音 楽 情 報 科 学 11－9
（1 9 9 5. 7. 2 2）

演繹オブジェクト指向に基づく ジャズピアノ知識ベースシステムの試作

平田 圭二
NTT 基礎研究所
hirata@nefertiti.brl.ntt.jp

本論文では，ジャズピアノの知識を演繹オブジェクト指向の枠組みに基づいて形式化する手法について述べる．さらに試作したプロトタイプシステムに関する実験結果についても述べる．本研究の動機は，音楽知識を形式的に表現する手法を開発することである．さらにその手法の有効性を確認するために知識ベースを構築する．本研究はジャズのソロピアニストの知識活動に焦点をおき，表現力，柔軟性，理論的な基礎において優れている演繹オブジェクト指向 (DOO) の枠組みを採用する．研究の出発点として，あるジャズピアニストのソロの実演の譜面とそれに対する注釈を提示する．その中では，様々な音楽的概念，例えば 1 音，和音，和音名などが現れ，相互に関連付けられている．DOO の手法に従い，それら音楽的概念の表現を 1 つ 1 つ検討していく．本手法の利点を示すために，現在制作中の試験システムにおけるサンプルセッションの様子を示す．

Prototyping A Jazz Piano Knowledge Base System With A Deductive Object-Oriented Approach

Keiji Hirata
NTT Basic Research Laboratories

This paper presents a method of formalizing jazz piano knowledge that is based on the deductive object-oriented approach. Further, the preliminary results of an experimental system are reported. The motivation of this research is to develop a formal representation method for musical knowledge. Moreover, a knowledge base is constructed to evaluate the method. This research focuses on the knowledge activities of a jazz solo pianist and adopts a deductive object-oriented (DOO) approach because of the advantages offered by the DOO approach: expressibility, flexibility and theoretical basis. As a starting point, the transcription and the annotations of an actual solo performance by a jazz pianist are given, where the various musical concepts, such as notes, chords and chord names, appear and are related to each other. Their representation are examined one after another based on the DOO approach. To show the advantages of this approach, the sample sessions of the experimental system that the author is now developing are demonstrated.

1 Introduction

This paper presents a method of formalizing jazz piano knowledge that is based on the deductive object-oriented approach. Further, the preliminary results of an experimental system are reported.

In general, a musician needs to manipulate and organize various kinds of musical knowledge not only during performance but also in exercises and rehearsals. Those knowledge activities are research subjects in the fields of artificial intelligence (AI) and knowledge processing (KP) [1] [7]. Thus, many research results of AI and KP should be helpful in the formalizing of musician's activities.

This research focuses on the knowledge activities of a jazz solo pianist, since the musical knowledge that a jazz pianist handles is familiar to the author. Toward constructing a jazz pianist knowledge base, first, we should develop a formal knowledge representation method for jazz piano solo performance [4] [5]. Of the many knowledge representation methods have been studied in AI and KP. This research adopted a deductive object-oriented (DOO) approach [8] [6]. The advantages offered by the DOO approach are that the relations among objects are considered as a partial ordering (in the mathematical sense), an object whose information is partially known can be naturally specified, and the expression of rules in a clausal form enables deductive inference.

As a starting point, the transcription of an actual solo performance by a jazz pianist is presented, explained, and analyzed [3]. Once the musical knowledge contained in these materials is extracted and represented using the DOO method, we can treat the knowledge in a formal way. To show the advantages of this approach, the sample sessions of the experimental system are demonstrated. Then, the perspectives and the problems are discussed.

2 "Autumn Leaves" played by Herbie Hancock

Figure 1 shows the first two bars of a solo performance by a jazz pianist; the tune title is Autumn Leaves [3]. The pianist and an editor added the following comment to his performance:

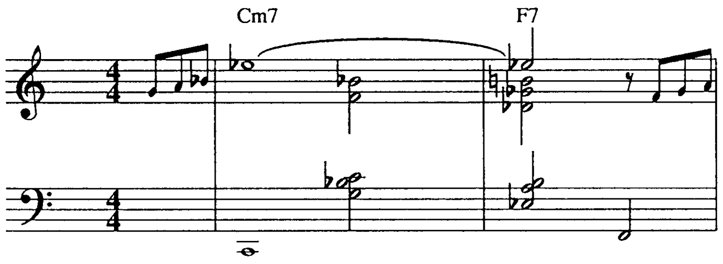


Figure 1: The First Two Bars of Autumn Leaves

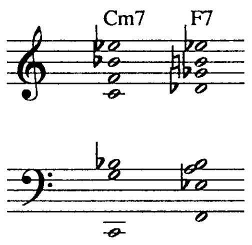


Figure 2: Supplementary Score

> The voicing of C_{m7} does not belong to the conventional 3rd-build system[1]. It has three perfect 4th intervals ($C5 - F5 - B^b5 - E^b6$). Some of the inside voices of C_{m7} move to the inside voices of the subsequent F_7 chord by a semitone up ($B^b5 \to B5$, $F5 \to G^b5$, $C5 \to D^b5$, $B^b4 \to B4$). The voicing of F_7 also has two 4th intervals ($D^b5 - G^b5 - B5$).

Figure 2 shows a supplementary score for this comment.

An apprentice analyzes the scores and the comment, studies the new musical concepts, and applies them in other situations later. How should these knowledge activities be formalized? What kind of representation method efficiently describes these knowledge activities? In the following, let us consider a knowledge representation method for jazz piano, using Figures 1 and 2 and the above comment as a sample problem.

We assume that the piano performance analysis consists of abstraction and association. An apprentice abstracts the actual performance in Figure 1, and finally reaches Figure 2 and the comment previously quoted. The comment tells us that there are various associations among musical concepts, e.g., a set of notes is associated with a chord name. Next, it is noted that the abstraction from Figure 1 to Figure 2 includes three kinds of abstraction: *grouping*, *removing* and *relativizing*. In the first bar of Figure 1, the E^b6 note and the B^b5 and $F5$ notes have different onset times, but in Figure 2, they make a group and are considered as a single chord conceptually. Although three 8th notes of the opening melody $G5A5B^b5$ occur in Figure 1, they are removed in Figure 2 because they are not so important from the musical structure point of view. In Figure 1, the onset time and the duration of each note are specified; the timing information is determined absolutely. However, in Figure 2, the timing information is abstracted and only the relative position of each note is specified.

3 Representing Musical Concepts

3.1 Deductive Object-Oriented Approach

Usually, we represent a relation by a predicate such as $p(t_1, \cdots, t_n)$. The DOO framework can be considered as an extension of predicate logic from the data model point of view. The extension means the introduction of a record type which produces higher expressibility. Then, let us rewrite

[1]A harmony is composed of two or more adjacent 3rd intervals.

a predicate to $(l_0 = p, l_1 = t_1, \cdots, l_n = t_n)$ or $p(l_1 = t_1, \cdots, l_n = t_n)$. This new notation has the following advantages: there is flexibility in terms of argument positions and the number of arguments, and it is easy to introduce an ordering between objects.

3.2 Object Terms

A fundamental concept is represented by *a basic object term*; e.g., *note*, *chord_name* and *jazz_tune*. Next, the addition of attribute names and their values to a basic object term can represent a more complex object. It is called *a complex object term*; the examples are shown:

$$\begin{array}{l} note(pitch = C, octave = 5) \\ chord\_name(root = G, name = 7, tension = \{9, \flat 13\}) \\ chord(notes = \{note(pitch = C, octave = 5), note(pitch = D\flat, octave = 6)\}) \end{array}$$

Here, *pitch*, *octave*, *root* and so on are labels, and C, 5 and G are their values which are possibly object terms defined later. For instance, $note(pitch = C, octave = 5)$ represents a note whose pitch class is C and which is positioned within the 5th octave. The label and value pairs specify the intrinsic properties of a complex object term. *An object term* is either a basic object term or a complex object term. Note that our framework regards an object term itself as an object identifier.

3.3 Subsumption Relation

Between two basic object terms, an ordering, denoted by $\preceq$, is defined.

$$\begin{array}{l} Gershwin \preceq American \\ standard\_tune \preceq jazz\_tune \\ NTT \preceq telephone\_company \preceq company \end{array}$$

As you see, the symbol $\preceq$ represents the ordering between a class and its instance or between a general object and a more special object. We can interpret that $\preceq$ has the meanings like *is_a*, *kind_of* and *part_of*.

To compare two object terms, the ordering $\preceq$ is extended to a subsumption relation, denoted by $\sqsubseteq$. The ordering principle for $\sqsubseteq$ is as follows:

- If the number of labels of an object term is more than that of another, the object is more special and more instantiated.
- If the value of a label, included in an object term, is more general, so is the object term itself.

It is important that the above statement is formalized as the following deductive rules $\mathbf{R_{sub}}$:

For two (ground) object terms $o(k_1 = v_1, \cdots, k_m = v_m)$ and $p(l_1 = w_1, \cdots, l_n = w_n)$,
$o \preceq p \wedge (\forall l_j \exists k_i \; k_i = l_j \wedge v_i \sqsubseteq w_j) \rightarrow o(k_1 = v_1, \cdots, k_m = v_m) \sqsubseteq p(l_1 = w_1, \cdots, l_n = w_n)$,
where $1 \leq i \leq m$ and $1 \leq j \leq n$ $\mathbf{R_{sub}}$

Since this rule is higher-order and is applied pairwise to every objects, this rule can be seen as a part of background knowledge. The examples are shown:

$$\begin{array}{l} note(pitch = C, octave = 5) \sqsubseteq note(pitch = C) \\ standard\_tune(composer = Gershwin) \sqsubseteq jazz\_tune(composer = American) \\ worker(affiliation = NTT) \sqsubseteq worker(affiliation = company) \end{array}$$

Intuitively, "*instantiated* $\sqsubseteq$ *abstract*" is stated.

3.4 Ordering of Sets

A set is universal and is one of primitive data structures in knowledge representation. In general, to represent a concept, a set is used in two ways:

- A class is a set of its instances and a class is more abstract than its instances: e.g., $\{a,b,c\} \sqsubseteq \{a,b,c,d,e\}$. This is called *the Hoare ordering*, denoted by $\sqsubseteq_H$. The definition is: for two sets $\{o_1,\cdots,o_m\}(=S_o)$ and $\{p_1,\cdots,p_n\}(=S_p)$, $S_o \sqsubseteq_H S_p \overset{\text{def}}{=} \forall o_i \in S_o, \exists p_j \in S_p \; o_i \sqsubseteq p_j$
- The whole consists of its parts and the whole is more specific and more instantiated than its parts: e.g., $\{a,b,c,d,e\} \sqsubseteq \{a,b,c\}$. This is called *the Smyth ordering*, denoted by $\sqsubseteq_S$. The definition is: for S_o and S_p, $S_o \sqsubseteq_S S_p \overset{\text{def}}{=} \forall p_j \in S_p, \exists o_i \in S_o \; o_i \sqsubseteq p_j$

The examples are shown:

$$\begin{array}{ll} A. & \{note(pitch = C), note(pitch = E), note(pitch = G), note(pitch = B\flat)\} \\ & \quad \sqsubseteq_S \{note(pitch = E), note(pitch = B\flat)\} \\ B. & note(pitch = C, time =_H \{2\}) \sqsubseteq note(pitch = C, time =_H \{1,2,3\}) \\ C. & chord\_name(root = G, name = 7, tension =_S \{9, \flat 13\}) \\ & \quad \sqsubseteq chord\_name(root = G, name = 7, tension =_S \{9\}) \end{array}$$

The example A. means that the whole chord $CEGB\flat$ includes $EB\flat$ as a subchord and the subchord is more general. The example B. means that a note whose onset time is 2 is regarded more instantiated than a note whose onset time is either 1, 2, or 3. The example C. means that the chord name $G_7(9,\flat 13)$ is more special than $G_7(9)$.

3.5 Attribute Terms and Inheritance

Our DOO framework further extends an object term to *an attribute term* which has extrinsic attribute names and their values in addition to intrinsic ones. An attribute term can specify additional properties; two examples are shown:

$$\begin{array}{l} chord\_name(root = G, name = 7)/(voicings = \{\{F, B\}\}) \\ chord(notes = \{F, B\})/(chord\_name = \{G_7, D\flat_7\}) \end{array}$$

Since the extrinsic attribute names and their values are not used as an object identifier, the ordering between two attribute terms is the same as the ordering between the object terms which occur in these two attribute terms. The left-hand side of '/' in an attribute term is an object term and corresponds to its object identifier. The right-hand side represents extrinsic attributes and their values of an object.

Then, property inheritance can be realized by using attribute terms. In general, the inheritance in the DOO framework is defined by the following deductive rule:

$$o \sqsubseteq p \;\wedge\; o/(l = a) \;\wedge\; p/(l = b) \rightarrow a \sqsubseteq b$$

The value of the attribute l is inherited from the upper-class object p to the lower-class object o, and the value a should be constrained by this rule. Simultaneously, extrinsic attribute values can be inherited from a lower object to upper objects inversely. An example is shown:

Suppose $o = standard\_tune(composer = Gershwin)$ and $p = jazz\_tune(composer = American)$, $o/(played\_by = pianist) \wedge p/(played\_by = musician) \rightarrow pianist \sqsubseteq musician$

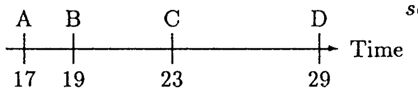
As you noticed, the inheritance rule is also higher-order.

4 Analysis of Actual Performance

This section formalize the observation mentioned in Section 2 with the DOO approach to achieve "*an actual performance* $\sqsubseteq$ *supplementary score*" and the association of musical concepts.

4.1 Representing Chronological Order

First, to represent a chronological order for an event sequence, a new object term, $seq(i = I, l = L, r = R)$, is introduced, which means that first an event L occurs and, after an interval of I, an event R occurs. For instance, an event sequence "$A, B, C, D \cdots$" is represented by the following object term:



$$\begin{array}{l} seq(i = 2, l = A, \\ \quad r = seq(i = 4, l = B, \\ \qquad r = seq(i = 6, l = C, \\ \qquad\quad r = seq(l = D, r = \cdots)))) \end{array}$$

Moreover, a set of constraint rules $\mathbf{R_{seq}}$ is given to prescribe the *seq* object.

$$\begin{array}{ll} seq(l = seq(M = L), r = R) \sqsubseteq seq(l = L, r = R) & \\ seq(l = L, r = seq(M = R)) \sqsubseteq seq(l = L, r = R) & \\ seq(r = X) \equiv seq(l = X) & \mathbf{R_{seq}} \\ seq(M = X) \sqsubseteq X \quad \text{if } X \text{ is of form either } seq(l = L, r = R) \text{ or } seq(N = Y) & \end{array}$$

where L, R and Y are object terms and M, N is either l or r

The rules for the *seq* object terms with the i attribute explicitly specified are slightly complicated.

4.2 Abstraction

As stated before in Section 2, the score abstraction consists of grouping, removing and relativizing.

Grouping: Suppose that, given A and B, there exists C such that $A \sqsubseteq C$ and $B \sqsubseteq C$.

$$seq(l = A, r = seq(l = B, r = D)) \sqsubseteq seq(l = C, r = D)$$

is derived from the rules $\mathbf{R_{seq}}$ and $\mathbf{R_{sub}}$ (Section 3.3). Two events that occur adjacently are merged into one event; the example is shown:

$$seq(l =_H \{n(p = 60)\}, r = seq(l =_H \{n(p = 72)\}, r = X)) \sqsubseteq seq(l =_H \{n(p = 60), n(p = 72)\}, r = X)$$

Removing: Removing is simply realized by $\mathbf{R_{sub}}$. That is, to remove an attribute name and its value pair straightforwardly abstracts an object term; the example is shown:

$$seq(l = A, r = seq(l = B, r = seq(l = C, r = \cdots))) \sqsubseteq seq(l = A, r = seq(r = seq(l = C, r = \cdots)))$$

Relativizing: Relativizing is also realized by simply removing the i attributes from the *seq* object terms; the example is shown:

$$\begin{array}{ll} & seq(i = 2, l = A, r = seq(i = 4, l = B, r = seq(i = 6, l = C, r = \cdots))) \\ \sqsubseteq & seq(l = A, r = seq(l = B, r = seq(l = C, r = \cdots))) \end{array}$$

Demonstration: Figure 3 demonstrates the abstraction from Figure 1 to Figure 2, using *note* objects, *seq* objects and the $\sqsubseteq$-ordering. First, the *seq* object term (1) in Figure 3 stands for the actual solo performance shown in Figure 1. For space efficiency, $n(P, O, D)$ is an abbreviation of $note(pitch = P, octave = O, duration = D)$ and $\{n(P, O, D), \cdots\}$ is of $chord(notes =_H \{n(P, O, D), \cdots\})$. For simplicity, the duration of an 8th note is supposed to be 1 clock unit and ties are not taken into account. Next, abstraction by removing the parts of melody leads to (2) in Figure 3. Next, abstraction by relativizing in terms of time leads to (3). Next, abstraction by grouping leads to (4). Finally, abstraction by removing the duration of each note leads to (5), which corresponds to the supplementary score in Figure 2. Here, $n(P, O)$ is an abbreviation of $note(pitch = P, octave = O)$. It is important that (1) $\sqsubseteq$ (2) $\sqsubseteq$ (3) $\sqsubseteq$ (4) $\sqsubseteq$ (5) holds in Figure 3.

(1)
$seq(i = -3, l = seq(i = 1, l = \{n(G, 5, 1)\}, r = seq(i = 1, l = \{n(A, 5, 1)\}, r = seq(l = \{n(B^\flat, 5, 1)\}))),$
$\quad r = seq(i = 4, l = \{n(E^\flat, 6, 8), n(C, 3, 8)\},$
$\quad\quad r = seq(i = 4, l = \{n(B^\flat, 5, 4), n(F, 5, 4), n(C, 5, 4), n(B^\flat, 4, 4), n(G, 4, 4)\},$
$\quad\quad\quad r = seq(i = 4, l = \{n(E^\flat, 6, 4), n(B, 5, 4), n(G^\flat, 5, 4), n(D^\flat, 5, 4), n(B, 4, 4), n(A, 4, 4), n(E^\flat, 4, 4)\},$
$\quad\quad\quad\quad r = seq(i = 1, l = \{n(F, 3, 4)\},$
$\quad\quad\quad\quad\quad r = seq(i = 1, l = \{n(F, 5, 1)\}, r = seq(i = 1, l = \{n(G, 5, 1)\}, r = seq(l = \{n(A, 5, 1)\}))))))))$

(2)
$seq(i = -3,$
$\quad r = seq(i = 4, l = \{n(E^\flat, 6, 8), n(C, 3, 8)\},$
$\quad\quad r = seq(i = 4, l = \{n(B^\flat, 5, 4), n(F, 5, 4), n(C, 5, 4), n(B^\flat, 4, 4), n(G, 4, 4)\},$
$\quad\quad\quad r = seq(i = 4, l = \{n(E^\flat, 6, 4), n(B, 5, 4), n(G^\flat, 5, 4), n(D^\flat, 5, 4), n(B, 4, 4), n(A, 4, 4), n(E^\flat, 4, 4)\},$
$\quad\quad\quad\quad r = seq(i = 1, l = \{n(F, 3, 4)\})))))$

(3)
$seq(r = seq(l = \{n(E^\flat, 6, 8), n(C, 3, 8)\},$
$\quad\quad r = seq(l = \{n(B^\flat, 5, 4), n(F, 5, 4), n(C, 5, 4), n(B^\flat, 4, 4), n(G, 4, 4)\},$
$\quad\quad\quad r = seq(l = \{n(E^\flat, 6, 4), n(B, 5, 4), n(G^\flat, 5, 4), n(D^\flat, 5, 4), n(B, 4, 4), n(A, 4, 4), n(E^\flat, 4, 4)\},$
$\quad\quad\quad\quad r = seq(l = \{n(F, 3, 4)\})))))$

(4)
$seq(r = seq(l = \{n(E^\flat, 6, 8), n(C, 3, 8), n(B^\flat, 5, 4), n(F, 5, 4), n(C, 5, 4), n(B^\flat, 4, 4), n(G, 4, 4)\},$
$\quad\quad r = seq(l = \{n(E^\flat, 6, 4), n(B, 5, 4), n(G^\flat, 5, 4), n(D^\flat, 5, 4), n(B, 4, 4), n(A, 4, 4), n(E^\flat, 4, 4) n(F, 3, 4)\})))$

(5)
$seq(r = seq(l = \{n(E^\flat, 6), n(B^\flat, 5), n(F, 5), n(C, 5), n(B^\flat, 4), n(G, 4), n(C, 3)\},$
$\quad\quad r = seq(l = \{n(E^\flat, 6), n(B, 5), n(G^\flat, 5), n(D^\flat, 5), n(B, 4), n(A, 4), n(E^\flat, 4) n(F, 3)\})))$

Figure 3: Abstraction of Actual Performance

4.3 Association

The analysis of jazz piano performance consists of association as well as abstraction. The association is to relate different musical concepts represented by objects to each other. It is implemented by creating links of extrinsic attributes.

Chord Name: The chord names of the two chord in (5) of Figure 3 are C_{m7} and F_7; their tension notes are $11th$ and $\flat 9, \# 11, \flat 13ths$, respectively. Our method connects the distinct concepts, *chord*, *chord_name* and *tension*, by using extrinsic attributes in the following way:

$$M_1 = chord(notes = \{n(E^\flat, 6), n(B^\flat, 5), n(F, 5), n(C, 5), n(B^\flat, 4), n(G, 4), n(C, 3)\})$$
$$M_2 = chord(notes = \{n(E^\flat, 6), n(B, 5), n(G^\flat, 5), n(D^\flat, 5), n(B, 4), n(A, 4), n(E^\flat, 4) n(F, 3)\})$$
$$M_3 = chord\_name(root = C, name = m7, tension =_S \{11\})$$
$$M_4 = chord\_name(root = F, name = 7, tension =_S \{\flat 9, \# 11, \flat 13\})$$

$$M_1/(chord\_name =_H \{M_3\}) \quad M_3/(voicings =_H \{M_1\})$$
$$M_2/(chord\_name =_H \{M_4\}) \quad M_4/(voicings =_H \{M_2\})$$

Note that the right-hand side of '/' represents the additional information which is not necessary for object identification.

Voice Leading: The comment quoted in Section 2 states voice leading as "Some of the inside voices of C_{m7} move to the inside voices of the subsequent F_7 chord by a semitone up". Our method of forming associations by attributes translates this part as

For $M_1 = seq(l = chord(notes =_S \{n(B^\flat, 5), n(F, 5), n(C, 5), n(B^\flat, 4)\}),$
$\quad\quad r = seq(l = chord(notes =_S \{n(B, 5), n(G^\flat, 5), n(D^\flat, 5), n(B, 4)\})))$
(5) in Figure 3 $\sqsubseteq M_1$ and $M_1/(diff = 1)$

That is, for every pair of notes occurring in M_1 and M_2, the pitch difference is 1 semitone. (5) occurs in Figure 3.

Subchord Structure: In terms of voicing, the comment also states that C_{m7} has three 4th intervals. Our method translates the comment to

$$\begin{aligned}&\text{For } M_1 = chord(notes =_S \{n(E^\flat,6), n(B^\flat,5), n(F,5), n(C,5)\}) \text{ and}\\ &\quad M_2 = chord(notes =_S \{n(E^\flat,6), n(B^\flat,5), n(F,5), n(C,5)n(B^\flat,4), n(G,4), n(C,3)\})\\ &M_2 \sqsubseteq M_1 \text{ and } M_1/(interval = 5)\end{aligned}$$

That is, every pitch difference of adjacent notes in M_3 that is a subchord of C_{m7} is 5 semitones (4th degree).

5 Sample Sessions

At present, the author is prototyping an experimental system in KLIC [2]; KLIC is a compiler-based language processing system on UNIX for a concurrent logic programming language KL1. The experimental system consists of about 1800 lines in KL1. The input data to the experimental system is the transcription of an actual solo performance by Herbie Hancock and the comment of the performance [3]; he played "Autumn Leaves" for one chorus. As a result, 135 attribute terms are stored into the experimental system in total. Then, a user issues queries to retrieve and deduce the knowledge of jazz solo piano. The sample sessions are shown below.

Query 1 : First, the chords which are more instantiated than a given chord, $chord(notes =_S \{\})$, are collected (referred to as M). Then, all the labels of each chord in M are collected. Since $chord(notes =_S \{\})$ is the most abstract chord, this query is equivalent to collecting all the labels which occur in every chord stored in the system. The answer is:

$$[interval, chord\_name, scale, ust, pedal]$$

Here, *ust* is an abbreviation of "upper structure triad".

Query 2 : First, given a chord name, $chord\_name(name = m7)$, its voicings are obtained (referred to as M); this operation is implemented by the dereference along the *voicings* attribute. Then, each voicing in M is relativized with respect to its root note. The answer is:

$$\begin{aligned}&[[0,19,22,24,29,34,39],\ [-2,0,3,7,12],\ [0,10,15,19,22,26],\\ &[0,3,10,15,19,22],\ [0,3,10,14,19,22],\ [0,7,12,14,15,19,24]]\end{aligned}$$

Query 3 : First, given a chord, $chord(notes =_S \{n(C,5), n(F,5), n(B\flat,5)\})$, the chords which are more instantiated are collected (referred to as M). Then, the chord names of each chord in M are collected. The answer is:

$$\begin{aligned}&[cn(C,m7,[11]), cn(E^\flat,maj7,[9,13]), cn(B^\flat,7,[9]), cn(E^\flat,maj7,[9,13]),\\ &cn(G,min7,[11]), cn(root = D, numerator = cn(G,m7,[9,11])),\\ &cn(root = D, numerator = cn(B^\flat,maj7,[9])), cn(D,7,[\flat 9,\#9,\flat 13])]\end{aligned}$$

Here, *cn* is an abbreviation of *chord_name*, and $cn(root = R, numerator = C)$ represents a fractional chord C/R.

Query 4 : The derivation by the deductive rules $\mathbf{R_{sub}}$ and $\mathbf{R_{seq}}$ is examined. First, all the object terms which are derived from a given object term, $seq(l = \{a\}, r = seq(l = \{b\}, r = \{c\}))$, are collected (referred to as M). For readability, Figure 4 shows only a part of the $\sqsubseteq$-hierarchy built by the answer M. In the figure, $seq(L,R)$ is an abbreviation of $seq(l = L, r = R)$. Note that, along the $\sqsubseteq$-ordering, extrinsic properties are inherited upward and downward.

$$\begin{tikzcd}
 & & seq(l=\{b\}) & & & seq(r=\{c\}) & & & \\
seq(l=\{a,b\}) & & seq(r=seq(l=\{b\})) \arrow[u] & & seq(\{b\},\{c\}) \arrow[ur] & & seq(r=seq(r=\{c\})) \arrow[ul] & & seq(l=\{a\}) \\
 & seq(\{a\},seq(l=\{b\})) \arrow[ul] \arrow[ur] & & & seq(r=seq(\{b\},\{c\})) \arrow[u] & & seq(\{a\},seq(r=\{c\})) \arrow[u] \arrow[urr] & & seq(\{a,b\},\{c\}) \\
 & & & & & seq(\{a\},seq(\{b\},\{c\})) \arrow[ullll] \arrow[ul] \arrow[ur] \arrow[urrr] & & &
\end{tikzcd}$$

Figure 4: Part of the $\sqsubseteq$-Hierarchy

6 Concluding Remarks

As mentioned earlier, a transcription from an actual performance by a jazz pianist, its explanation, and analysis to the performance are given as the starting point of this work. If the jazz piano knowledge included in these materials is put into a knowledge base by using the DOO method presented in this paper, a user can issue queries and execute various operations, such as (deductive) inference, searching, updating and inheritance along the $\sqsubseteq$-ordering.

The author thinks that future work is to put more performance scores and the annotations into the experimental system, to experience various queries and responses and to improve the system (e.g., a relevant query primitive set and GUI).

The analysis by abstraction and association will be automated to some extent. The current experimental system is supposed to interactively and incrementally manipulate jazz piano knowledge, especially harmony theory for jazz solo piano. In future, the application area will be expanded to other musical theories.

Acknowledgment: The author thanks Dr. Ken'ichiro Ishii and Dr. Rikio Onai of NTT for supporting the author's research environment.

References

[1] M. Balaban, K. Ebcioğlue and O. Laske (eds.), *Understanding Music with AI: Perspectives on Music Cognition*, The MIT Press, 1992.

[2] T. Chikayama, *KLIC-2.000 User's Manual*, ICOT, Available through anonymous FTP from `http://www.icot.or.jp` (1995).

[3] Herbie Hancock, Private Piano Lessons, *jazz Life*, No.144-No.150, Ritto-sha, 1989 (in Japanese).

[4] K. Hirata, Study on Music Description Based on Deductive Object-Oriented Approach, *In Proc. of SIGMUS 94-MUS-8*, pp.57-62, IPSJ, 1994 (in Japanese).

[5] K. Hirata, Towards Formalizing Jazz Piano Knowledge with a Deductive Object-Oriented Approach, *to appear in Proc. of IJCAI95 Workshop on AI and Music*, 1995.

[6] M. Kifer and G. Lausen, F-Logic: A Higher-Order Language for Reasoning about Objects, Inheritance, and Scheme, *ACM SIGMOD International Conference on Management of Data*, *SIGMOD Record*, Vol.18, No.2, 1989.

[7] S. Schwanauer and D. Levitt (eds.), *Machine Models of Music*, The MIT Press, 1993.

[8] K. Yokota and H. Yasukawa, Towards an Integrated Knowledge-Base Management System – Overview of R&D on Databases and Knowledge-Bases in the FGCS Project, *In Proc. of FGCS'92*, ICOT, 1992.