# Expectation-based Parsing for Jazz Chord Sequences

**Yuta Ogura**
Tokyo University of Science
6319508@ed.tus.ac.jp

**Hidefumi Ohmura**
Tokyo University of Science
ohmura@is.noda.tus.ac.jp

**Yui Uehara**
JAIST
yuehara@jaist.ac.jp

**Satoshi Tojo**
JAIST
tojo@jaist.ac.jp

**Kouichi Katsurada**
Tokyo University of Science
katsurada@rs.tus.ac.jp

## ABSTRACT

Many grammar theories of music have been proposed compared to language, and thus, the syntax has been revealed in a tree. In their parsing processes, trees are usually constructed by the generative rules given by the whole span of a music piece in a top-down manner. In this paper, we propose an incremental parser of music, from the left to the right of the scores, building partial trees. In this process, we can predict the succeeding phrase from the beginning part. Therefore, the generation process of the tree itself represents our human expectation process of music. We apply our method to jazz music represented by chord sequences, implement the parser, and exemplify the incremental generating process in several standard pieces.

## 1. INTRODUCTION

We recognize a progression of music by *Gestalt*. This means that while we cannot recognize a single tone as music, a group of sequential tones provides us with a meaningful phrase. For example, in jazz, a sequence of chords II-V-I (two-five-one) creates an explicit notion of progression, where II starts the progression, V makes us feel tense and the final I releases us from the tension. However, each single chord cannot be meaningful on its own.

In the dynamic process of forming Gestalt in our minds, Meyer [1] claimed that an emotion appears from mutual references among melodies and phrases. When a listener recognizes a phrase, he/she predicts how the phrase proceeds in the next time-step consciously or unconsciously. This prediction is called *expectation* and it enables us to discover the meaning of music.

It is widely accepted that music and language have the same origin [2], and Charles Darwin suggested that the music had been a media to pass love messages between men and women in the ancient era [3]. Therefore, various linguistic ideas have been considered to be imported into music analysis. A typical example is the idea of grammar in music. We can compose a constructive structure in music to represent such relations as mutual references, which often result in hierarchical tree. For example, in a II-V-I progression such as Dmin7-G7-Cmaj7, G7 possesses the function of dominant, and together with Dmin7, it forms a dominant phrase. A dominant phrase tends to proceed to a tonic chord of Cmaj7, and all the three chords form a tonic phrase (Figure 1) [4].

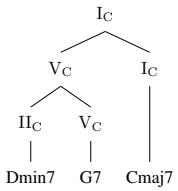


Figure 1. Hierarchical analysis of the most popular chord sequence II-V-I.

When we analyze such a structure in Figure 1, we need to consider the whole piece at a time as one static structure. Namely, if we assume a virtual machine to output such a hierarchical tree, we feed the whole piece as an input. This kind of static recognition does not reflect Meyer's view of expectation. If we are to implement such a process of forming an expectation, we need to feed a piece of music by one note by another consecutively to the virtual machine, and the tree structure should be composed dynamically, along with the progression of music, that is, from left to right on the score.

In this paper, we consider music as a dynamic cognitive process, and propose an incremental parser that composes a tree from a local phrase of the beginning part to a larger phrase including the succeeding part. This incremental parser explicitly represents our expectation, which suggests what follows from a half-way of a music piece.

In natural language processing, there have been studies on the incremental parser [5]. This parser composes a tree, one word by another from left to right, called Earley's algorithm [6] or in general a *chart parser* [7]. We employ this parser in our music recognition and expectation, thereby proposing a set of context-free grammar (CFG) rules. We implement a chart parser and apply it to the analysis of jazz-standards.

This paper is organized as follows: In the following section, we survey related works; In section 3, we detail the mechanism of our incremental parser; In section 4, we show our analyses with several examples; In section 5, we discuss the cognitive meaning of our parser; and in section

6, we summarize our contributions.

## 2. RELATED WORK

Linguistic approaches to music have been basically based on Chomskian generative syntactic theory [8, 9] with the notion of context-free grammar (CFG); each rule consists of one non-terminal symbol (or a variable) on the left-hand side of an arrow '$\rightarrow$' to *produce* a sequence of non-terminal or terminal (constant) symbols on the right-hand side. Since any of the non-terminal symbol on the left-hand side, regardless of its surrounding context (where the symbol appears), can be replaced by the sequence on the right-hand side. The rule is called *context-free*.

In this paper, we focus on Chomsky Normal Form (CNF) of CFG, where any CFG-rule can be rewritten by the following binary production.

$$\begin{cases} X \rightarrow Y\ Z \\ X \rightarrow a \end{cases}$$

where $X, Y$ and $Z$ are non-terminal symbols and $a$ is a terminal symbol. Furthermore, Chomsky proposed $X$-bar theory [10]; in each of a binary production in Chomsky Normal Form (CNF), one of the right-hand side non-terminal should be the head daughter category $X$ that would be the parent category $X'$ on the left-hand side, as follows:

$$\begin{cases} X' \rightarrow Spec\, X \\ X' \rightarrow X\, Comp \end{cases}$$

where $Spec$ stands for specifier and $Comp$ for complement. Namely, *Spec* rule represents the left-branching and *Comp* rule does the right-branching, respectively, among the binary trees.

Winograd [11] made efforts of applying CFG to music. Some works were based on specific grammar theories; Tojo employed Head-driven Phrase Structure grammar (HPSG) [12] and Steedman et al. did Combinatory Categorial Grammar (CCG) [13, 14]. Furthermore, In recent years, two distinguished works are shown; one is exGTTM by Hamanaka et al. [15] based on Generative Theory of Tonal Music (GTTM) [16] and the other is Generative Syntax Model (GSM) by Rohrmeier [4].

In GSM, we introduce the basic sets of regions, functions, key names, and chord names. The $P$ (piece) is the start symbol of production rules. It introduces $TR$ (tonic region), as $P \rightarrow TR$. In the next level, $TR$ generates $DR$ (dominant region) and $SR$ (subdominant region). In the further downward level they result in $t$ (tonic), $d$ (dominant), or $s$ (subdominant).

The current GSM employs abstract context free grammar (ACFG) [17], which is an extension of CFG, and can take advantage of the algebraic structure of music-specific categories. ACFG is defined by the 4-tuple: $G = (T, C, S, \Gamma)$ where $T$ is a set of *terminal symbols* which make up the actual content of the sentence / musical sequence; $C$ is a *constituent categories*; $S$ is a start symbol $S \subseteq C$; and a set of partial functions $\Gamma := \{r \mid r : C \rightarrow (T \cup C)^*\}$ is called rewrite rules.

For example, we apply ACFG to jazz chord progression. In the following, $\mathbb{Z}_n$ denotes the ring of integers modulo n $\in \mathbb{N}$. We introduce the terminal symbol $T$ as a set of chord symbols such as Cmaj7, Dmin7, .... The categories $C$ are modeled as pairs of scale degrees and keys: $C = \mathbb{Z}_7 \times k$, where $\mathbb{Z}_7$ indicates scale degrees that is designated by roman numerals and $k$ means an arbitrary key. The start symbol S directly introduces a category with degree I (e.g. $S \rightarrow \mathrm{I_C}$). The set of rewrite functions consists of $x_k \rightarrow (x + 4 \bmod 7)_k\ x_k$, where $x \in \mathbb{Z}_7$ is a scale degree. According to the definition above, we can get tree structure of chord progression as in Figure 1.

Tojo [18] proposed to employ the modal logic to represent internal references in music. Such modal operators as:

$\Box$ : For all the time points in some neighborhood region
$\Diamond$ : A time point exists in any neighborhood region

denote the *accessibility* from each pitch event in music to specified regions, and thus, we can clarify the inter-region relationship in a piece. For example, '$\Diamond$' operator represents the notion of expectation in the near future. Furthermore, '$\Box$' operator characterizes the local region (neighborhood) by a key, a local modulation, or a chord function. Considering $X$-bar rule, we regard the rule '$X' \rightarrow Spec\, X$' as an expectation of $X$ after observing *Spec*. Therefore, the future $X$ is written as $\Diamond X$. '$X' \rightarrow X\, Comp$' can be viewed as prolongation, and $X'$ area is characterized by $\Box X$. In this paper, we do not consider explicitly such modal operators nor $X'$ notations. However, we should be aware of such logical semantics behind the syntactic parser.

## 3. INCREMENTAL CHART PARSING

In this research, we implement an incremental chart parser, based on [19]. This parser consists of the binary branching rules *viz.*, CNF of CFG. By observing the first non-terminal symbol in the right-hand side, the parser predicts the second non-terminal symbol. These non-terminal symbols in production rules are called *categories*. The possible succeeding structures are generated along the parsing process by the dynamic programming.

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

An *edge* combines one node to 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 a 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 $X$ and $Y$, respectively. On the contrary, an edge can possess multiple terms, that is, there may be multiple parse trees on the edge. Thus, there might be different interpretations on the edge.

A term displayed by $[\langle u \rangle]_X$ is called an undecided term, where the contents of category $X$ is not decided. When

**Algorithm 1** Algorithm of incremental parsing

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

  *step1 Lexicon Consultation*
  **for** $X \in$ Lexicon **do**
    **if** $w = X$ **then**
      $L\_chart \leftarrow L\_chart \cup \{[w]_X\}$

  *step2 Rule Application*
  **for** $\sigma \in L\_chart$ **and** $A \rightarrow XY \ldots Z \in$ Rules **do**
    **if** $\sigma = X$ **then**
      $L\_chart \leftarrow$
        $L\_chart \cup \{[\sigma[\langle u \rangle]_Y \ldots [\langle u \rangle]_Z]_A\}$

  *step3 Term Replacement*
  **for** $\phi \in G\_chart$ **and** $\psi \in L\_chart$ **do**
    **if** $\exists \gamma\ \gamma = lut(\phi) \wedge \gamma = \psi$ **then**
      replace $lut(\phi)$ with $\psi$
      $temp \leftarrow temp \cup \{\phi\}$
  $G\_chart \leftarrow temp$  *Global charts*
  **return** $G\_chart$

*main*
$G\_chart \leftarrow [\langle u \rangle]_S$
**for** i=1,...,last **do**
  $w_i \leftarrow$ input_chord
  $G\_chart \leftarrow$ CHART_PARSING($G\_chart, w_i$)

an undecided term resides on an edge, the edge is called *active*; otherwise, *inactive*. In addition, We denote the leftmost undecided term of term $\phi$ as $lut(\phi)$.

In general, a chart parser employs two rules, *bottom-up* rule and *fundamental* rule. The charts obtained by the former rule are *local charts*, and by the latter, *global charts*. Therefore, the local charts represent local dependencies while the global charts represent the global dependencies from the beginning of a given sentence.

In Algorithm 1, we demonstrate the algorithm of an incremental chart parser. The addition of charts is executed by the following three steps: In Algorithm 1, an edge is represented by a pair of indices in the array; hence, an edge is not mentioned explicitly in the algorithm. Furthermore, $\sigma, \psi, \phi$, and $\gamma$ represent terms, and when terms are connected by equals (=), it indicates that the outermost categories are equal.

**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 $[[\ldots]_X[\langle u \rangle]_Y \ldots [\langle u \rangle]_Z]_A$ on $(i-1, i)$ exhaustively.

**Term Replacement** Let $[\langle u \rangle]_X$ 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 term that replaces the leftmost undecided term of $\phi$ with $\psi$ to $(0, i)$.

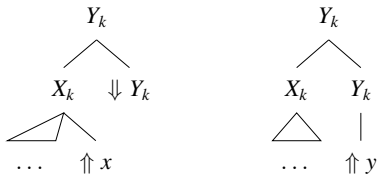


Figure 2. Incomplete sub-tree (left) and sub-complete tree (right)

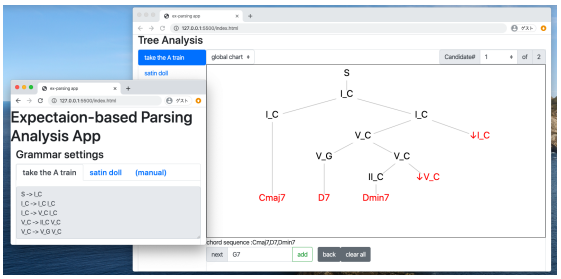


Figure 3. Web Application screenshot

For each input of a new word (chord), the above operations are executed.

A general incremental chart parser treats all rules equivalently in **Rule Application**, however, when there is a left-recursion rule[1] the process falls in an infinite loop. To avoid this, we place the upper limit of the number of application of a left-recursion rule; in this paper, we restrict the number to a maximum of one.

In this process, we may obtain those complete, closed trees that are half-way in a sentence/music piece. Since these are inactive, we name such trees as *sub-complete* trees; regarding them as stable structures residing in a music piece. To visualize such sub-completeness, we employ an uparrow ($\Uparrow$) beside the node to show the current input chord. Conversely, a down-arrow ($\Downarrow$) indicates undecided term. Since no undecided terms exist in a sub-complete tree, there is no down arrow as depicted in the right side of Figure 2.

The proposed algorithm was implemented as a web application and is publicly available[2] . The application was implemented by HTML and JavaScript. We confirmed that it works with the latest version of Google Chrome. The application has an interface as shown in Figure 3.

## 4. SAMPLE ANALYSES

In this section, we apply our incremental parser to two jazz-standards; *Take The A Train* and *Satin Doll*, both in C major, and clarify our expectations.

[1] A rule such that $X \rightarrow X \cdots$ .
[2] https://github.com/yutaogura/expectation-based-parsing

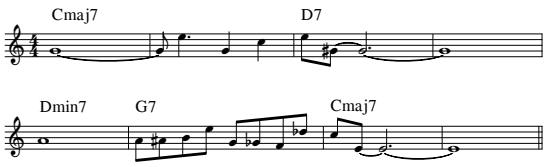


Figure 4. The sheet of *Take The A Train* A-Part

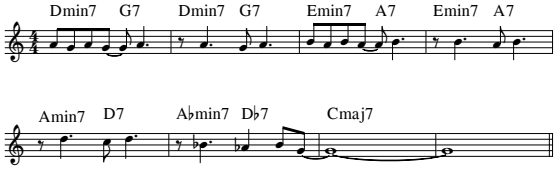


Figure 5. The sheet of *Satin Doll* A-Part

| | Grammar | | Lexicon |
|---|---|---|---|
| (r1) | $\mathrm{S} \rightarrow \mathrm{I_C}$ | (l1) | $\mathrm{I_C} \rightarrow \mathrm{Cmaj7}$ |
| (r2) | $\mathrm{I_C} \rightarrow \mathrm{I_C\ I_C}$ | (l2) | $\mathrm{II_C} \rightarrow \mathrm{Dmin7}$ |
| (r3) | $\mathrm{I_C} \rightarrow \mathrm{V_C\ I_C}$ | (l3) | $\mathrm{V_C} \rightarrow \mathrm{G7}$ |
| (r4) | $\mathrm{V_C} \rightarrow \mathrm{II_C\ V_C}$ | (l4) | $\mathrm{V_G} \rightarrow \mathrm{D7}$ |
| (r5) | $\mathrm{V_C} \rightarrow \mathrm{V_G\ V_C}$ | | |

Table 1. Grammar and lexicon for *Take The A Train*.

In these sample analyses, we have employed those rules proposed in [4, 17]. Since this research focuses on the expectation model, the number of rules is slight and the rules are only restricted to the intrinsic ones.

### 4.1 *Take The A Train*

*Take The A Train* is a form of AABA, and consists of 32 bars. For brevity, we analyze only A-part of this section, in which the chord progression is:

Cmaj7 – D7 – Dmin7 – G7 – Cmaj7

we present the sheet music of A-Part of *Take The A Train* in Figure 4 as reference.

We employ the grammar rules and the lexicon in Table 1 to analyze the *take the A train*. These grammars characterize harmonic relationships on an abstract level and have the following meanings: (r1) presents the beginning of parsing and start the song. (r2) is for the prolongation, which admits the continuation of the same chords. (r3), (r4), and (r5) are the preparation rules, that arouse expectations to the following chord, given an initial chord. Consequently, the lexicon is used to connect each chord representation to a surface chord name.

We demonstrated the process of our parser in Table 3 and Table 4. Each row represents an edge of the charts; the ID is added for convenience of explanation, which represents the identifier of the pair (edge and term).'rules' are the employed rules. For example, when the parser observes D7, it checks the lexicon, and the terms in (8) are generated. Consequently, for each term in (8), rules are applied and terms of (9) to (13) are generated. Here, we omit multiple applications of the left-recursion rule to enable (r2) to only generate (l2). Next, the parser replaces terms. Prior to this step, the surviving active edge is only (7); since the category of (7) is S, the parser replaces (10) and (12) with (14) and (15).

Following the tree generation process with the global chart, We observe that two trees (6) and (7) are generated when the initial chord of Cmaj7 is the input. Here, it can be concluded that (6) is a *sub-complete* tree because it has no undecided terms. Subsequently, D7 is input. (6) cannot grow because there are no undecided terms, and only (7) generates the two trees (14) and (15). When Dmin7 and G7 are input, each candidate tree grows. Then, when Cmaj7 is input, (29) becomes (35), and it is *sub-complete*. At Cmaj7, there are candidate trees (36) to (38), which suggests the possibility of subsequent musical development.

| | Grammar | | Lexicon |
|---|---|---|---|
| (r1) | $\mathrm{S} \rightarrow \mathrm{I_C}$ | (l1) | $\mathrm{I_C} \rightarrow \mathrm{Cmaj7}$ |
| (r2) | $\mathrm{I_C} \rightarrow \mathrm{V_C\ I_C}$ | (l2) | $\mathrm{II_C} \rightarrow \mathrm{Dmin7}$ |
| (r3) | $\mathrm{V_C} \rightarrow \mathrm{II_C\ V_C}$ | (l3) | $\mathrm{V_C} \rightarrow \mathrm{G7}$ |
| (r4) | $\mathrm{V_C} \rightarrow \mathrm{V_G\ V_C}$ | (l4) | $\mathrm{II_D} \rightarrow \mathrm{Emin7}$ |
| (r5) | $\mathrm{V_C} \rightarrow \mathrm{V_{G\flat}}$ | (l5) | $\mathrm{V_D} \rightarrow \mathrm{A7}$ |
| (r6) | $\mathrm{V_D} \rightarrow \mathrm{II_D\ V_D}$ | (l6) | $\mathrm{II_{G\flat}} \rightarrow \mathrm{A\flat min7}$ |
| (r7) | $\mathrm{V_{G\flat}} \rightarrow \mathrm{II_{G\flat}\ V_{G\flat}}$ | (l7) | $\mathrm{V_{G\flat}} \rightarrow \mathrm{D\flat 7}$ |
| (r8) | $\mathrm{V_G} \rightarrow \mathrm{II_G\ V_G}$ | (l8) | $\mathrm{II_G} \rightarrow \mathrm{Amin7}$ |
| (r9) | $\mathrm{V_G} \rightarrow \mathrm{V_D\ V_G}$ | (l9) | $\mathrm{V_G} \rightarrow \mathrm{D7}$ |

Table 2. Grammar and lexicon for *Satin Doll*

### 4.2 *Satin Doll*

Similar to *Take The A Train* anlysis, *Satin Doll* is a form of AABA, which consists of 32 bars. However, we analyze only A-part in this section. Further, we drop chord repetitions concisely, thus the chord progression is as follows:

Dmin7 – G7 – Emin7 – A7 – Amin7 – D7 –
A♭min7 – D♭7 – Cmaj7

The sheet music of A-Part of *Satin Doll* is displayed in Figure 5. To analyze *Satin Doll*, we add several grammar rules as in Table 2. The basic set of the rules is common with that of *Take The A Train*, however, we have added unary rewriting rule for (r5). This is a substitution, preserving the same function between both side of the arrow.

We demonstrate growth of tree in Figure 6, instead of permitting the chart representation. In Figure 6, the tree growth is shown in order of (a) → (b) → (c) → (d) → (e) → (f) → (g) → (h) → (i).

*Satin Doll* does not begin from the tonic chord, as opposed to *Take The A Train*, and thus (a) in the figure is not sub-complete. According to the progression of the music, the undecided terms are filled with corresponding chords and the tree becomes finally sub-complete when part A completes. Actually, there appear more candidate sub-trees in the midway of analysis. However, the majority of them are discarded.

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

Table 3. Local chart of *Take The A Train*

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

Table 4. Global chart of *Take The A Train*

## 5. DISCUSSION

We have visualized the growth of trees dynamically with their properties in the figures. We regard the undecided term in the parser as expectation; hence, we could elucidate the two different expectations as follows: One is a chord assignment in a local chart, that is, a local expectation. This operation represents our cognitive process of expecting the consecutive chord after listening to a certain chord, in accordance with given grammar rules such as preparation or prolongation. Secondly is the global expectation, which appears in the global chart, and is resolved as the whole tree in the figure. These processes deeply interconnected to the context of chord progression, and reveal the existence of our innate intuition towards such progression.

The strength of the expectation can be measured by the depth of the corresponding undecided term in the whole tree. For example, $\mathrm{I_C}$ in *Take The A Train* or the same $\mathrm{I_C}$ in *Satin Doll* are long retained undecided and those subtrees become sub-complete when the final solution chord appears. Therefore, a long retained unresolved chord in a considerable shallow position in the tree indicates a very significant role in the music.

In this research, we have focused on the incremental construction of a tree to restrict the grammar rules to a simple set. However, if this research is developed to a more practical one, acquiring the rules in a larger scale with machine learning should be considered. For this purpose, we consider accumulating annotated data by learned musicologists for such learning. The annotation by expectation may be different from the usual chord assignment. Such expectation may vary according to the genre and the experience of the annotators. Thus, we would require an extra care to normalize the annotated data.

On the other hand, merely grammar extension may cause

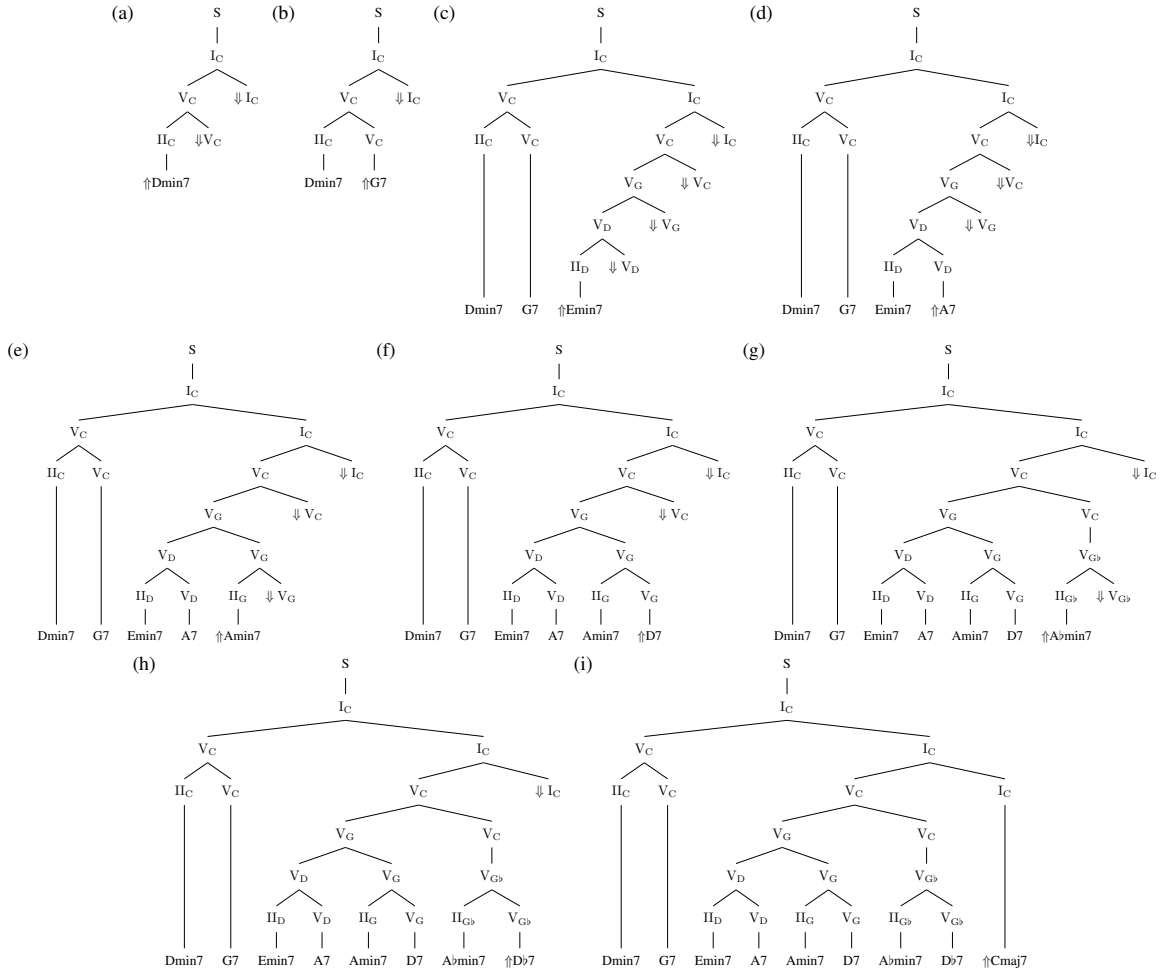


Figure 6. Tree display of incremental analysis *Satin Doll*

an explosive increase of intermediate sub-trees. Therefore, we will introduce probabilistic context-free grammar (PCFG) and update the parsing Algorithm to optimize the candidate tree. The introduction of the notion of probability may give a clearer indication of the strength of expectation.

In Table 1 and 2, we have presented the rules and the lexicon with the suffixes of key, to clarify how each rule is grounded to the chord names at terminal level. We can absorb such redundancy (attaching key labels), employing the grammar formalism with internal features, as *e.g.*, HPSG (Head-driven Phrase Structure Grammar), where '$V_C \rightarrow II_C\ V_C$' and '$V_D \rightarrow II_D\ V_D$' are summarized as

$$V_{[key=X]} \rightarrow II_{[key=X]}\ V_{[key=X]}$$

providing those internal values for $X$ independently.

## 6. CONCLUSION

Our prime contribution in this paper is that we have implemented an incremental parser based on a chart algorithm, which composes a tree structure dynamically from the beginning of a given music piece, assuming that music consists of Chomskian context-free grammar (CFG). The process generates multiple possible parse trees from the first phrase, and thus inevitably includes multiple intermediate local trees. In this research, we have applied our method in sequences of chords in jazz-standards.

The incremental parser is considered the realization of our notion of expectation. As mentioned previously, music is regarded as meaningful when it generates an expectation as to the future progression of music during listening to a piece. Therefore, our parser returns to us the resultant analysis of music structure, and our cognitive construction process of the parse tree.

Since this paper focused on the expectation model, the grammar size was small and restricted. In future work, we will combine our framework with those of automatic grammar acquisition, employing PCFG towards annotated data. Furthermore, we the expectation-based incremental analysis has the potential to be applied to various MIR tasks in the futre, and we will continue our broad investigation.

**Acknowledgments**

This work is supported by JSPS Kaken 16H01744 and 19K12024.

## 7. REFERENCES

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

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

[3] P. Ball, *The music instinct: how music works and why we can't do without it*. Random House, 2010.

[4] M. Rohrmeier, "Towards a generative syntax of tonal harmony," *Journal of Mathematics and Music*, vol. 5, no. 1, pp. 35–53, 2011.

[5] A. Köhn, "Incremental natural language processing: Challenges, strategies, and evaluation," in *Proc. of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Aug. 2018, pp. 2990–3003.

[6] J. Earley, "An efficient context-free parsing algorithm," *Communications of the ACM*, vol. 13, no. 2, pp. 94–102, 1970.

[7] M. Kay, "Algorithm schemata and data structures in syntactic processing," *Technical Report*, 1980.

[8] N. Chomsky, *Syntactic structures*. Mouton, 1957.

[9] ——, *Aspects of the Theory of Syntax*. MIT Press, 1965.

[10] N. Chomsky, R. A. Jacobs, and P. S. Rosenbaum, "Readings in english transformational grammar," *Remarks on nominalization*, pp. 184–221, 1970.

[11] T. Winograd, "Linguistics and the computer analysis of tonal harmony," *Journal of Music Theory*, vol. 12, no. 1, pp. 2–49, 1968.

[12] S. Tojo, Y. Oka, M. Nishida *et al.*, "Analysis of chord progression by hpsg." in *Proc. Artificial Intelligence and Applications*, 2006, pp. 305–310.

[13] M. Granroth-Wilding and M. Steedman, "A robust parser-interpreter for jazz chord sequences," *Journal of New Music Research*, vol. 43, no. 4, pp. 355–374, 2014.

[14] M. Steedman, "Combinatory grammar and human sentence processing," *Modurality in Knowledge Representation and Natural language Understanding*, 1987.

[15] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing "a generative theory of tonal music"," *Journal of New Music Research*, vol. 35, no. 4, pp. 249–277, 2006.

[16] R. S. Jackendoff *et al.*, *A Generative Theory Of Tonal Music*. MIT Press, 1983.

[17] D. Harasim, M. Rohrmeier, and T. J. O'Donnell, "A generalized parsing framework for generative models of harmonic syntax." in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 152–159.

[18] S. Tojo, "Modal logic for tonal music." CMMR, 2019.

[19] 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.