# Estimating Unexpectedness in Jazz Harmony with a Probabilistic Incremental Parser

**Yuta Ogura** (小椋裕太)[1,2], **Hidefumi Ohmura** (大村英史)[2], **Satoshi Tojo** (東条敏)[3], and **Kouichi Katsurada** (桂田浩一)[2]

[1]**Yamaha Corporation**, 10-1 Nakazawa-cho, Naka-ku, Hamamatsu, Shizuoka, 430-8650, Japan
[2]*Department of Information Sciences*, **Tokyo University of Science**, 2641 Yamazaki, Noda, Chiba, 278-8510, Japan
[3]*Graduate School of Information Science*, **Japan Advanced Institute of Science and Technology (JAIST)**, 1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan

## ABSTRACT

Cognitive music theory analyzes the listeners' understanding of music. The generative syntax model (GSM) has shown that the structure of expectation-realization in a harmonic progression becomes recursive and hierarchical, in terms of context-free grammars. However, GSM only takes into consideration the cognitive structure after listening and does not discuss the dynamic process during listening, but given that music is a temporal structure of sound, dynamic changes in cognitive structure are more important. In this study, we extend the GSM by using probabilistic context-free grammar to represent the cognitive structure for each successive chord. Furthermore, we implemented a harmonic analysis system based on the extended model. We use a jazz standard, a genre of music in which harmonic progression is particularly important, as a case study, analyze it, and show its efficacy. The experimental result quantified its unexpectedness, appearing in the middle of a piece of music.

## 1. INTRODUCTION

The origin of music is said to be closely related to the evolution of language [1], and thus, "what is music?" is a historically abstruse question. The first theorized music seems Pythagoras' pitch in ancient Greece, however, along with the history the music has diversified into various genres, and in accordance with the theories also have been complicated. On the other hand, many fundamental questions, such as "how do we understand music?" still remain unclear. Cognitive music theory focuses on such questions. Cognitive music theory analyzes music based on the cognitive processes of the "listener", whereas a general music theory is used as a tool for the music "creator", that is, to compose and arrange music [2, 3].

In cognitive music theory, there is a method of analyzing music as a hierarchical structure. The notion of a hierarchical structure of a piece of music originated from the reduction hypothesis proposed by Schenker [4]. The reduction hypothesis states that "a listener of a piece of music will try to organize all pitch events (notes and chords) into a hierarchical structure of relative importance." Lerdahl and Jackendoff's *A generative theory of tonal music* (GTTM) [5] analyzes the melody of a piece as a hierarchical structure [6–8]. *The generative syntax model* (GSM) by Rohrmeier focuses on harmony and defines context-free rules for harmonic progressions [9]. This allows the hierarchical structure of harmonic cognition to be represented as a tree structure, as shown in Figure 1.

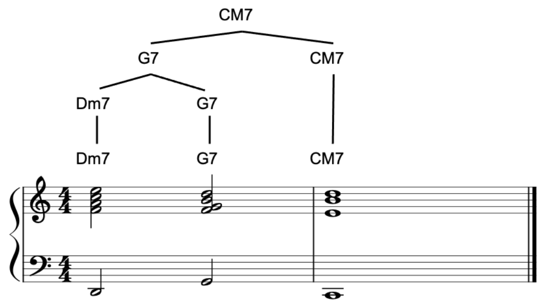


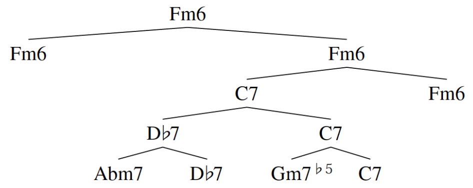

Figure 1. Structural analysis of ii-V-I based on GSM

However, GSM only takes into consideration the cognitive structure after listening and does not discuss the dynamic process during listening, but given that music is a temporal structure of sound, dynamic changes in cognitive structure are more important. The philosopher Meyer states that "the meaning of music arises from the relation of sounds in which the preceding sound somehow **expects** the following sound, and the embodiment of the following sound tries to confirm or review the preceding sound [10]." The interest in music is formed by incremental cognition.

Furthermore, in the original GSM, we cannot compare the tree structure if multiple analyses are probable due to generative syntax. This is because there is no concept of probability. The degree of expectation in the middle of a piece can be expressed by probabilities.

In this study, we focused on the incremental cognition of music. To clarify the cognitive structure for each successive chord, we propose the application of incremental chart parsing [11] to GSM. In addition, we extend the grammatical rules of GSM to probabilistic context-free grammar, to enable a quantitative discussion of unexpectedness in a harmonic progression. This makes it possible to compare the importance of different tree structures. The proposed method is implemented on a computer, and incremental analysis is performed on a jazz piece to discuss where the unexpectedness occurs within the music.

This paper is organized as follows: In the following section, we summarize the theory of GSM; In section 3, we detail the mechanism of incremental parser and probabilistic context-free grammar; In section 4, we propose a method for the evaluation of unexpectedness; In section 5, we show an example of incremental analysis with a jazz chord sequence and discuss the unexpectedness; In section 6, we summarize our contributions.




(a) Harmonic syntax tree based on GSM


(b) Harmonic expectation–realization structure

Figure 2. Syntax tree example about the final chords of the jazz standard Birks's Works [16]

## 2. GENERATIVE SYNTAX MODEL

### 2.1 Overview

The GSM [9, 12] is a cognitive music theory proposed by Rohrmeier. GSM is a model that represents the cognitive structure of a musical piece as a tree structure, similar to GTTM [5], a well-known cognitive music theory. Whereas GTTM proceeds without explicit context-free rules, GSM is strongly based on Chomsky's generative grammar theory [13–15], and proceeds with explicit context-free rules for harmonic progressions.

GSM makes the following assumptions about harmonic cognition: one chord has a dependency relationship with the chords before and after it. In particular, an adjacent chord has a "functional head," in which the dominant chord governs a broader time interval absorbing surrounding pitch events.

There are several versions of the phrase structure rules presented in the GSM, depending on the type of music. In the following, we will focus on the rule [12, 16] proposed for jazz music, which is the subject of this study.

All syntactic rules presented in the GSM are said to follow either the **Prolongation principle** or the **Preparation principle** . Figure 2 shows the GSM analysis of the jazz standard Birk's Works (Fm6 A♭m7 D♭7 $\mathrm{Gm7}^{\flat 5}$ C7 Fm6) In the following, We explain the principle of Prolongation and Preparation using this analysis.

The initial Fm6 established the tonic and as such creates the expectation that the progression ends with Fm6. The chords A♭m7 and D♭7 function as the tritone-substituted subdominant and dominant of C7, respectively. They therefore create expectation that resolves in the (temporally distant) chord C7. $\mathrm{Gm7}^{\flat 5}$ can be thought of as a subdominant chord in the F minor key. It herefore creates expectation that resolves with the dominant chord C7 wich itself resolves into the last tonic chord Fm6. We say that the tonic chords constitute a **Prolongation**. The subdominant chords **Prepare** the dominant chords and the dominant chords **Prepare** the tonic chord. Abstractry, we say that a chord $X$ *refers* to a chord $Y$ if $X$ either prolongs or prepares $Y$.

Figure 2b illustrates the structure of expectation realization in the harmonic progression. Chord pairs, represented by arrows in the diagram, are based on the preparation principle, in the sense that the former chord serves as a preparatory function for the following chord, while pairs such as Fm6 - Fm6, without arrows, are based on the prolongation principle. This is in one-to-one correspondence with the tree structure in Figure 2a, so it can be said that the tree structure of harmonic progressions reveals the structure of expectation–realization in harmonic progressions.

There are also two types of prolongation principles: strong prolongation and weak prolongation. A strong prolongation is an extension with the same chord type of the same root, while a weak prolongation is an extension of a chord with the same function (e.g., prolongation by C and Am in the key of C major). A strong prolongation is represented as $X \to X\ X$ for any chord symbol $X$ (e.g., Fm6 → Fm6 Fm6). Weak prolongation is represented as $X \to Y\ X$ or $X \to X\ Y$ with respect to functionally equal chord symbols $X, Y$ (e.g., Fm6 → A♭ Fm6). The preparation principle is represented by $X \to Y\ X$ for chord symbols $X, Y$ that are not functionally equal (e.g., Fm6 → C7 Fm6).

The prolongation and preparation principles can be summarized as follows:

$$\begin{array}{rlcl} \textbf{Strong Prol.} & X & \to & X\ X \\ \textbf{Weak Prol.} & X & \to & Y\ X \mid X\ Y \\ \textbf{Preparation} & X & \to & Y\ X \end{array}$$

These syntactic rules are characterized by the fact that non-terminal symbols do not have their own category but are expressed in the form of a binary tree, where the left-hand side symbol always appears on the right-hand side. This feature can also be seen in grammatical theories such as dependency grammar [17] and combinatorial category grammar (CCG) [18]. The symbol appearing on both sides of the arrow is called the **head**. In the setting of GSM, the prolonged (resp. prepared) chrod is the head. Therefore, the preparation rule is always right-headed, but the weak prolongation rule can be left-headed or right-headed de-

**Algorithm 1** Algorithm of incremental parsing

```
function CHART_PARSING(G_chart, w)
    L_chart ← {}        \*Local charts*\
    temp ← {}

    \*step1 Lexicon Consultation*\
    for α ∈ Lexicon do
        if w = α then
            L_chart ← L_chart ∪ {[w]_α}

    \*step2 Rule Application*\
    for σ ∈ L_chart and β → β_1β_2 ... β_n ∈ Rules do
        if σ = β_1 then
            L_chart ←
              L_chart ∪ {[σ[?]_β2]_β]}

    \*step3 Term Replacement*\
    for φ ∈ G_chart and ψ ∈ L_chart do
        if γ = lut(φ) ∧ γ = ψ then
            replace lut(φ) with ψ
            temp ← temp ∪ {φ}
    G_chart ← temp        \*Global charts*\
    return G_chart

\*main*\
G_chart ← [?]_S        \*initialize*\
for i=1,...,last do
    w_i ← input_chord
    G_chart ← CHART_PARSING(G_chart, w_i)
```

pending on the interpretation.

### 2.2 Jazz Harmony Treebank

The Jazz Harmony Treebank (JHT)[1] [16] is a dataset annotated with the results of the hierarchical analysis of harmonic progressions in jazz standards by experts. Hierarchical analysis was based on the aforementioned GSM principles. In this study, it was used as a corpus to estimate the probability of applying the probabilistic context-free grammar described below.

The analysis of JHT is based on 150 jazz tunes in the genres of Swing, Bossa Nova, Jazz Blues, Bebop, Cool Jazz, and Hard Bop, and does not include non-tonal genres such as Modal Jazz, Free Jazz, and Modern Jazz.

## 3. INCREMENTAL STRUCTURAL ANALYSIS

### 3.1 Incremental Chart Parsing

We have proposed a model that displays the tree structure, for each successive chord, by incrementally analyzing harmonic progressions [19]. This model was realized by applying a natural language parsing method, incremental chart parsing [11], to the GSM. Here, we explain this algorithm, which is a natural language processing technique, we use the word "word" to describe it, but in harmony, "word" refers to a chord symbol (e.g., CM7, G7, etc.).

[1] https://github.com/DCMLab/JazzHarmonyTreebank

In natural language processing, the input is a sequence of words spaced by blanks. Each word is positioned by numbers, called *nodes*, placed at the blanks between words; thus, word $w_i$ resides between node $i-1$ and node $i$.

An *edge* combines one node with another. A tree is represented by a data structure, called *term*; when $\alpha$ belongs to category $X$, we write it as $[\alpha]_X$. Here, $\alpha$ is either a word (chord), a term, or a list of terms. A *chart* consists of an edge and term. For example, when a chart is $(i, j)$ and $[[\alpha]_Y[\beta]_Z]_X$, it represents a (local) tree obtained by an application of production rule '$X \rightarrow Y\ Z$' between nodes $i$ and $j$ to the sequence of $\alpha\beta$, being recognized by $\alpha$ and $\beta$ belonging to $Y$ and $Z$, respectively. In contrast, an edge can possess multiple terms; that is, there may be multiple parse trees on the edge. Thus, there might be different interpretations of edges. The term displayed by $[?]_X$ is called an undecided term, where the content of category $X$ is not decided. When an undecided term resides on an edge, the edge is called *active*; otherwise, *inactive*.

In incremental chart analysis, when the $i$-th word $w_i$ is input, the following operations are performed sequentially:

**Lexicon Consultation** When the category of $w_i$ is $X$, add an inactive edge labeled by term $[w_i]_X$ on $(i-1, i)$.

**Rule Application** When there exists an active edge labeled by term $[\ldots]_X$ on $(i-1, i)$, for all grammar rules such as $A \rightarrow XY \ldots Z$, add an edge labeled by term $[[\ldots]_X[?]_Y \ldots [?]_Z]_A$ on $(i-1, i)$.

**Term Replacement** Let $\phi$, $\psi$ be terms, and $[?]_X$ be the leftmost undecided term of $\phi$ labeled on $(0, i-1)$. If the category of $\psi$ labeled on $(i-1, i)$ is $X$, add an edge labeled by a term that replaces the leftmost undecided term of $\phi$ with $\psi$ to $(0, i)$.

Algorithm 1 shows the above operations in pseudo-code. In pseudo-code, an edge is represented by a pair of indices in the array; hence, an edge is not mentioned explicitly. Furthermore, $\sigma, \psi, \phi$, and $\gamma$ represent terms, and when terms are connected by equals (=), it indicates that the outermost categories are equal. In addition, we denote the left-most undecided term of term $\phi$ as $lut(\phi)$.

In general, chart parsing takes the whole sentence as an input and constructs a tree. On the other hand, in incremental chart parsing, parts of a sentence are input sequentially, and the tree is constructed incrementally. There are two types of algorithms for chart analysis: **bottom-up** and **top-down**. The bottom-up algorithm starts with a word and builds a tree toward the start symbol S. The top-down algorithm starts with the start symbol S and builds a tree toward the leaves, that is, the word. A combination of bottom-up and top-down algorithms can be used to deal with sequential inputs. Therefore, incremental chart parsing introduces two top-down operations into the bottom-up chart analysis, namely, the operation of applying a grammar rule to an active arc and the operation of replacing the leftmost undecided term, of a term labeled with an active arc, with a term labeled with another active arc. In the actual system, only the global chart at each stage is displayed. In this study, we refer to these terms as **candidate trees**.

In addition, it is necessary to initialize the global chart with an undecided term whose category is the start symbol S. In this paper, following tonic chord for all 12 keys are used for start symbol S.

$$\mathrm{S} = \{\mathrm{C}, \mathrm{D}\flat, \mathrm{D}, \ldots, \mathrm{B}\} \times \{\mathrm{M}, \mathrm{M7}, \mathrm{m}, \mathrm{m7}\} \tag{1}$$

### 3.2 Probabilistic Context Free Grammar

Probabilistic context-free grammar (PCFG) models extend context-free grammars and can calculate the probability of occurrence of a syntax tree [20]. This model assigns the following conditional application probabilities to each generative rule in the grammar $\mathrm{A} \to \alpha$.

$$P(\mathrm{A} \to \alpha | \mathrm{A}) \tag{2}$$

Since it is a conditional probability, the following equation holds.

$$\sum_{\alpha} P(\mathrm{A} \to \alpha | \mathrm{A}) = 1 \tag{3}$$

In other words, the sum of the probabilities of applying the generative rules, with the same non-terminal symbol (pre-terminal symbol[2] ) on the left side was 1. The simplest way to calculate such a probability is to use a parsed corpus. The probability of applying the generative rule can be calculated as follows:

$$P(\mathrm{A} \to \alpha | \mathrm{A}) = \frac{\text{Number of } \mathrm{A} \to \alpha \text{ in the corpus}}{\text{Number of A in the corpus}} \tag{4}$$

In Equation (4), the denominator is the number of occurrences of non-terminal symbols in the corpus, and numerator is the number of times the generative rule is used.

Obviously, expression 3 is satisfied. In addition, given the application probabilities in this way, the generation probability of a certain tree structure $t$ can be given by the product of the application probabilities of all the generative rules that make up the tree structure.

To prevent the exponential increase in analysis time with longer sentences in the incremental chart analysis, we performed branch trimming, using the generation probability of the tree structure at each word stage. The terms stored in the global chart, at the time of each word, up to the top 100 terms in probability, were retained for the analysis of the next word.

### 3.3 Expectation-based Chord Sequence Analyzer

In this study, we implemented a GUI application called **expectation-based chord sequence analyzer** (ECSA)[3] . The main purpose of this application is to intuitively understand the harmonic structure.

Figure 3 shows ECSA's main view. When we enter a chord sequences in the text box on the top page, the results of the tree structure analysis for the input are displayed in multiple lines. Each line shows the result of the analysis up to the point when the chord was entered. Figure 3 is the analysis result of jazz standard *Cute*, which will be explained in section 5. The first line shows the results of the analysis at the stage when chords up until Dm7 are input, and the second line shows the results when up to when Dm7 and G7 are input. Each parse tree, at that time, is displayed in a slider (carousel) panel. At the top of each panel, the generation probability of the parse tree is shown, and the parse trees are sorted from left to right with the highest probability. Also, the number next to the label of each chord name shows the unexpectedness measure $U$ that will be described in section 4.

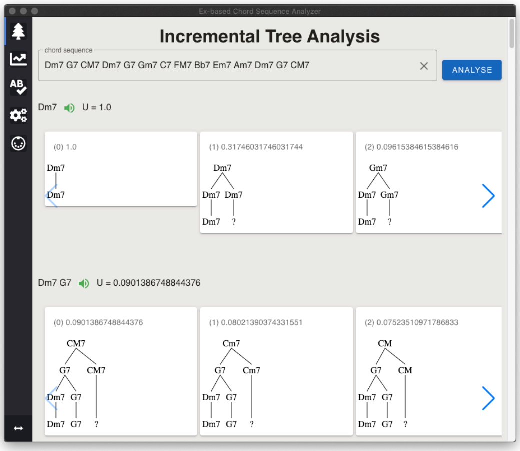



Figure 3. Appearance of Expectation-based Chord Sequence Analyzer

## 4. EVALUATION OF UNEXPECTEDNESS

In this study, unexpectedness is considered to arise from expectation–realization and expectation–deviation. A harmonic progression with "low" unexpectedness is a harmonic progression in which the expectation of the preceding chord is realized by the following chord. A harmonic progression with "high" unexpectedness is a harmonic progression in which the chord deviates from the expectation set up by the preceding chord.

This expectation–realization and expectation–deviation depends on the growth process of the tree. In a harmonic syntax tree in the middle of a piece of music, the chords expected to follow are represented as categories of undecided terms such as $[?]_{\mathrm{CM7}}$. In the next step, we consider an expectation–realization to have occurred when chord CM7 is actually input, and an expectation-deviation to have occurred when another chord is input. In the following, we refer to the stage in the middle of a piece, where a certain chord is input as the **chord step**.

The change in the generation probability of the tree structure is also important for unexpectedness. In general, there are multiple candidate trees for each chord step, and the

[2] This is the equivalent of phrases such as NP and VP in natural language processing. In this study, we follow the example of [12, 16] and use a grammar rule that equates non-terminal with pre-terminal symbols, namely there are no lexicons.

[3] https://github.com/yutaogura/Ex-based-Analyzer

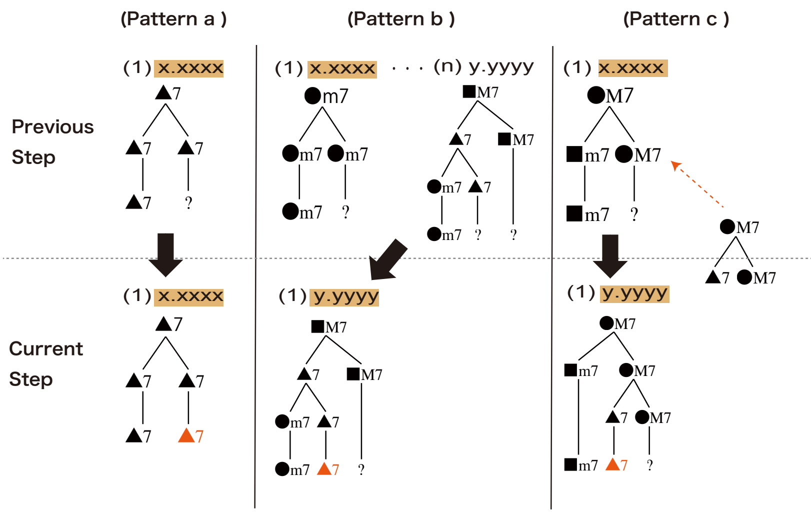



Figure 4. Tree structure change pattern with maximum generation probability

candidate trees are ranked by their generation probabilities. In this study, we focus on the tree structure with the highest generation probability, at each chord step, as a representative of the cognitive structure at each chord step.

Figure 4 shows a schematic diagram of how the tree structure, with the highest generation probability, changes at a given chord step. The numbers in parentheses indicate the rank of the probability at that chord step. In the following, the tree structure from which a certain tree structure is grown is called the derivation tree structure. The thick solid arrows show the relationship between the source and destination of the tree.

**Pattern a** and **b** in Figure 4 show how the chord expected in the previous chord step is realized in the current step. In **Pattern a**, there is no change in the value of the probability of generating the tree with the highest probability, but in **Pattern b**, the value of the probability of generating the tree with the highest probability has changed because the rank of the tree structure that was n-th in the previous step has become the first. **Pattern c** shows the input of a chord that was not expected in the previous step. In this case, a new substructure is added (dotted arrow in the figure), and the value of the probability of generating the tree with the highest probability changes.

Based on the above discussion, we formulate a measure of unexpectedness that considers the degree of increase in rank and the addition of substructures. Let $t^{(n)}$ denote the candidate tree structure at a certain chord step n, and let $t^{(n)}_{\mathrm{maxprob}}$ denote the tree structure with the highest generation probability, and $P(t^{(n)}_{\mathrm{maxprob}})$ denote the probability value. The tree structure from which $t^{(n)}$ is derived is denoted as $t^{(n-1)}$. In this case, the unexpectedness $U^{(n)}$ of a chord step n is given as follows:

$$t^{*} = t^{(n)}_{\mathrm{maxprob}} \tag{5}$$

$$A = \frac{P\left(t^{*(n-1)}\right)}{P\left(t^{(n-1)}_{\mathrm{maxprob}}\right)} \tag{6}$$

$$B = \frac{P\left(t^{(n)}_{\mathrm{maxprob}}\right)}{P\left(t^{*(n-1)}\right)} \tag{7}$$

$$U^{(n)} = \begin{cases} P\left(t^{(n)}_{\mathrm{maxprob}}\right) & (n = 1) \\ A \times B = \frac{P\left(t^{(n)}_{\mathrm{maxprob}}\right)}{P\left(t^{(n-1)}_{\mathrm{maxprob}}\right)} & (n > 1) \end{cases} \tag{8}$$

The closer the measure of unexpectedness $U$ is to 1, the more the expectation–realization has occurred, meaning a "low" unexpectedness, and the closer it is to 0, the more the expectation–deviation has occurred, meaning a "high" unexpectedness. The $A$ represents the scarcity of the source tree structure in the previous chord step, that is, the increase in rank (Eq. 6). Also, $B$ represents the probability of generating the newly added substructure (Eq. 7). The unexpectedness measure $U$ is a combination of these. In the actual calculation, the $P(t^{*(n-1)})$ parts cancel each other out, so in the end, $U$ is just the ratio of the probability of generating the tree with the highest probability before and after the target chord step.

## 5. CASE STUDY WITH A JAZZ CHORD SEQUENCE

In this study, we present an example of incremental structural analysis using an actual jazz standard *Cute*.

*Cute* consists of 32-bars ABAC form. In this section, we analyze the AC part, which is the second half of the 16-bars. The chord progression and lead melody are shown in Figure 5.

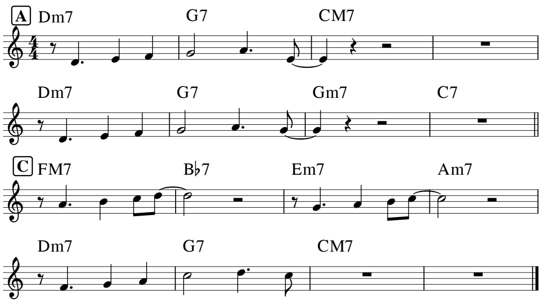



Figure 5. The chord progression and lead melody of *Cute* (second half 16-bars)

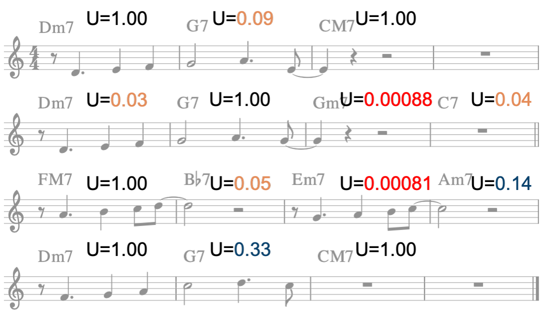



Figure 6. $U$ value at each chord step of *Cute*

Before discussing the unexpectedness of the piece, using incremental structural analysis, let us review the basic characteristics of the piece from the perspective of conventional music theory [21, 22]. *Cute* is in the key of C major. This is evident from the fact that the last chord of the song ends with CM7. The first four measures, Dm7 – G7 – CM7, are two-five-one of the C major key. The following bars 5 and 6 are also two-five-one in the key of C major, and in the seventh bar, chord CM7, which is tonic, is expected to come, but the chord Gm7 is inserted, and from here, Gm7 – C7 – FM7, two-five-one in the key of F major, begins. This is followed by B♭7, a subdominant minor, and Em7, a diatonic chord in the key of C major, and then iii – vi – ii – V – I, leading to tonic CM7.

The results of the tree structure analysis are shown in Figure 7. The figure shows the tree structure output by the system for each chord step. The system can display up to the maximum number of candidate trees for branch trimming at each chord step, but only a few of them are shown for the sake of space limitations. (a)–(h) show the chord steps in the chord progression of *Cute*. In the upper part of each tree structure, the generation probability of the candidate tree and the ranking of the generation probability for each chord step are shown in parentheses. In the following, a candidate tree whose probability of generation is *n*-th in chord step (a) is denoted as (a–n). '?'(question mark) denotes an undecided term and indicates the next expected chord or category. The rank in parentheses with an asterisk (*) indicates an inactive arc, that is, a closed tree structure.

If we look at the growth process of the tree structure in order, we can see that at each chord step, various chords are expected to be realized in the next step, and the tree structure is recombined. Looking at the evolution of the tree structure, up to (d–1), we can see that the tree grows as (a–4)→(b–1)→(c–2)→(d–1). CM7 is the tonic in this piece, and the progression of Dm7 – G7 – CM7 forms a group. Thus, the closed tree structure is considered to be harmonically stable at this point.

Next, the unexpectedness value $U$ is calculated for all the chord steps, as shown in Figure 6. It can be seen that the value of $U$ decreases from the 6th to the 7th bars, which is the part where two-five-one in the key of F Major appears. In the actual tree structure, G7 is expected to be the parent of Gm7 in (f–1), but in (g–1), the insertion of C7 causes a recombination of the tree structure, and FM7 is expected to be the parent.

Then, FM7 at the 9th bar is inserted such that the expectation of 8th bars is realized, and the value of $U$ is lowered again in the following B♭7 and E7. This is thought to be caused by the non-diatonic chord B♭7. In general music theory, B♭7 is considered to be a sub-dominant minor chord. It comes from the iv chord in key of C minor, which is the parallel key[4]. Together with FM7, FM7–B♭7 this progression is famous for the formation of a subdominant-subdominant minor chord progression. In this case, FM7 is often analyzed as working as a pivot chord[5], while tonally it remains in C major. Therefore, the fact that the value of $U$ is lower in Em7, which is often analyzed as a tonic in C major, should be reconsidered as whether it has cognitive reality[6] or not.

## 6. CONCLUSION

In this study, we focused on the cognitive structure for each successive chord and proposed an incremental structural analysis of jazz harmony, based on the generative syntax model [9, 12]. Especially, we have employed probabilistic context-free grammar (PCFG) instead of traditional CFG, and thus, we could externalize the unexpectedness $U$, concerning the growth process of syntactic tree. Through the analysis of jazz music, using the implemented system ECSA, it became possible to quantitatively evaluate the position of unexpectedness in the music.

The importance of expectation-realization in music cognition has been discussed in Narmour's implication-realization model [23], but it was limited to the analysis of an entire piece. The main contribution of this study

[4] A major scale and a minor scale that have the same tonic are called parallel keys. In this case, C major and C minor is parallel key.

[5] A chord that has a function across multiple tonalities. In the case of FM7, one is I (tonic) in the key of F major, another is IV (subdominant) in the key of C major.

[6] When a concept or model can rationally explain a cognitive or psychological phenomenon, it is said that the concept or model has cognitive reality.

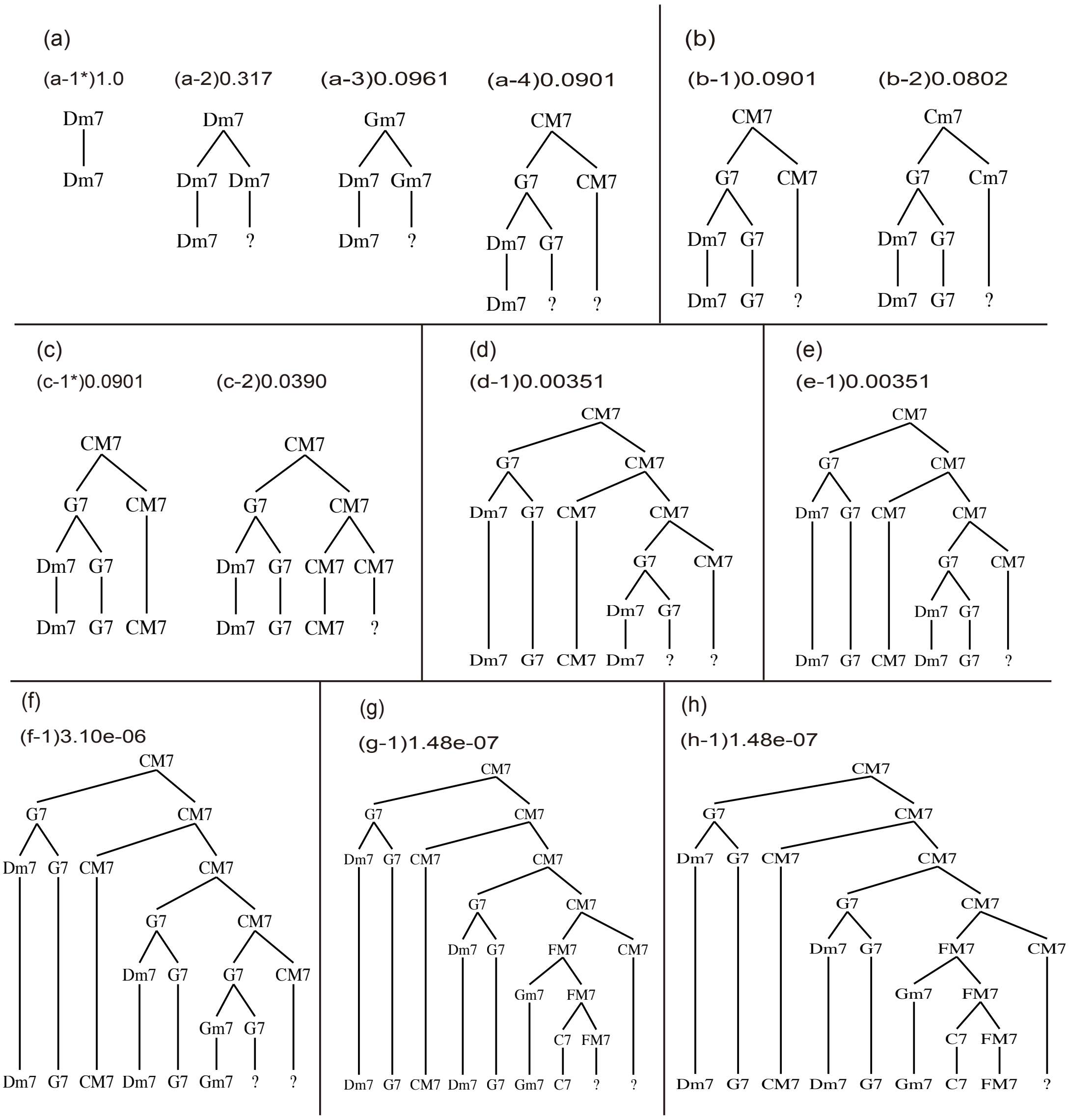



Figure 7. Incremental structural analysis of *Cute*

is to propose an analysis method that is closer to human music cognition by representing expectation-realization or expectation-deviation for each successive chord, using an incremental structural analysis method.

As a future work, we believe there is an advantage in investigating the cognitive reality of the measurement of unexpectedness $U$ by experiment. The reason for this is that there is a difference between the position of a piece of music, that we consider surprising based on conventional music theory, and the position where unexpectedness occurs quantitatively using the measurement of unexpectedness $U$. The position of unexpectedness in a piece of music is thought to vary greatly depending on the individual's musical experience. Therefore, we need to creat a measure reflecting the cognitive differences between individuals, to go back to the grammatical rules themselves and to examine their rationality.

As a possible application, we are considering incorporating it into real-time applications such as automated session systems, taking advantage of the incremental analysis of the sequential interpretation of music flow.

**Acknowledgments**

This work was supported by JSPS Kaken 16H01744 and 20K12126.

## 7. REFERENCES

[1] N. L. Wallin, B. Merker, and S. Brown, *The origins of music*. MIT Press, 2000.

[2] S. Tojo and K. Hirata, *Music · Mathmatics · Language -The horizons of music opened up by information science-*. Kindaikagaku co., 5 2017.

[3] K. Ohgushi, S. Kuwano, S. Namba *et al.*, *Music Perception Handbook: Science that challenges the mysterious perception of music*. Kitaohji Shobou co., 2020.

[4] H. Schenker, *Der Freie Satz. Neue musikalische Theorien und Phantasien, Margada, Liège, Belgium*, 1935.

[5] F. Lerdahl and R. S. Jackendoff, *A Generative Theory of Tonal Music*. The MIT Press, 1996.

[6] S. Tojo, Y. Oka, and M. Nisida, "Analysis of chord progression by HPSG." ACTA Press, 2006, pp. 305–310.

[7] M. Granroth-Wilding and M. Steedman, "A robust parser-interpreter for jazz chord sequences," *New Music Research*, vol. 43, pp. 354–374, 2014. [Online]. Available: http://jazzparser.

[8] T. Fukunari, S. Arn, and S. Tojo, "CCG analyzer with tonal pitch space for non-classical chords." Institute of Electrical and Electronics Engineers Inc., 11 2016, pp. 239–246.

[9] M. Rohrmeier, "Towards a generative syntax of tonal harmony," *Journal of Mathematics and Music*, vol. 5, pp. 35–53, 2011.

[10] L. B. Meyer, "Meaning in music and information theory," *The Journal of Aesthetics and Art Criticism*, vol. 15, no. 4, pp. 412–424, 1957.

[11] S. Matsubara, S. Asai, K. Toyama, and Y. Inagaki, "Chart-based parsing and transfer in incremental spoken language translation," in *Proc. the 4th Natural Language Processing Pacific Rim Symposium*, 1997, pp. 521–524.

[12] M. Rohrmeier, "The syntax of jazz harmony: Diatonic tonality, phrase structure, and form," *Music Theory and Analysis (MTA)*, vol. 7, 2020.

[13] N. Chomsky, "Syntactic structures," 1957.

[14] ——, "Aspects of the theory of syntax," 1965.

[15] ——, *The minimalist program*. MIT press, 1995.

[16] D. Harasim, C. Finkensiep, P. Ericson, T. J. O'donnell, and M. Rohrmeier, "The jazz harmony treebank," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020. [Online]. Available: https://github.com/DCMLab/JazzHarmonyTreebank

[17] J. Nivre, "Dependency grammar and dependency parsing," *MSI report*, vol. 5133, 2005.

[18] M. Steedman and J. Baldridge, *Combinatory Categorial Grammar*, 2011.

[19] Y. Ogura, H. Ohmura, Y. Uehara, H. Tojo, and K. Katsurada, "Expectation-based parsing for jazz chord sequences," in *Proceedings of the 17th Sound and Music Computing Conference*, 2020, pp. 350–356.

[20] M. Nagao *et al.*, *Natural Language Processing*. Iwanami Shoten co., 1996.

[21] D. Oyama, *Jazz Theory Workshop*. Musashino Conservatory Publishing Department, 2004.

[22] M. Levine, *The Jazz Theory Book*. Sher Music, 1995.

[23] E. Narmour, *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. The University of Chicago Press, 1992.