilistic Context Free Grammar; PCFG) は構文木、及び文の生起確率を計算するためのモデルである [30]。このモデルでは文法中の各生成規則 $A \rightarrow \alpha$ に対して、 A

¹ $X \rightarrow X \dots$ のように構文規則の右辺第 1 項に左辺と同じものがくる規則

が与えられた場合にその生成規則が適用される条件付き適用確率

$$P(A \rightarrow \alpha | A) \quad (4.1)$$

を付与する。条件付き確率であることから、

$$\sum_{\alpha} P(A \rightarrow \alpha | A) = 1 \quad (4.2)$$

が成り立つ。すなわち、同一の非終端記号(前終端記号²)を左辺に持つ生成規則の適用確率の和は1である。このような確率を計算する最も簡単な方法として構文解析済のコーパスを使用する方法があげられる。コーパス中の非終端記号の出現回数と生成規則の使用回数を数えることによって生成規則の適用確率は次のように計算される。

$$P(A \rightarrow \alpha | A) = \frac{\text{コーパス中の } A \rightarrow \alpha \text{ の使用回数}}{\text{コーパス中の } A \text{ の出現回数}} \quad (4.3)$$

この時、明らかに式4.2は満足される。また、このようにして適用確率が与えられると、ある木構造 t の生成確率はその木構造を構成する全生成規則の適用確率の積で与えることができる。例えば、表4.1に示す文法と辞書に対して、表4.3のように確率値が与えられたとする。このとき、グローバルチャートの項(14)の構成に用いられた文法規則は、(r1), (r2), (r3), (r5), (l1), (l4)であるので³、その生起確率は $1.0 \times 0.6 \times 0.3 \times 0.2 \times 0.1 \times 1.0 = 0.0036$ であると計算できる。本研究ではこの木構造の生起確率を比較することによってその妥当性を検討する。

表 4.3: 文法規則の確率

文法規則	確率
(r1), (l2), (l4)	1.0
(r2)	0.6
(l3)	0.5
(r3), (r4)	0.3
(r5)	0.2
(l1)	0.1

²終端記号より一階層上にある、自然言語では品詞にあたる部分で辞書規則の親となるもの。GSM[19]の一部では、TR(トニックリージョン)やDR(ドミナントリージョン)と呼ばれる概念があり、コードシンボル(終端記号)を生成する I_C といったディグリーネームで表される範疇(前終端記号)と区別される。本研究で実装するシステムでは、[23, 24]に習って、非終端記号と前終端記号を同一視した(辞書規則の存在しない)文法規則を用いることとする。

³チャートのruleの欄を逆順にたどると使用した文法規則がわかる。

$$\begin{aligned}
 (14) &\Rightarrow (7)(10) \\
 &\Rightarrow (1)(5)(r3)(9) \\
 &\Rightarrow (1)(r1)(4)(r3)(r5)(8) \\
 &\Rightarrow (1)(r1)(r2)(2)(r3)(r5)(l4) \\
 &\Rightarrow (1)(r1)(r2)(l1)(r3)(r5)(l4)
 \end{aligned}$$

(1)はグローバルチャートの初期状態のため無視して良い。

4.3 Ex-based Chord Sequence Analyzer

本研究では木構造の直感的な理解を目的として、楽曲のコード進行を入力すると、それに対する漸進的な構造解析結果を出力するような GUI アプリケーション、Expectation-based Chord Sequence Analyzer(ECSA)(読み：エクサ)を作成した。本節では、本アプリケーションの概要について説明する。

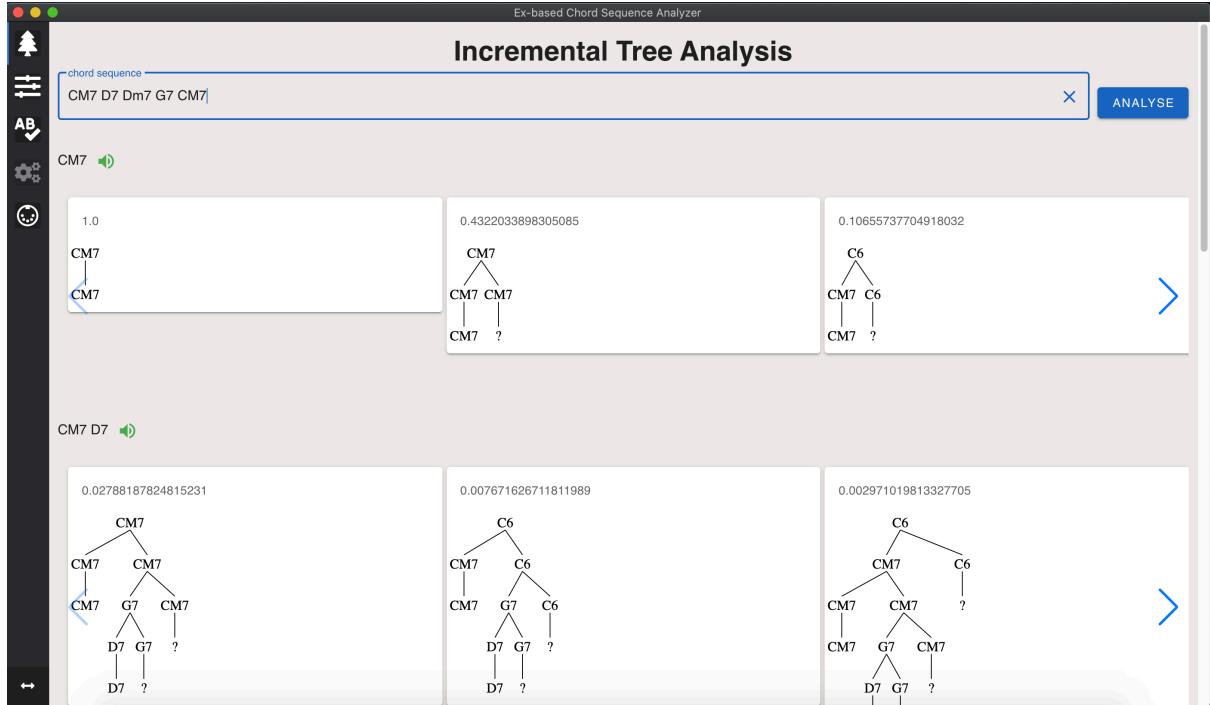


図 4.3: Expectation-based Chord Sequence Analyzer の実行画面

まず、本アプリケーションの実行画面は図 4.3 のようになっている。トップページのテキストボックスにユーザがコード進行を入力すると、入力に対する木構造解析の結果が複数行に渡って表示される。各行にはそのコードが入力された時点までの解析結果が表示されている。図 4.3 では *Take the A Train* のコード進行 (CM7 D7 Dm7 G7 CM7) に対して解析を行っており、1 行目では CM7 まで入力された段階の解析結果が、2 行目には CM7 D7 まで入力された時の解析結果が表示されている。またその時点での複数の構文木はそれぞれスライダー(カルーセル)のパネルに表示されている。各パネルの上部にはその構文木の生成確率が表示されており、構文木は確率の高いものを左から順に並べている。

図 4.4 に表示したページでは、ユーザが各構文規則の適用確率値を確認することが可能である。これによって、どの構文規則より重要であるかなどを考察することができる。

本アプリケーションの実装はフロントエンドに Electron、バックエンドに Python3 を用いている。Electron とは Windows/macOS/Linux とマルチプラットフォーム対応のデスクトップアプリを開発することができるフレームワークである。Electron で作られたアプリケーションは Chromium と呼ばれる Google Chrome のベースとなったオープンソースのウェブブラウザ上で

The screenshot shows the 'Check Grammar' interface of the Ex-based Chord Sequence Analyzer. The top bar displays the title 'Check Grammar'. On the left sidebar, there are icons for file operations (New, Open, Save, etc.) and a gear icon for settings. A vertical scroll bar is on the right side of the main content area.

Prolongation rule

rule	prob.
A7 -> A7 A7	0.0979020979020979
AM7 -> AM7 AM7	0.2857142857142857
Ab6 -> Ab6 Ab6	0.2
Ab7 -> Ab7 Ab7	0.1333333333333333
AbM7 -> AbM7 AbM7	0.5

Rows per page: 5 1-5 of 65 < >

Preparation rule

rule	prob. ↓
Abm6 -> Fm7 Abm6	1.0
Absus -> Cm7 Absus	1.0
Bb07 -> GM Bb07	1.0
Bhdim7 -> C7 Bhdim7	1.0

図 4.4: 適用確率確認画面

動作するため、HTML + CSS + JavaScript といった Web 開発の技術を使って開発できるのが特徴である。著者がこの修士論文を執筆するためにも使用した、優れたテキストエディタである Visual Studio Code にも採用されているフレームワークである。一方、漸進的チャート解析の実装や木構造画像の生成といったバックエンドの処理は Python で実装している。木構造の描画などが必要ない場合は Python プログラム単独でも動作可能である。

なお、実装したアプリケーションは <https://github.com/yutaogura/Ex-based-Analyzer> でダウンロードすることができる。

第5章 ECSA を用いた楽曲解析実験

本節では、4節で述べた提案手法を実装したシステムを用いた実験について述べる。まず、5.1節で実験の内容と実験条件について述べ、次いで5.2節で実際にいくつかの楽曲で解析を行った結果とその考察を行う。

5.1 実験内容と条件

実験では、ジャズスタンダード楽曲のコード進行の一部を本システムへの入力として、その出力である解析結果を示し、考察を行う。本システムの構成は、図4.1のようになっており、PCFGのパラメータ推定には3.3節で説明した、Jazz Harmony Treebankを使用する。実験で入力として使用する楽曲のコード進行はジャズスタンダードとし、ジャンルはJHTの楽曲選定基準と同じとした。また、複数の小節にまたがるコードの連続については、一つのコードにまとめて入力した。

また、2.2などでも説明したように、音楽解析における正解というものを断定する困難さのために、様々な楽曲の解析結果を総合した定量的な評価を与えることは現時点では難しいと考える。そこで、代表的ないくつかの楽曲に対する解析結果を個別に紹介し、各々の結果に対して解説や考察を与えることとする。

また、今回global chartの初期状態として登録しておく最初の項としては全てのルートに対するM, M7, m, m7を採用した。これをroot chordと呼ぶと、すなわち、

$$\text{root chord} = \{C, Db, \dots, B\} \times \{M, M7, m, m7\}$$

である。

また、4.2.1節で述べたアルゴリズムを基に、各コードが入力された状態（以下、コードステップとよぶ）における木構造を示す漸進的解析を行っているわけであるが、今回の実験では、コードステップごとに次のコードステップに残すglobal chart内の項を制限する枝刈りという処理を追加した。これは、解析対象のコード進行が長くなると、global chartによって保持する項の数が指数関数的に増え、計算量も指数関数的に増えることへの対処として導入したものである。今回は、枝刈りの上限数として、各コードステップにおいて、木構造の生成確率の上位100位までを次のコードステップの解析に残した。

5.2 解析結果と考察

5.2.1 コードステップごとの木構造について

漸進的解析を行うことにより、楽曲途中の木構造を表示することができるようになる。この節では、まずコードステップごとの木構造に着目し、“解釈の多様性”という観点からの考察を行う。

(1) Take the A Train

Take the A Train は 32 小節からなる AABA 形式の楽曲である。ここでは、冒頭 A パートの解析を行う。A パートのコード進行は以下の通りである。また該当部分の楽譜を図 5.1 に示す。

CM7 – D7 – Dm7 – G7 – CM7

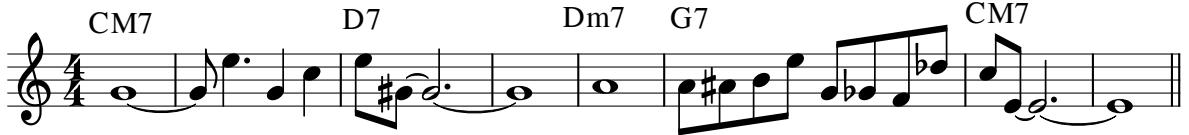


図 5.1: *Take the A Train* のリードシート (A パート)

木構造解析の結果を図 5.2 に示す。図ではコードステップごとにシステムが出力した木構造を行ごとに示している。なお、システム上ではコードステップごとに、枝刈りの上限数までの候補木を表示することが可能であるが、ここでは紙面の都合上、確率順に上位のいくつかのみ、また今後の説明で必要な候補木のみを抜粋して示す。各木構造の上部には、ステップ毎におけるその木構造の順位が括弧内に、その候補木の生成確率（有効数字 2 桁）が括弧外に記載されている。また、説明のため今後、あるコードステップのある順位の候補木を組で表す。例えば、Dm7 が入力されたコードステップにおける順位 1 の候補木を (c -1) といったように表す。括弧内の順位に*(アスタリスク) がついているものは完全部分木であることを示している。また、4.2.1 節と同様に? (疑問符) は未決定項を示し、次に期待されているコードまたはその範疇を示している。

(a) から (e) までを順に見ていくと、コードステップ毎にあるコードが期待されてそれが次のステップで実現される様子を確認することができる。今回は (e -1*) に向けて、(a -2) → (b -1) → (c -1) → (d -1) → (e -1*) というように木が成長している。また、木の成長過程は (a -2) → (b -1) → (c -1) というような新たな枝が追加される過程と、(c -1) → (d -1) → (e -1*) というような新しく枝は追加されず、直前に予測されていた未決定項が実現される過程という 2 つに分類することができる。また、CM7 が入力されたコードステップでは、(a -1*) や (e -1*) といった完全部分木が候補木として挙げられていることがわかる。CM7 はこの楽曲におけるトニックである。よってここから、完全部分木が候補木として列挙されるコードステップは、和声的に安定しているということが考えられる。

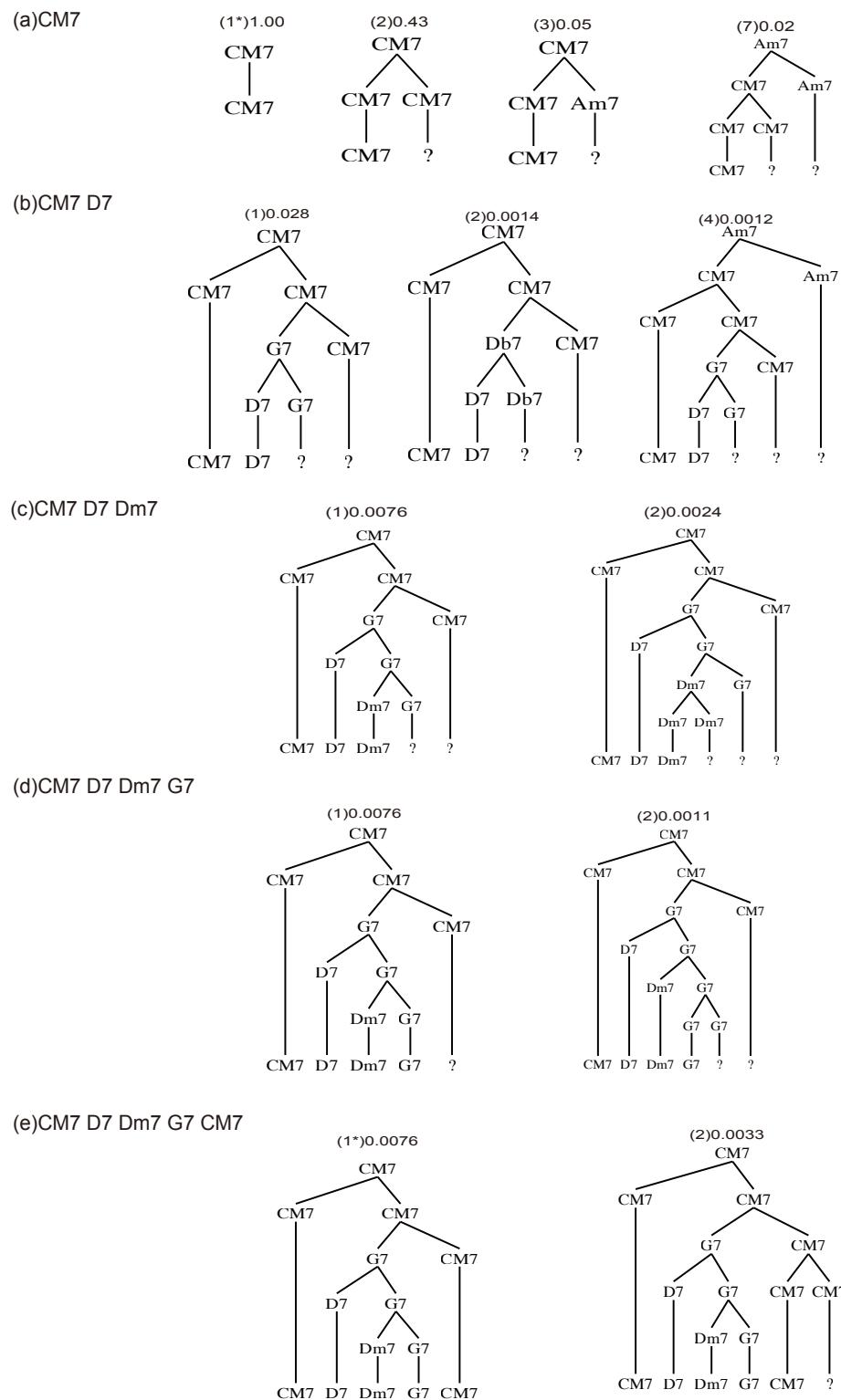


図 5.2: *Take the A Train* の木構造解析結果 (抜粋)

(2) Autumn Leaves

Autumn Leaves は 32 小節からなる AABC 形式の楽曲である。ここでは、冒頭 A パートの解析を行う。A パートのコード進行は以下の通りである。また該当部分の楽譜を図 5.3 に示す。

Cm7 – F7 – BbM7 – EbM7 – Ahdim7 – D7 – Gm7

図 5.3: *Autumn Leaves* のリードシート (A パート)

木構造解析の結果を図 5.4 と 5.5 に示す。各時点での解析結果については、確率順に上位のいくつかのみを抜粋して示す。図の読み方については、図 5.2 と同様である。

この解析結果においても、先ほどの例と同様に、コードステップ毎にあるコードが期待されてそれが次のステップで実現される様子を確認することができる。最終的に完全部分木として示されている (g -1) までに至る過程を順に見ていく (a -4) → (b -2) → (c -3) → (d -4) → (e -1*) → (f -1) → (g -1*) という流れであることがわかる。今回の解析では、*Take the A Train* の解析に比べ、最終的な完全部分木に至るまでの、コードステップごとの順位の変動が多いことがわかる。一般的な音楽理論では、*Autumn Leaves* は Bb メジャーと G マイナーという平行調を交互に行き交う曲であると解説される。A パート冒頭の Cm7 – F7 – BbM7 は Bb メジャーキーのトゥー・ファイブ・ワンであり、続く Ahdim7 – D7 – Gm7 は G マイナーキーのトゥー・ファイブ・ワンである。このように、複数の調性が混じり合った曲では、順位の変動が多く起こるのではないかということが予想される。

Autumn Leaves が複数の調性から成り立っていることを、木構造のルート (根) として予測されるものの変遷という観点からも考える。図 5.6 は、木構造のルートとして予測されるものの数がどのように変遷しているかを表したものである。各ステップでは 100 個の木構造を保持しており、最終的に予測された木構造の数が多かった上位 6 カテゴリの個数の変異を示している。これを見ると、様々なコードが木構造のルートとして予測されているが、4 の段階 (Ahdim7 が入力された段階) で大きく予測数が減少し、そこから BbM7 がルートとして予測される木構造の数が多くなっていくことがわかる。また、最終的な 6 の段階では、Gm7 がルートとして予測される個数も増加しており、BbM7 と Gm7 はこの曲 (セクション) において重要なコードであるといえよう。

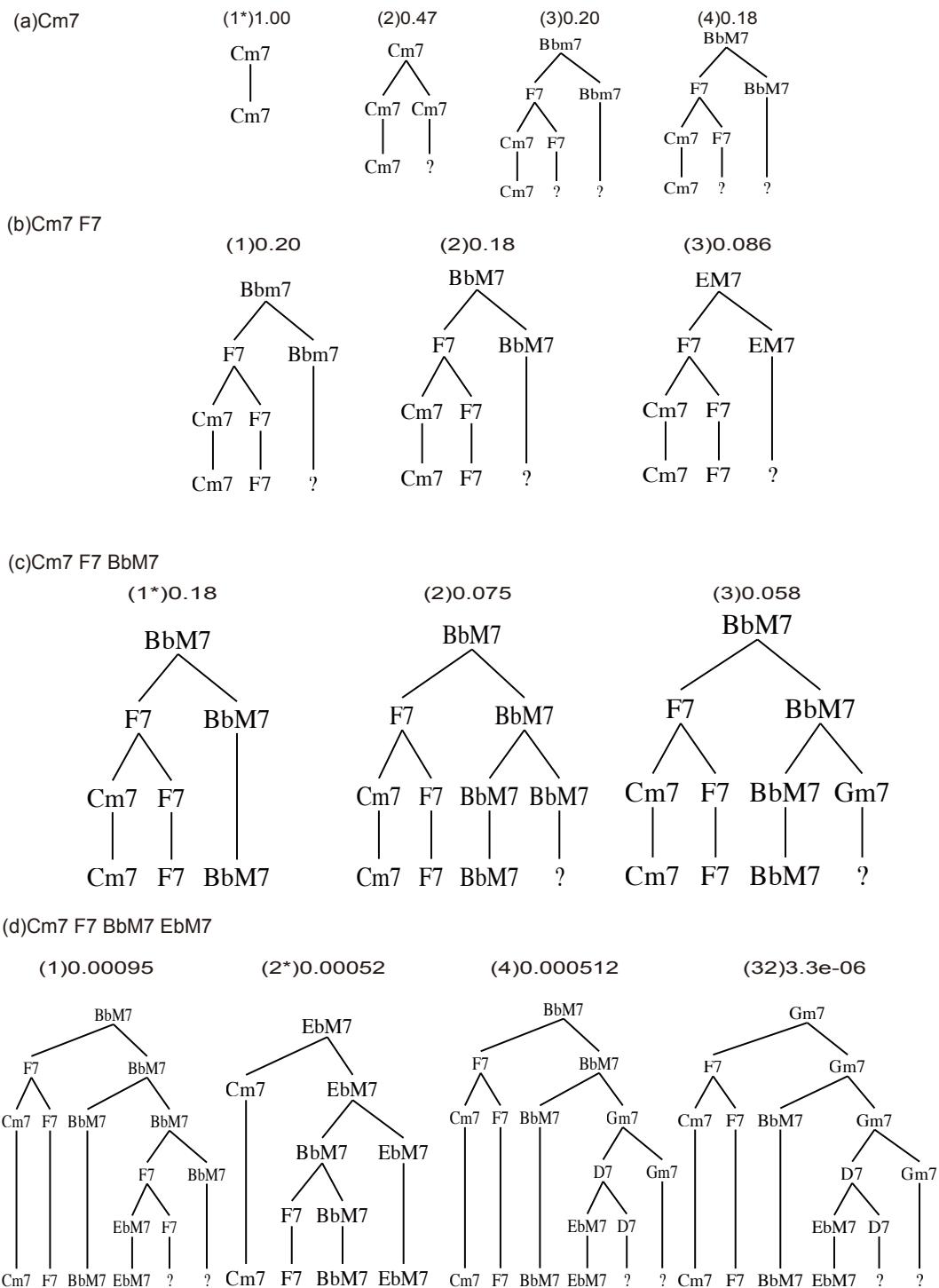
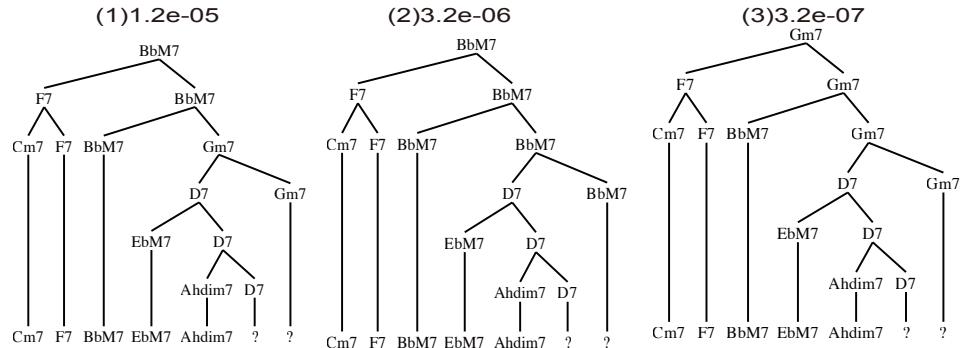
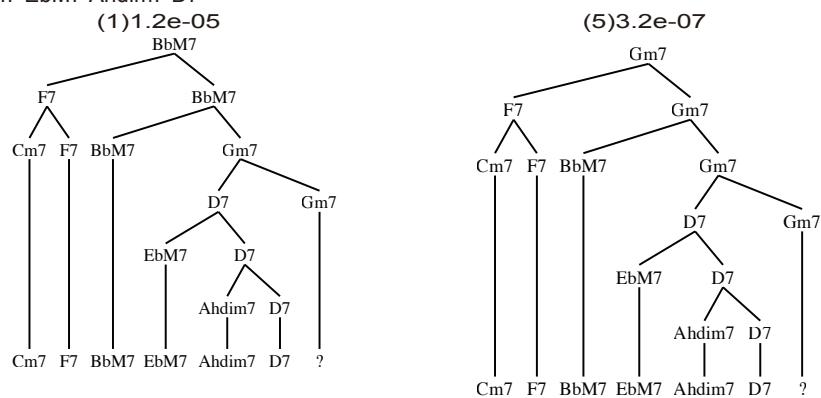


図 5.4: *Autumn Leaves* の木構造解析結果 (抜粋・前半)

(e) Cm7 F7 BbM7 EbM7 Ahdim7



(f) Cm7 F7 BbM7 EbM7 Ahdim7 D7



(g) Cm7 F7 BbM7 EbM7 Ahdim7 D7 Gm7

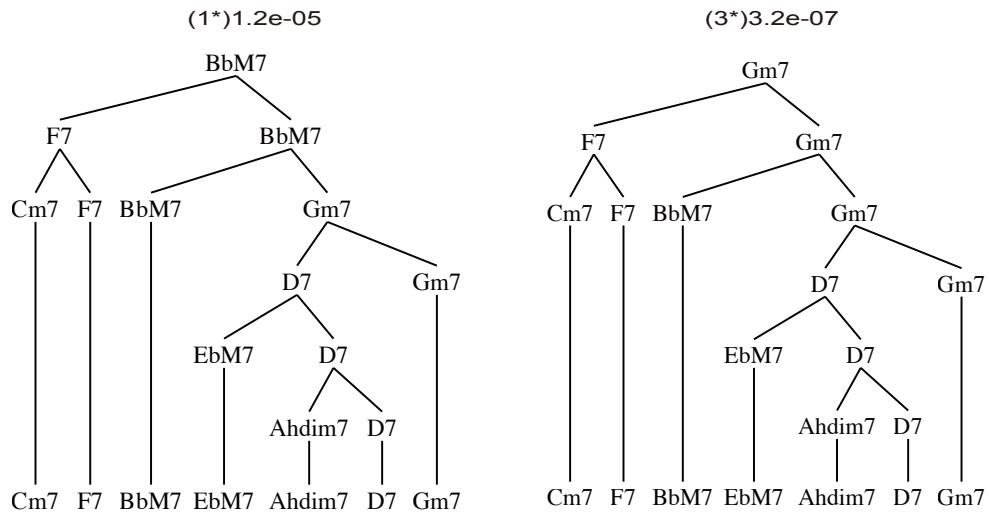


図 5.5: Autumn Leaves の木構造解析結果 (抜粋・後半)

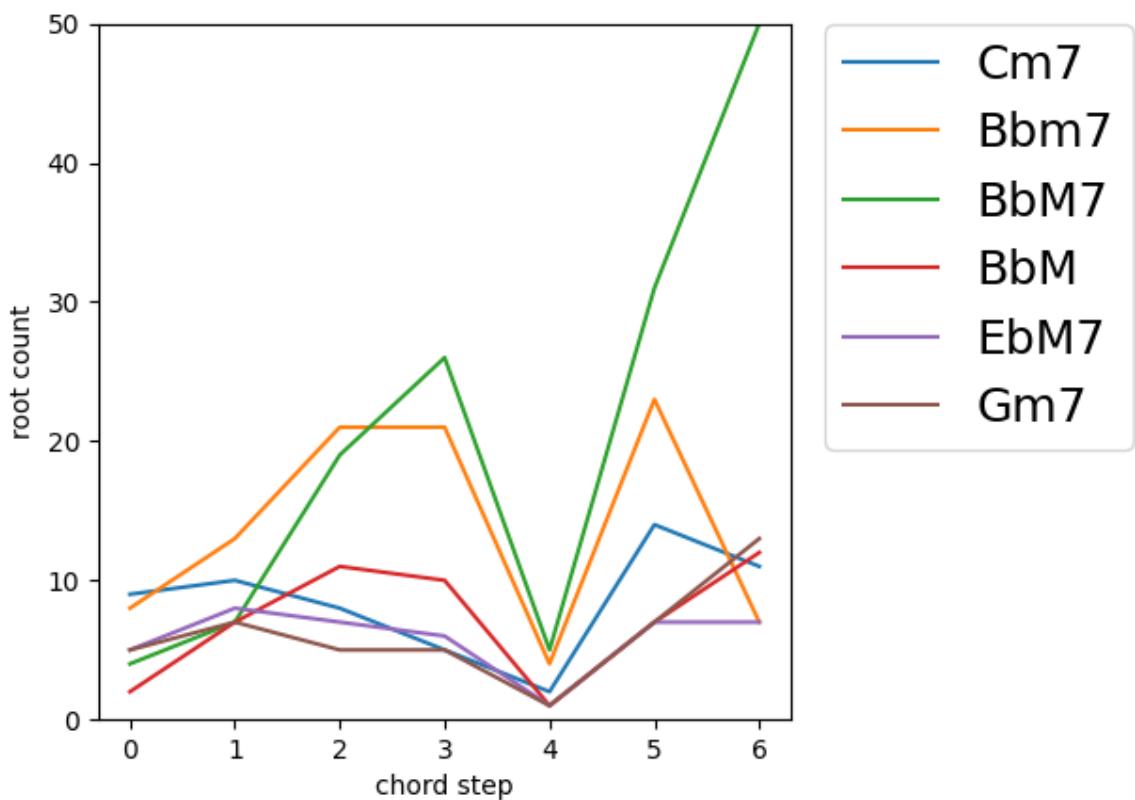


図 5.6: *Autumn Leaves* の木構造解析における根の予測結果数の遷移（上位 6 カテゴリ）

5.2.2 意外性の検出

この節では、前節と同様、コードステップごとの解析結果を分析しながら、木構造の確率値に着目し、“意外性”という観点からの考察を行う。

(1) Cute

Cute は 32 小節からなる ABAC 形式の楽曲である。ここでは、後半の 16 小節である AC パートの解析を行う。該当パートのコード進行は以下の通りである。また該当部分の楽譜を図 5.7 に示す。

Dm7 – G7 – CM7 – Dm7 – G7 – Gm7 – C7 –
FM7 – Bb7 – Em7 – Am7 – Dm7 – G7 – CM7

図 5.7: *Cute* のリードシート (後半 16 小節)

木構造解析の結果を図 5.8 と 5.9 に示す。図の構成は、図 5.2 と同様である。木構造解析を用いて意外性について考察する前に、従来の音楽理論の見解からこの楽曲の基本的な特徴を確認する。まず、*Cute* は基本的にキー C メジャーの曲であるといえる。これは、曲の最後のコードが CM7 で終了していることからも明らかである。はじめの 4 小節、Dm7 – G7 – CM7 はキー C メジャーのトゥー・ファイブ・ワンである。続く 5, 6 小節目もキー C メジャーのトゥー・ファイブであ

り，7小節目にIの和音であるCM7が来るのかと思ひきや，Gm7というコードが挿入され，これからGm7-C7-FM7というキーFメジャーのトゥー・ファイブ・ワンが始まっている。その後サブドミナントマイナーであるBb7を挟んでキーCメジャーのダイアトニックコードであるEm7に続き36251(ディグリーネーム)という進行でトニックであるCM7に帰着している。

ここで，楽曲途中における，あるコードの意外性を表す指標としてU(unexpectedness)を導入する。これは，あるコードステップにおいて最大確率を持つ木構造を基準として木構造の予測の変遷を評価するものである。あるコードステップnにおける木構造の候補を $t^{(n)}$ とし，その中で最大の生成確率を持つ木構造を $t_{\maxprob}^{(n)}$ ，その確率値を $P(t_{\maxprob}^{(n)})$ と表す。また， $t^{(n)}$ の導出元となった木構造を $t^{(n-1)}$ というように表す。この時，あるコードステップn(で入力されるコード)の意外性 $U^{(n)}$ を以下のように与える。

$$\begin{aligned} t^* &= t_{\maxprob}^{(n)} \\ U^{(1)} &= P(t_{\maxprob}^{(1)}) \quad (n=1 \text{ のとき}) \\ U^{(n)} &= \frac{P(t^{*(n-1)})}{P(t_{\maxprob}^{(n-1)})} \times \frac{P(t_{\maxprob}^{(n)})}{P(t^{*(n-1)})} \quad (n \geq 1 \text{ のとき}) \end{aligned}$$

意外性の指標Uは1に近いほど，“期待通りの(意外性のない)”進行であり，0に近いほど，“期待はずれの(意外性のある)”進行であるという意味である。Uは，一つ前のコードステップにおける木構造の希少さ(掛け算の左側)と新たに付け加えられた部分木の生成確率(掛け算の右側)を掛け合わせたものである。 $P(t^{*(n-1)})$ の部分は打ち消しあうため，結局，対象コードステップの前後における最大確率の比を求めれば良いという事になる。以下で，Uの算出法について具体的にみていく。

まず，コードステップ(c)や(e)，すなわち3小節目のCM7や6小節目のG7が入力された段階の意外性について考える。この時点の木構造は，(b-1)→(c-1)や(d-1)→(e-1)といった遷移によって成長した木構造である。これらは，予測順1位の木構造の中で予測されていたコードがそのまま実現される遷移，すなわち“期待通りの”進行であることがわかる。これらは遷移の前後で確率値が等しいため， $U = 0.0079/0.0079 = 1.00$ となる。このように前ステップで期待されたものがそのまま実現するような遷移の意外性は1である。

次に，コードステップ(b)や(d)，すなわち2小節目のG7や5小節目のDm7が入力された段階の意外性について考える，これは(a-4)→(b-1)や(c-2)→(d-1)という遷移で成長した木構造である。これらは，前ステップにおいてそれぞれ4位と2位の確率を持つ木構造が成長することで，現ステップにおいて1位の確率を持つようになった遷移であることがわかる。このような木構造の予測順位の変動を加味するものが，意外性の指標Uの計算における掛け算の左側である。また，(a-4)→(b-1)の遷移では，(a-4)の段階ですでに期待されていたG7が実現するように木が成長しているが，(c-2)→(d-1)の遷移では，期待されていた，CM7というコードを親として，新たにCM7→G7 CM7とG7→Dm7 G7という部分木が付加される形で，木が成長していることがわかる。このように新たに付加された予測部分を考慮するものが，指標Uの計算における掛け算の右側である。実際にUを求めるとき，コードステップ(b)でのUの値は $0.090/1.00 = 0.09$ ，コードステップ(d)でのUの値は $0.0035/0.090 = 0.03$ となる。これらの進行は和声構造が安定していた状態から新しい進行へと動き出す始まりのような部分であるといえるため，先ほどの(c)

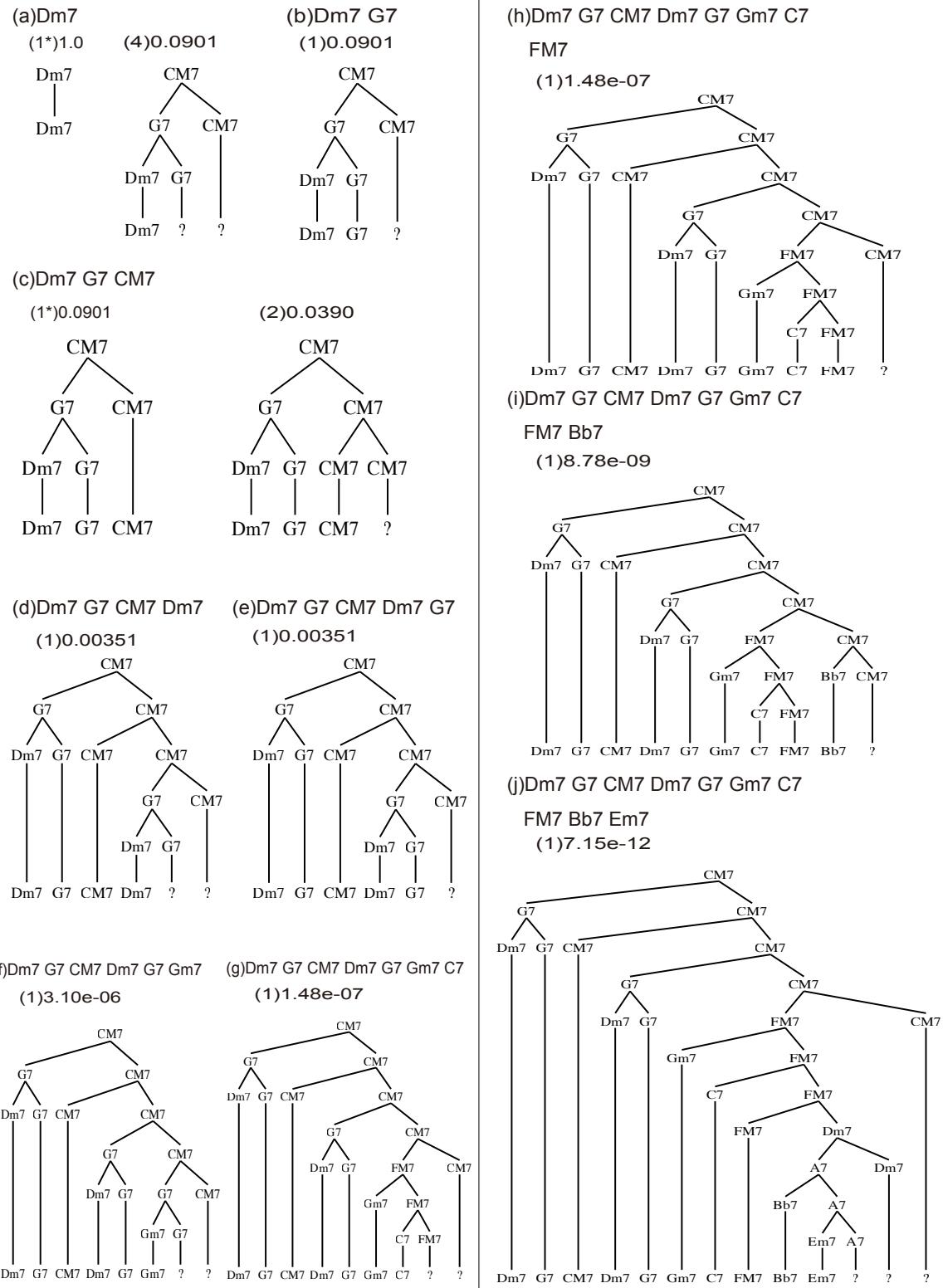
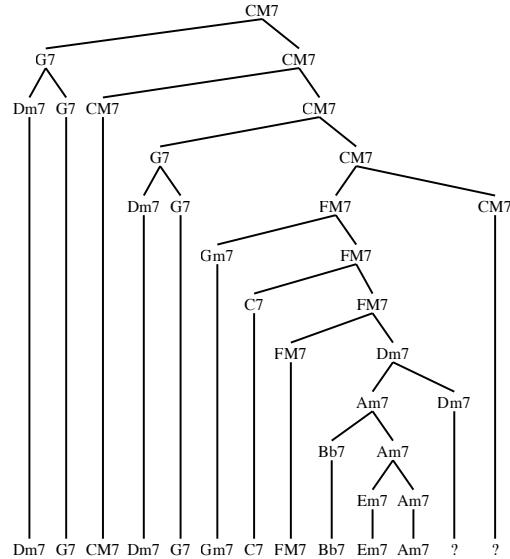


図 5.8: *Cute* の木構造解析結果(各 1 位のみ・前半)

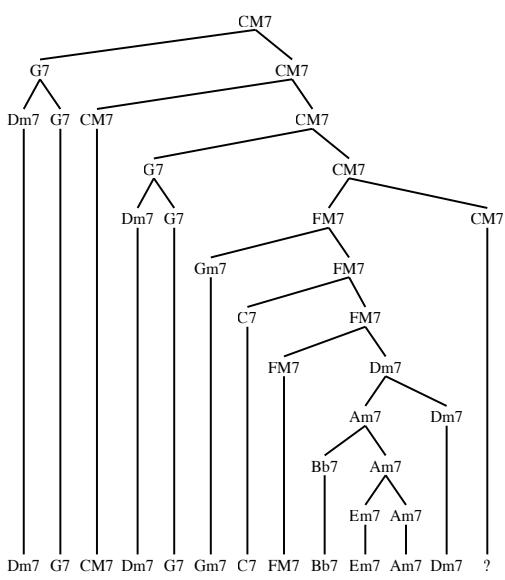
(k) Dm7 G7 CM7 Dm7 G7 Gm7 C7
FM7 Bb7 Em7 Am7

(1) 1.02e-12



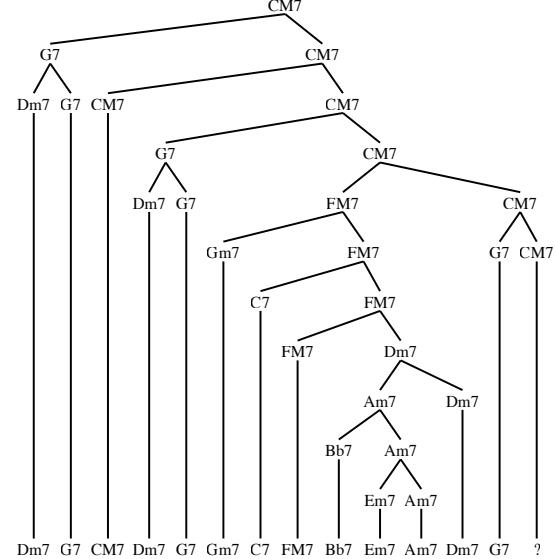
(l) Dm7 G7 CM7 Dm7 G7 Gm7 C7
FM7 Bb7 Em7 Am7 Dm7

(1) 1.02e-12



(m) Dm7 G7 CM7 Dm7 G7 Gm7 C7
FM7 Bb7 Em7 Am7 Dm7 G7

(1) 3.38e-13



(n) Dm7 G7 CM7 Dm7 G7 Gm7 C7
FM7 Bb7 Em7 Am7 Dm7 G7 CM7

(1*) 3.38e-13

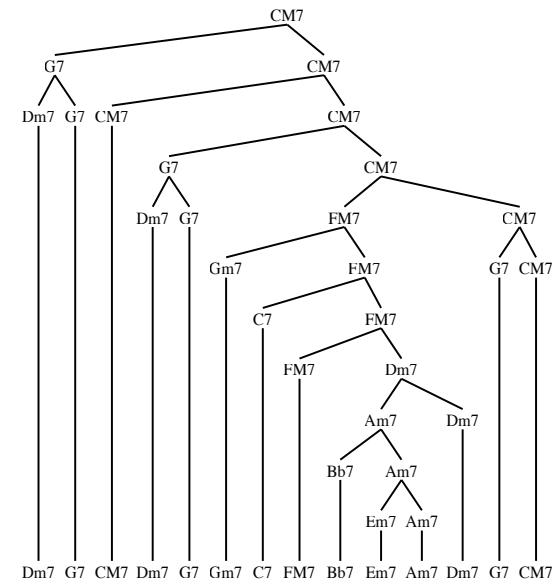


図 5.9: *Cute* の木構造解析結果 (各 1 位のみ・後半)

や (e) のステップよりかは “意外性のある” 進行だといえるであろう。

以上のような計算方法で、意外性を示す値 U を全てのコードステップにおいて算出したものを図 5.10 に示す。

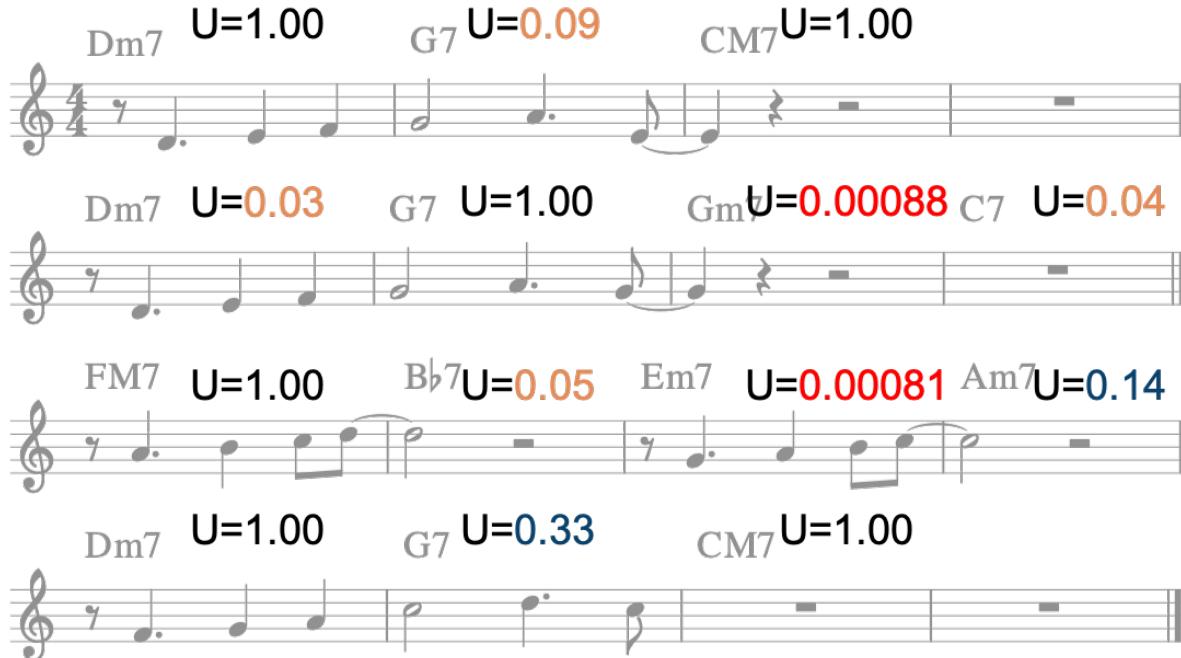


図 5.10: *Cute* の各コードステップにおける意外性 U の値

図 5.10 をみると、冒頭で述べた、キー F メジャーのトゥー・ファイブ・ワンが出現する部分である 7 小節目にかけて U の値が小さくなっていることがわかる。実際の木構造を見てみても (f -1) の段階では Gm7 の親として G7 が予測される形になっているが、(g -1) では C7 が挿入されることによって、木構造の予測順位が入れ替わり、FM7 が親として予測されるような形が 1 位のものとなっている。

続いて、8 小節目における FM7 期待が実現される形で、9 小節目が挿入され、続く Bb7, E7 で再度 U の値が低くなっている。これは、ノンダイアトニックコードである Bb7 に起因するものであると考えられる。一般的な音楽理論ではこの Bb7 はサブドミナント・マイナー、つまり同主調である C マイナーキーの IV の和音であると考えられ、一つ前の FM7 と合わせて、サブドミナント-サブドミナントマイナーというコード進行で有名である。今回のような場合は、FM7 がピボット・コード¹として働いているとして、調性的には C メジャーのままであるというふうに分析することが多い。しかし、木構造解析の結果をみると Bb7 Em7 Am7あたりでは A マイナーが親とする結果になっている。ここで親-子関係が認知的にどのような意味を持つのかについては、

¹複数の調性に跨って機能を持つコードのこと。今回の FM7 の場合は F メジャーキーの I(トニック) と C メジャーキーの IV(サブドミナント)。

さらに考察する必要があるだろう.

(2) Lady Bird

lady bird のコード進行は以下の通りである. また該当部分の楽譜を図 5.11 に示す.

CM7 – Fm7 – Bb7 – CM7 – Bbm7 – Eb7 – AbM7 – Am7 – D7 – Dm7 – G7 – CM7

The musical score consists of four staves of music in common time (indicated by a '4' in the top left corner). The first staff starts with a C major 7th chord (CM7). The second staff starts with an F minor 7th chord (Fm7), followed by a B flat major 7th chord (Bb7) indicated by a bracket over the next three measures. The third staff starts with an A flat major 7th chord (AbM7), followed by an A minor 7th chord (Am7) and a D7 chord. The fourth staff starts with a D minor 7th chord (Dm7), followed by a G7 chord, and ends with a C major 7th chord (CM7).

図 5.11: *lady bird* のリードシート

木構造解析の結果を図 5.12 と 5.13 に示す. 図の構成は、図 5.2 と同様である. また、*Autumn Leaves* の時と同様にルートの予測個数の遷移について図 5.14 に示す.

lady bird は基本的にキー C メジャーの曲であるが、Fm7 – Bb7 や Bbm7 – Eb7 といった C メジャー以外のトゥー・ファイブ (それぞれ Eb メジャーキー, Ab メジャーキーのトゥー・ファイブ) が現れることが特徴である. このあたりに、この楽曲の意外性のポイントがあることが、従来の音楽理論からも予測される.

意外性の指標 U を全てのコードステップにおいて求めた結果を図 5.15 に示す. 図 5.15 をみると、3, 7, 11 小節目のような、他のキーのトゥー・ファイブが始まる段階において U の値が小さくなっていることがわかる. また、そのような段階において、トゥー・ファイブのトニックとなりうるキーのコードがルートとして予測される個数が上昇していることが図 5.14 からもわかる.

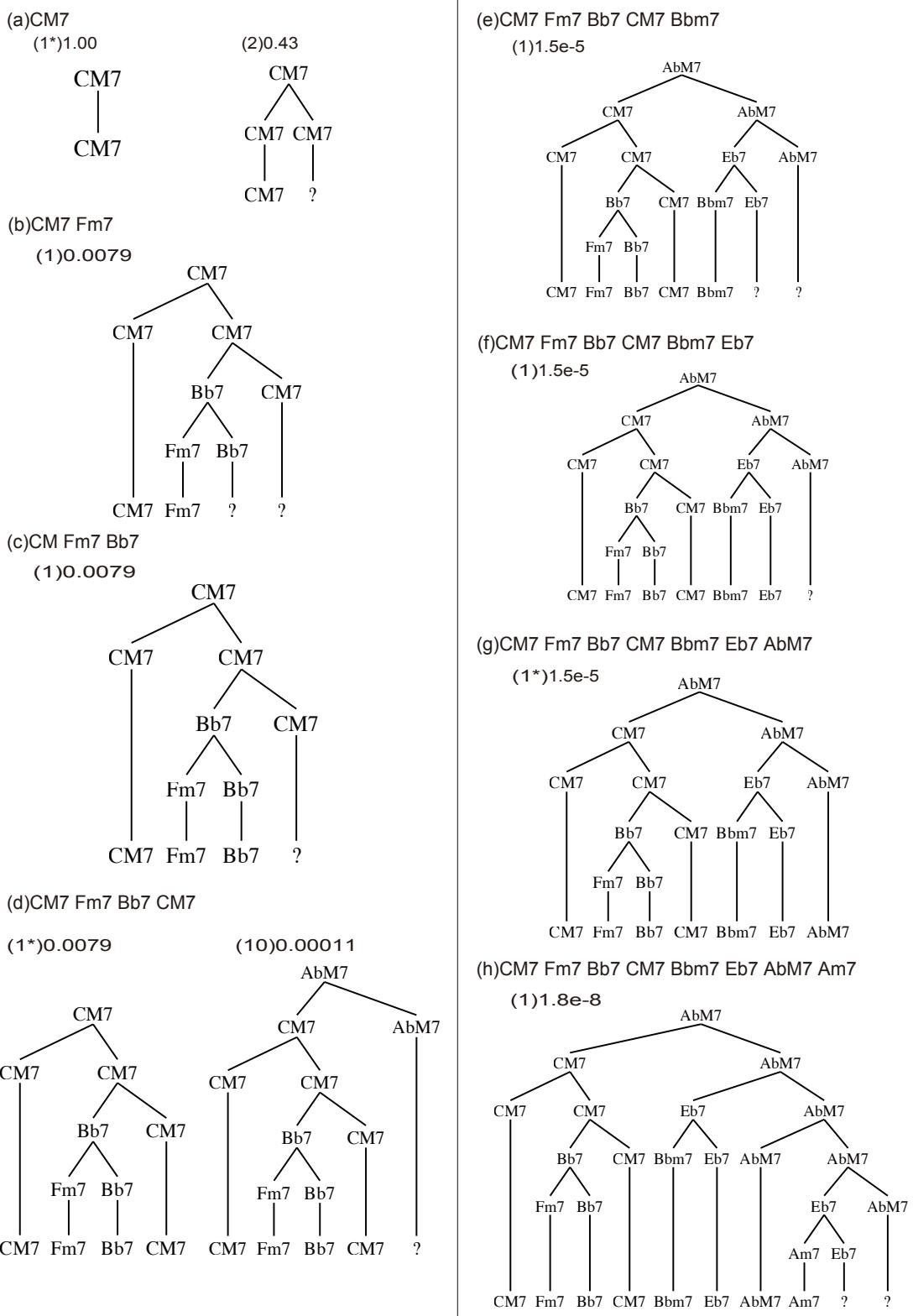
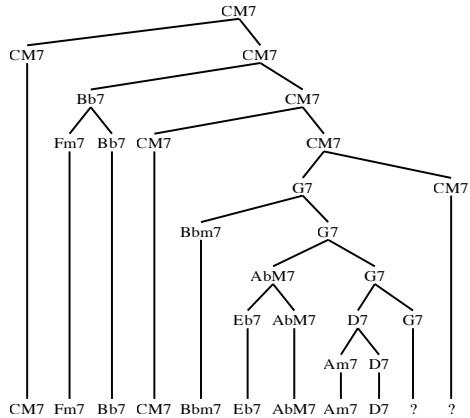
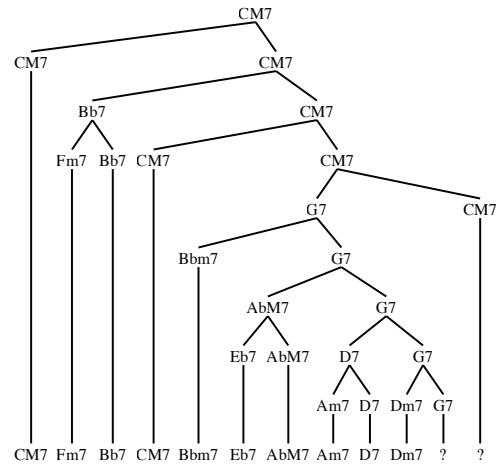


図 5.12: *Lady Bird* の木構造解析結果(各 1 位のみ・前半)

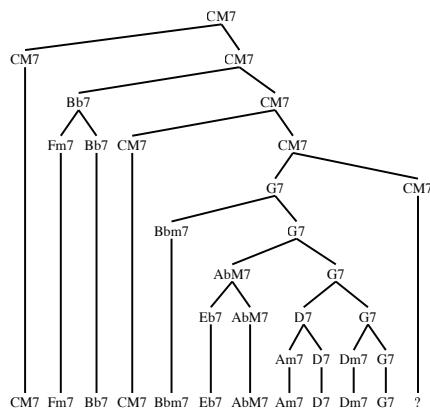
(i) CM7 Fm7 Bb7 CM7 Bbm7 Eb7 AbM7 Am7 D7
(1) 1.3e-9



(j) CM7 Fm7 Bb7 CM7 Bbm7 Eb7 AbM7 Am7 D7 Dm7
(1) 3.7e-10



(k) CM7 Fm7 Bb7 CM7 Bbm7 Eb7 AbM7 Am7 D7 Dm7 G7
(1) 3.7e-10



(l) CM7 Fm7 Bb7 CM7 Bbm7 Eb7 AbM7 Am7 D7 Dm7 G7 CM7
(1*) 3.7e-10

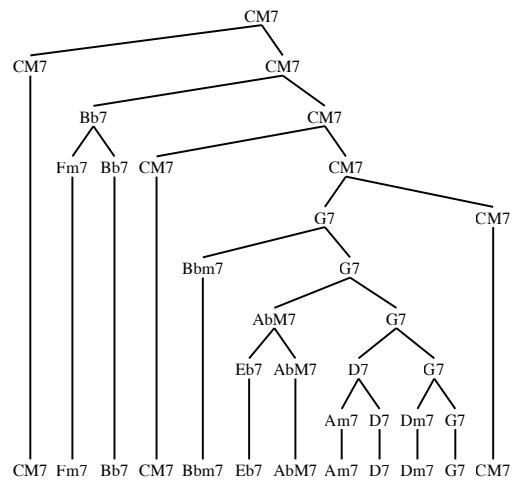


図 5.13: *Lady Bird* の木構造解析結果 (各 1 位のみ・後半)

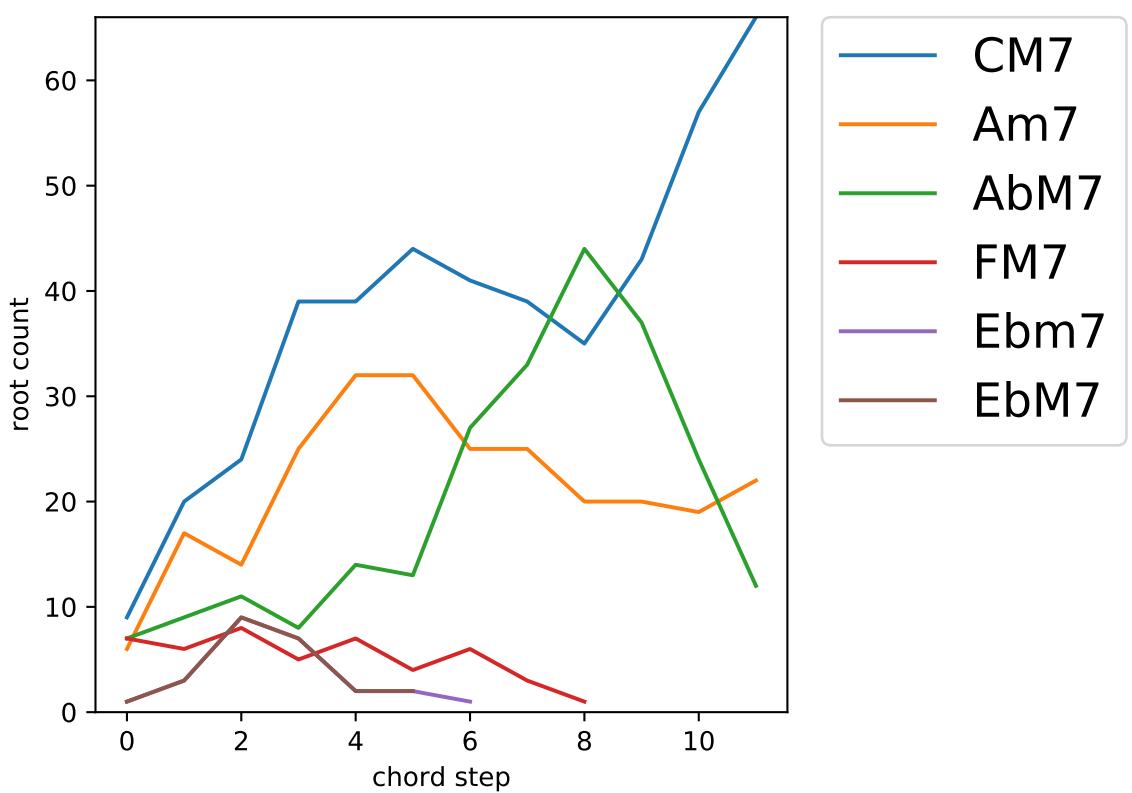


図 5.14: *Lady Bird* の木構造解析における根の予測結果数の遷移（上位 6 カテゴリ）

3, 7, 11 小節目の中では Fm7 が一番意外性が低いというような結果となった。これは、一般的な音楽理論において Fm7 – Bb7 – CM7 という進行が比較的頻度が高く使用されるということから類推できる。Fm7 – Bb7 のような iv7 から \flat VII7 のような動きは Backdoor ii – V という名前がついており、通常のトゥー・ファイブを代理コードで置き換えた進行である。そのため、7~9 小節目のような別のキーへの転調を伴うトゥー・ファイブよりかは意外性が低くなつたと考えられる。

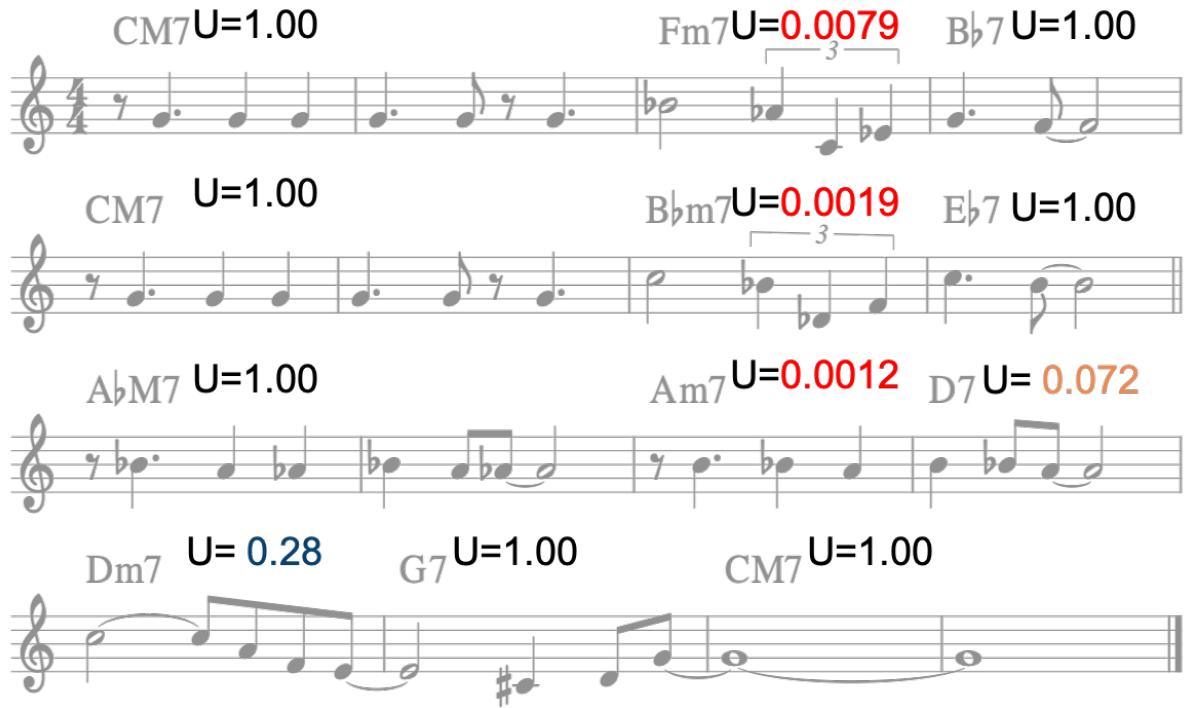


図 5.15: *Lady Bird* の各コードステップにおける意外性 U の値

このように各コードステップにおける意外性の指標 U の値を求ることによって、和声進行における期待-実現とその裏切り、つまり意外性について求めることができた。従来の音楽理論においてもコード進行のポイントのようなものは経験から指摘されていたが、この意外性の指標 U を用いることによって、より定量的に論じることができる。木構造やその確率値と認知的リアリティについての関係性については検討の余地があるが、漸進的構造解析によって表現できる木構造の段階ごとの比較によって、意外性について論じる枠組みを提案することができたと考える。

第6章 結論

本研究では、従来の認知的音楽理論においてこれまで議論がなされてこなかった楽曲途中の階層構造に着目し、Generative Syntax Model(GSM)[23]に基づくジャズ和声の漸進的構造解析システムを実装した。これまでの認知的音楽理論についての議論においても、解釈の多様性や暗意・実現モデルのような音楽的期待の重要性が言われていたが、それらを GTTM[8] や GSM[23] といった木構造解析の枠組みに取り込んで、議論をするような手法を提案し、実装したことが本研究の主たる成果である。

漸進的構造解析は自然言語処理における手法 [29] を基に、ジャズ和声に対する GSM[23, 24] で提案されているジャズ和声に対する句構造規則を用いて実装を行った。また、構文規則を確率文脈自由文法 (PCFG) に拡張することで、構文木の生成確率をシステムが出力できるようにした。PCFG のパラメータ推定には Jazz Harmony Treebank を用いた。またシステムは、GUI アプリケーションとして実装し、Expectation-based Chord Sequence Analyzer と名付けた。本アプリケーションでは、ユーザがコード進行を入力すると、それに対する漸進的な解析結果が木構造の画像と共にグラフィカルに表示することができる。また、ユーザによって各構文規則の確率値の確認や再設定ができるようになっている。

本システムを用いて、ジャズスタンダードと呼ばれるいくつかのジャズ楽曲で構造解析の実験を行った。コードが入力されるたびに複数の構文木が出現し、楽曲途中における解釈の多様性が確認できた。その中でも、構文木の生成確率が高いものは、より妥当な（一般的な）和声構造を表していると考えられる。また、コードが入力されていく過程を考えると、後続として既に期待されたものが入る場合と、期待された物は入らず、木構造の組み替えが生じる場合の 2 パターンがあることが確認できた。また、独自に定義した意外性の指標を用いることによって、コード進行のポイントとなる部分について議論した。

今後の課題として、楽曲全体といった長い入力への対応や、リズムや楽曲形式の考慮、ユーザ実験によるシステムの評価といったことがあげられる。現状のアルゴリズムでは、入力長の増加に伴って計算量が指数関数的に増加し、実行時間が長くなってしまう問題がある。今回は確率値の低い候補木の枝刈りを行うことによって対処をしたが、ある程度のかたまりをセクションとしてグループ化して解析を仕切り直したりする工夫も有効であろうと考えられる。これは、人間が楽曲を認知する場合は、楽曲の A セクションや B セクションといった和声構造より高次の階層のグループ化も同時に行っていると考えられることからも必要であると思われる。また本システムを使用することで、ユーザは楽曲全体を一度に解析した時には気づかなかったであろう、楽曲途中の認知構造を発見することができると考えられる。このような和声学習における実用性などについてシステムを用いたユーザ実験を行い、ジャズ和声の学習により有用であるシステムへと発展させていく必要があると考える。

将来的な展望として、漸進的解析という新たな解析の手法や考え方はリアルタイムアプリケーションへの応用が期待される。自動採譜や自動コード推定といった音楽認識系のタスクを逐次的に実行するには、人間が楽曲を認識するように前から漸進的に解析を行う必要があると考えられる。これまで音楽認識のタスクが逐次的に行われることはなかったが、この解析方法を応用することによってリアルタイム音楽認識といった新たな分野を創出することも可能ではないかと考えられる。さらに本システムは、生成文法を用いていることから生成系のアプリケーションへも親和性が高いと考えられる。ユーザに意外性を提示するような自動セッションシステムといったものも、漸進的な解析とそれに基づく生成によって可能になるであろう。

このように、漸進的解析によって、人間と同じ時間の流れで音楽を聴きながら、感動を共有するような機械の創出が可能であると考えている。

謝辞

本研究を進めるにあたり、日頃よりご指導・ご助言をいただいた桂田浩一准教授、大村英史講師に心より深く感謝申し上げます。

北陸先端科学技術大学院大学、東条敏教授には、本研究のテーマをご紹介いただき、背景知識に関する多くのことをご教授いただきました。厚く御礼を申し上げ、心より感謝いたします。

私がジャズサックス始めた頃、最初にジャズという音楽の理論的奥深さとその重要性について、懇切丁寧に教えていただいたジャズサックス奏者の多田誠司先生に心より感謝いたします。また、フルート奏者の松村拓海先生にもジャズ理論をご指導いただきました。深く感謝いたします。

そして、ジャズという音楽の楽しさ、難しさを共有し、共に練習に励んでくれた東京理科大学ジャズ研究会の皆様、早稲田大学フュージョンマニアの皆様に感謝いたします。

また、桂田研究室の皆様には日頃よりお世話になりました。厚く御礼申し上げます。

最後に、両親をはじめとして、これまでの大学生活を様々な面で支えてくださった全ての方々に心より深く感謝いたします。

参考文献

- [1] 小川容子, 谷口高士, 中島祥好, 星野悦子, 三浦雅展, 山崎晃男, 大串健吾, 桑野園子, 難波精一郎. 音楽知覚認知ハンドブック : 音楽の不思議の解明に挑む科学. 北大路書房, 2020.
- [2] 石輪悠貴. 改良 TPSに基づいた GTTM における Cadential Retention (修士論文), 2018.
- [3] 新編 音楽中辞典. 音楽之友社, 2002.
- [4] 小山大宣. *Jazz Theory Workshop*. 武蔵野音楽院出版部, 2004.
- [5] 東条敏, 平田圭二. 音楽・数学・言語 -情報科学が拓く音楽の地平-. 近代科学社, 5 2017.
- [6] Mark Levine. *The Jazz Theory Book*. Sher Music, 1995.
- [7] 菊地成孔, 大谷能生. 豪鬱と官能を教えた学校—【バークリーメソッド】によって俯瞰される 20世紀商業音楽史. 河出文庫, 2004.
- [8] Fred Lerdahl and Ray S. Jackendoff. *A Generative Theory of Tonal Music*. The MIT Press, 1996.
- [9] Fred Lerdahl. Genesis and architecture of the GTTM project. In *Music Perception*, Vol. 26, pp. 187–194, 2009.
- [10] 平賀譲. 音楽における構文性とパターン性. 人工知能学会全国大会論文集, 2015.
- [11] Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. ATTA: Automatic time-span tree analyzer based on extended gttm. 2005.
- [12] Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. The University of Chicago Press, 1992.
- [13] Terry Winograd. Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, Vol. 12, pp. 2–49, 1968.
- [14] Satoshi Tojo, Yoshinori Oka, and Masafumi Nishida. Analysis of chord progression by HPSG. pp. 305–310. ACTA Press, 2006.
- [15] Mark Granroth-Wilding and Mark Steedman. A robust parser-interpreter for jazz chord sequences. *New Music Research*, Vol. 43, pp. 354–374, 2014.

- [16] Takeshi Fukunari, Sean Arn, and Satoshi Tojo. Ccg analyzer with tonal pitch space for non-classical chords. pp. 239–246. Institute of Electrical and Electronics Engineers Inc., 11 2016.
- [17] Satoshi Tojo. Modal logic for tonal music. 2019.
- [18] 東条敏. 言語・知識・信念の論理. 株式会社 オーム社, 2006.
- [19] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, Vol. 5, pp. 35–53, 2011.
- [20] Noam Chomsky. Syntactic structures. 1957.
- [21] Noam Chomsky. Aspects of the theory of syntax. 1965.
- [22] Noam Chomsky. *The minimalist program*. MIT press, 1995.
- [23] Martin Rohrmeier. The syntax of jazz harmony: Diatonic tonality, phrase structure, and form. *Music Theory and Analysis (MTA)*, Vol. 7, , 2020.
- [24] Daniel Harasim, Christoph Finkensiep, Petter Ericson, Timothy J O'donnell, and Martin Rohrmeier. The jazz harmony treebank. In *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020.
- [25] Joakim Nivre. Dependency grammar and dependency parsing. *MSI report*, Vol. 5133, , 2005.
- [26] Mark Steedman and Jason Baldridge. *Combinatory Categorial Grammar*. 2011.
- [27] Daniel Shanahan and Yuri Broze. A diachronic analysis of harmonic schemata in jazz. pp. 909–917, 2012.
- [28] Yuta Ogura, Hidefumi Ohmura, Yui Uehara, Hidefumi Tojo, and Kouichi Katsurada. Expectation-based parsing for jazz chord sequences. In *Proceedings of the 17th Sound and Music Computing Conference*, pp. 350–356, 2020.
- [29] Shigeki Matsubara, Satoru Asai, Katsuhiko Toyama, and Yasuyoshi Inagaki. Chart-based parsing and transfer in incremental spoken language translation. pp. 521–524, 1997.
- [30] 長尾真. 自然言語処理. 岩波書店, 1996.